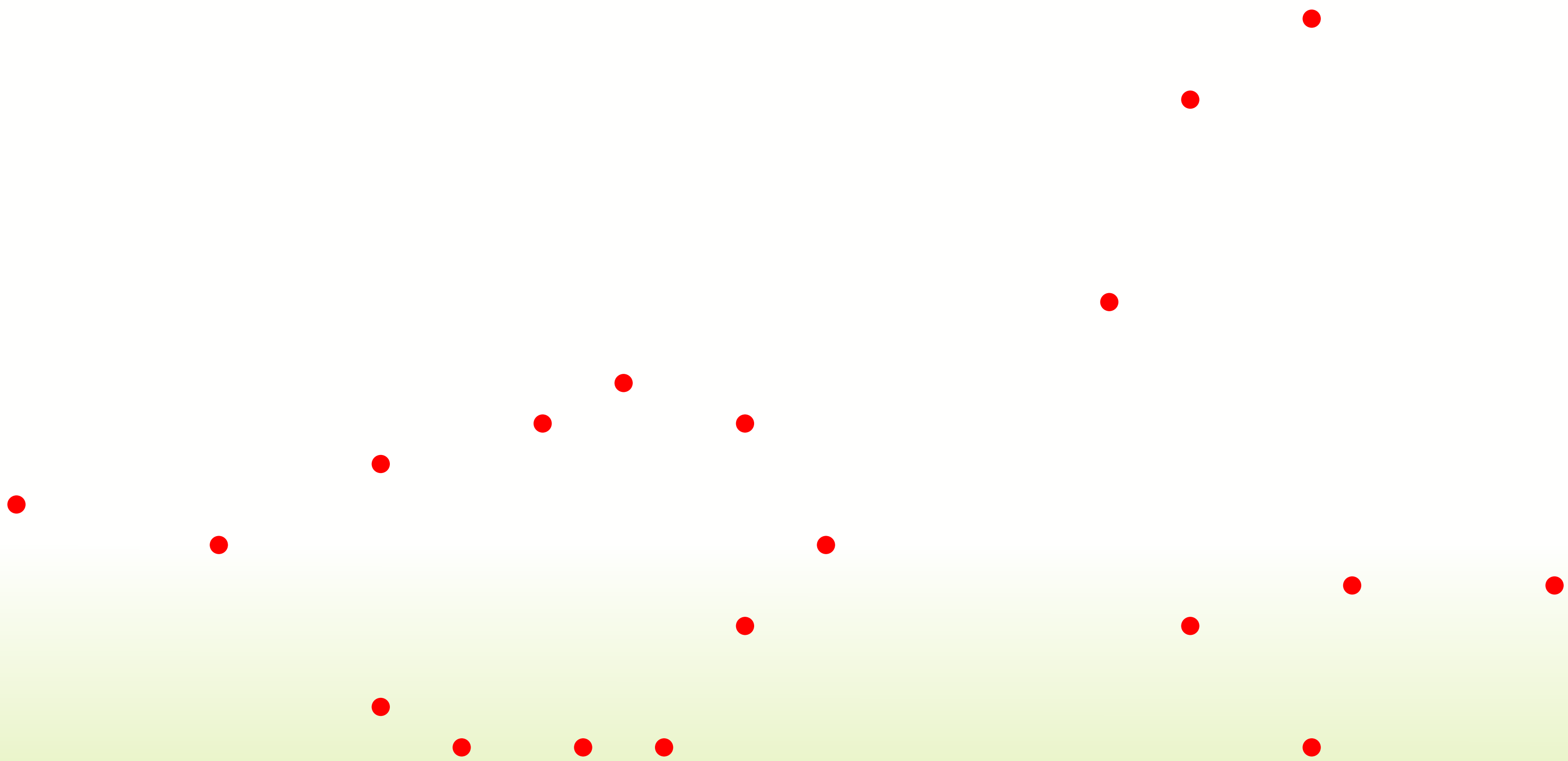
Efficient Spanner Construction for Disk Transmission Graphs

Haim Kaplan (Tel Aviv University)
Wolfgang Mulzer (Freie Universität Berlin)
Liam Roditty (Bar-Ilan University)
Paul Seiferth (Freie Universität Berlin)

Transmisson Graphs & Spanners

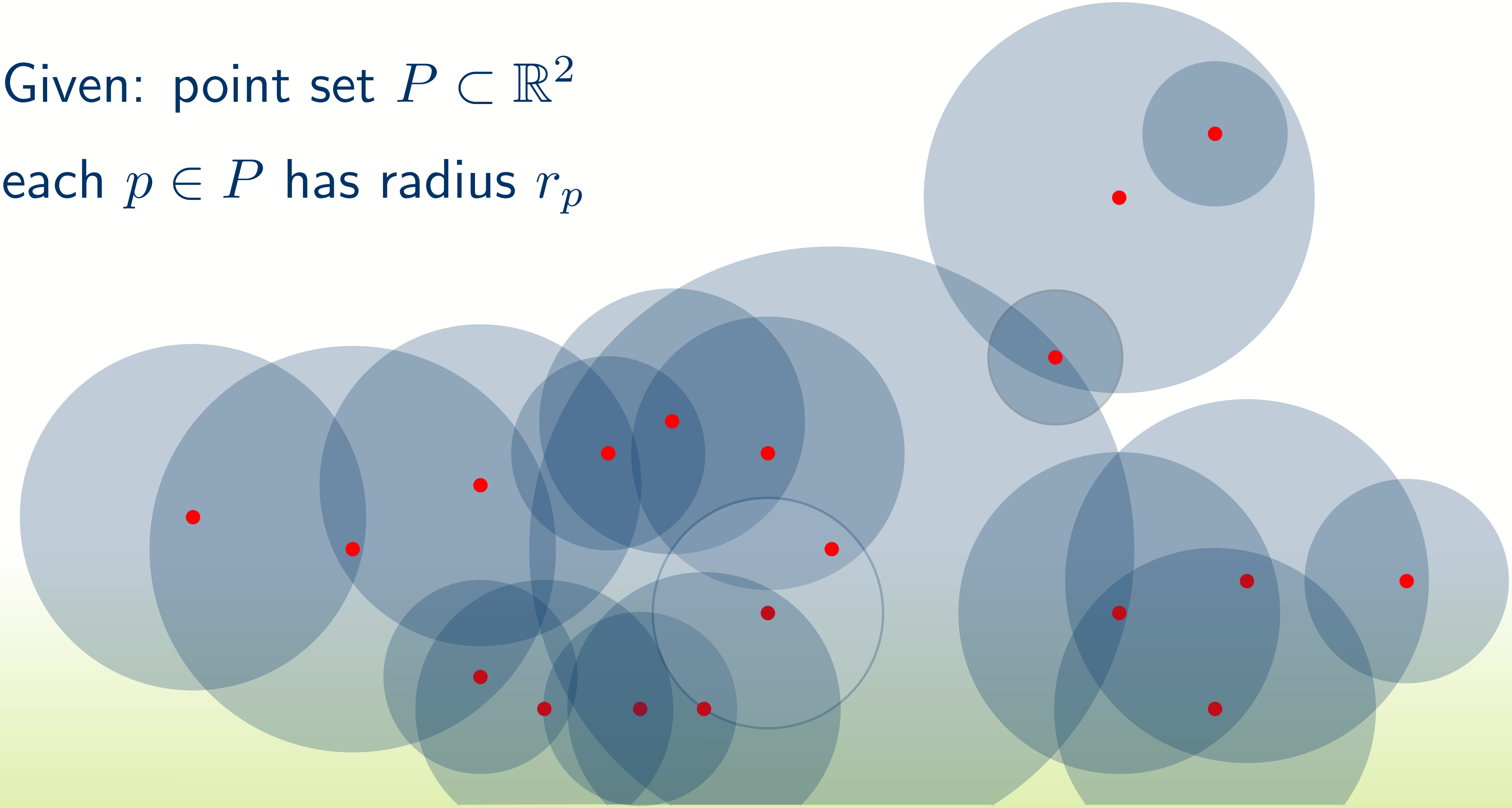
Given: point set $P \subset \mathbb{R}^2$



Transmisson Graphs & Spanners

Given: point set $P \subset \mathbb{R}^2$

each $p \in P$ has radius r_p

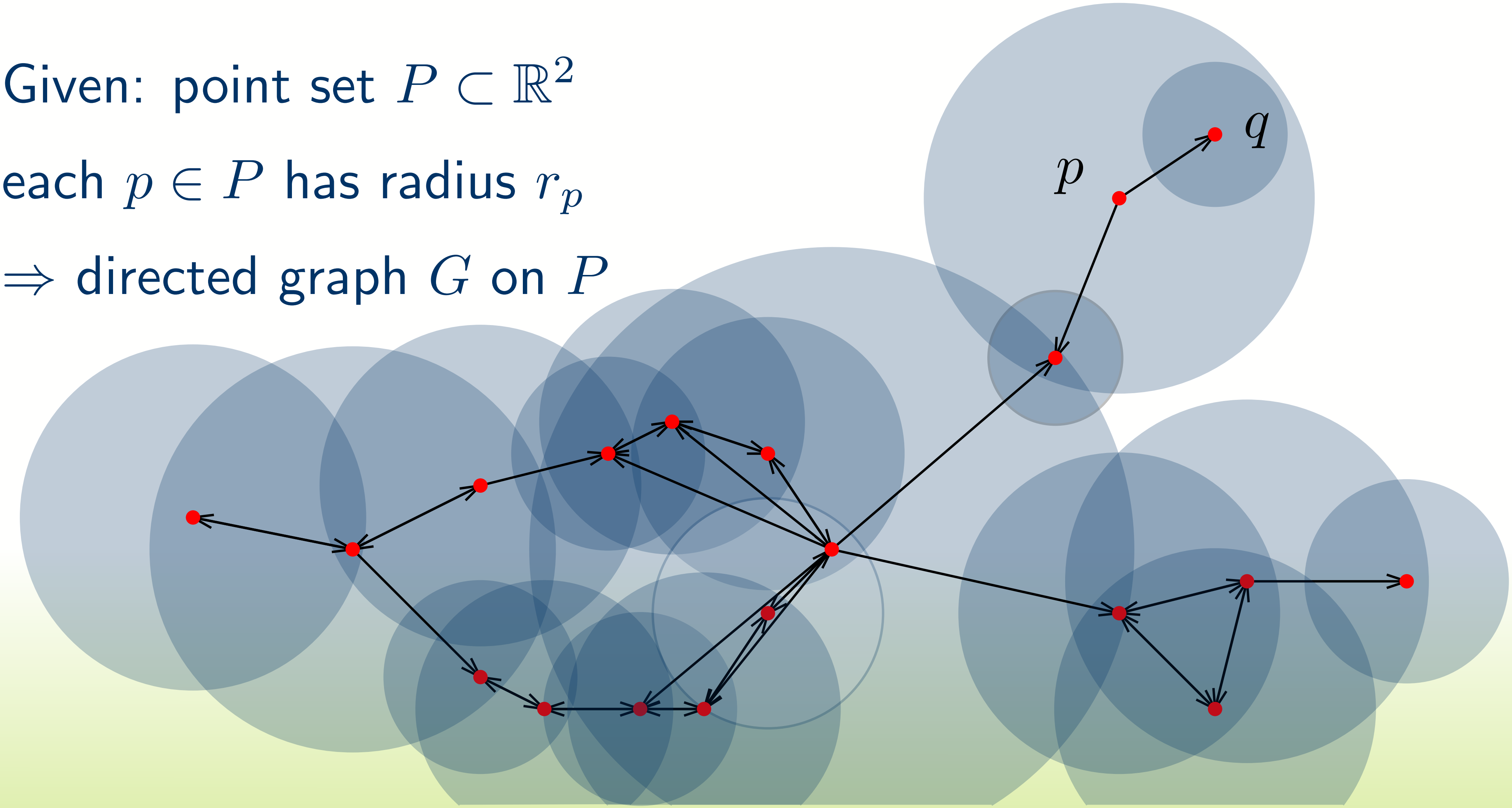


Transmisson Graphs & Spanners

Given: point set $P \subset \mathbb{R}^2$

each $p \in P$ has radius r_p

$\Rightarrow$ directed graph G on P

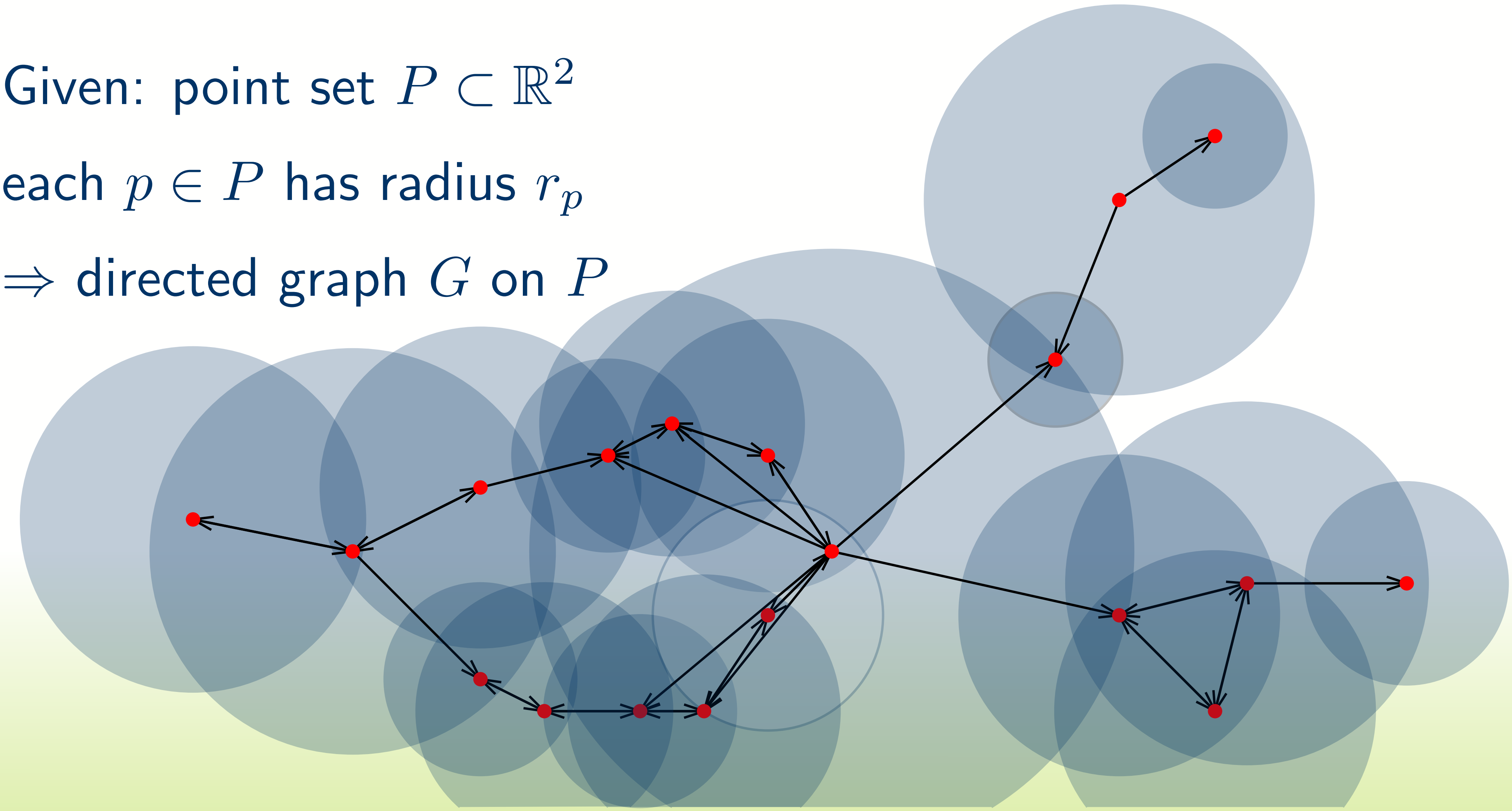


Transmisson Graphs & Spanners

Given: point set $P \subset \mathbb{R}^2$

each $p \in P$ has radius r_p

$\Rightarrow$ directed graph G on P



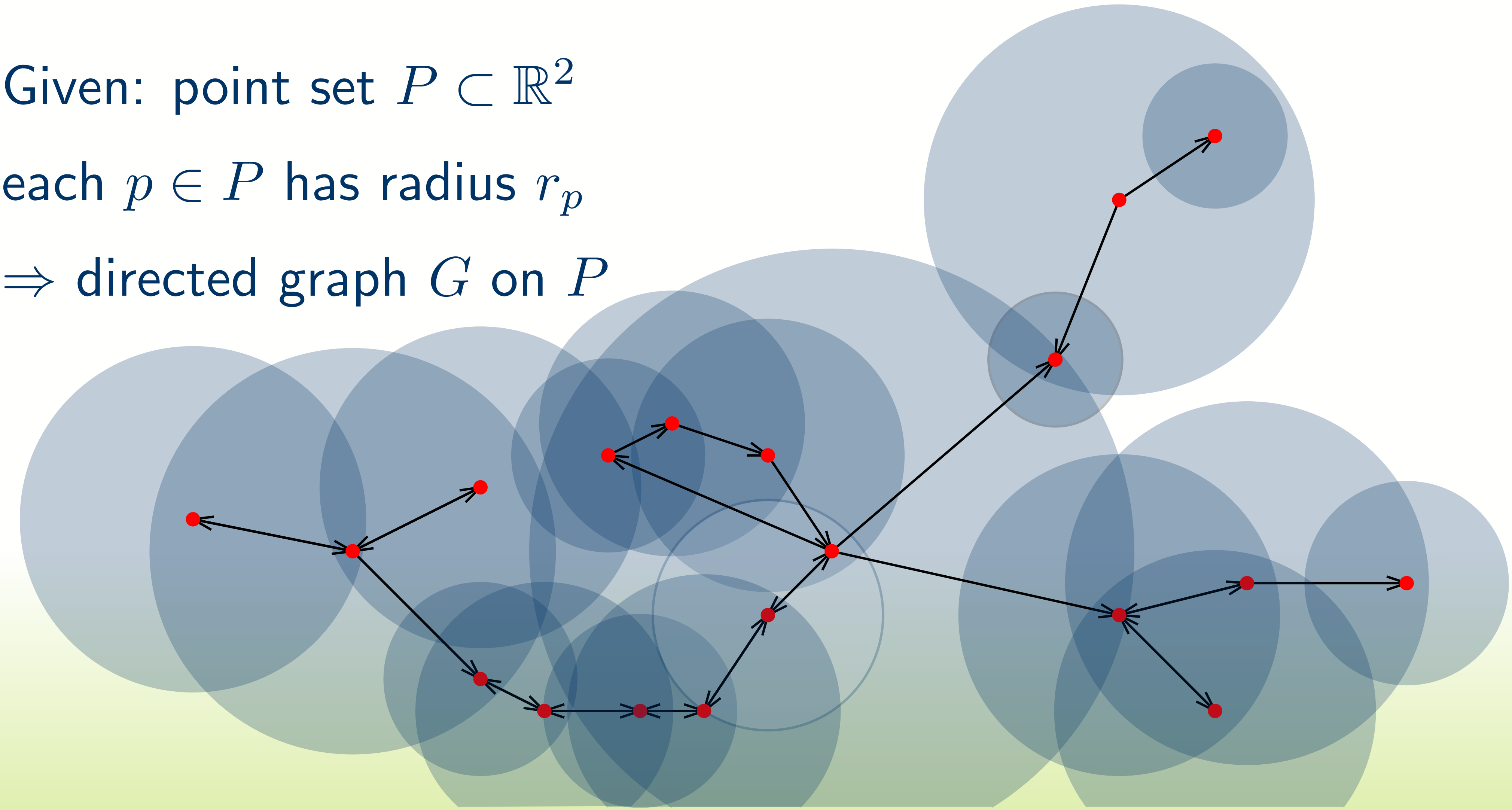
We want to compute a t-*spanner* $H \subseteq G$:
for any $p, q \in P$ we get $d_H(p,q) \leq t \cdot d_G(p,q)$

Transmisson Graphs & Spanners

Given: point set $P \subset \mathbb{R}^2$

each $p \in P$ has radius r_p

$\Rightarrow$ directed graph G on P



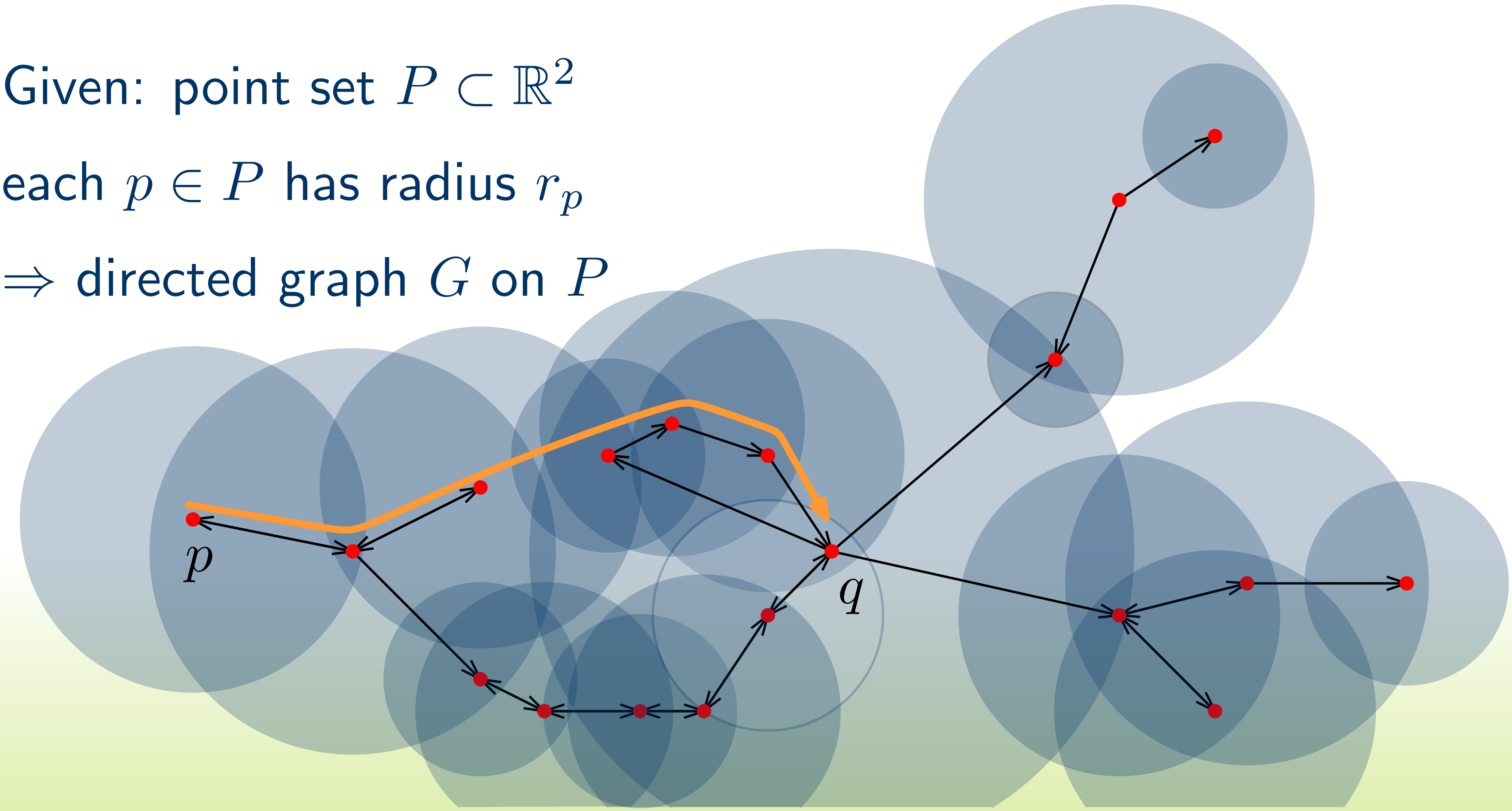
We want to compute a t-*spanner* $H \subseteq G$:
for any $p, q \in P$ we get $d_H(p,q) \leq t \cdot d_G(p,q)$

Transmisson Graphs & Spanners

Given: point set $P \subset \mathbb{R}^2$

each $p \in P$ has radius r_p

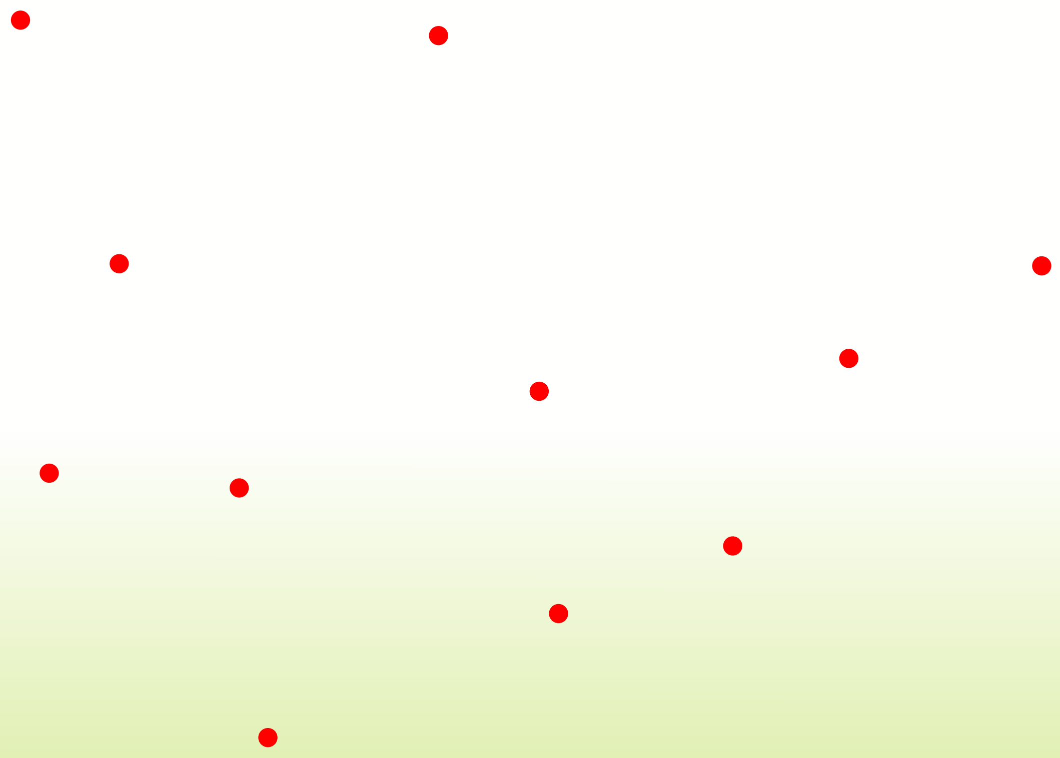
$\Rightarrow$ directed graph G on P



We want to compute a t-*spanner* $H \subseteq G$:
for any $p, q \in P$ we get $d_H(p,q) \leq t \cdot d_G(p,q)$

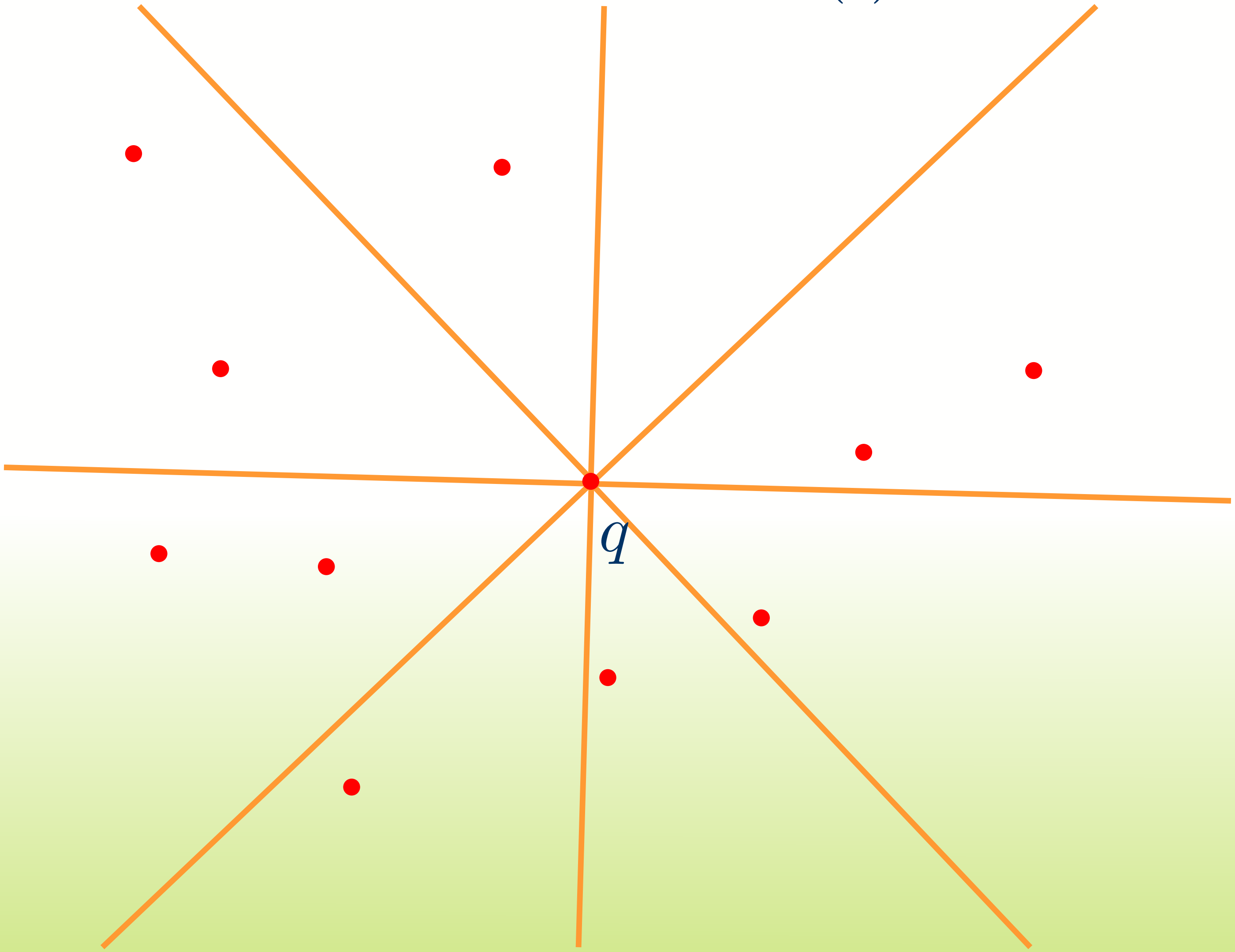
Modified Yao Graph

Given: point set $P \subset \mathbb{R}^2$, parameter $k(t)$



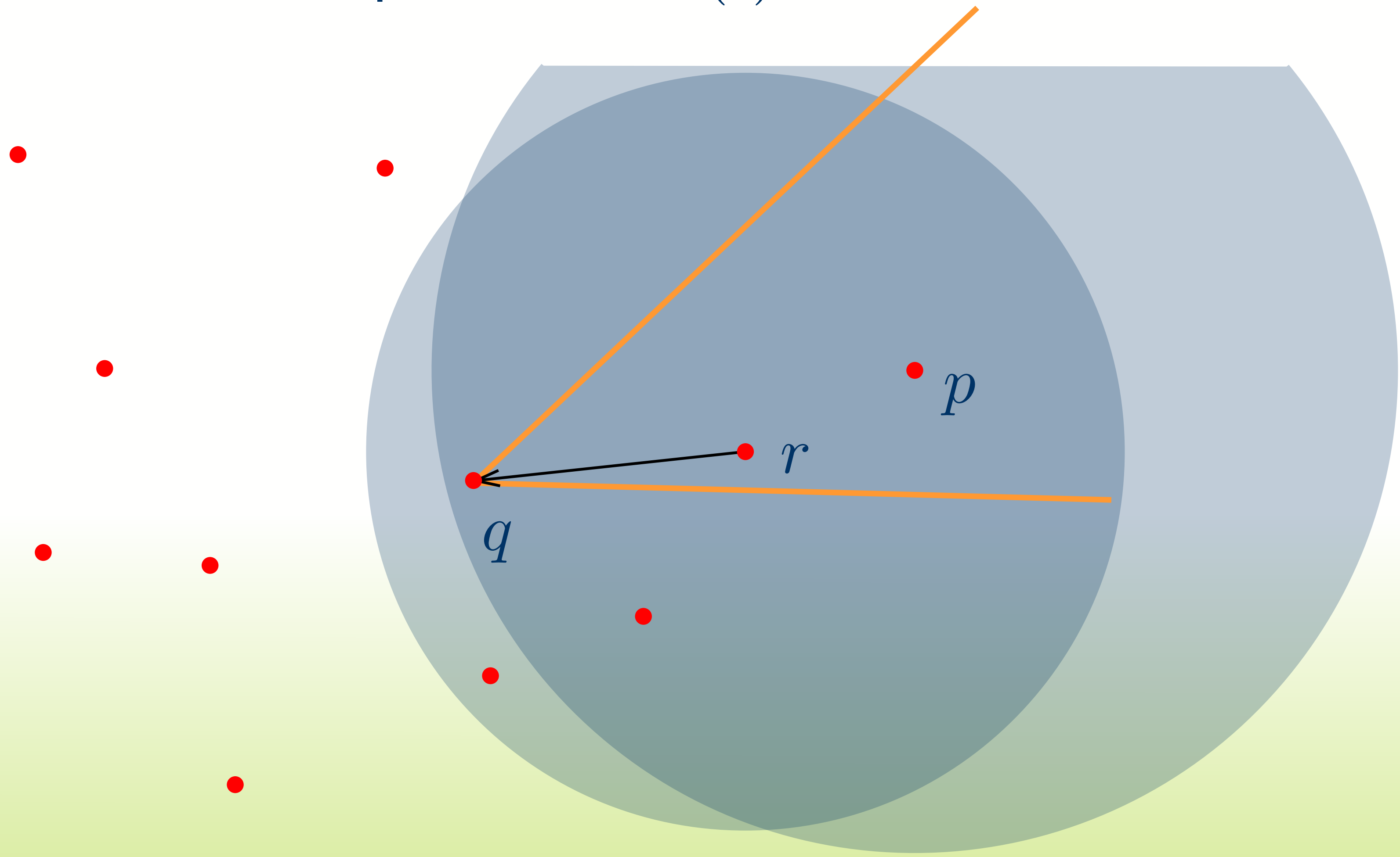
Modified Yao Graph

Given: point set $P \subset \mathbb{R}^2$, parameter $k(t)$



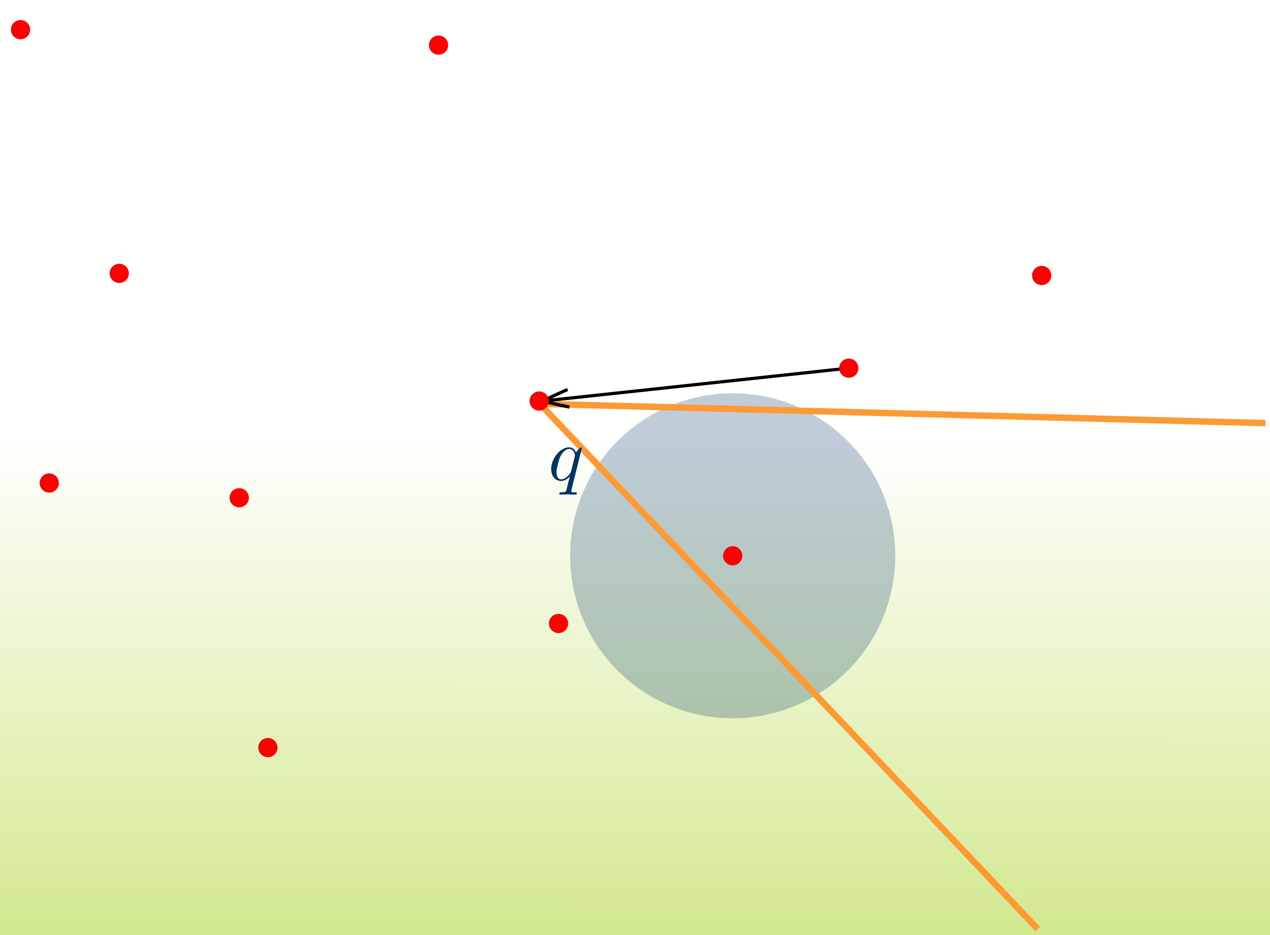
Modified Yao Graph

Given: point set $P \subset \mathbb{R}^2$, parameter $k(t)$



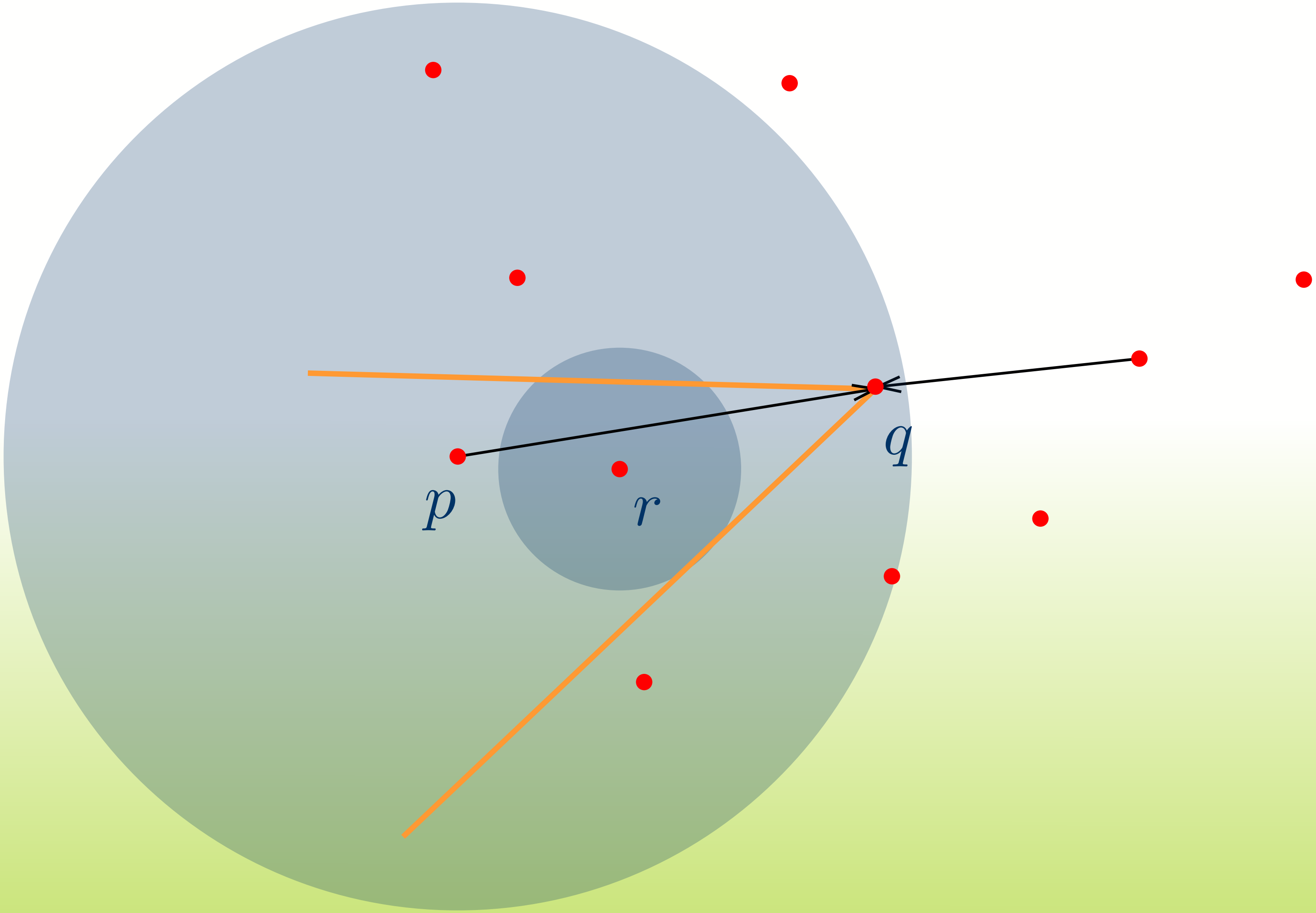
Modified Yao Graph

Given: point set $P \subset \mathbb{R}^2$, parameter $k(t)$



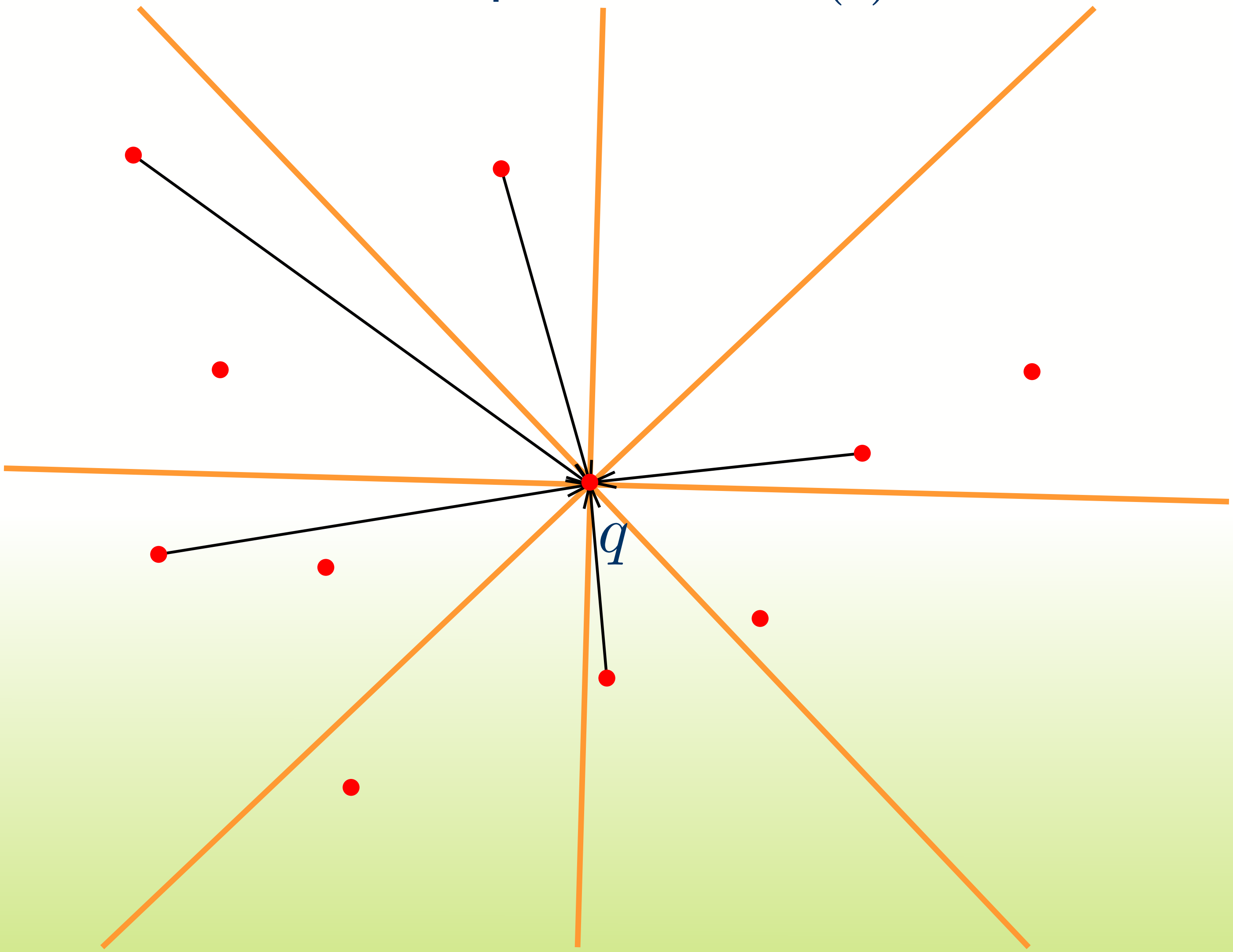
Modified Yao Graph

Given: point set $P \subset \mathbb{R}^2$, parameter $k(t)$



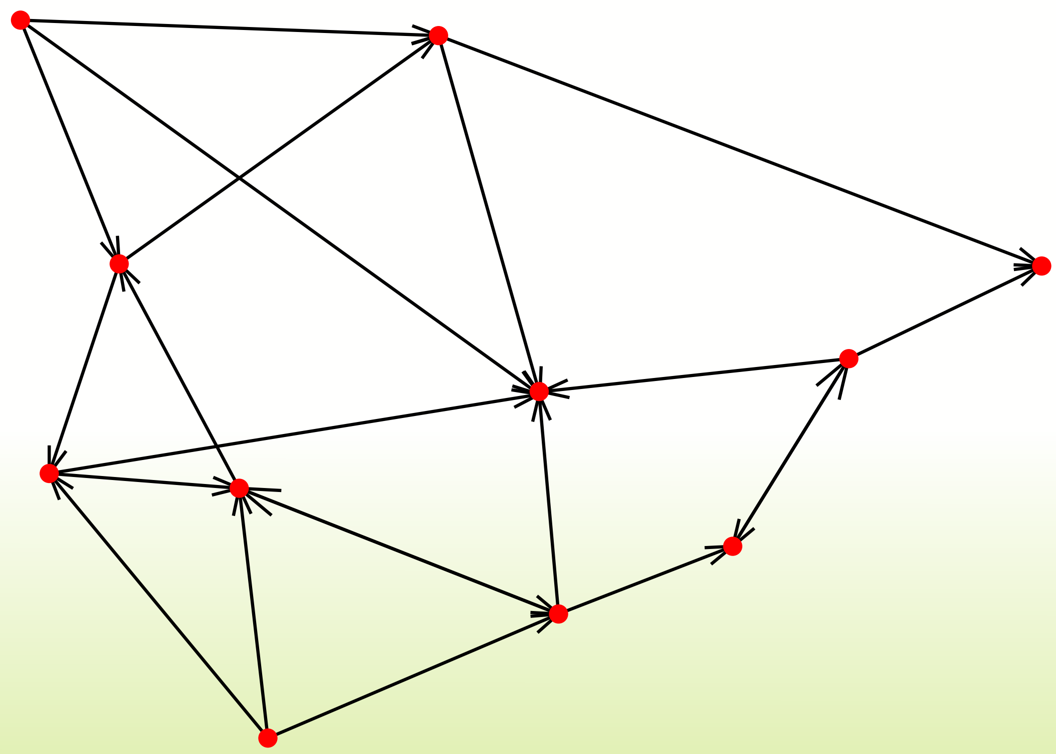
Modified Yao Graph

Given: point set $P \subset \mathbb{R}^2$, parameter $k(t)$



Modified Yao Graph

Given: point set $P \subset \mathbb{R}^2$, parameter $k(t)$

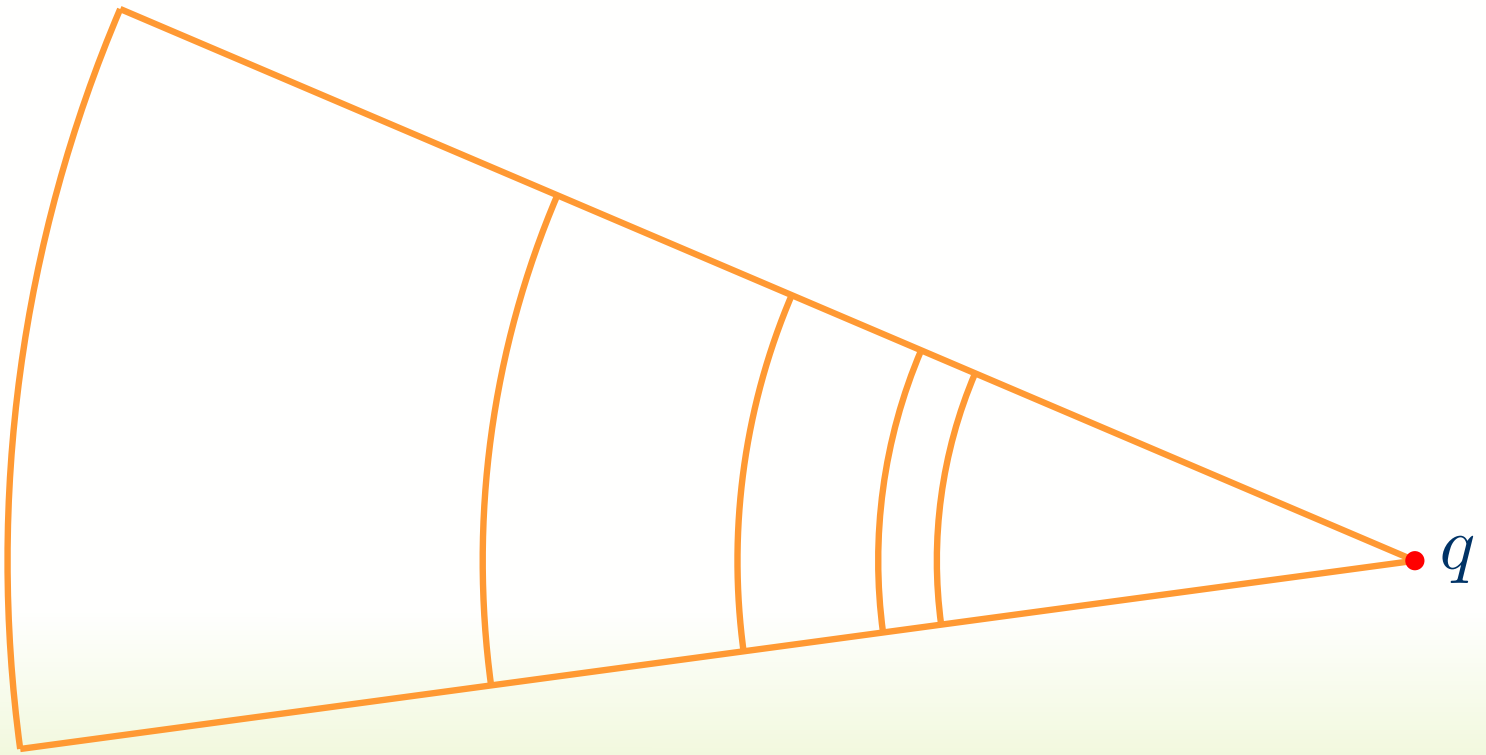


Results

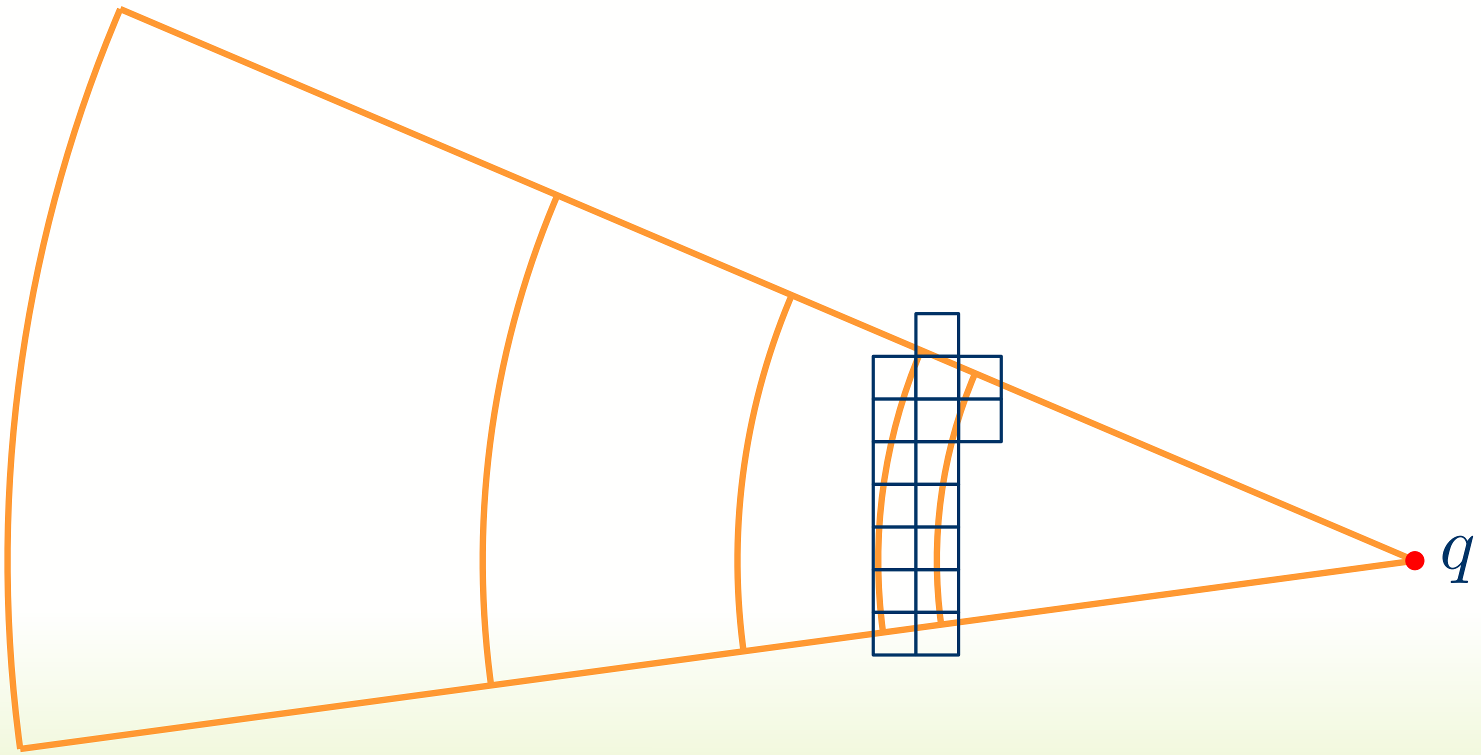
Theorem 1: Let $\boldsymbol{P} \subset \mathbb{R}^{\mathbf{2}}$ be a point set with radii and with **spread $\boldsymbol{\Phi}$**. Let G be the transmission graph of P. For any $t > 1$ we can compute a **t-spanner $\boldsymbol{H} \subseteq \boldsymbol{G}$** for G in **time $\boldsymbol{O(n(\log n + \log \Phi))}$**.

Theorem 2: Let $\boldsymbol{\Psi}$ be the **ratio of the largest and smallest radius** in P. We can compute a t-spanner $H \subseteq G$ in **time $\boldsymbol{O(n(\log n + \log \Psi))}$**.

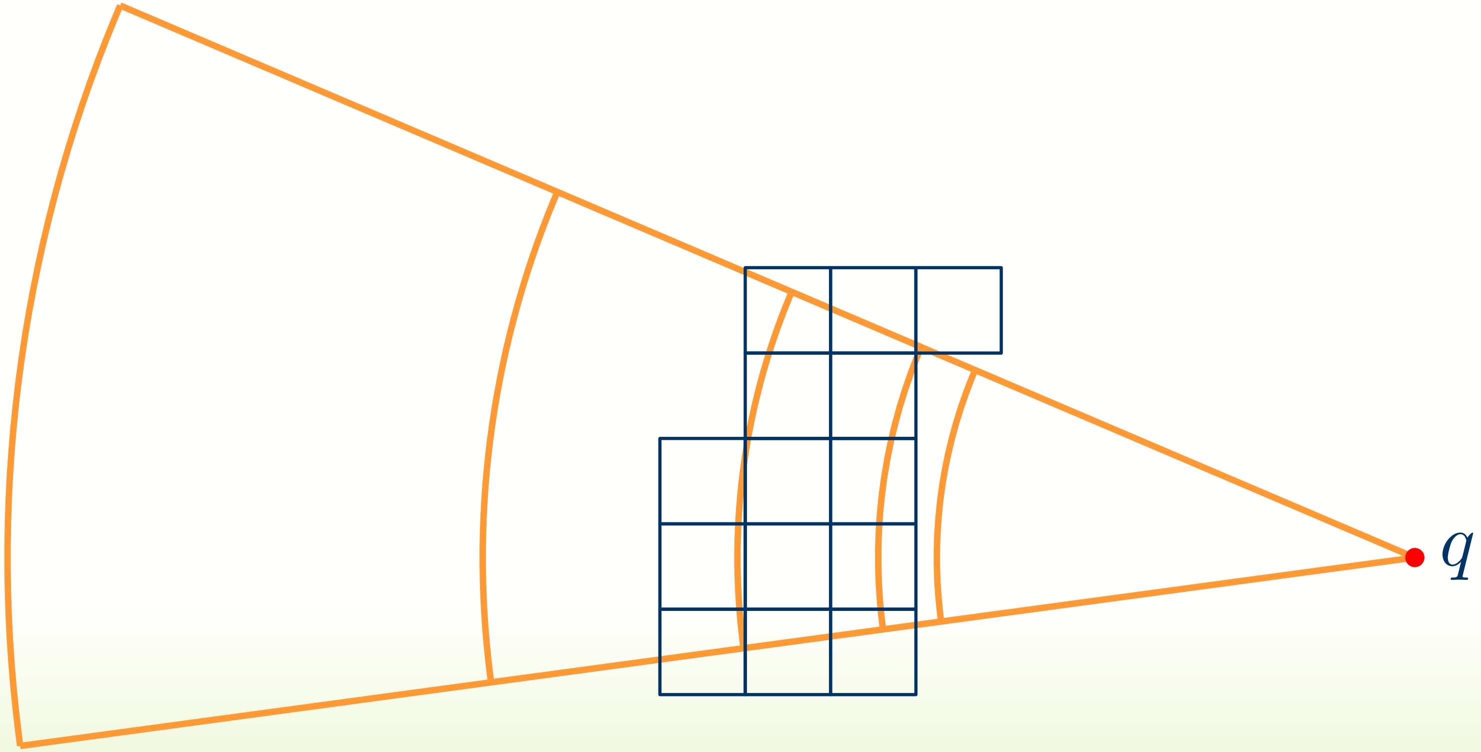
The Idea



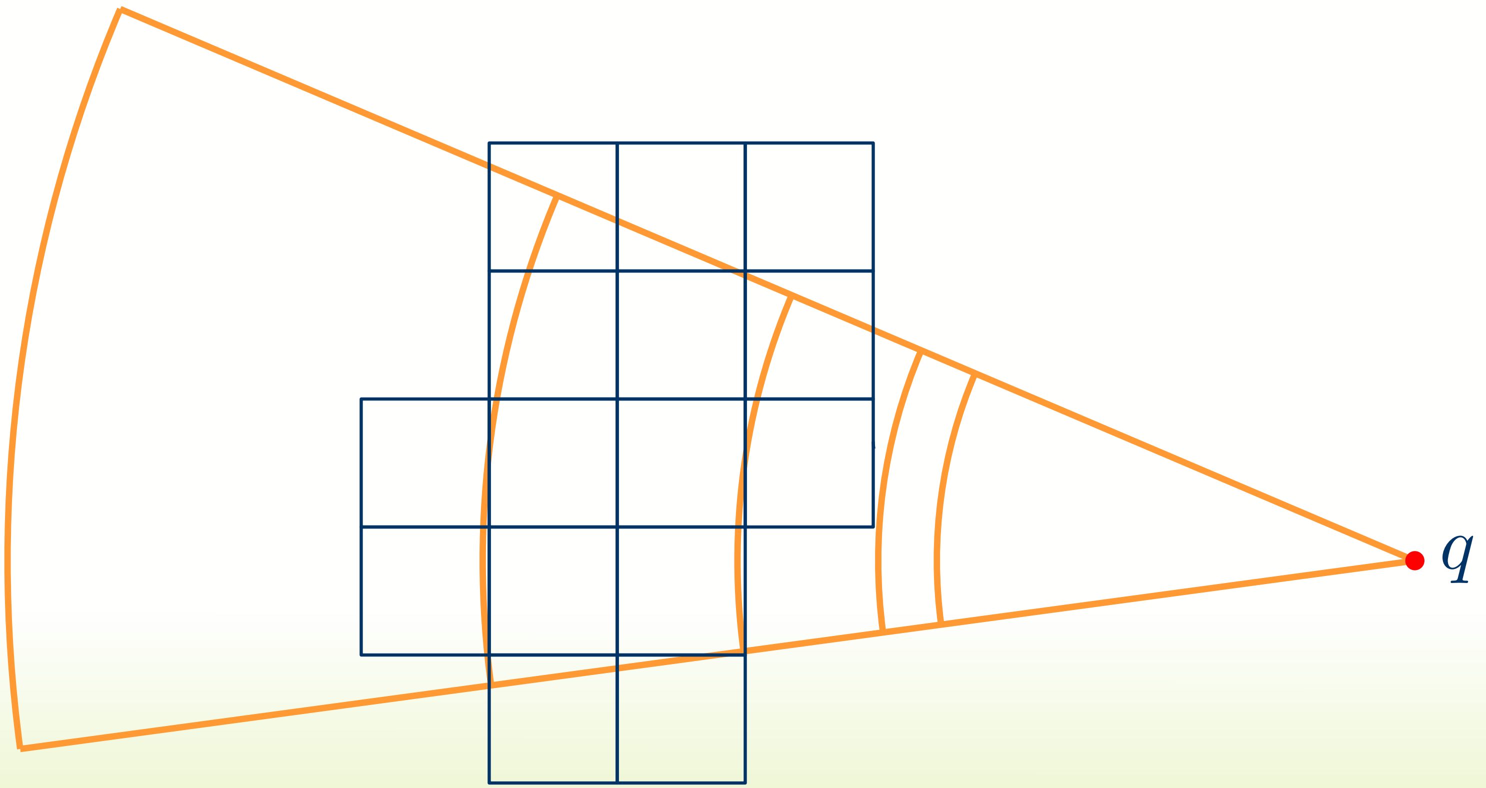
The Idea



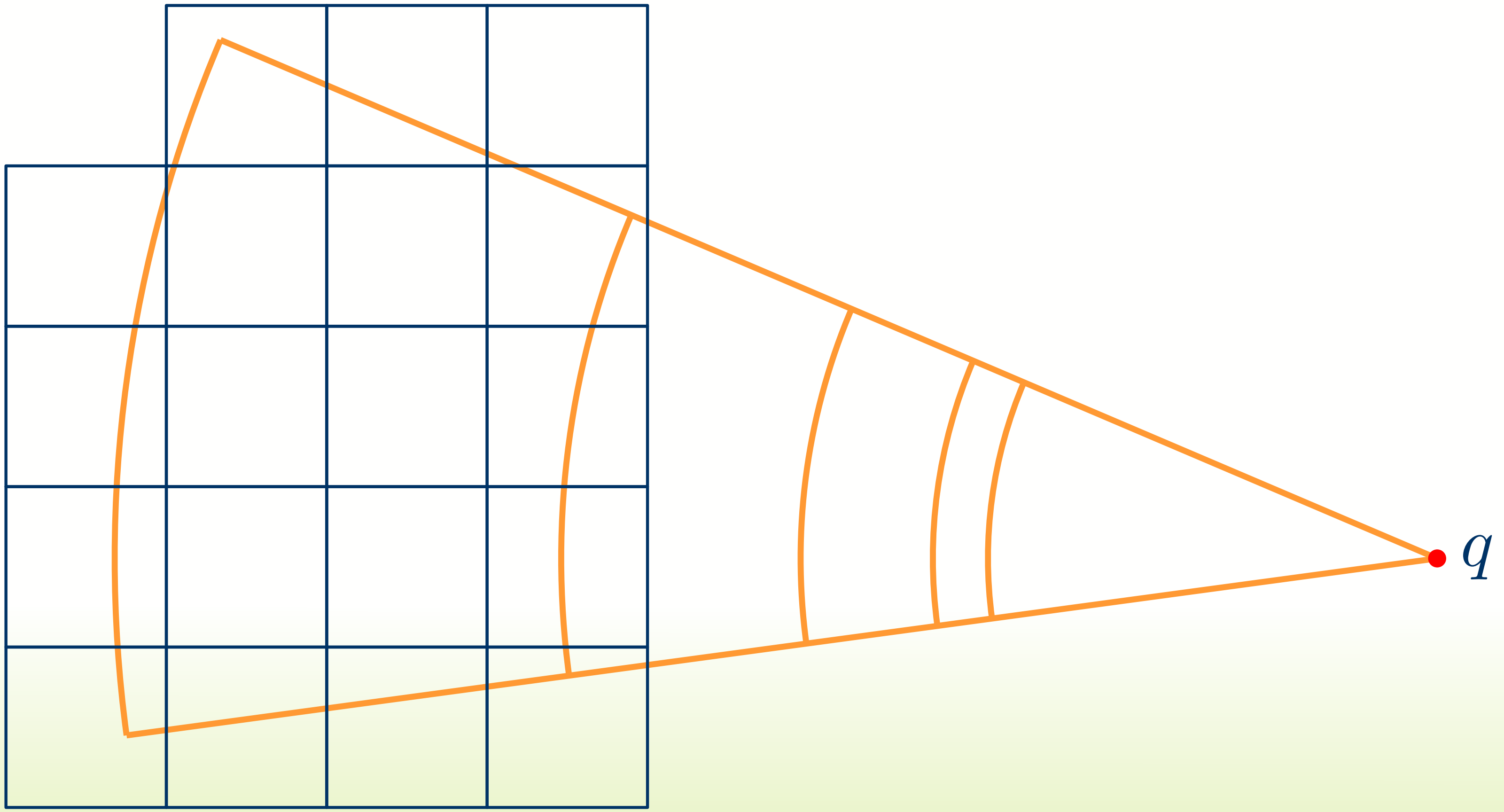
The Idea



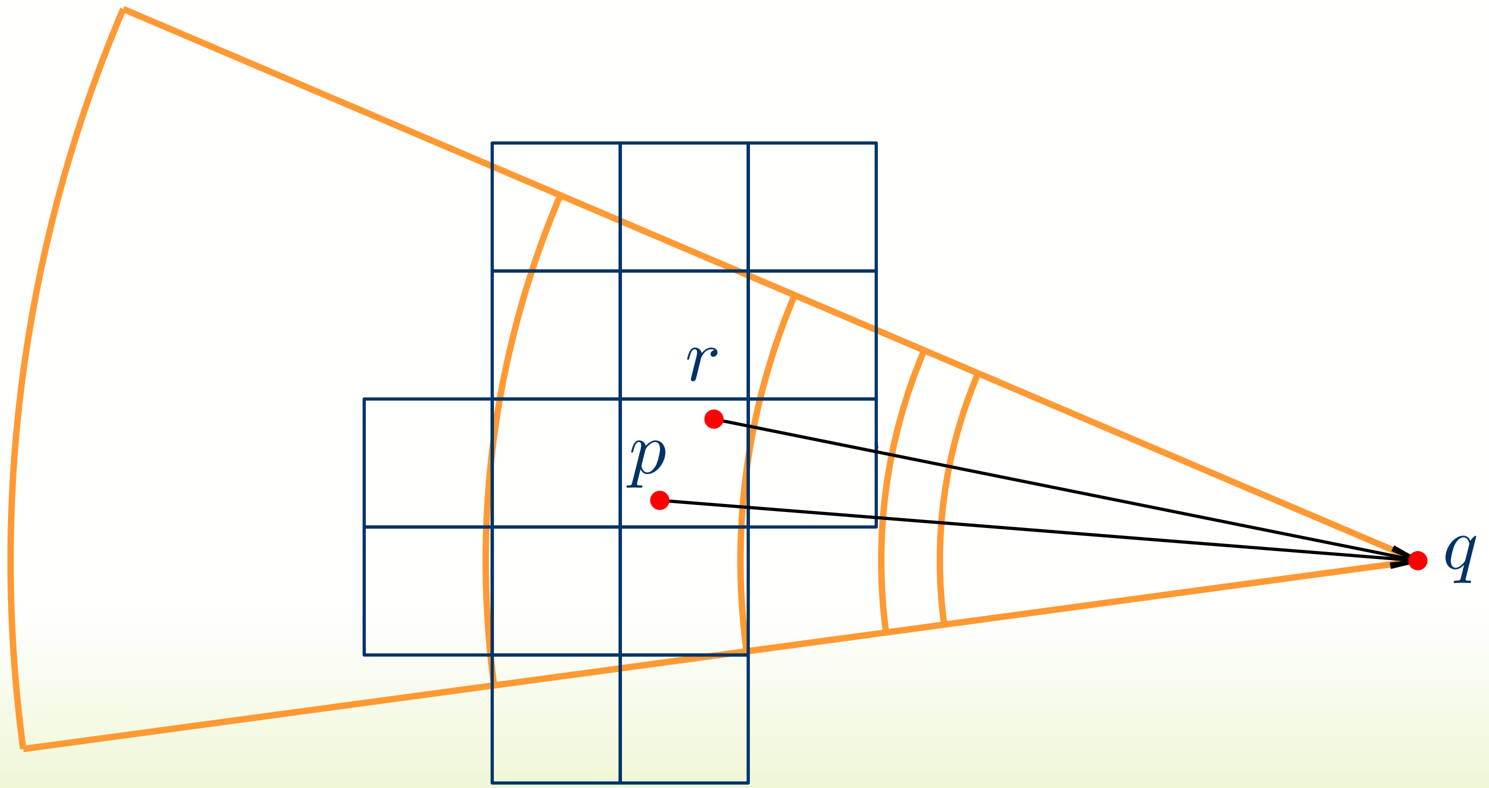
The Idea



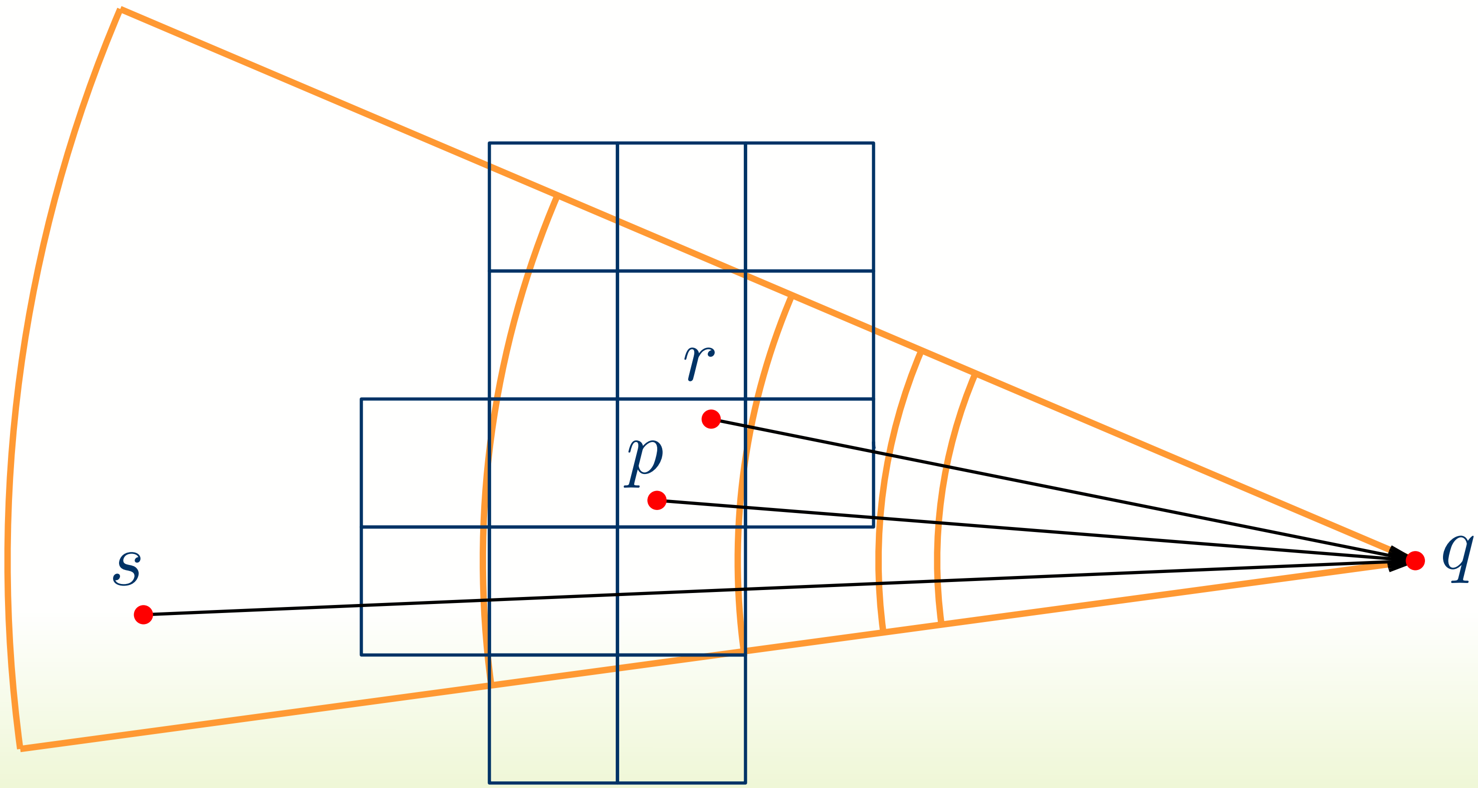
The Idea



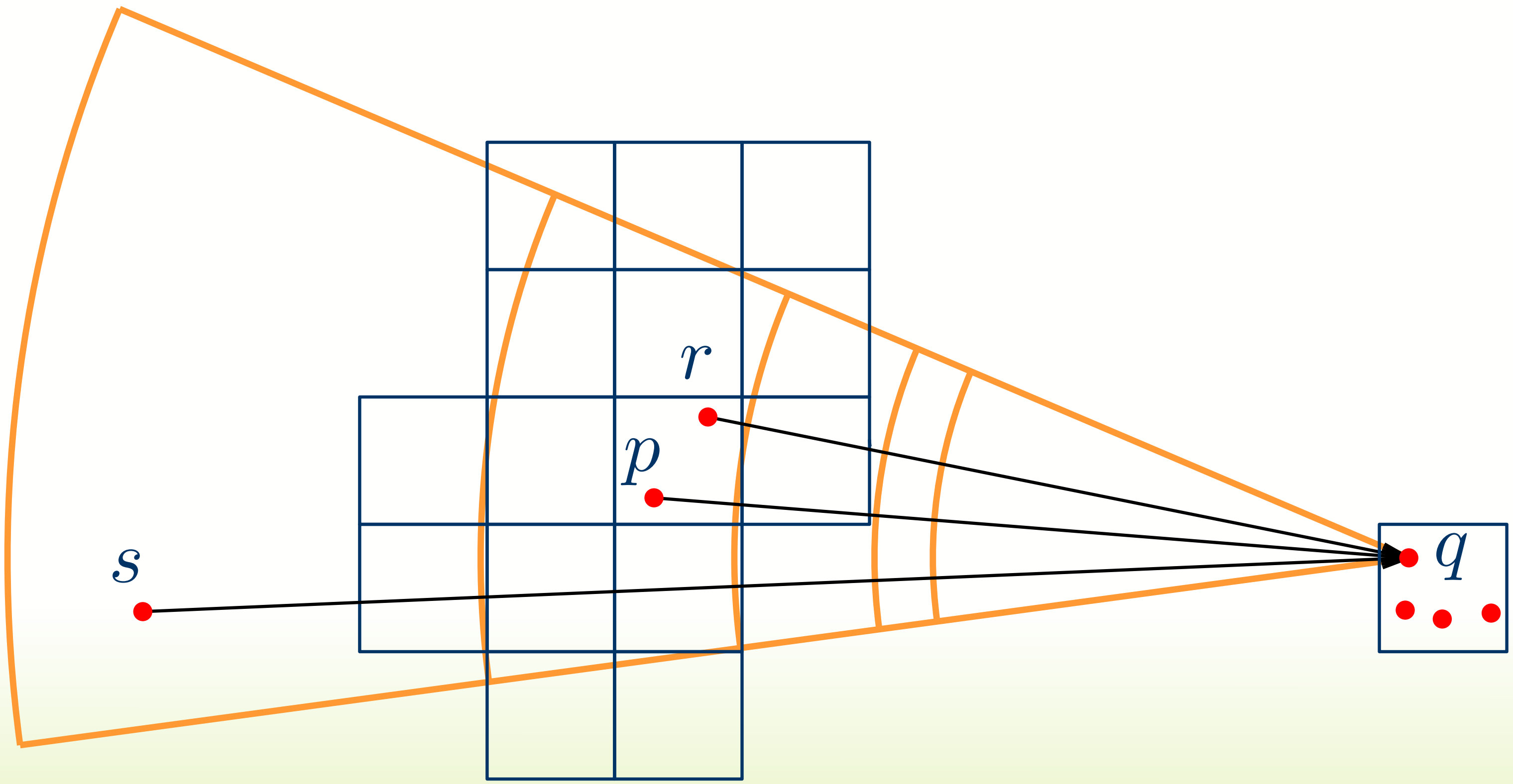
The Idea



The Idea

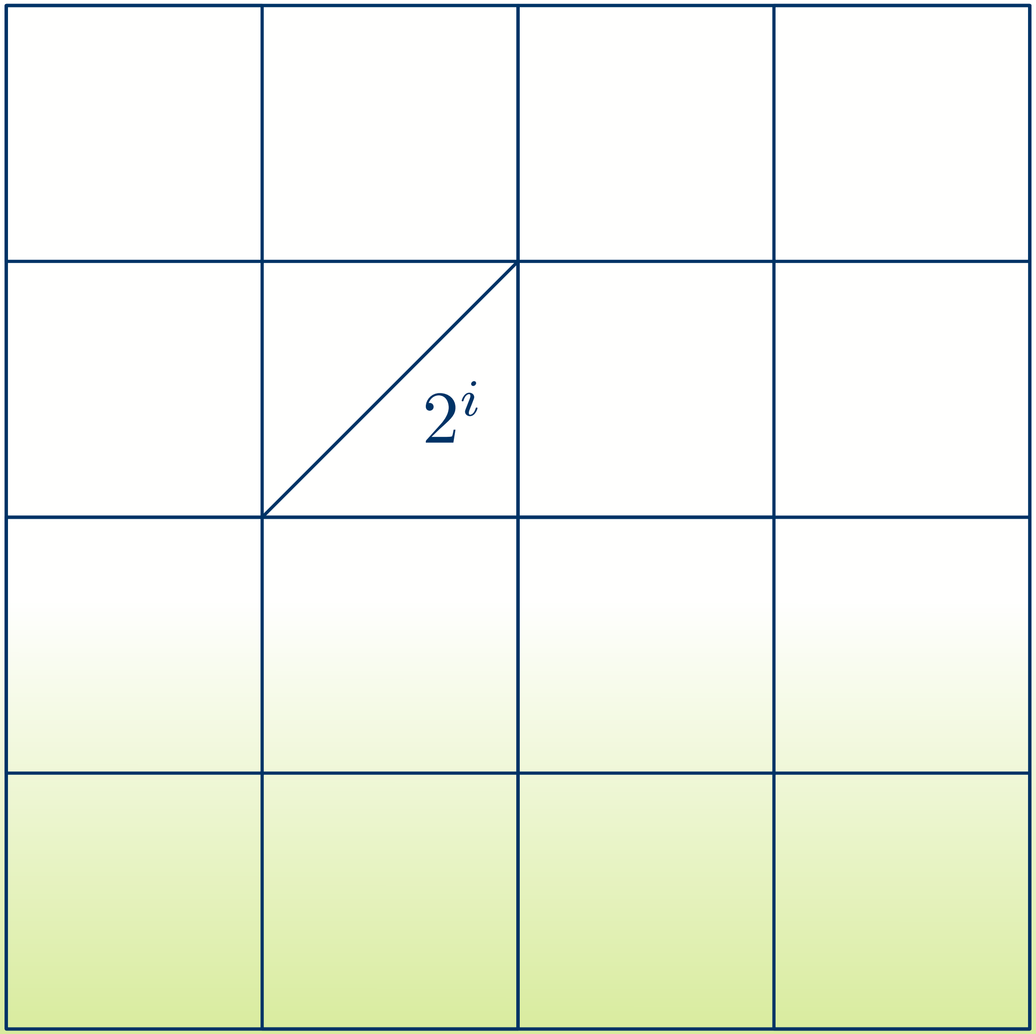


The Idea



A usefull definition

For $i \in \mathbb{N}$ let Q_i be the grid with diameter 2^i

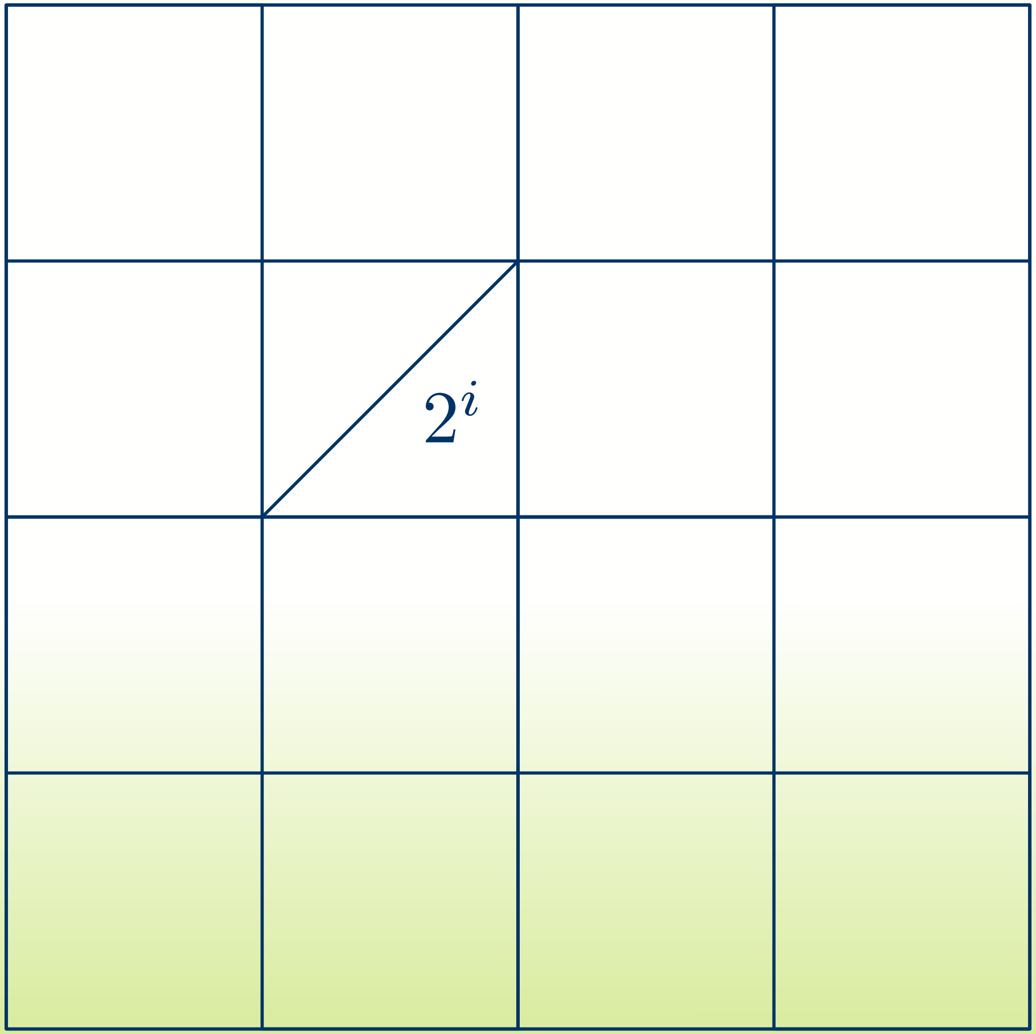


A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A c*-separated annulus decomposition* for G is

- finite set $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$,
- symmetric neighborhood relation $N \subseteq Q \times Q$,
- sets $R_\sigma \subseteq P \cap \sigma$ for $\sigma \in Q$



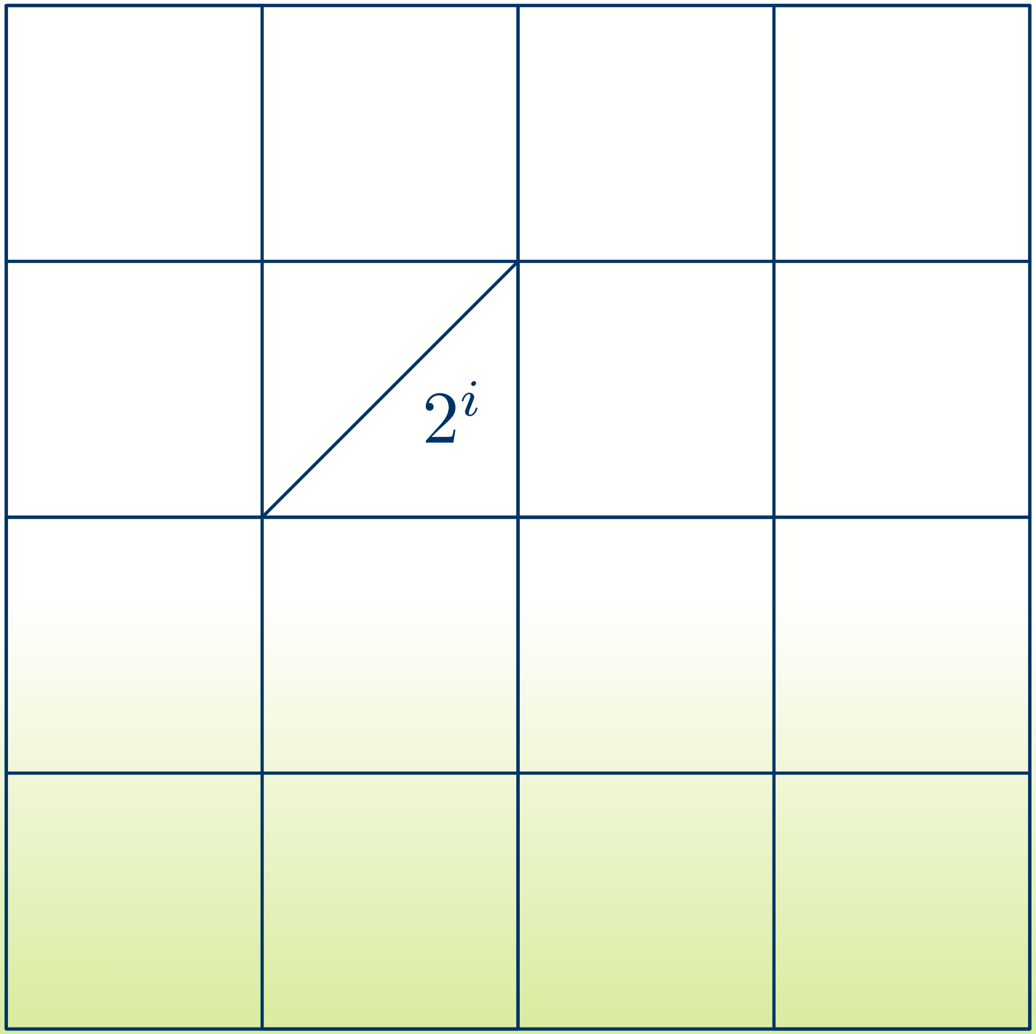
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A c-*separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



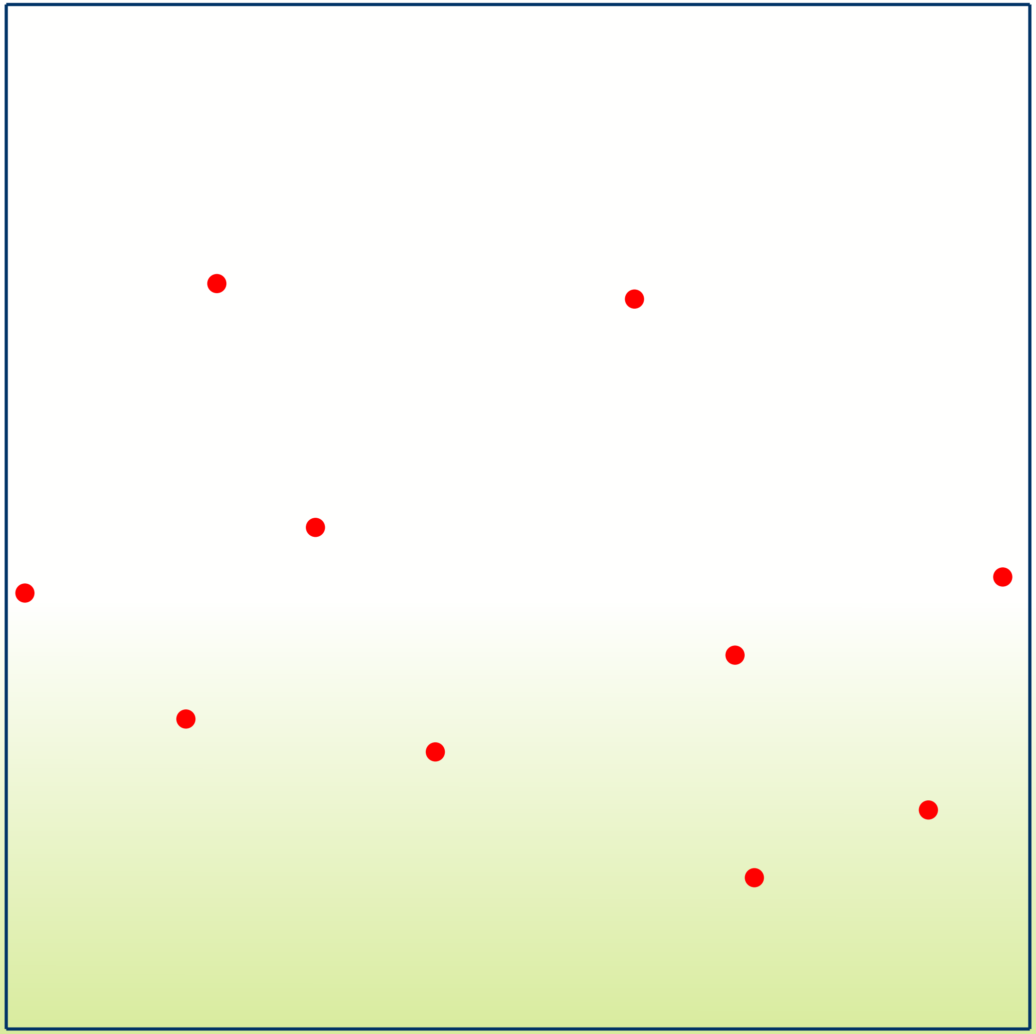
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



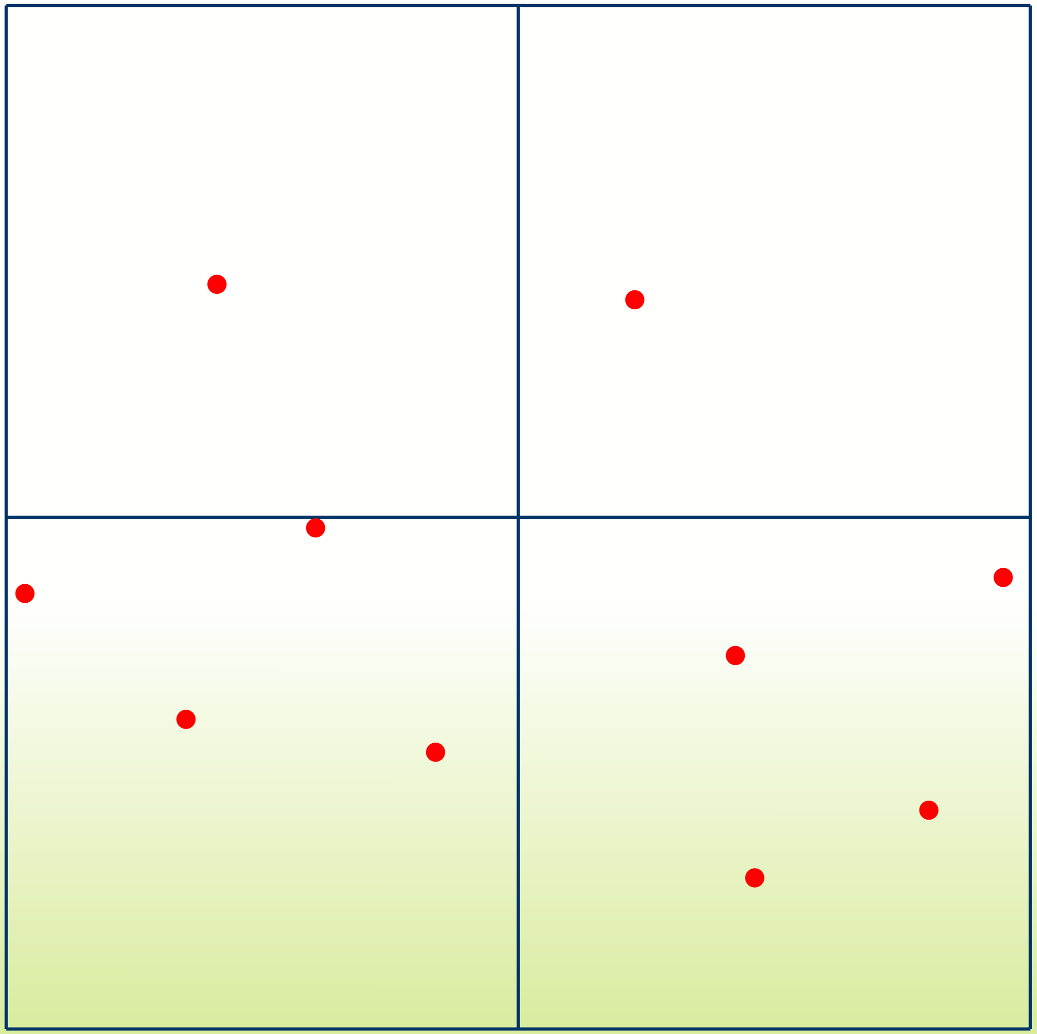
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



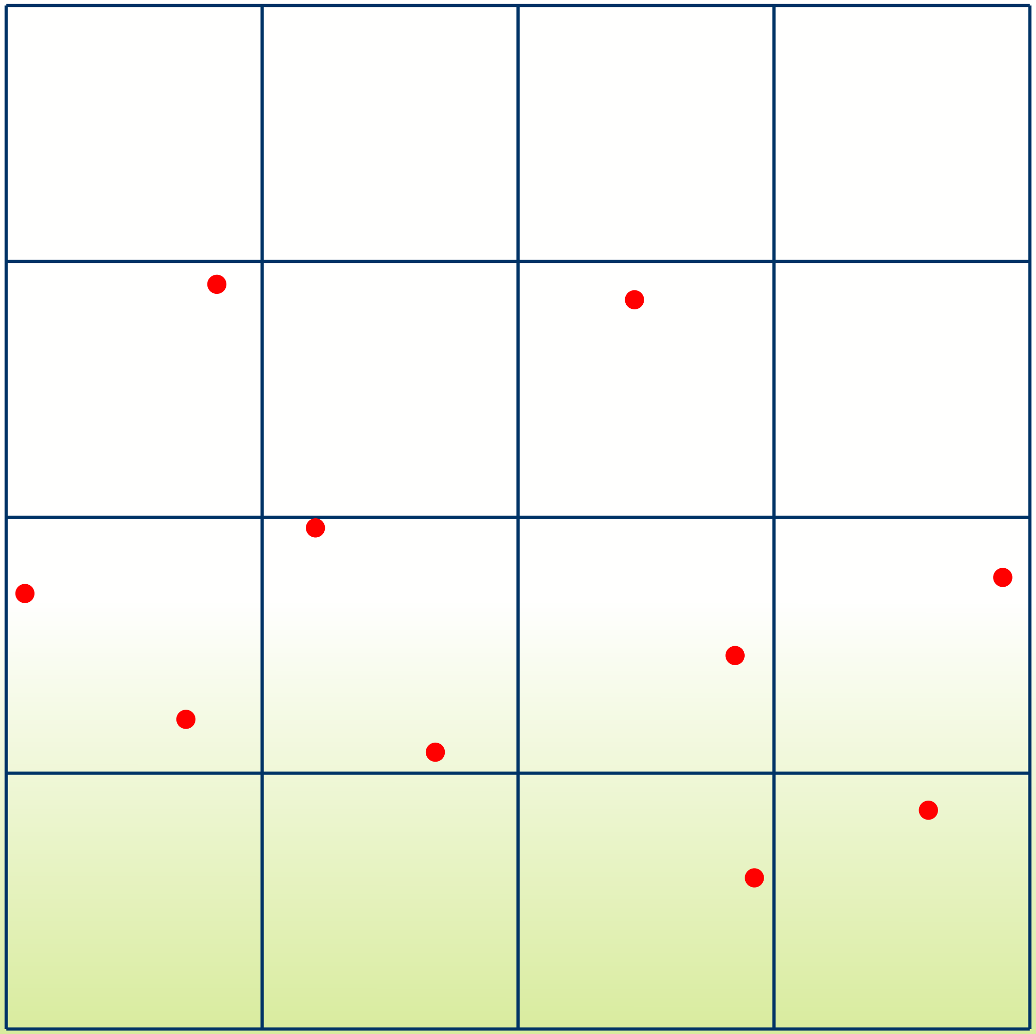
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A c*-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



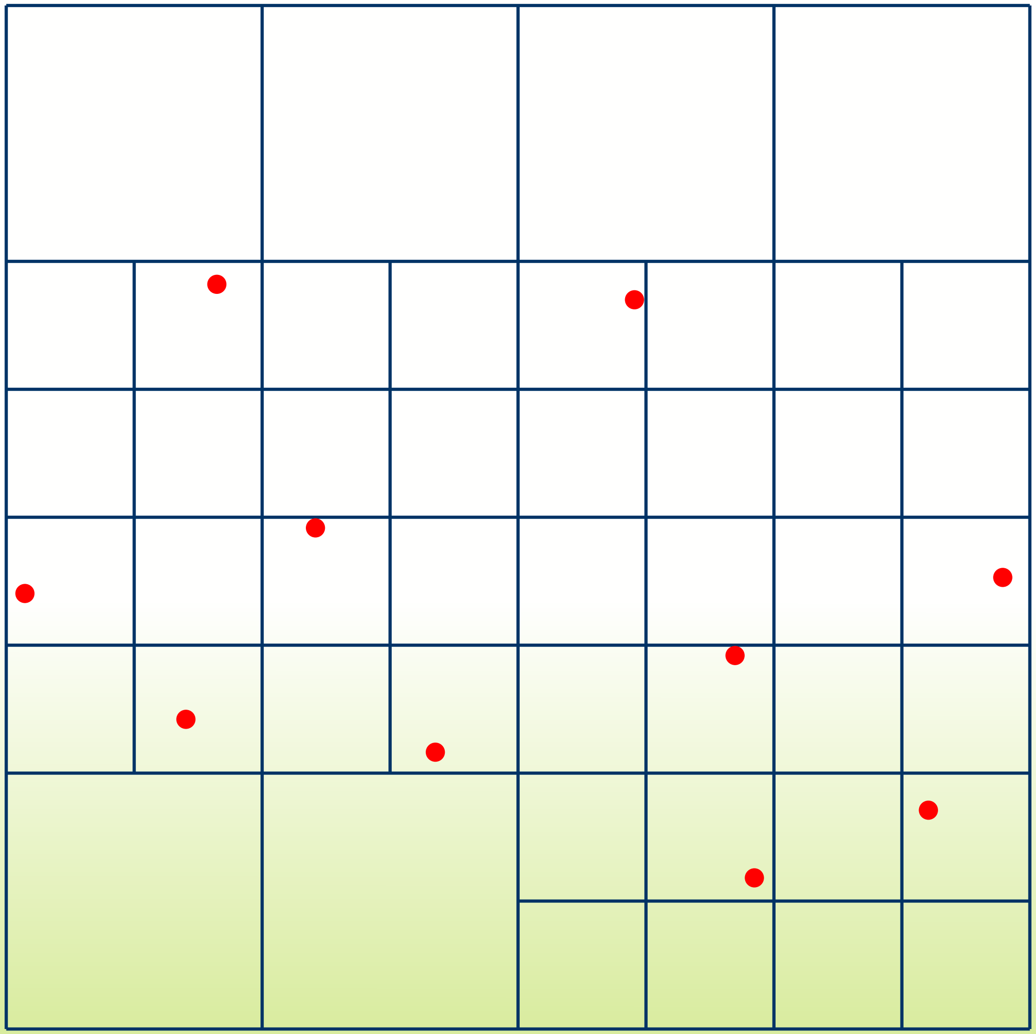
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



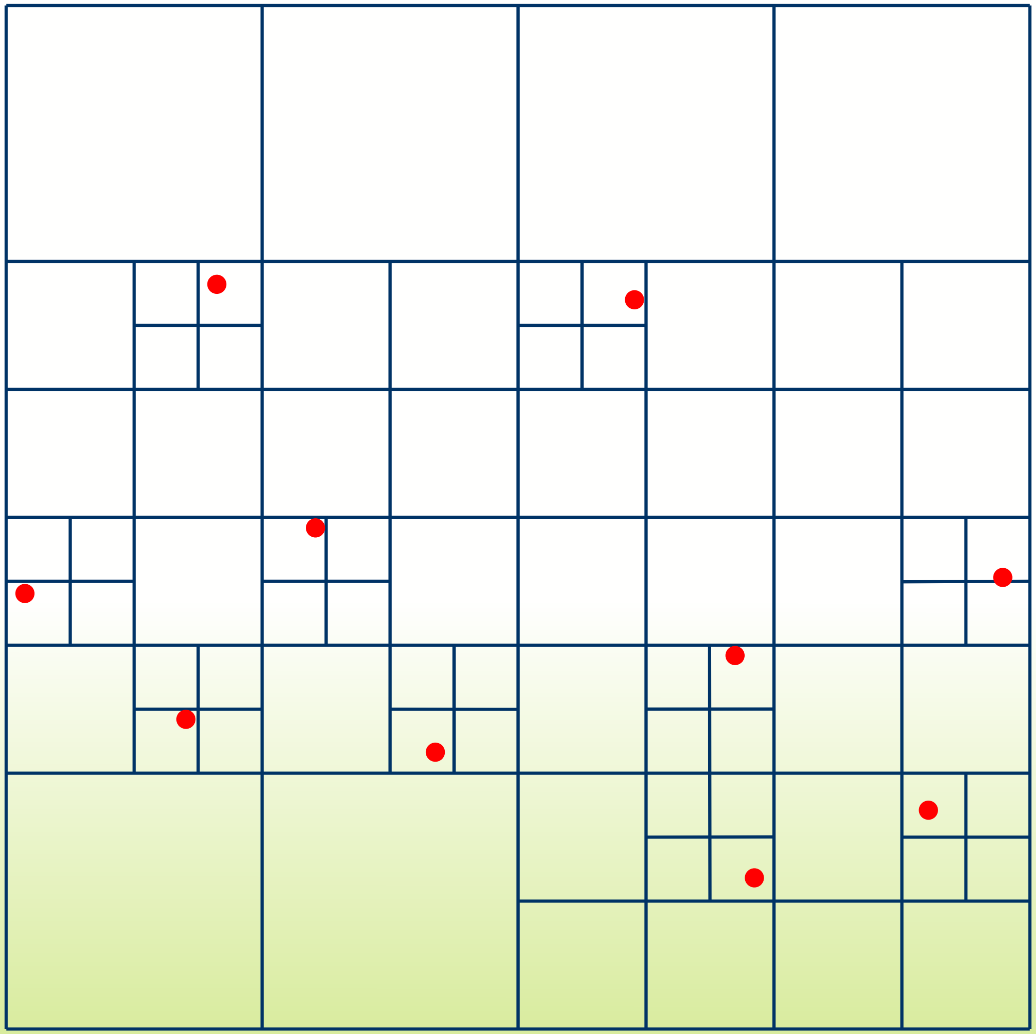
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A c-*separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



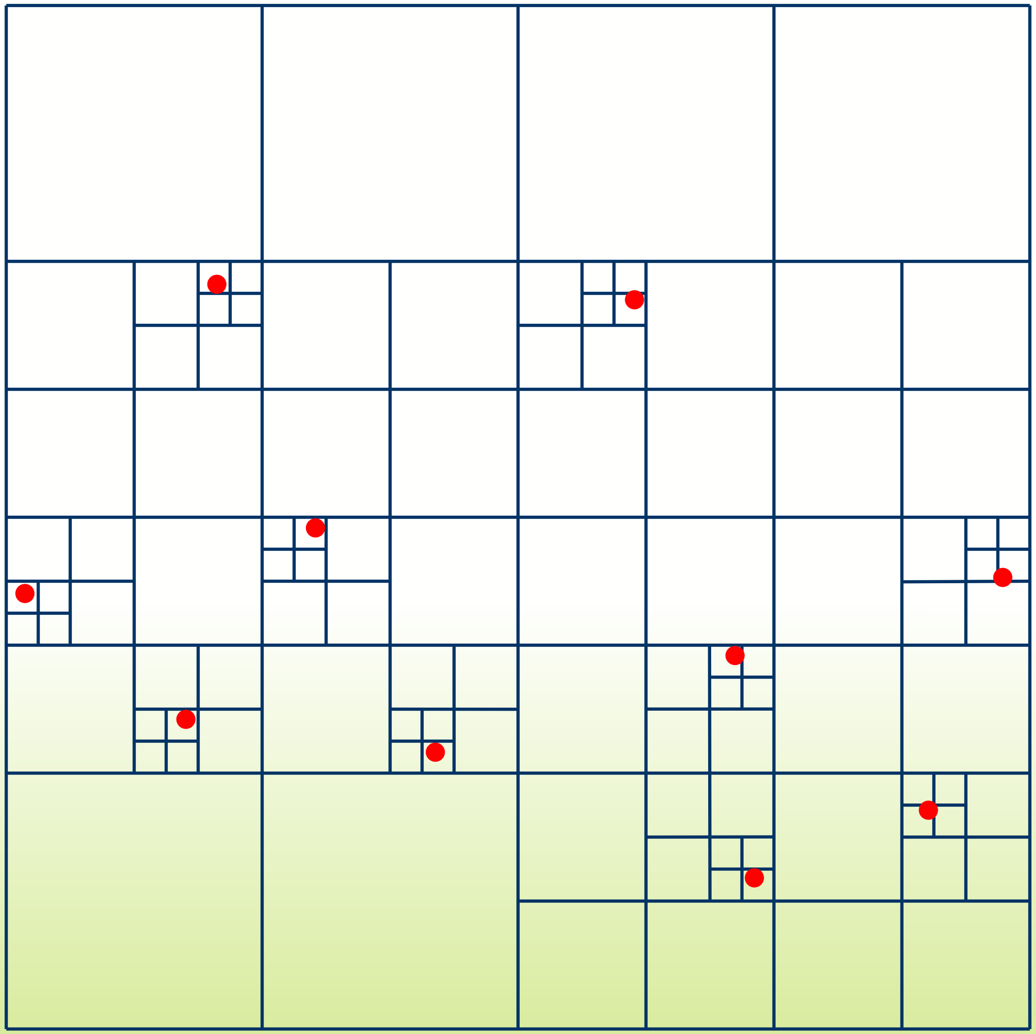
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A c-*separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



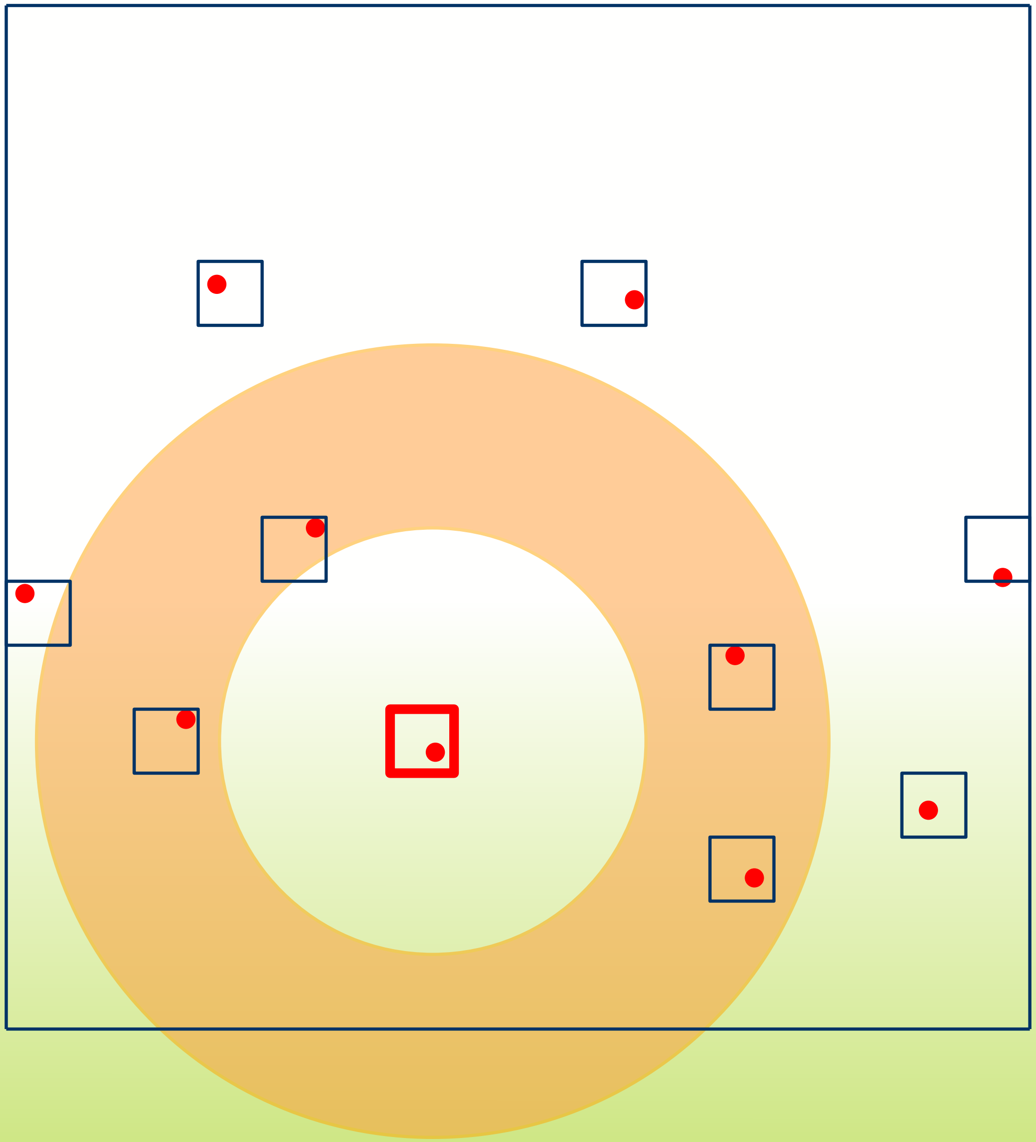
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A c-*separated annulus decomposition* for G is
$\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$,
$R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



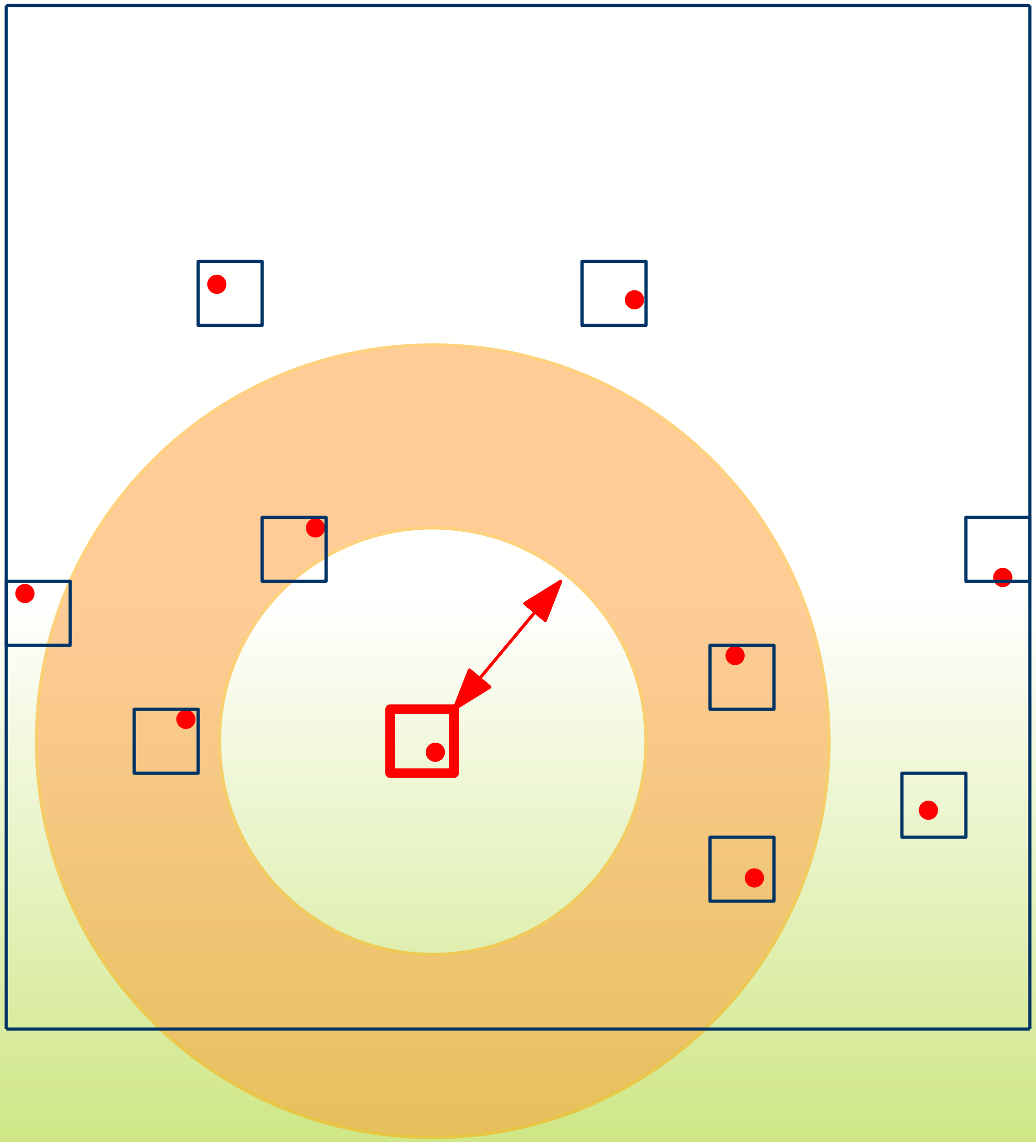
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



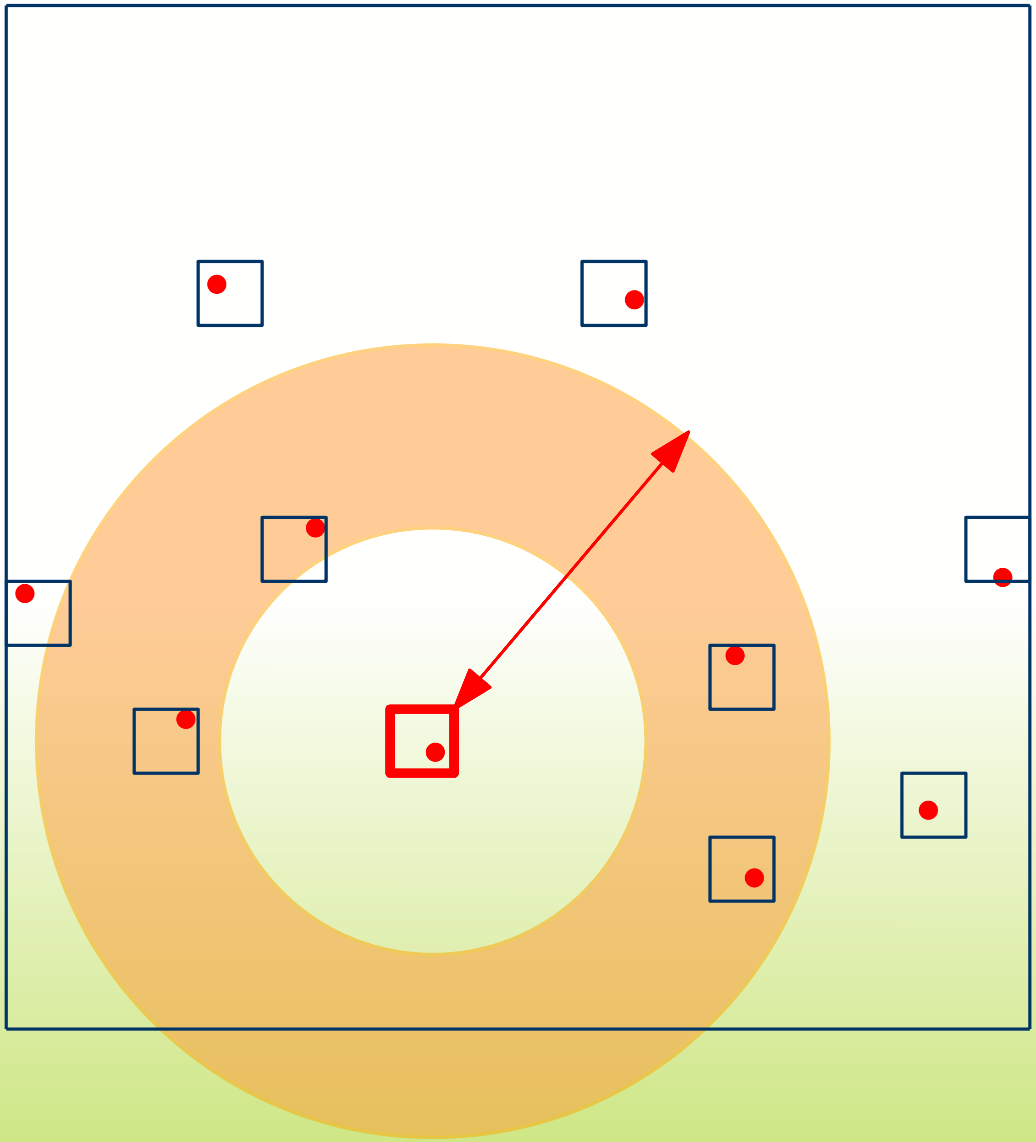
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



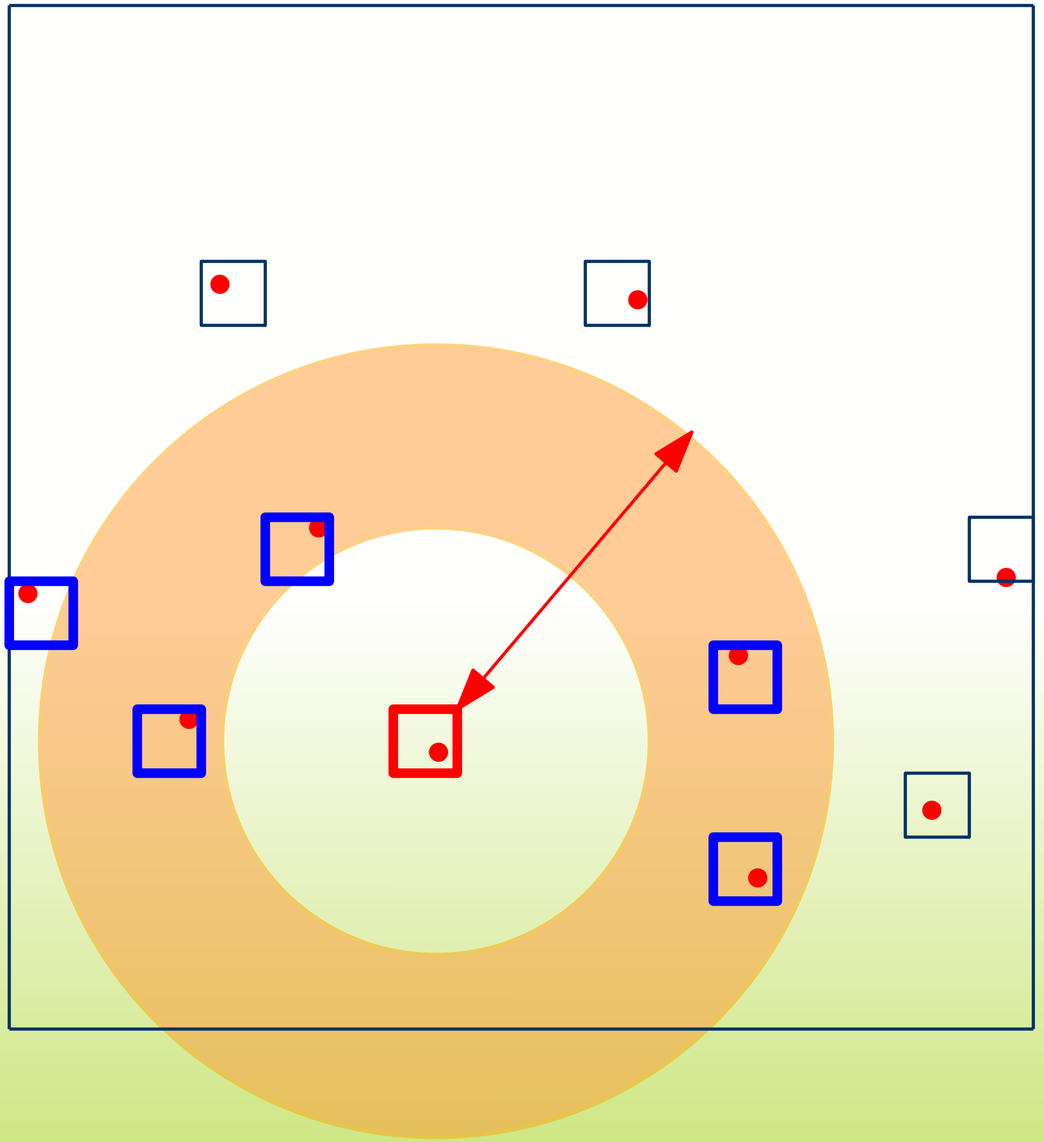
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



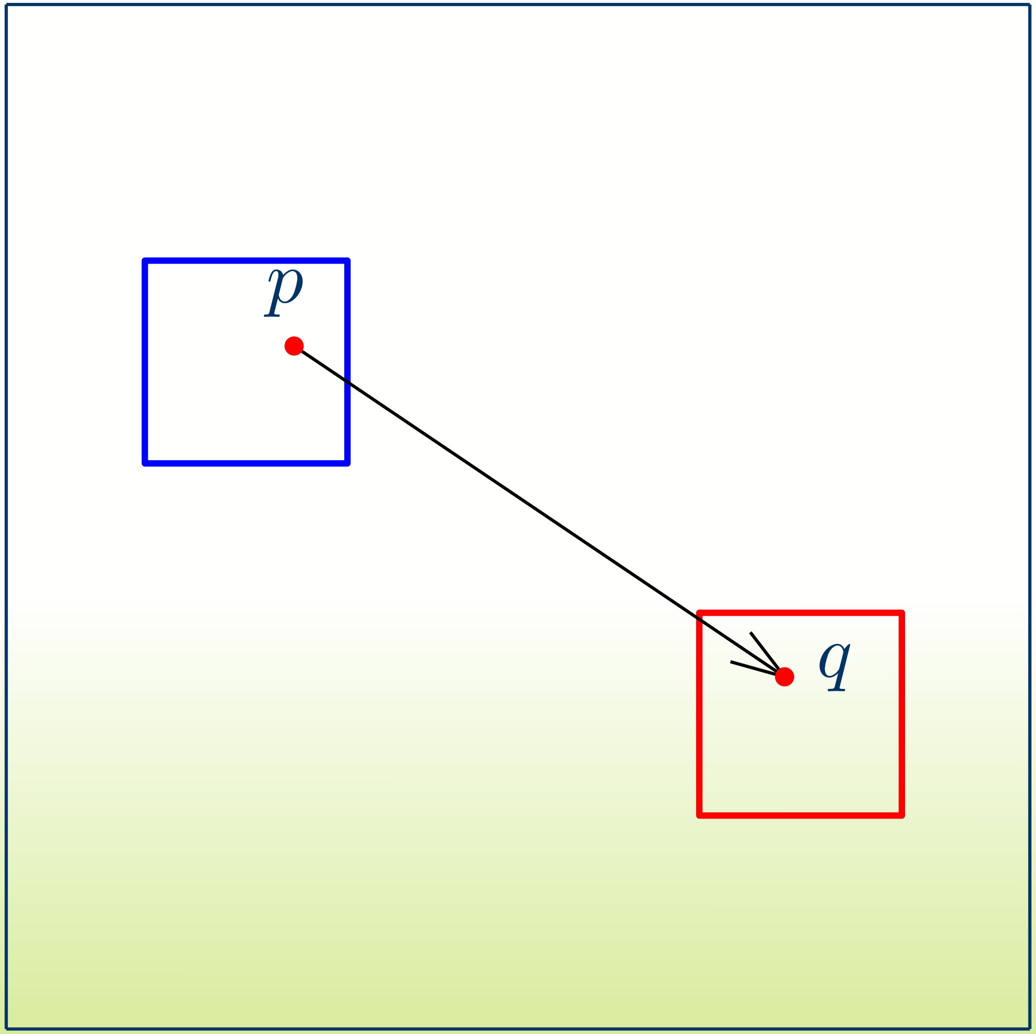
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A c*-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



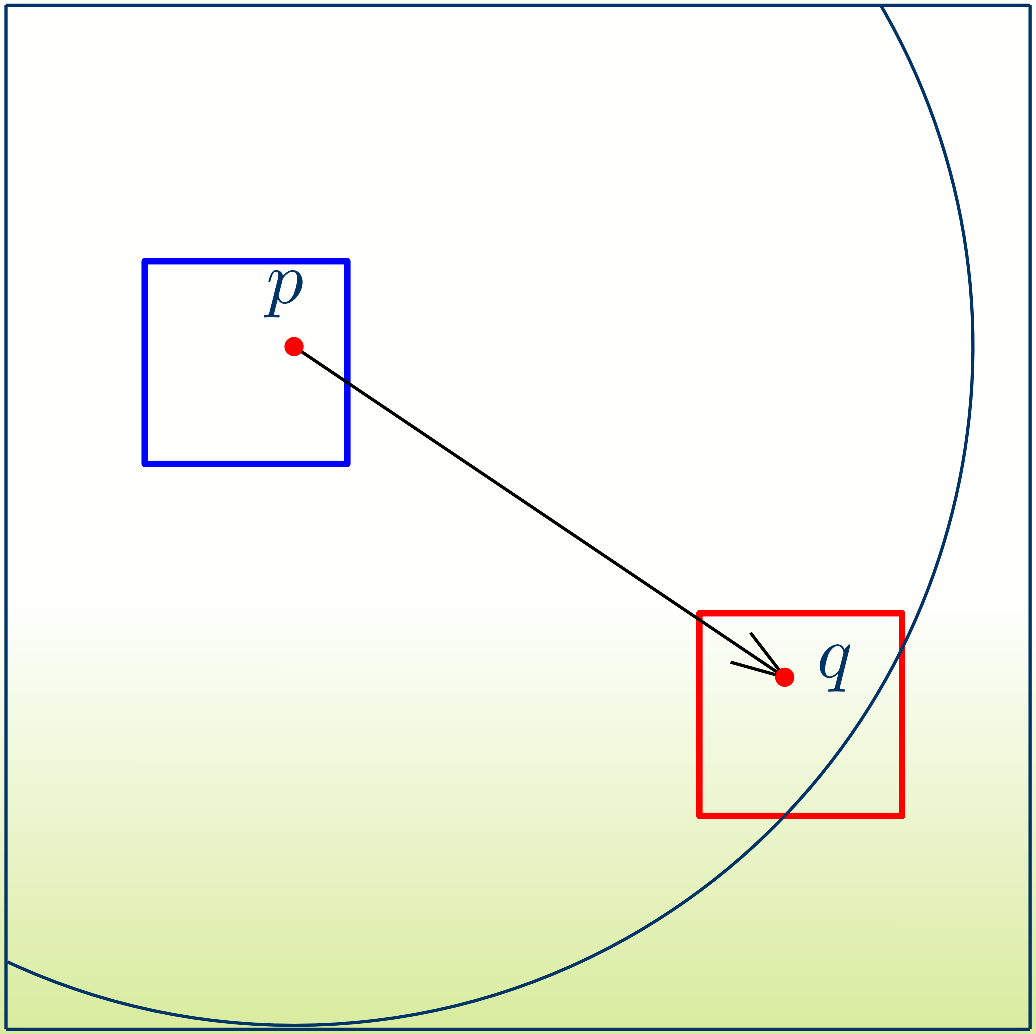
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is
$\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$,
$R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

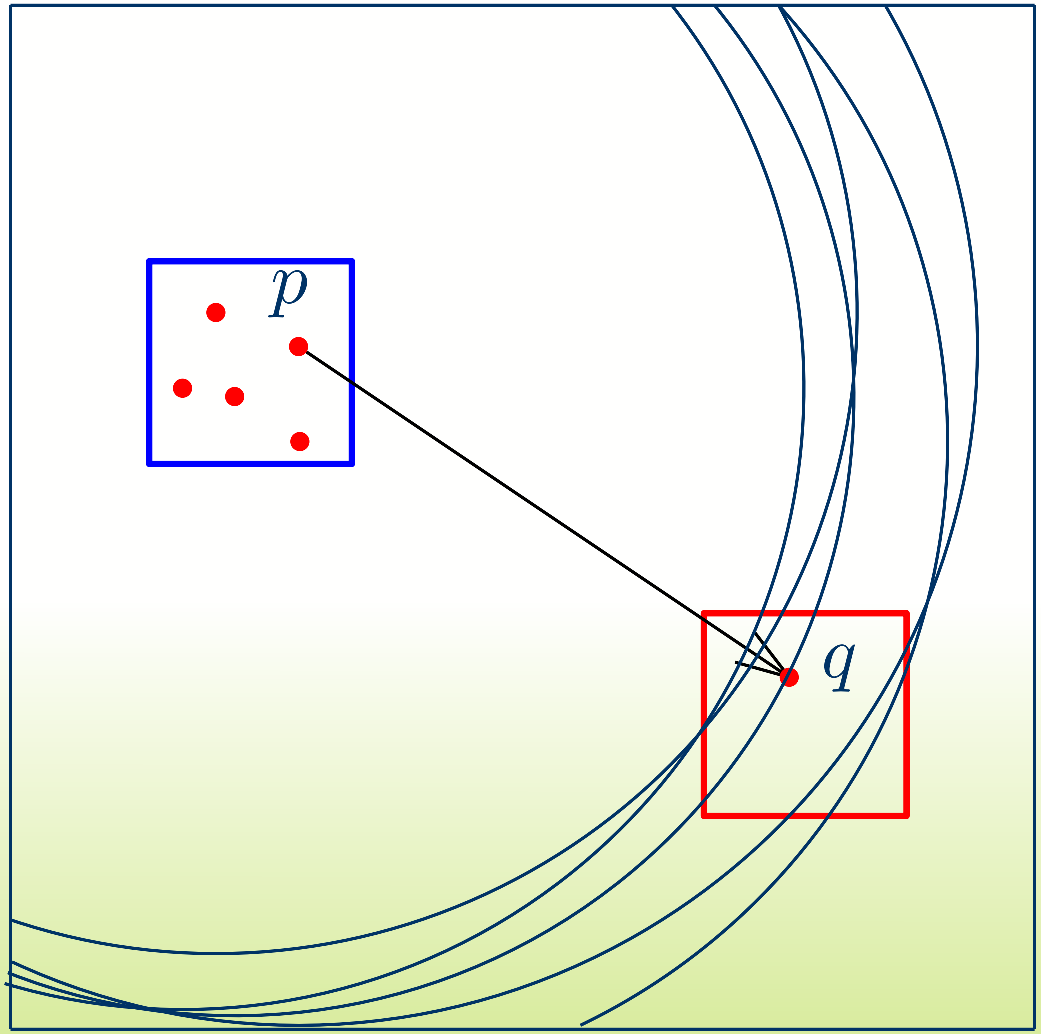
1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$



so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$

$R_{\sigma'} :=$ all points that might intersect $\sigma \in N(\sigma')$

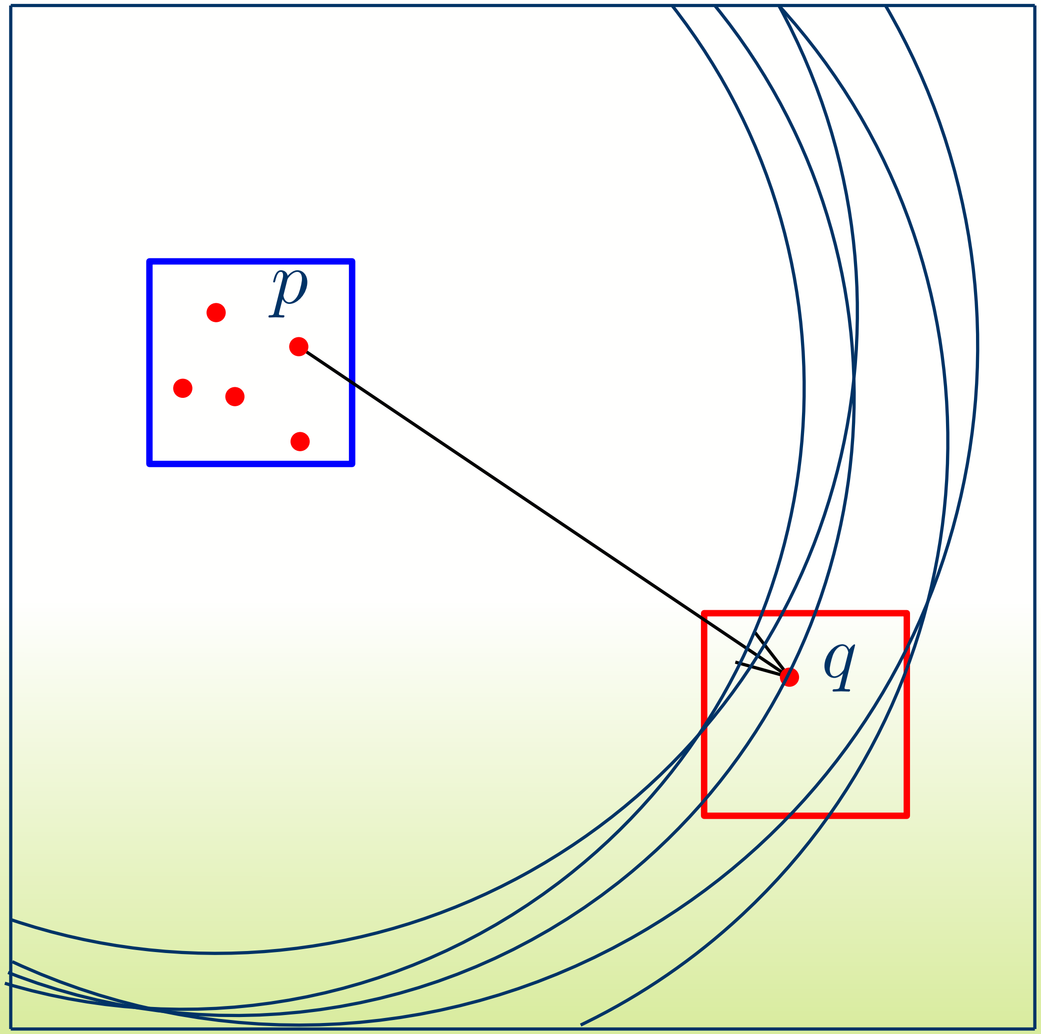
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



$R_{\sigma'} :=$ all points that might intersect $\sigma \in N(\sigma')$

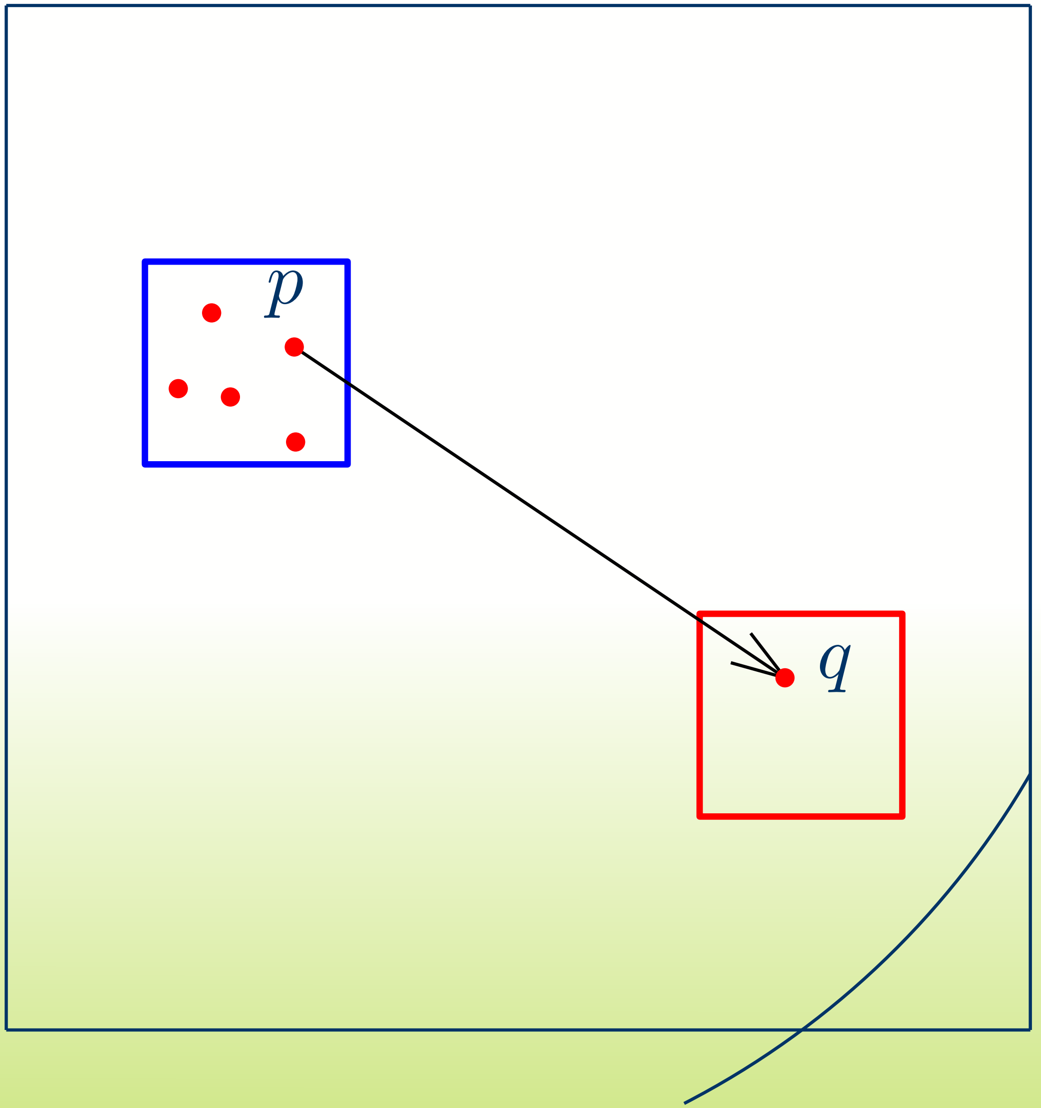
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is $\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$, $R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



$R_{\sigma'} :=$ all points that might intersect $\sigma \in N(\sigma')$

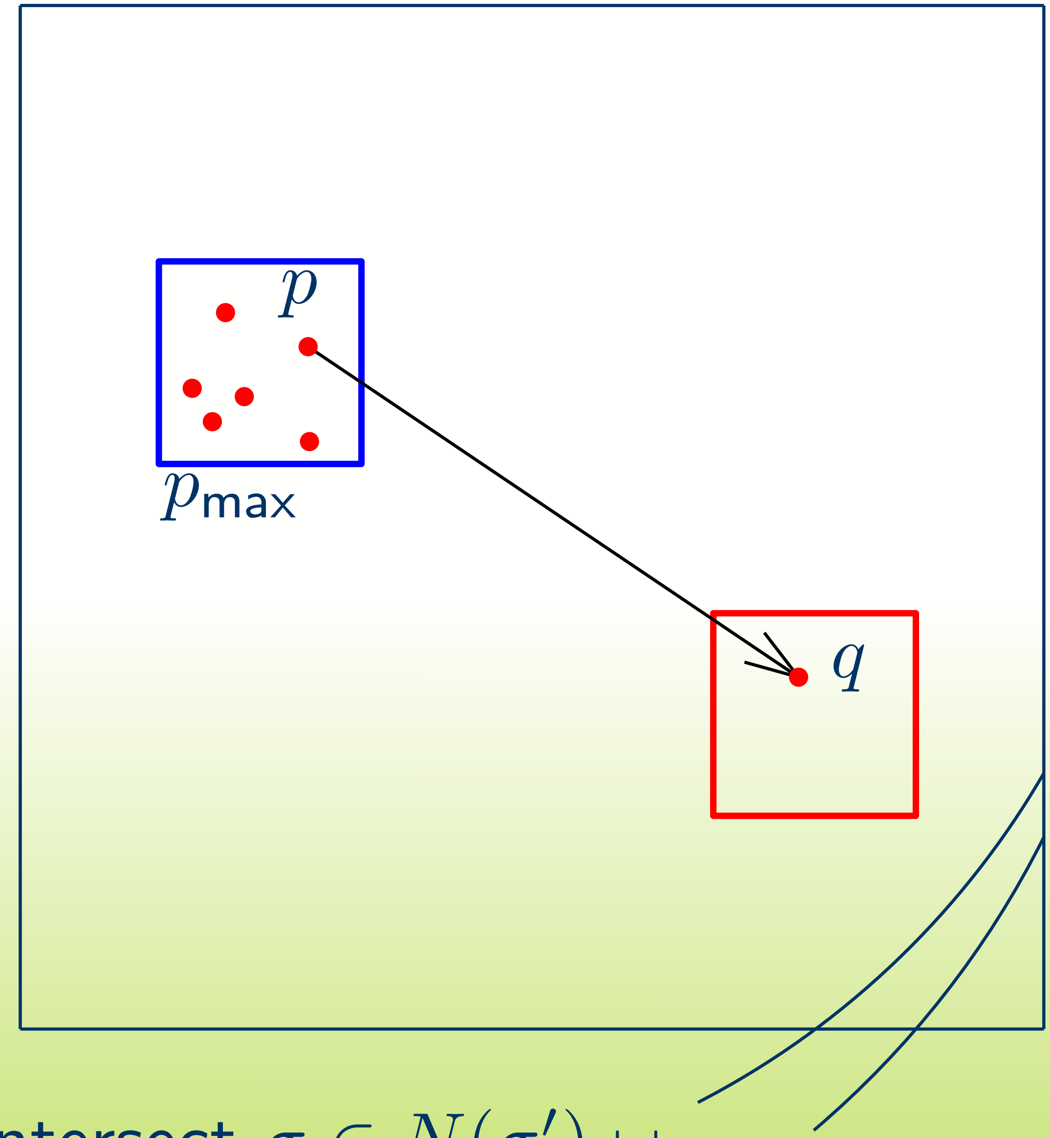
A usefull definition

For $i \in \mathbb{N}$ let $\mathcal{Q}_i$ be the grid with diameter 2^i

A *c-separated annulus decomposition* for G is
$\mathcal{Q} \subset \bigcup_{i=0}^{\infty} \mathcal{Q}_i$, $N \subseteq Q \times Q$,
$R_\sigma \subseteq P \cap \sigma$

so that for all $(\sigma, \sigma') \in N$,

1. $\text{diam}(\sigma) = \text{diam}(\sigma')$
2. $d(\sigma, \sigma') \in [(c-2)\text{diam}(\sigma), 2c\text{diam}(\sigma))$
3. for every edge $\overrightarrow{pq}$ of G, there is a $(\sigma, \sigma') \in N$ with $q \in \sigma$, $p \in \sigma'$, and with an $r \in R_{\sigma'}$ with $q \in D(r)$



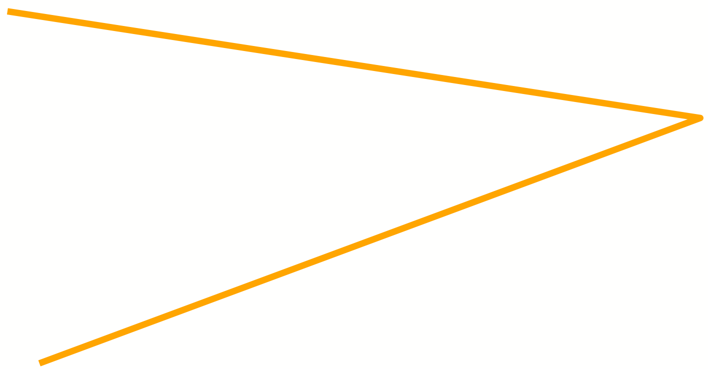
$R_{\sigma'}$:= all points that might intersect $\sigma \in N(\sigma')$ $\cup p_{\max}$

The Construction Algorithm

fix a cone C of the k cones

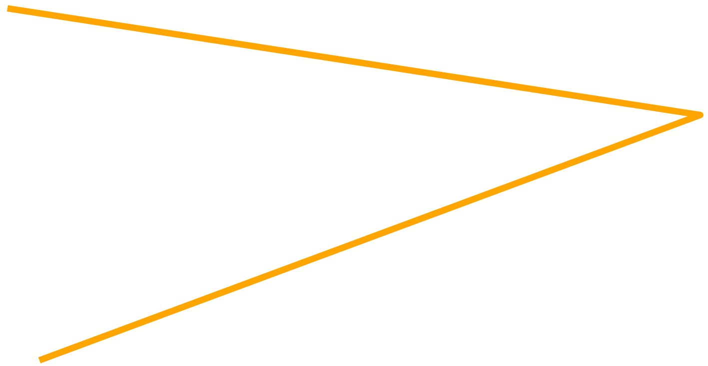
The Construction Algorithm

fix a cone C of the k cones



The Construction Algorithm

set all points in P to *active*



The Construction Algorithm

set all points in P to *active*

for each cell $\sigma \in \mathcal{Q}$ by increasing diameter

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

  for each lvl i cell σ

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

    for each lvl i cell σ

        for each cell $\sigma' \in N(\sigma)$

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

  for each lvl i cell σ

    for each cell $\sigma' \in N(\sigma)$

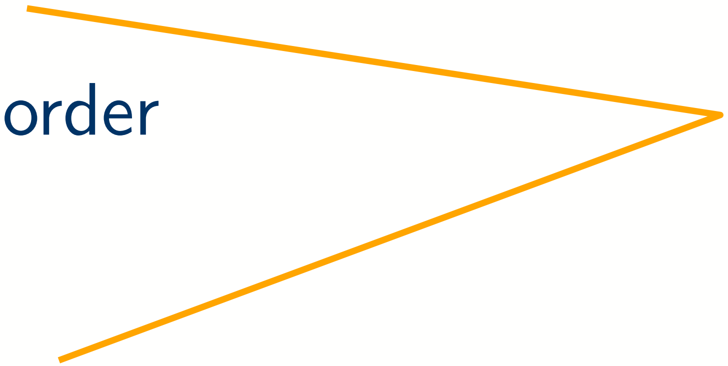
The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

    for each lvl i cell σ

        for each cell $\sigma' \in N(\sigma)$



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

  for each lvl i cell σ

    for each cell $\sigma' \in N(\sigma)$

      $Q \leftarrow$ *relevant* points in $P \cap \sigma$

      Sort Q in x/y-direction

      For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

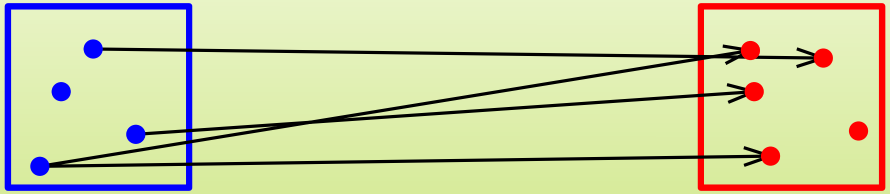
for each lvl i cell σ

for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

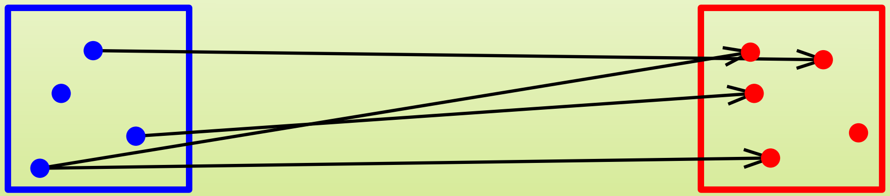
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

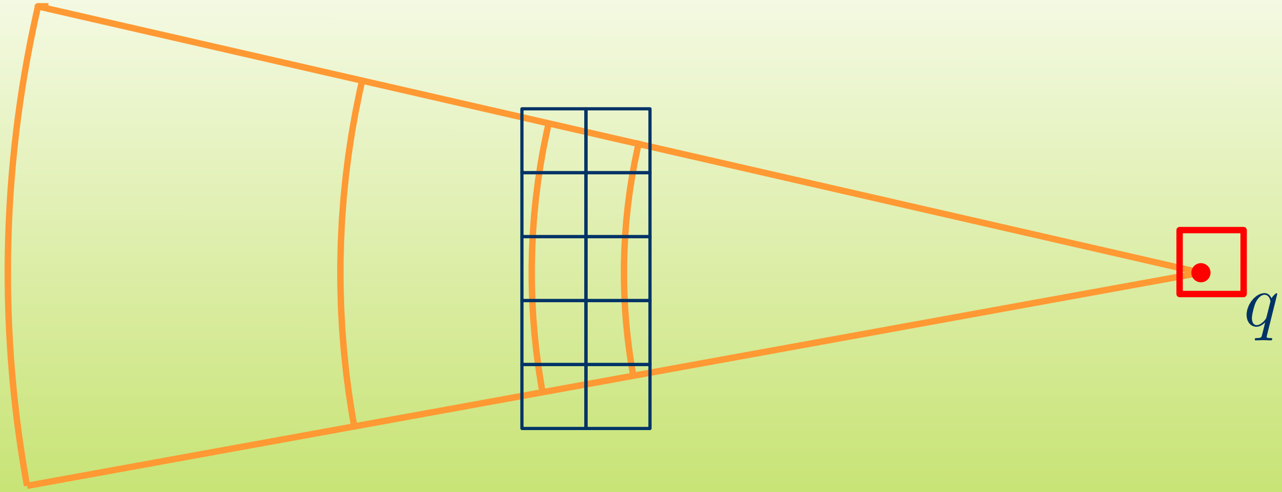
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

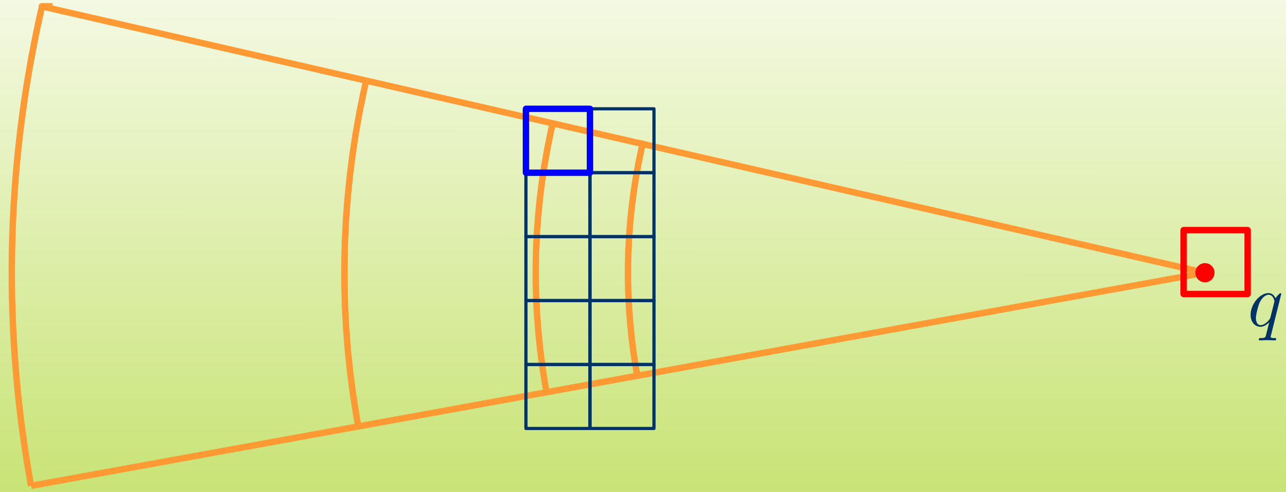
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

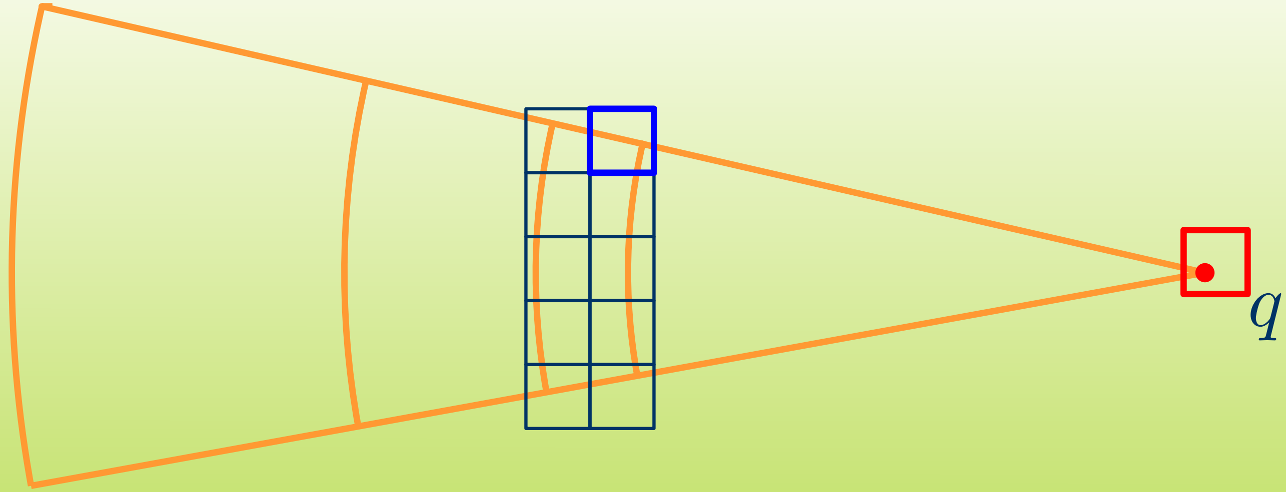
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

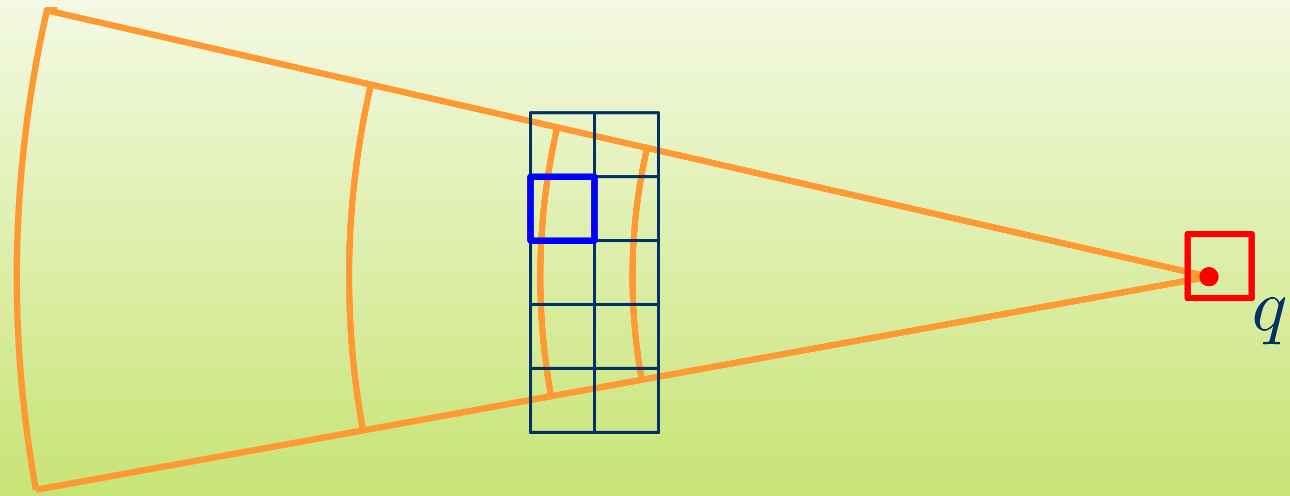
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

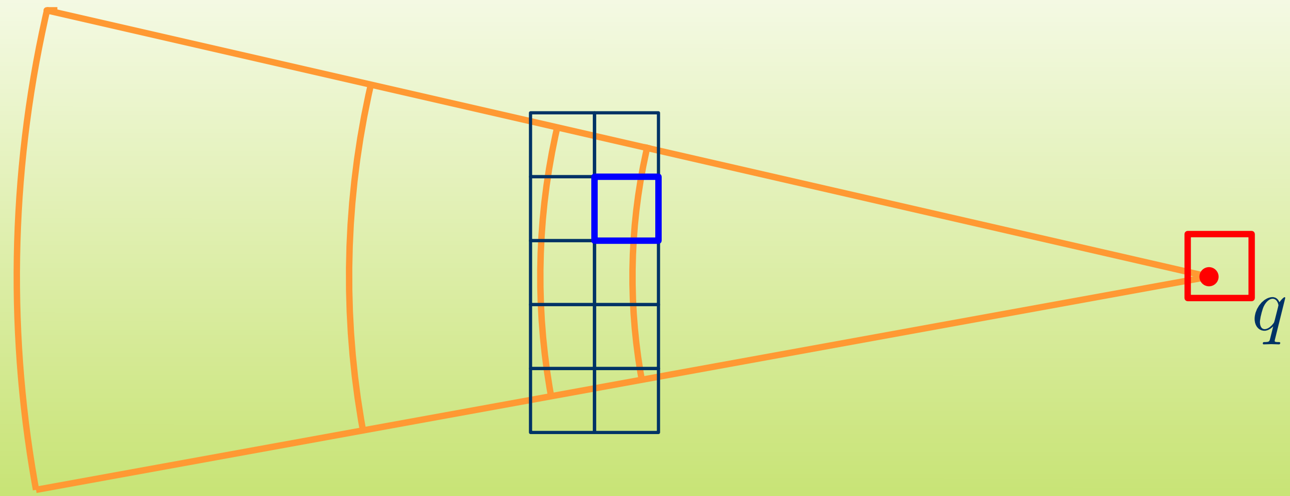
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*

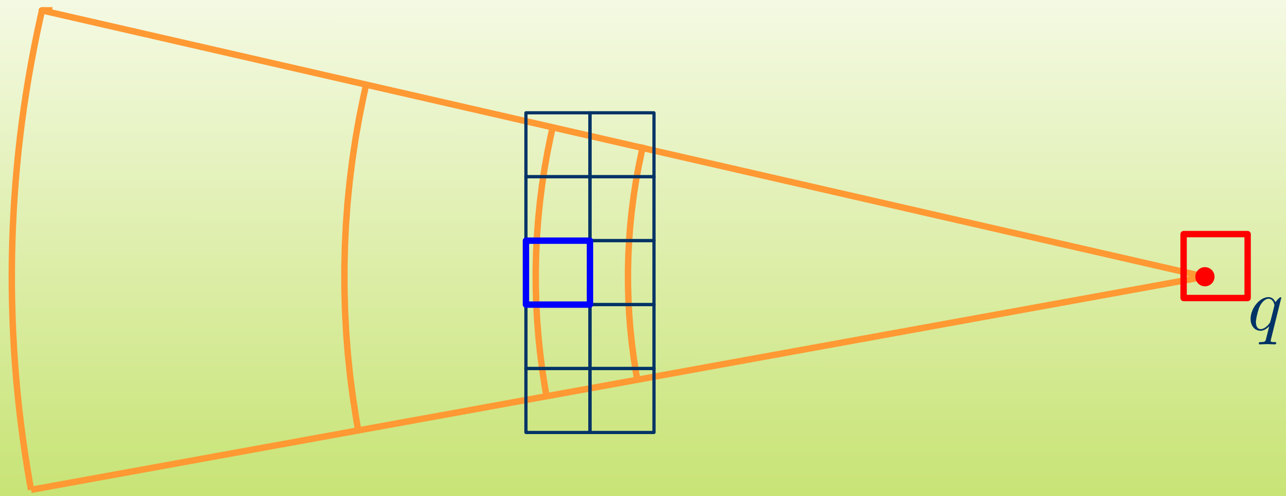


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*

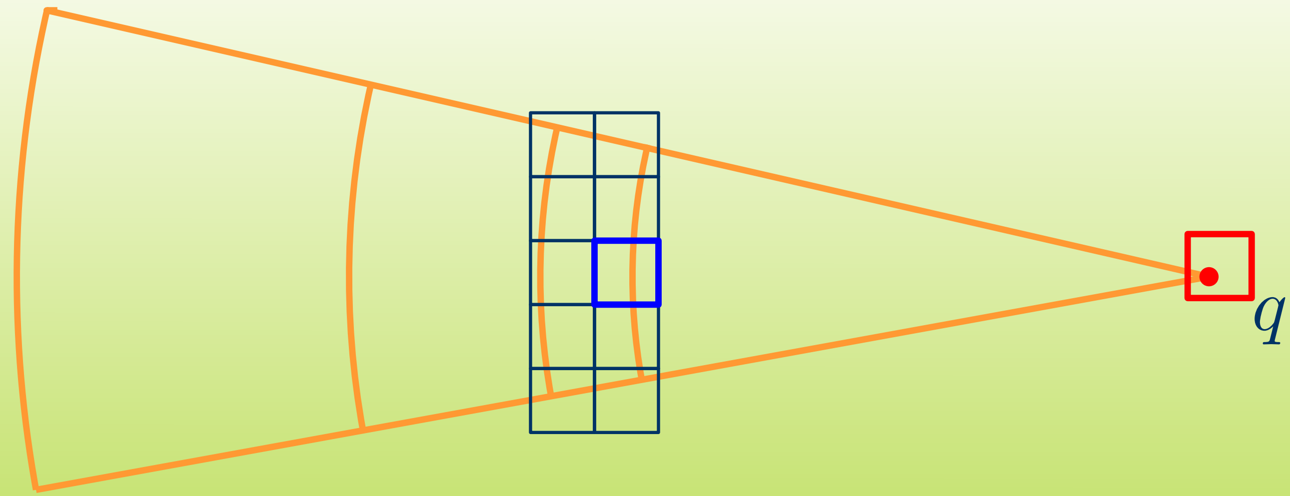


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

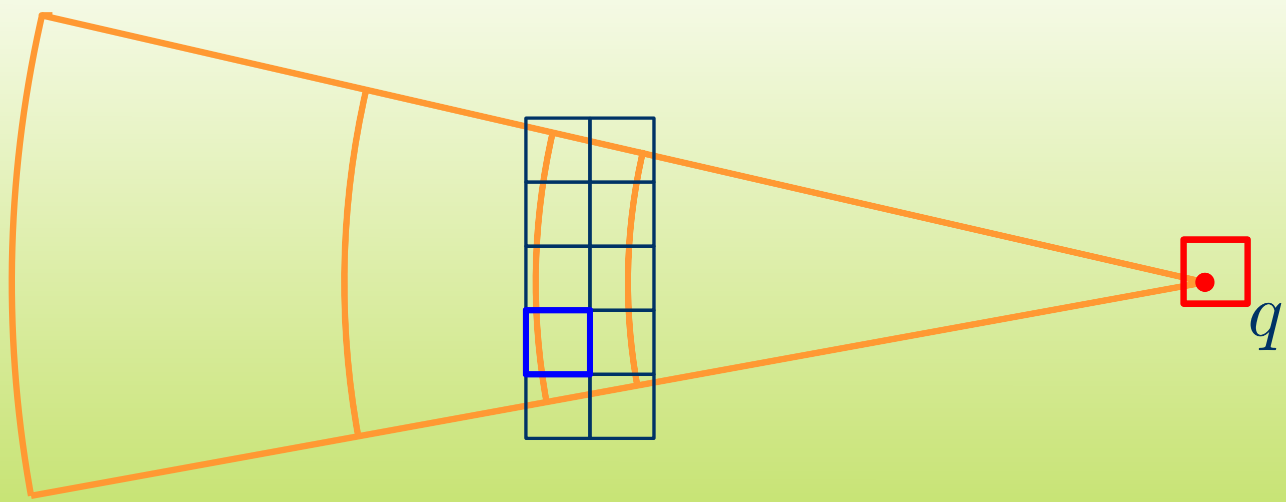
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*

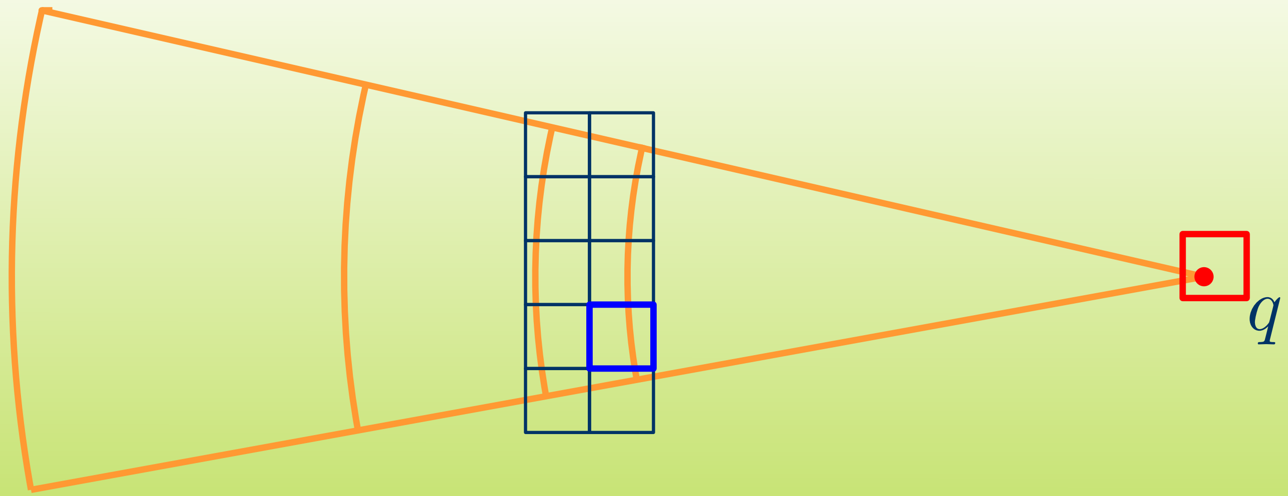


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*

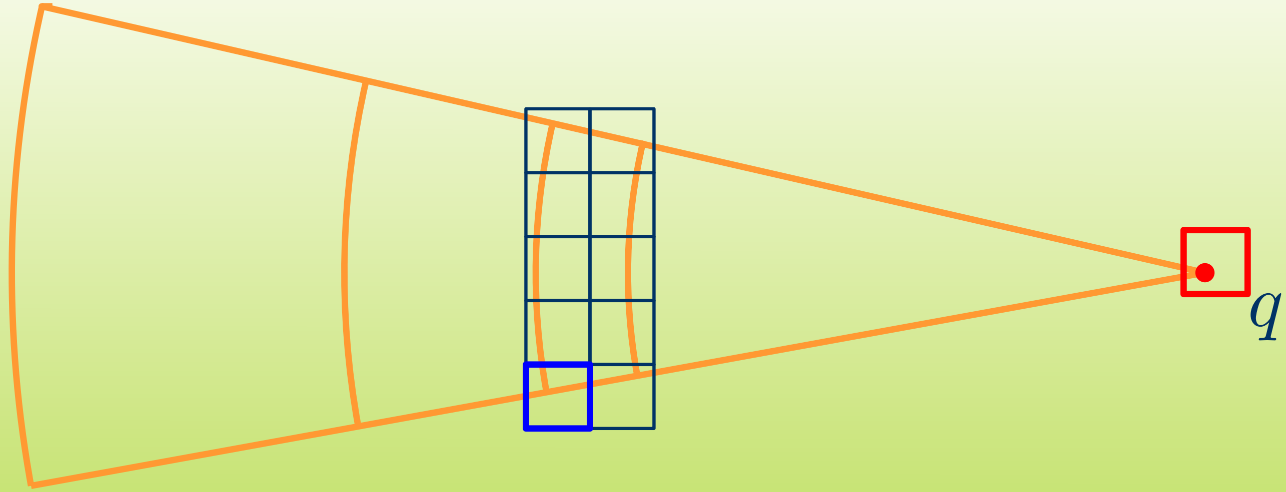


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

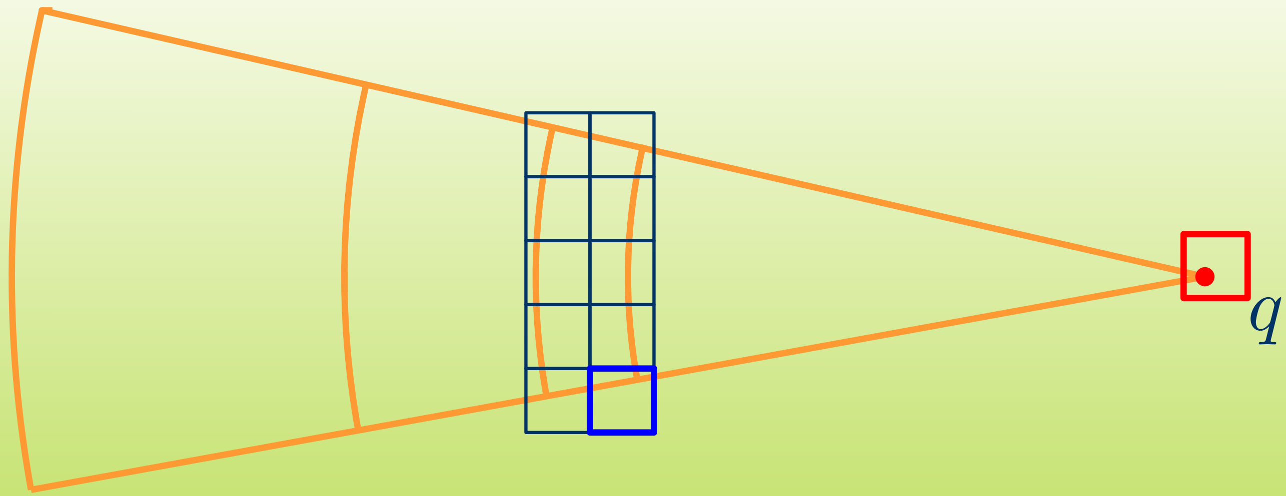
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*

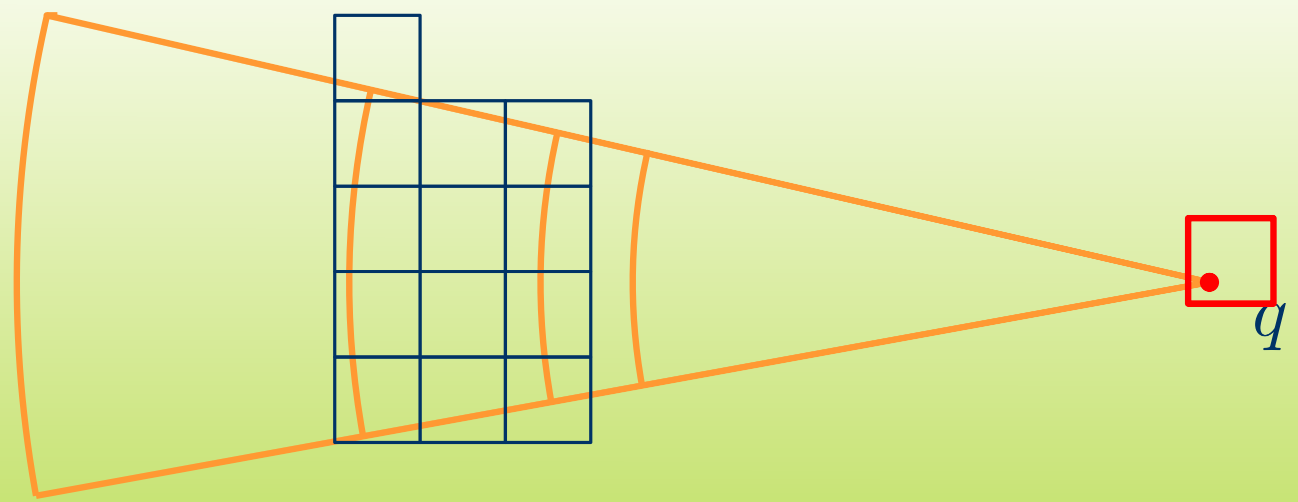


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*

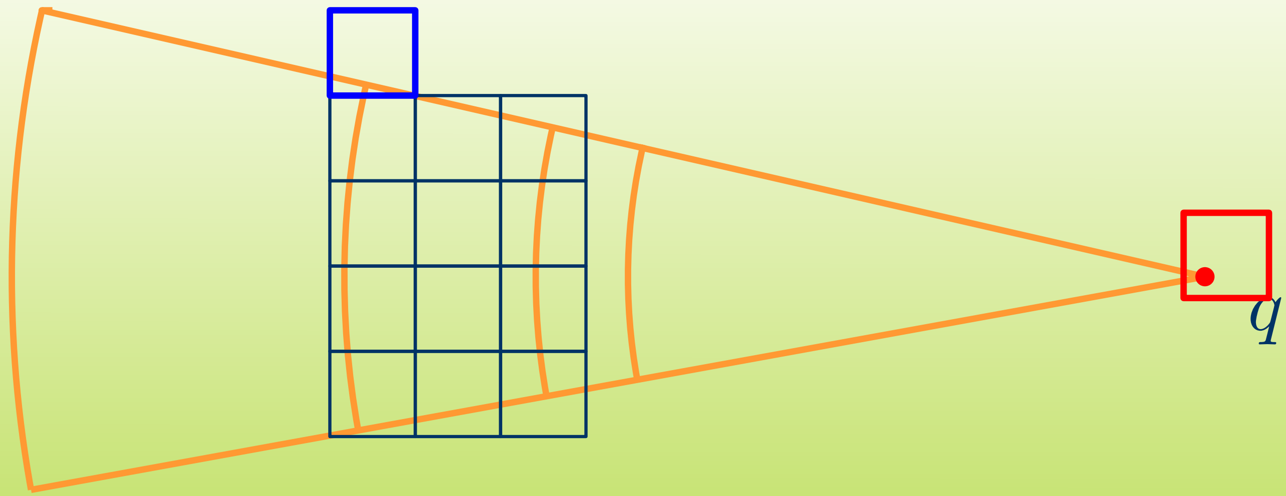


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

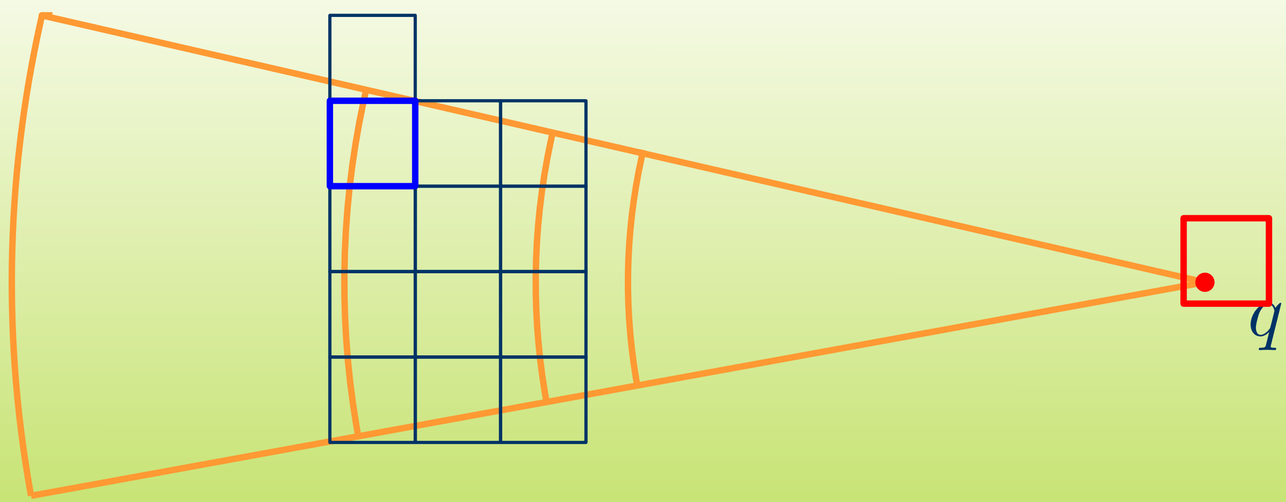
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

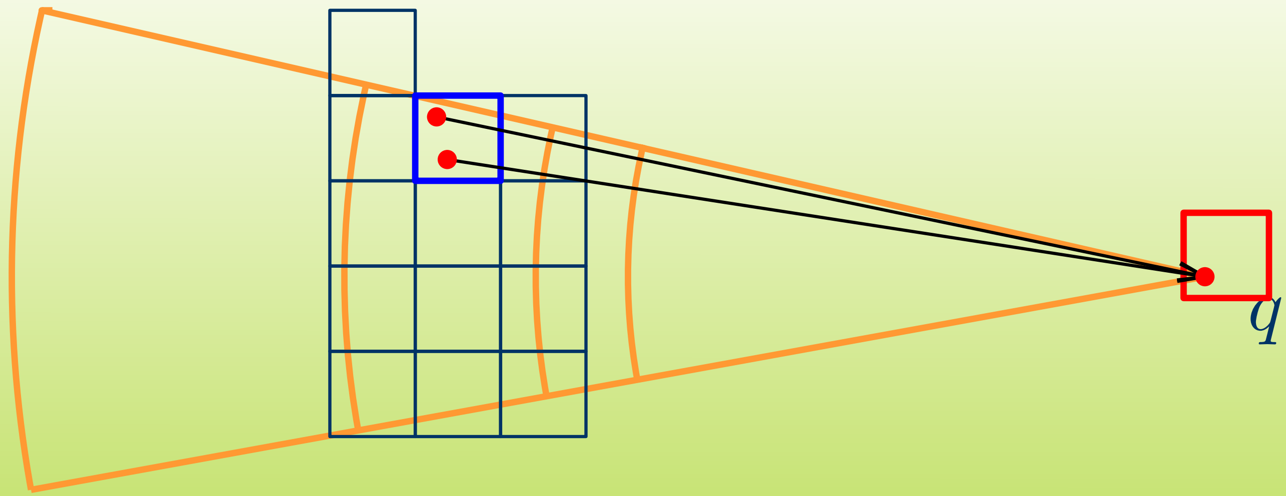
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

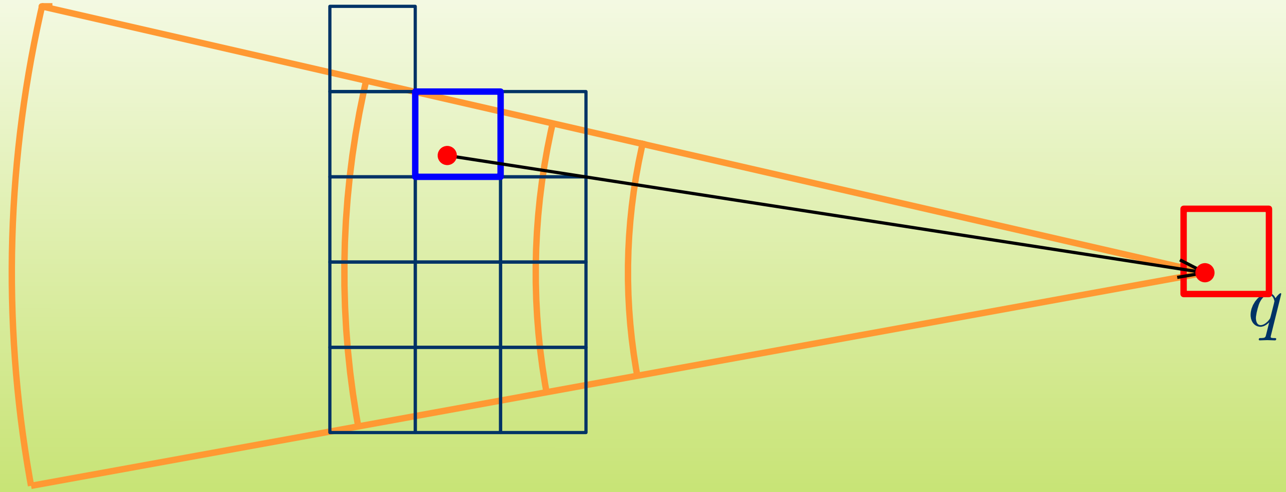
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

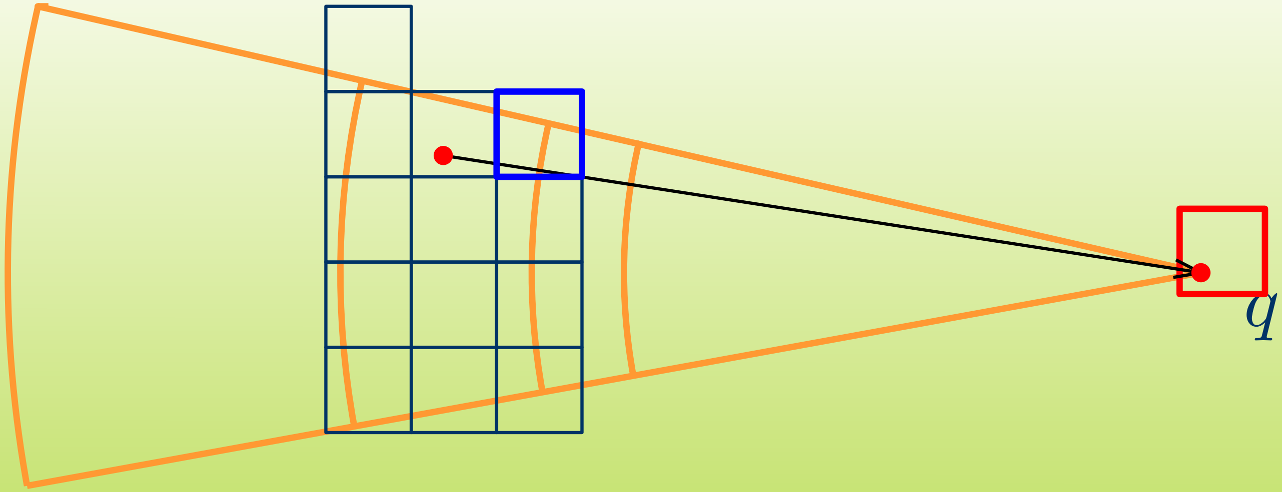
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*

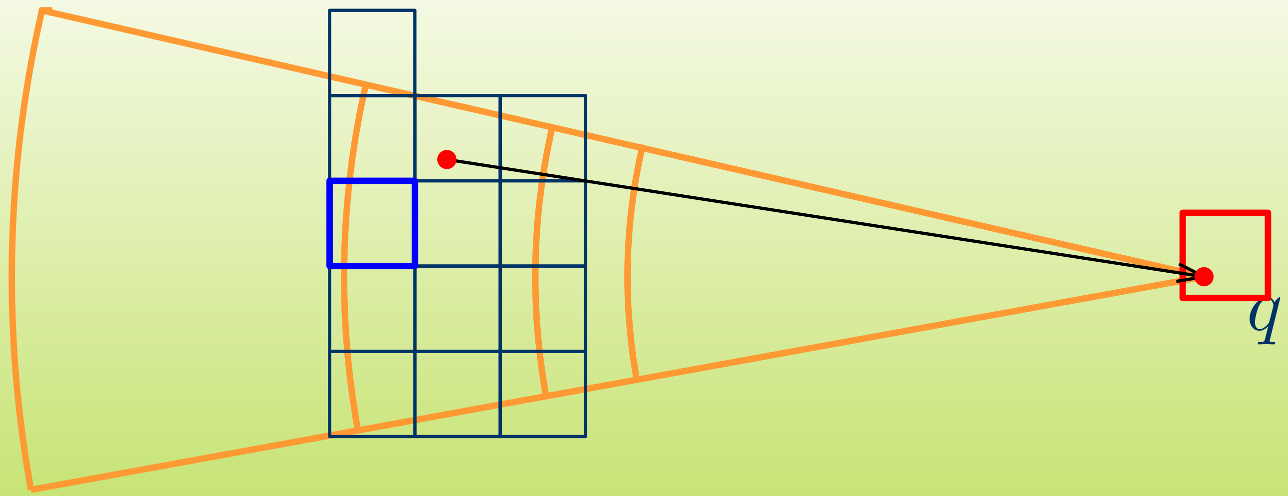


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

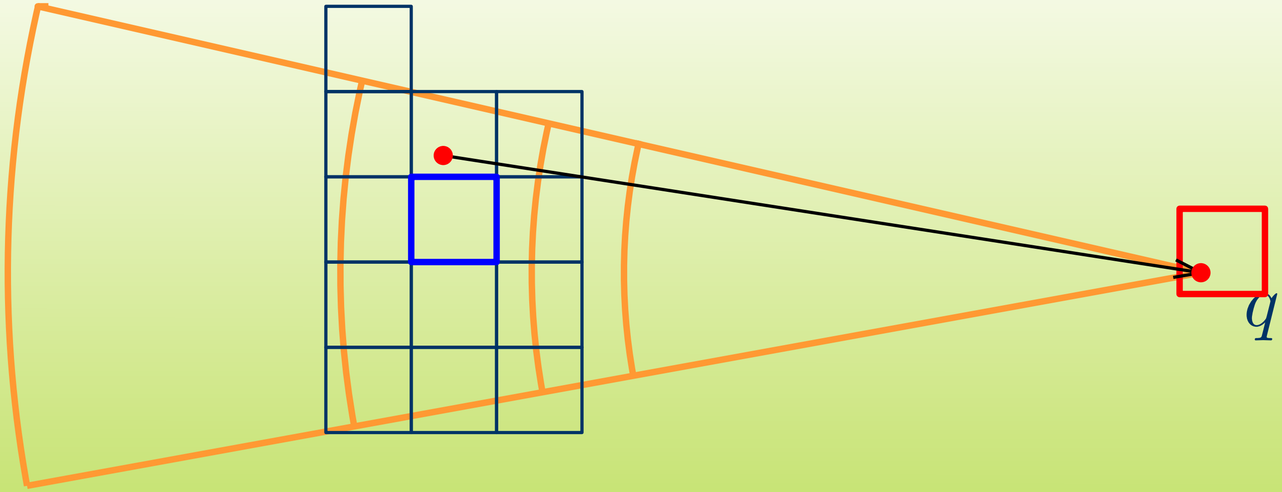
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*

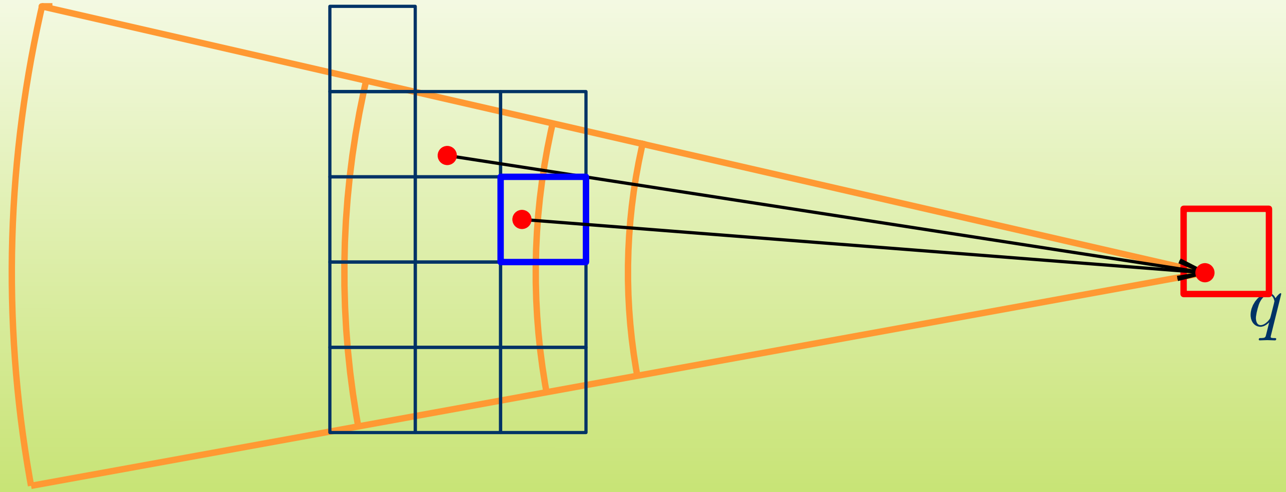


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

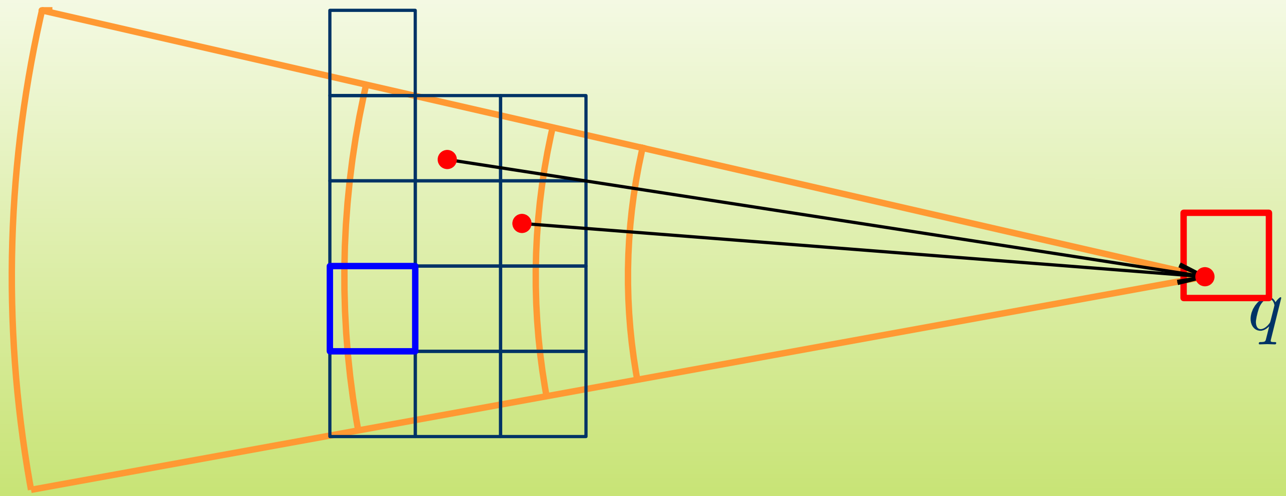
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*

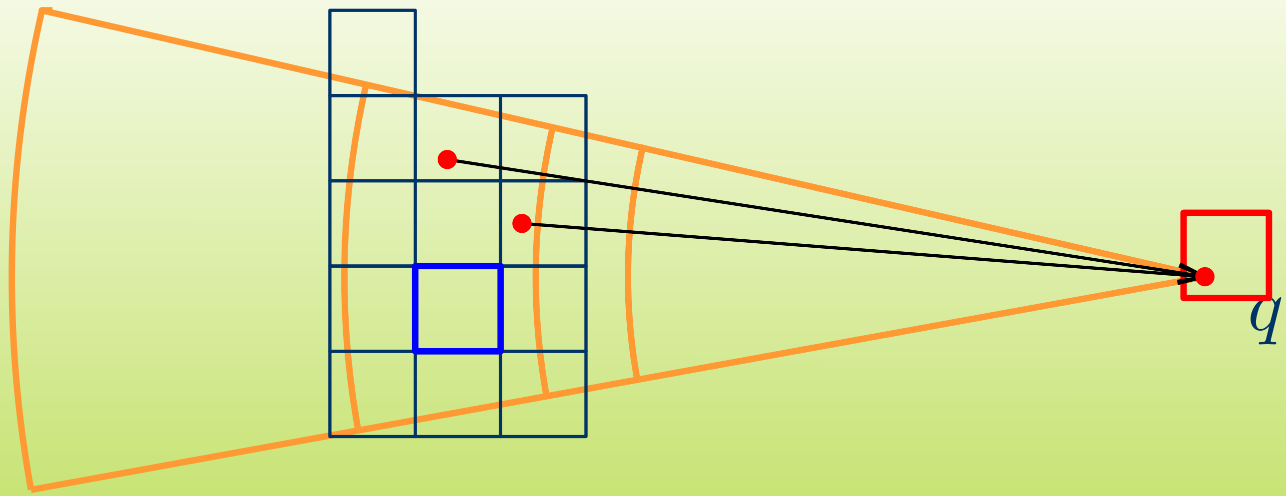


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

  for each lvl i cell σ

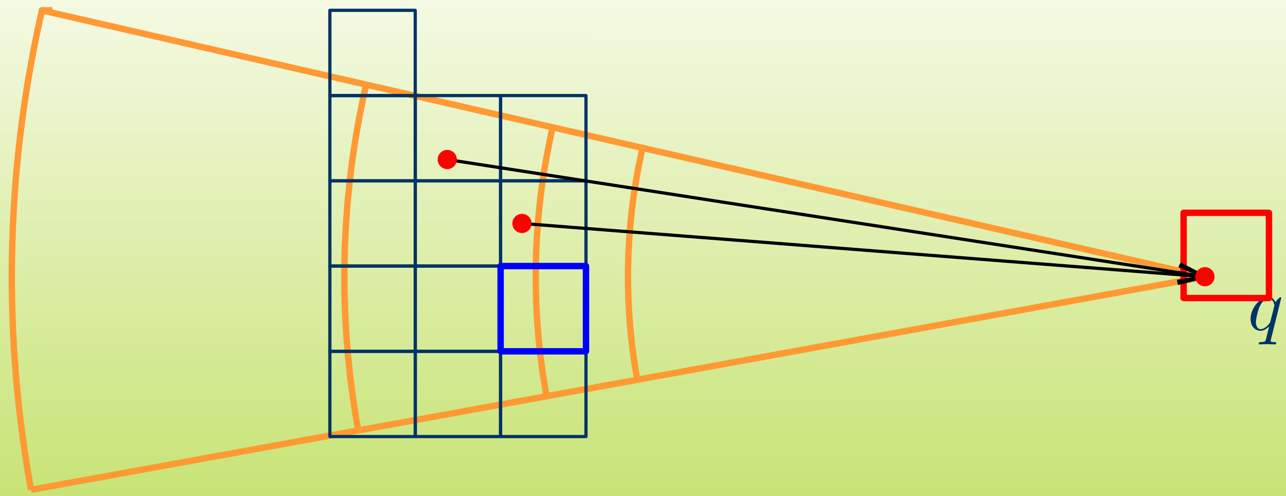
    for each cell $\sigma' \in N(\sigma)$

      $Q \leftarrow$ *relevant* points in $P \cap \sigma$

      Sort Q in x/y-direction

      For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

  set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

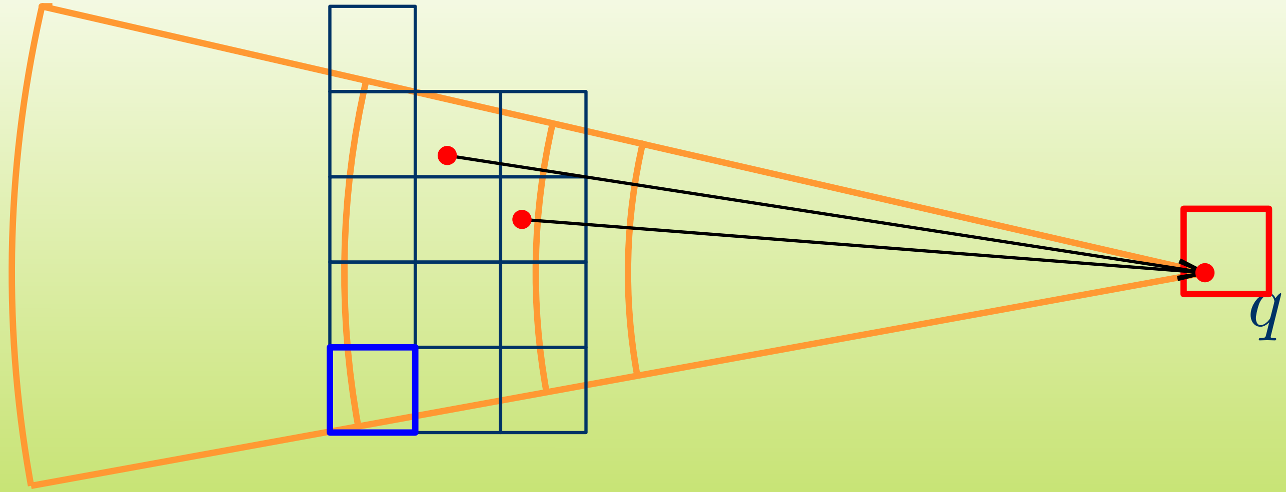
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*

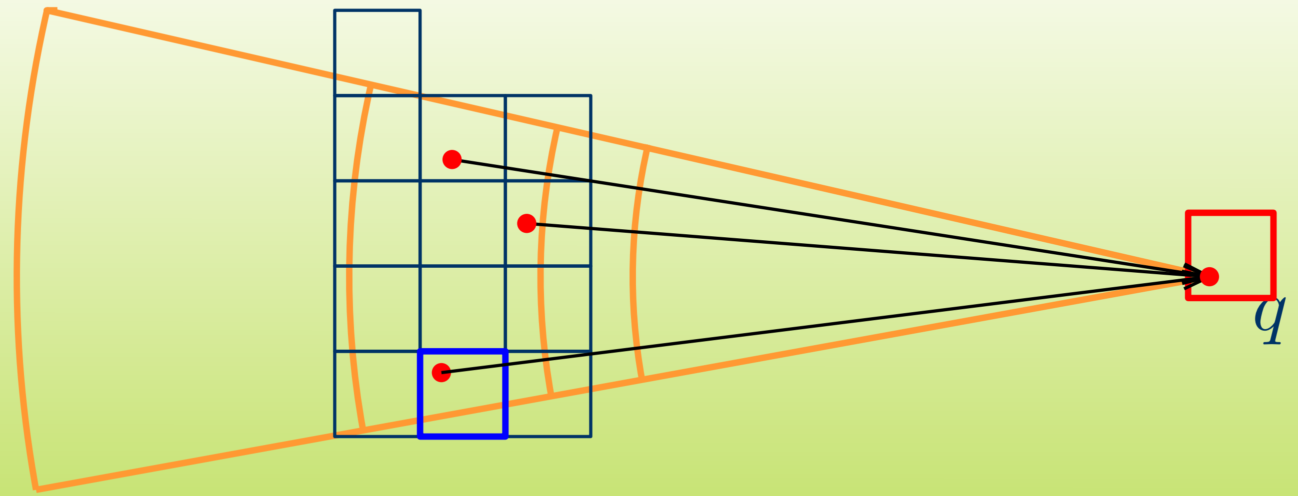


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

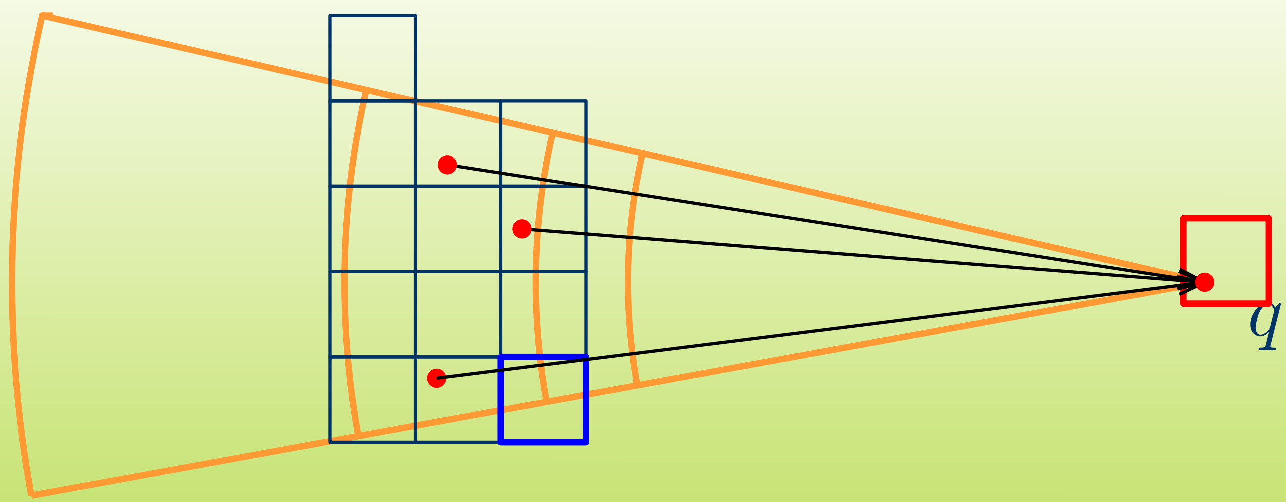
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*

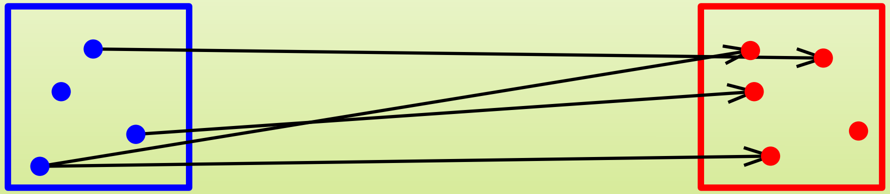


The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

- for each lvl i cell σ
  - for each cell $\sigma' \in N(\sigma)$
    - $Q \leftarrow$ *relevant* points in $P \cap \sigma$
    - Sort Q in x/y-direction
    - For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$
- set all points in Q with an incoming edge to *inactive*



The Construction Algorithm

set all points in P to *active*

for level i of the quadtree in increasing order

for each lvl i cell σ

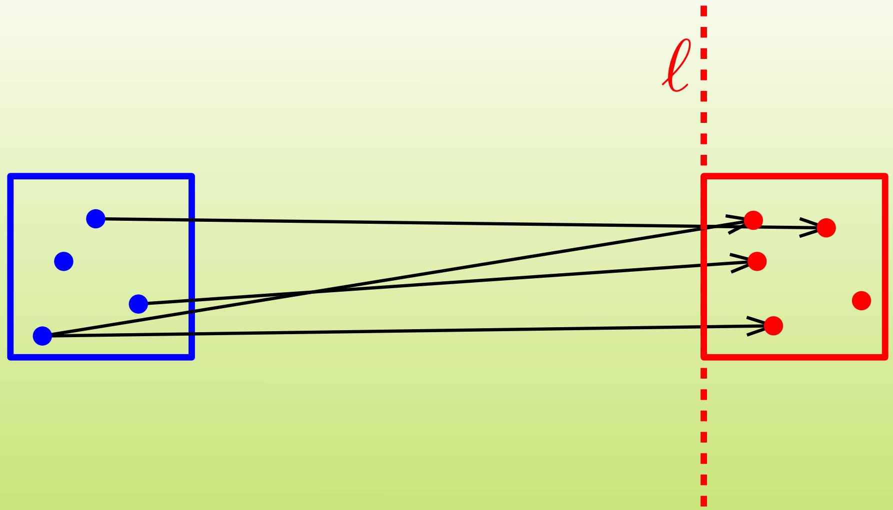
for each cell $\sigma' \in N(\sigma)$

$Q \leftarrow$ *relevant* points in $P \cap \sigma$

Sort Q in x/y-direction

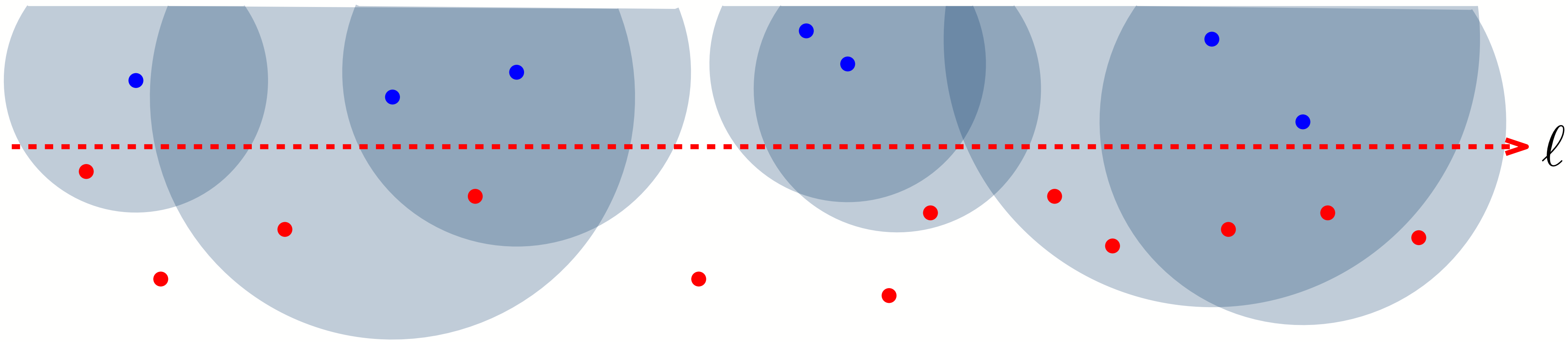
For each $q \in Q$, find an edge $\overrightarrow{rq}$ with $r \in R_{\sigma'}$

set all points in Q with an incoming edge to *inactive*



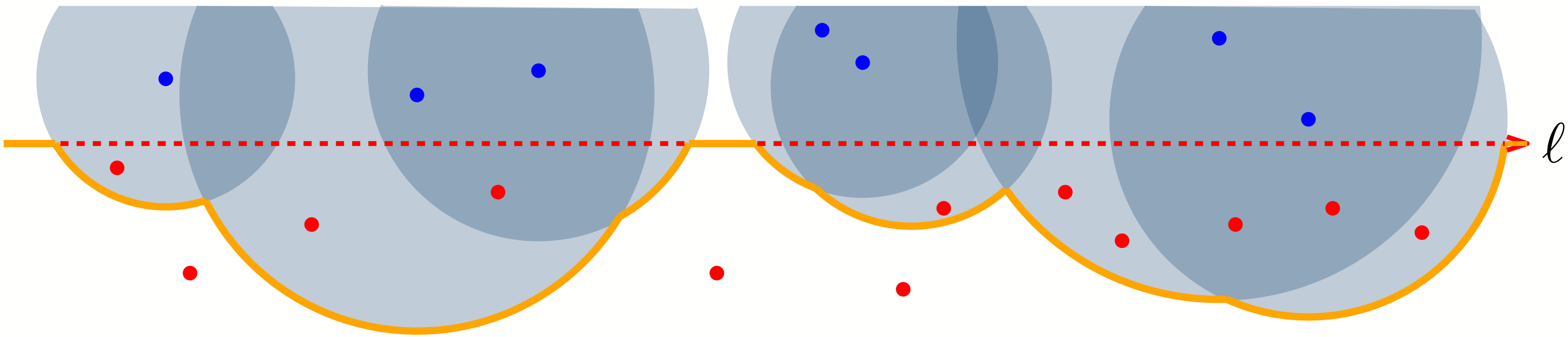
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



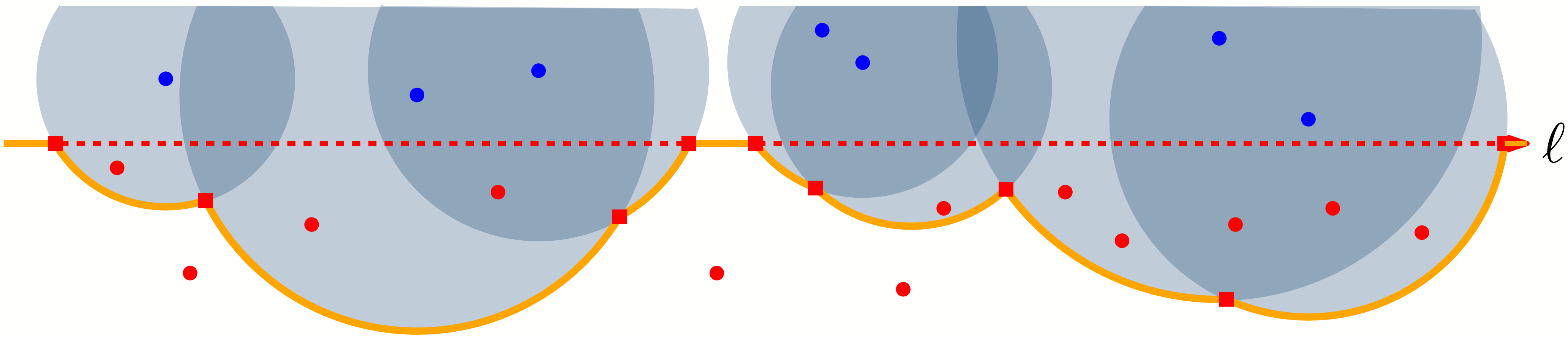
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



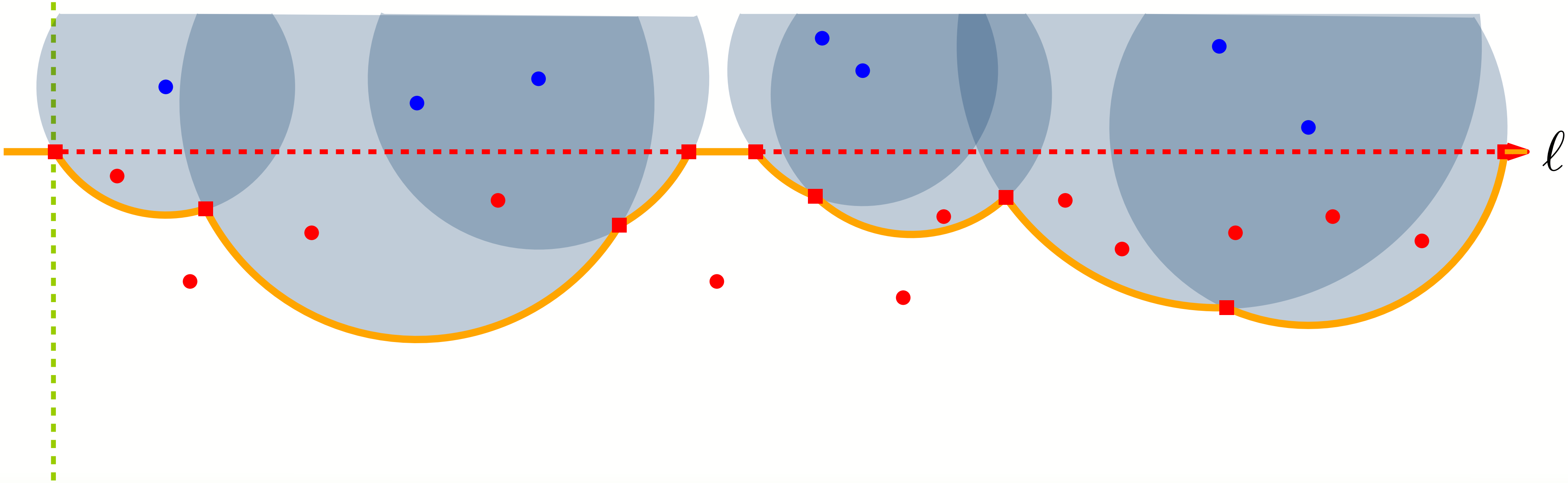
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



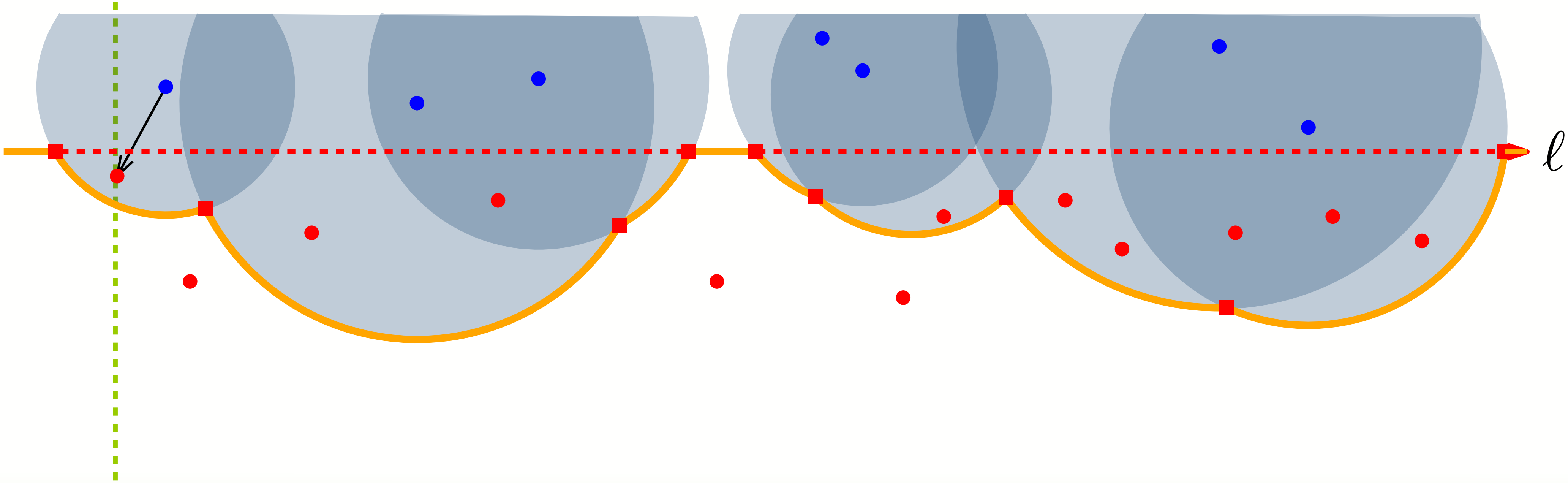
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



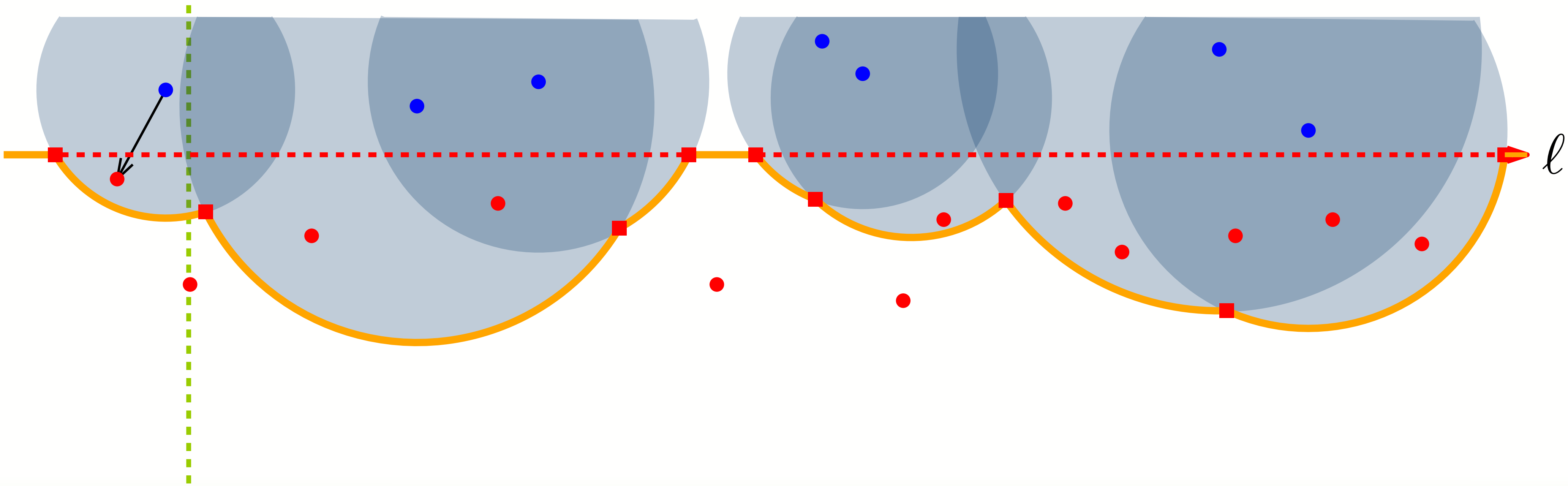
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



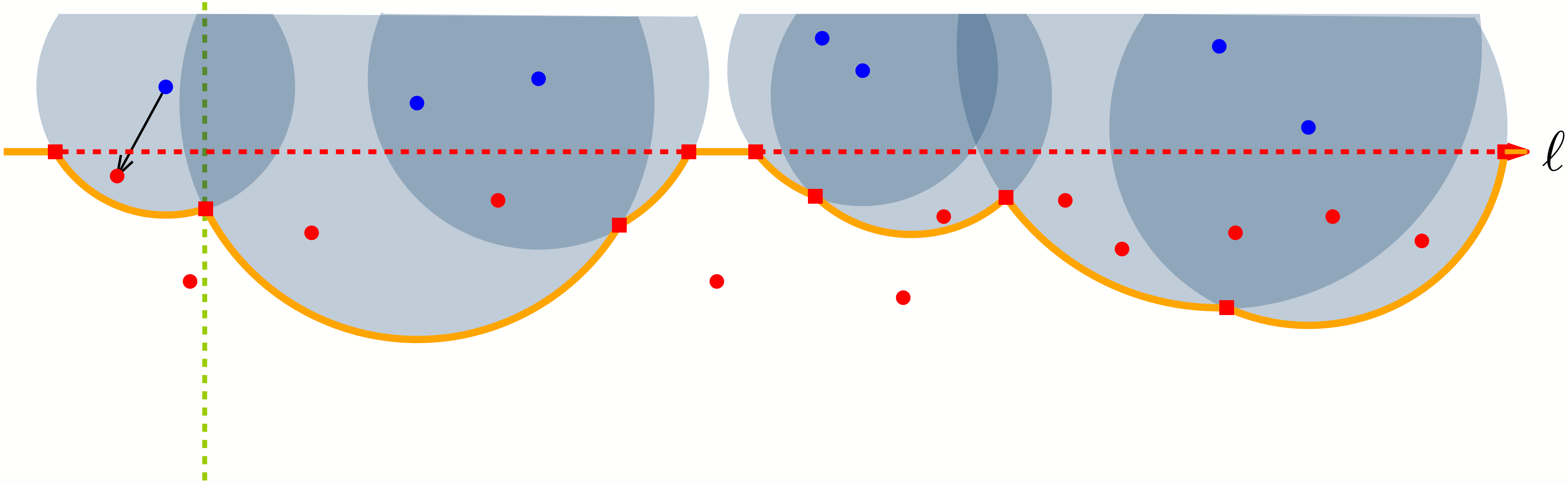
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



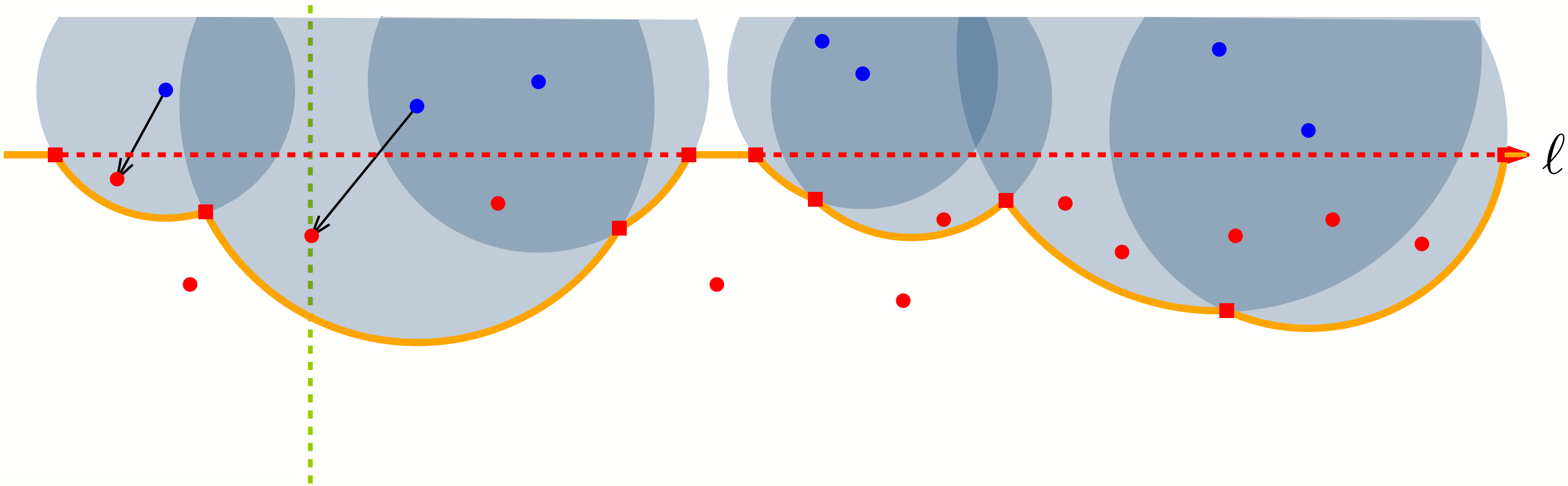
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



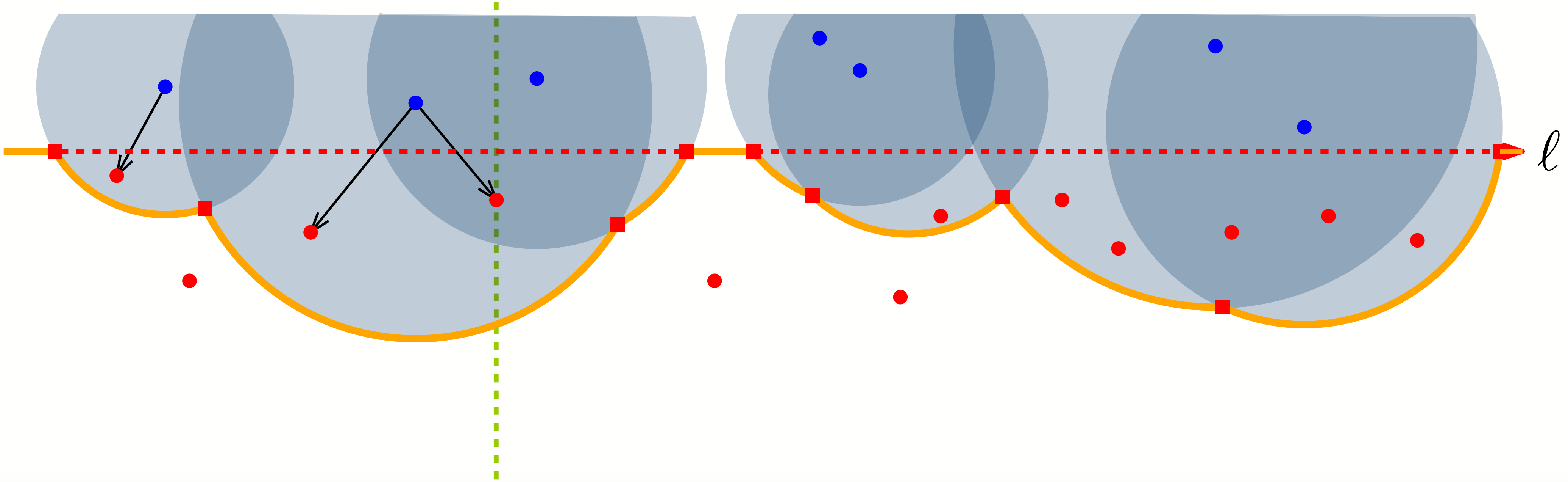
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



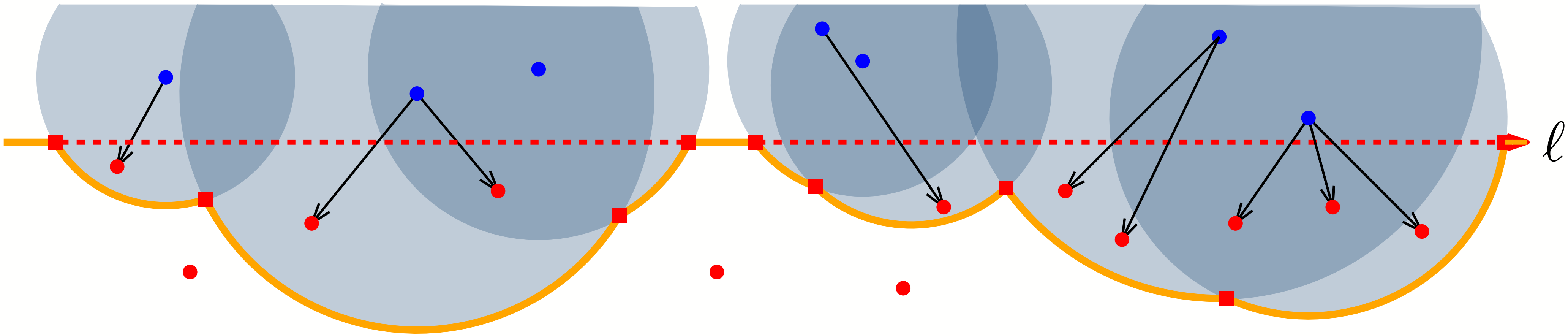
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



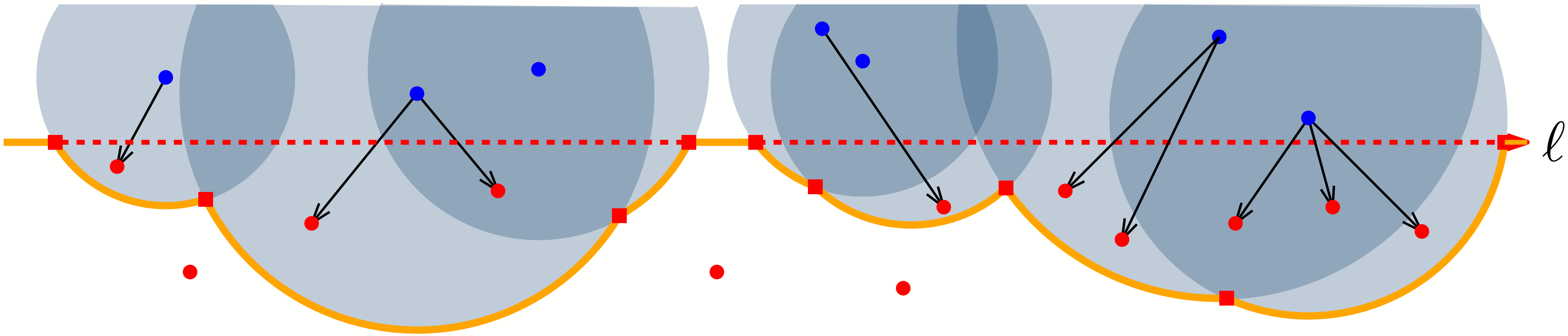
Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$

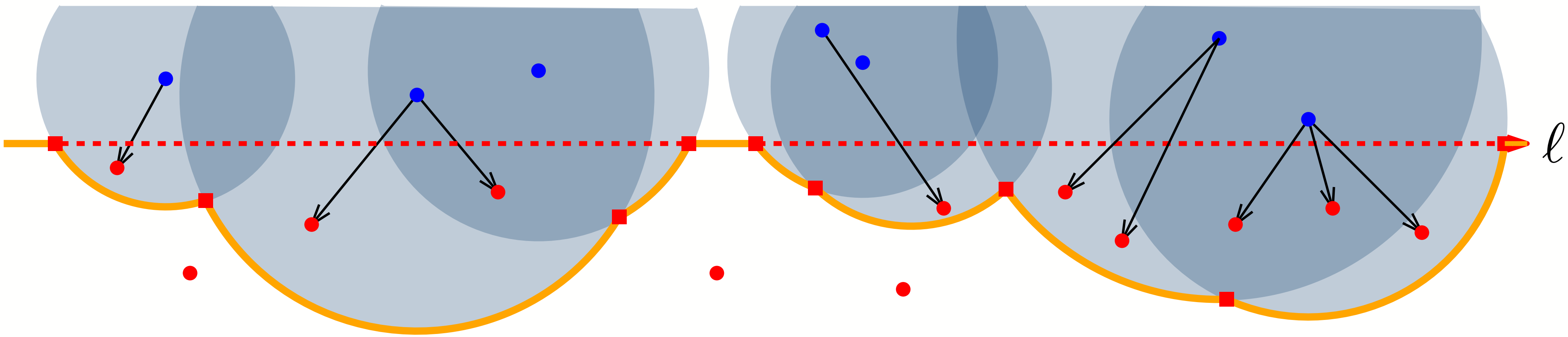


Total running time:

- $|N(\sigma)| = O(c^2)$

Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$

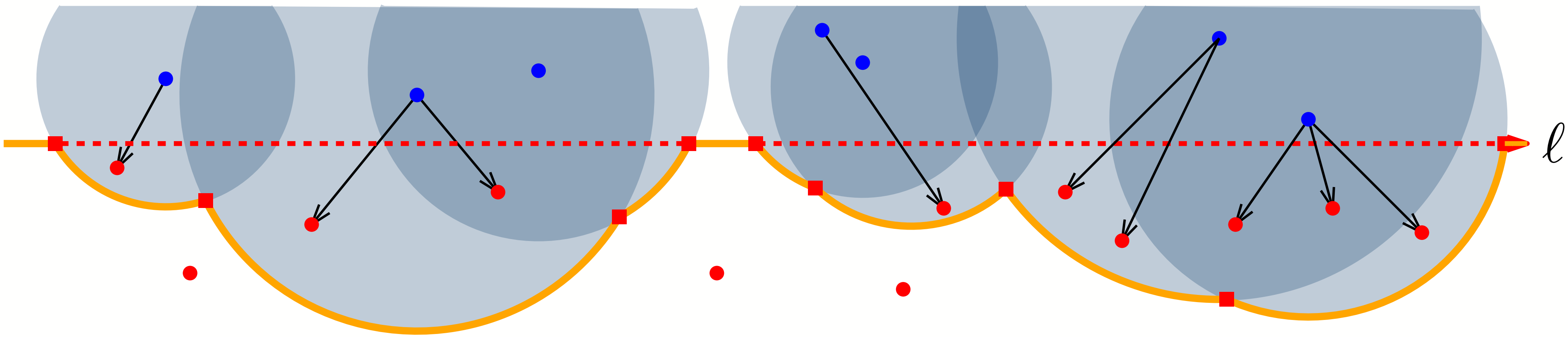


Total running time:

- $|N(\sigma)| = O(c^2)$
- each point $p \in P$ is in $O(\log \Phi)$ different sets Q

Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$

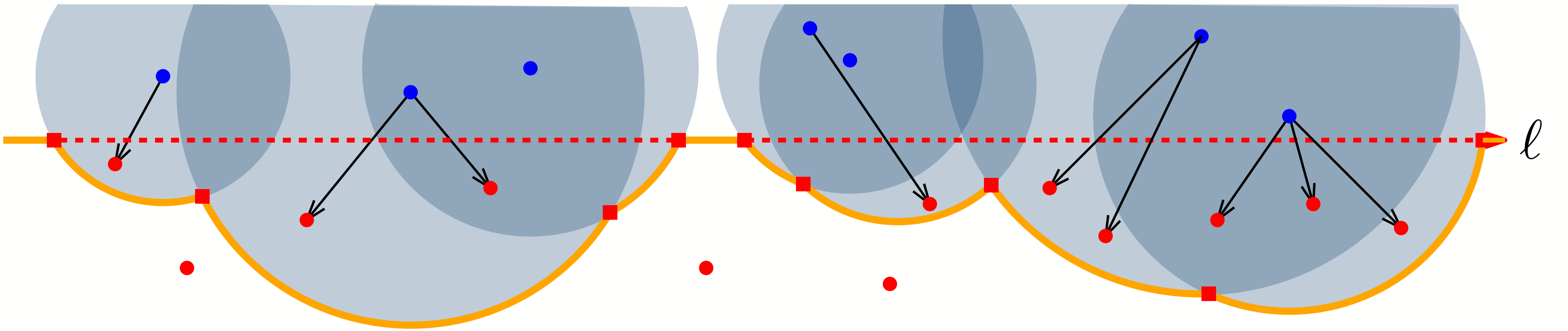


Total running time:

- $|N(\sigma)| = O(c^2)$
- each point $p \in P$ is in $O(\log \Phi)$ different sets Q
- each point $p \in P$ appears in $O(1)$ sets $R_{\sigma'}$

Fast Edge Selection and Running Time

This needs time $O(m \log m + n)$ where $m = |R_{\sigma'}|$ and $n = |Q|$



Total running time:

- $|N(\sigma)| = O(c^2)$
- each point $p \in P$ is in $O(\log \Phi)$ different sets Q
- each point $p \in P$ appears in $O(1)$ sets $R_{\sigma'}$

$$O(n(\log \Phi + \log n))$$

Final Results

Theorem 1: Let $\boldsymbol{P} \subset \mathbb{R}^{\mathbf{2}}$ be a point set with radii and with **spread** $\boldsymbol{\Phi}$. Let G be the transmission graph of P. For any $t > 1$ we can compute a **t-spanner** $\boldsymbol{H} \subseteq \boldsymbol{G}$ for G in **time** $\boldsymbol{O(n(\log n + \log \Phi))}$.

Theorem 2: Let $\boldsymbol{\Psi}$ be the **ratio of the largest and smallest radius** in P. We can compute a t-spanner $H \subseteq G$ in **time** $\boldsymbol{O(n(\log n + \log \Psi))}$.