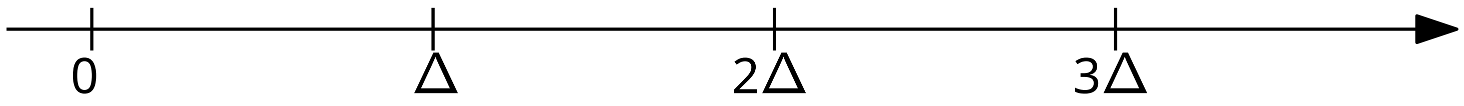
Shifting a grid over a point set

for simple and fast approximation algorithms

Shifted partition of the real line

Let $\Delta > 0$ and $b \in [0, \Delta]$ uniformly distributed.

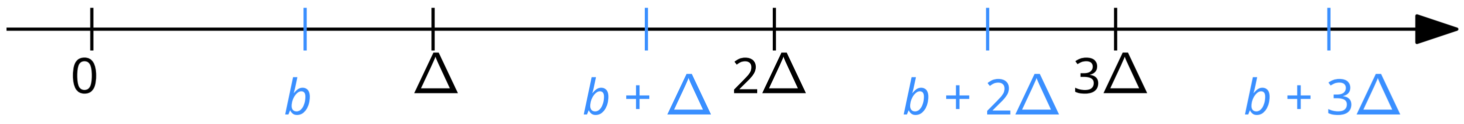
We shift the grid G_Δ by b



Shifted partition of the real line

Let $\Delta > 0$ and $b \in [0, \Delta]$ uniformly distributed.

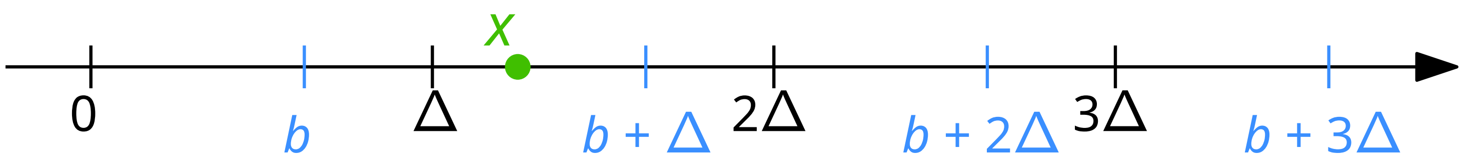
We shift the grid G_Δ by b



Shifted partition of the real line

Let $\Delta > 0$ and $b \in [0, \Delta]$ uniformly distributed.

We shift the grid G_Δ by b

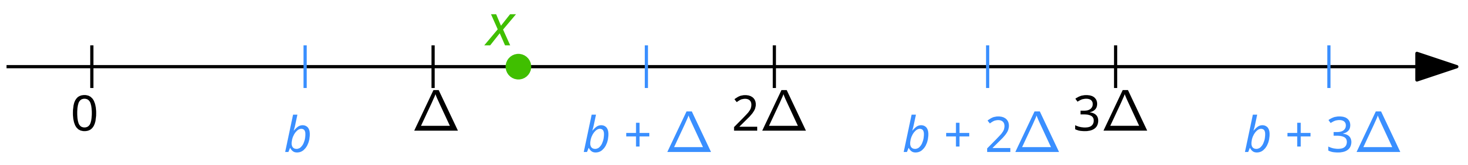


$h_{b,\Delta}(x) = \left\lfloor \frac{x - b}{\Delta} \right\rfloor$

Shifted partition of the real line

Let $\Delta > 0$ and $b \in [0, \Delta]$ uniformly distributed.

We shift the grid G_Δ by b



$h_{b,\Delta}(x) = \left\lfloor \frac{x-b}{\Delta} \right\rfloor$

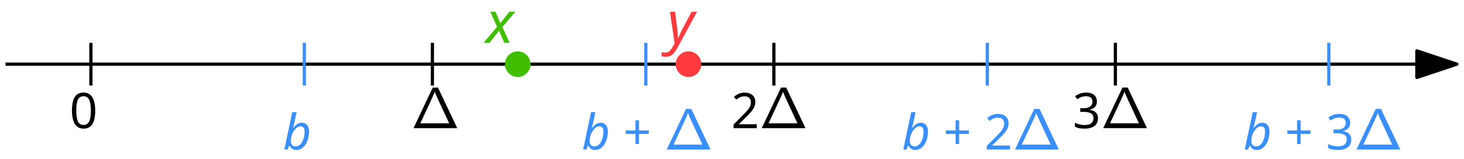
Remark: If $|b - b'| = i\Delta$ for some $i \geq 0$, then b and b' induce the same partition.

(Later we will use this to choose $b \in [y + i\Delta, y + j\Delta]$ for $i < j$.)

Shifted partition of the real line

Let $\Delta > 0$ and $b \in [0, \Delta]$ uniformly distributed.

We shift the grid G_Δ by b



$h_{b,\Delta}(x) = \left\lfloor \frac{x-b}{\Delta} \right\rfloor$

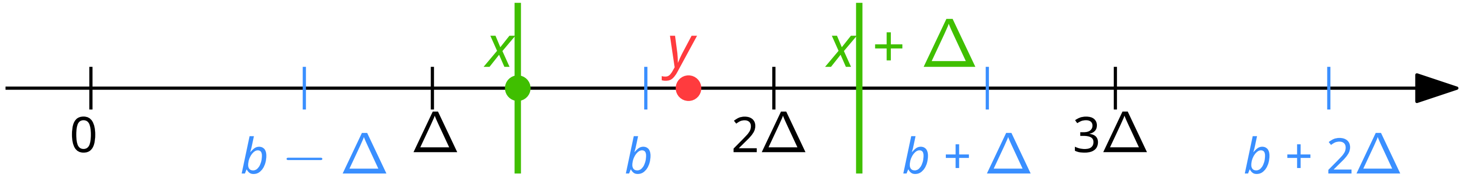
Remark: If $|b - b'| = i\Delta$ for some $i \geq 0$, then b and b' induce the same partition.
(Later we will use this to choose $b \in [y + i\Delta, y + j\Delta]$ for $i < j$.)

Lemma: For $x, y \in \mathbb{R}$ holds $\mathbb{P}\left[h_{b,\Delta}(x) \neq h_{b,\Delta}(y)\right] = min\left(\frac{|x-y|}{\Delta}, 1\right)$

Shifted partition of the real line

Let $\Delta > 0$ and $b \in [0, \Delta]$ uniformly distributed.

We shift the grid G_Δ by b



$h_{b,\Delta}(x) = \left\lfloor \frac{x-b}{\Delta} \right\rfloor$

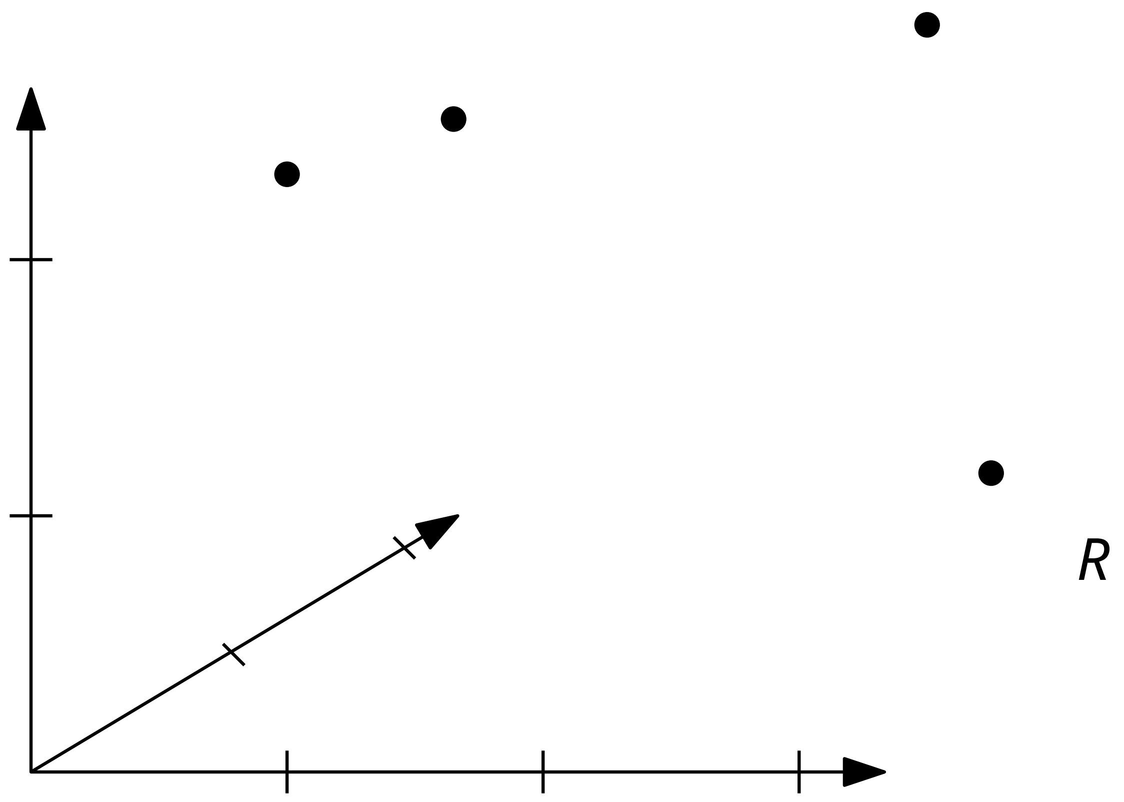
Remark: If $|b - b'| = i\Delta$ for some $i \geq 0$, then b and b' induce the same partition.
(Later we will use this to choose $b \in [y + i\Delta, y + j\Delta]$ for $i < j$.)

Lemma: For $x, y \in \mathbb{R}$ holds $\mathbb{P}\left[h_{b,\Delta}(x) \neq h_{b,\Delta}(y)\right] = min\left(\frac{|x-y|}{\Delta}, 1\right)$

Proof: Wlog $x < y$. Claim holds trivially if $|x - y| > \Delta$.
Otherwise assume $b \in [x, x + \Delta]$. Then $h_{b,\Delta}(x) \neq h_{b,\Delta}(y) \Leftrightarrow b \in [x, y]$.

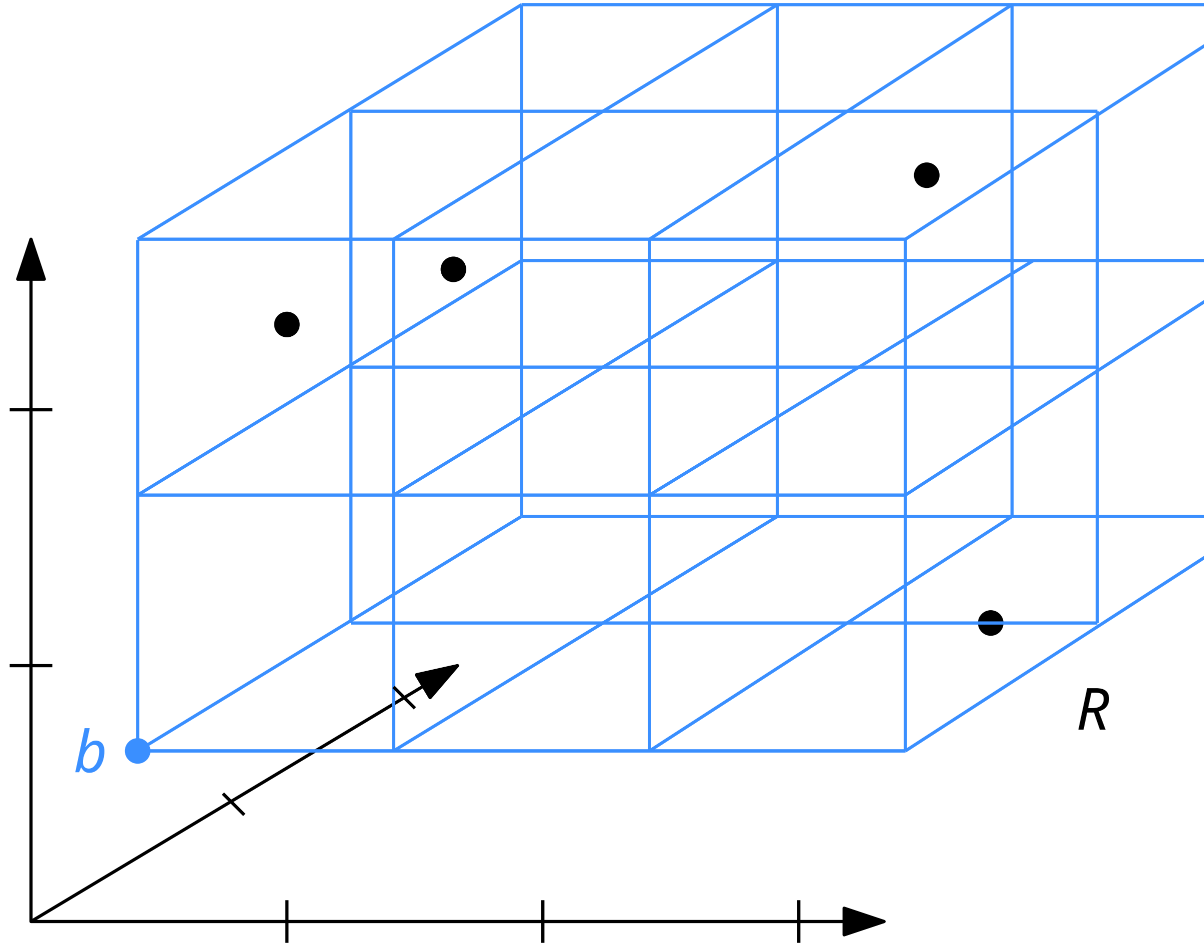
Shifted partition of space

Now let P be a point set in $\mathbb{R}^d$ and $b = (b_1, \dots, b_d)$ uniformly randomly choosen from the hypercube $[0, \Delta]^d$. Consider the (shifted) grid $G^d(b, \Delta)$ with origin in b and sidelength Δ.



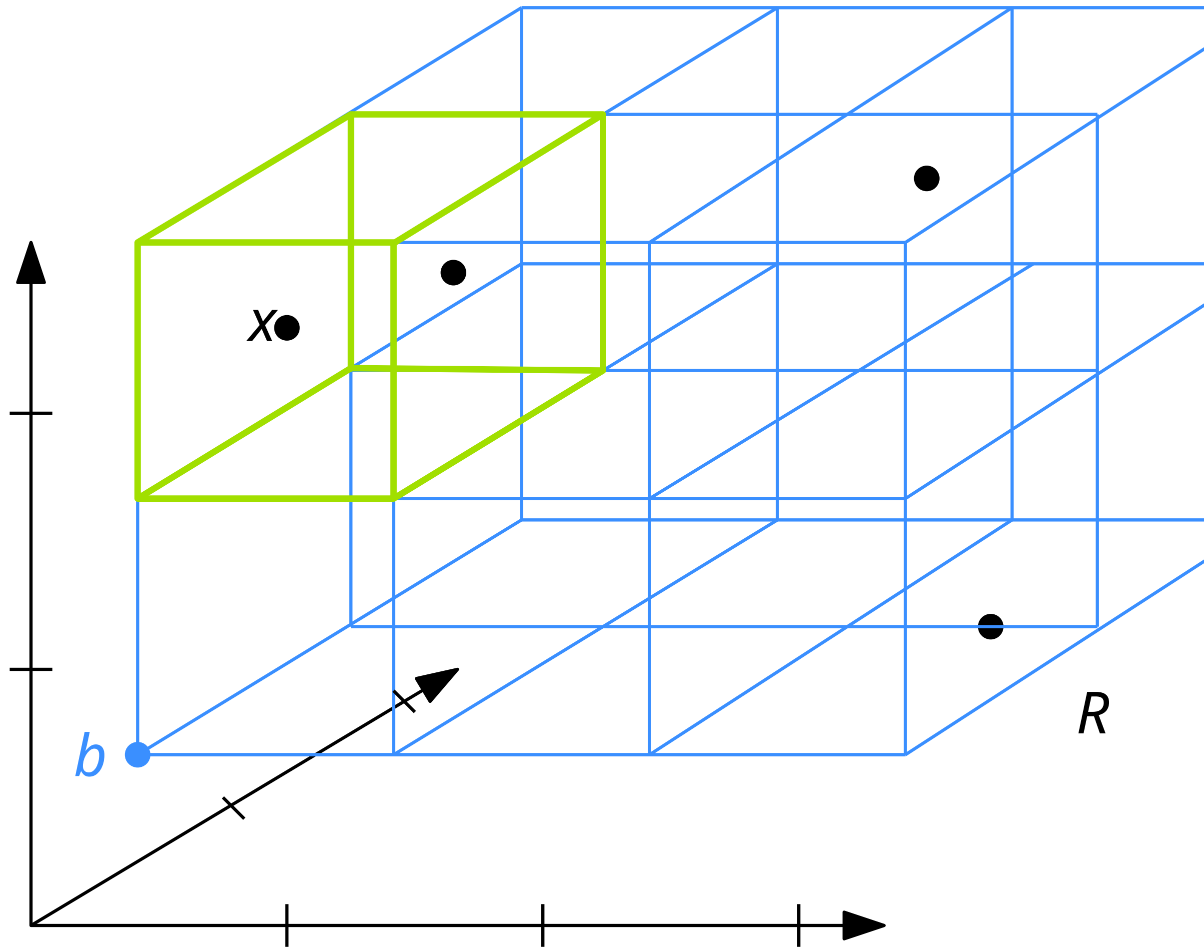
Shifted partition of space

Now let P be a point set in $\mathbb{R}^d$ and $b = (b_1, \dots, b_d)$ uniformly randomly choosen from the hypercube $[0, \Delta]^d$. Consider the (shifted) grid $G^d(b, \Delta)$ with origin in b and sidelength Δ.



Shifted partition of space

Now let P be a point set in $\mathbb{R}^d$ and $b = (b_1, \dots, b_d)$ uniformly randomly choosen from the hypercube $[0, \Delta]^d$. Consider the (shifted) grid $G^d(b, \Delta)$ with origin in b and sidelength Δ.

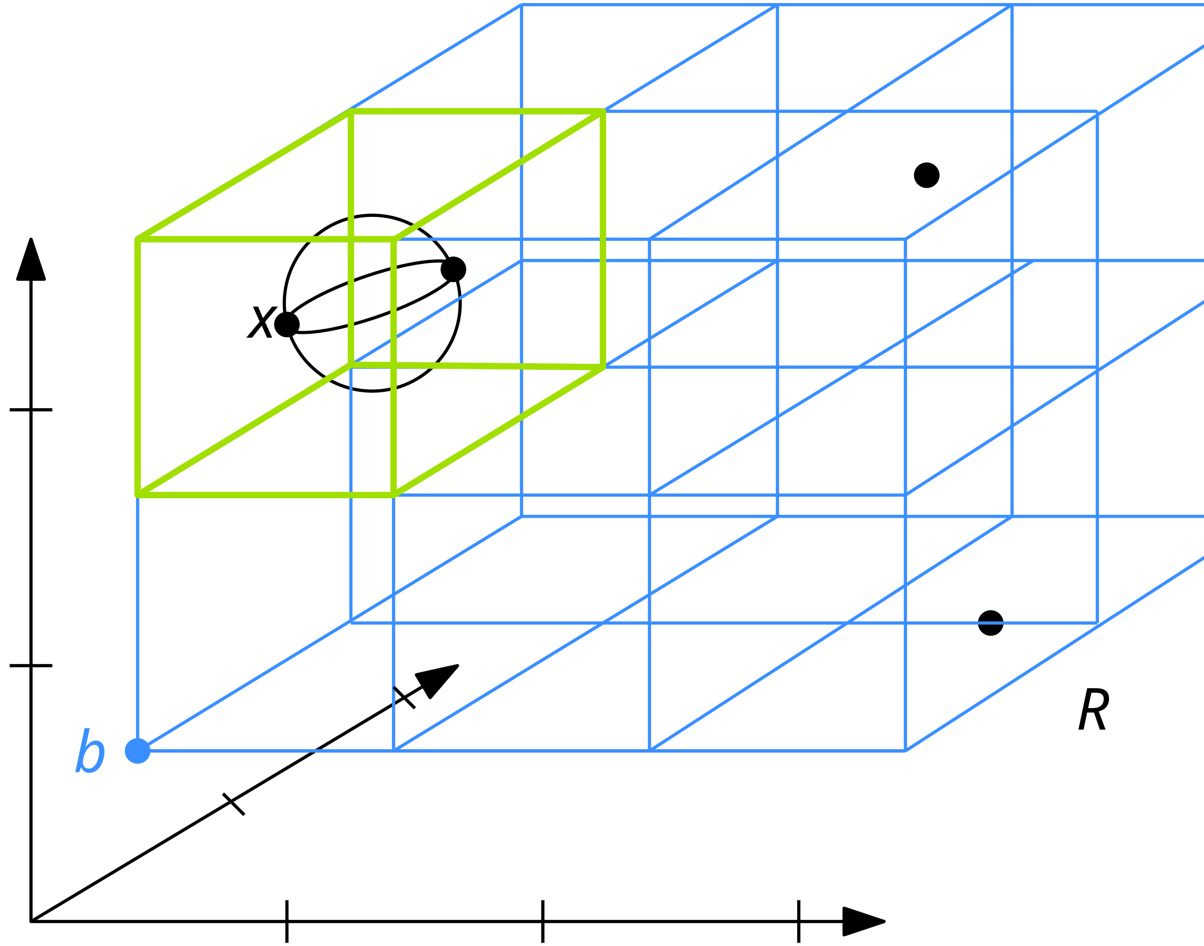


$$h_{b,\Delta}(x) = (h_{b_1,\Delta}(x_1), h_{b_2,\Delta}(x_2), \dots, h_{b_d,\Delta}(x_d))$$

Shifted partition of space

Now let P be a point set in $\mathbb{R}^d$ and $b = (b_1, \dots, b_d)$ uniformly randomly choosen from the hypercube $[0, \Delta]^d$. Consider the (shifted) grid $G^d(b, \Delta)$ with origin in b and sidelength Δ.

Lemma: Let B be a ball in $\mathbb{R}^d$ with Radius r (or an axis-parallel hypercube with sidelength $2r$). The probability that B is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.



$$h_{b,\Delta}(x) = (h_{b_1,\Delta}(x_1), h_{b_2,\Delta}(x_2), \dots, h_{b_d,\Delta}(x_d))$$

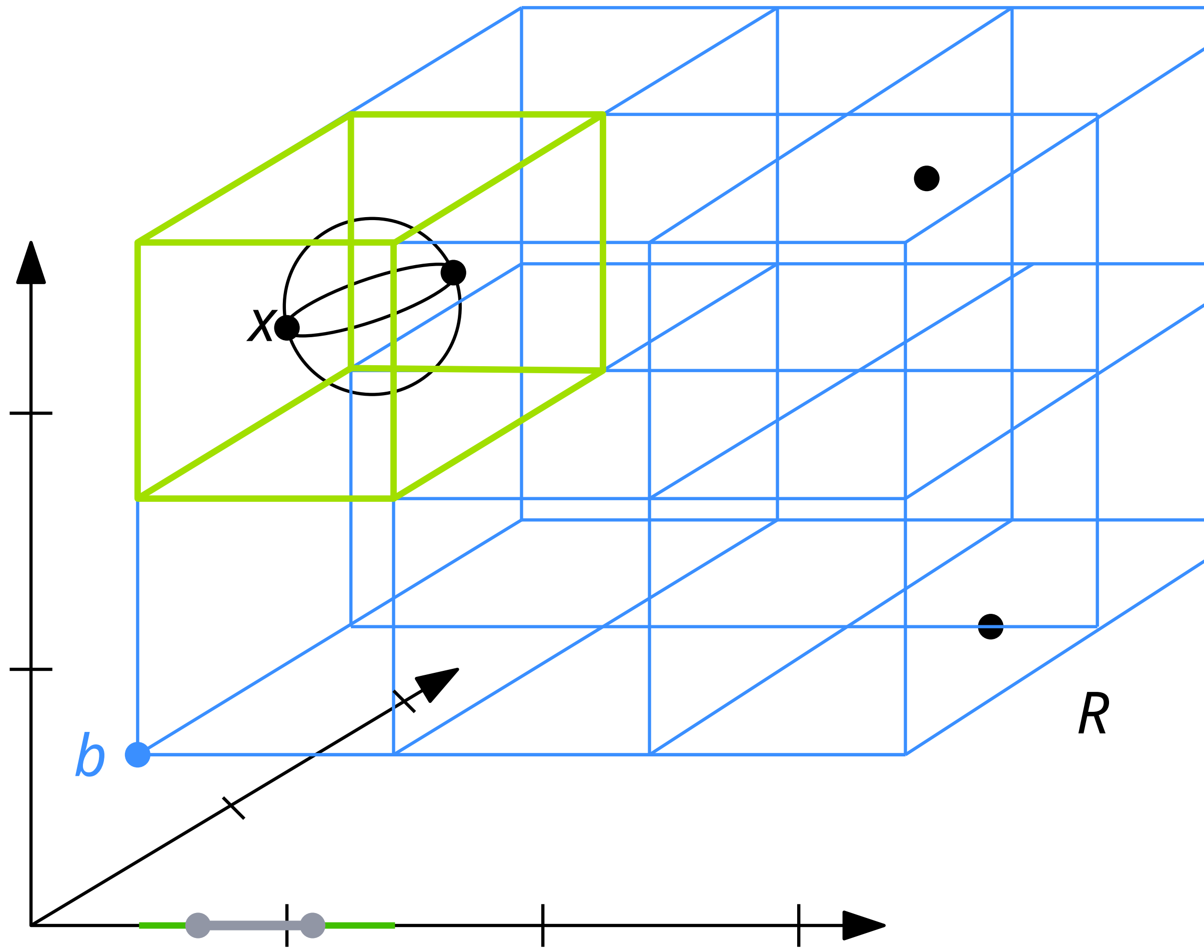
Shifted partition of space

Now let P be a point set in $\mathbb{R}^d$ and $b = (b_1, \ldots, b_d)$ uniformly randomly choosen from the hypercube $[0, \Delta]^d$. Consider the (shifted) grid $G^d(b, \Delta)$ with origin in b and sidelength Δ.

Lemma: Let B be a ball in $\mathbb{R}^d$ with Radius r (or an axis-parallel hypercube with sidelength $2r$). The probability that B is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.

Proof:

project B onto the i^{th} coordinate $\rightarrow$ interval B_i of length $2r$ and shifted 1-dim grid $G^1(b_i, \Delta)$.



$$h_{b,\Delta}(x) = (h_{b_1,\Delta}(x_1), h_{b_2,\Delta}(x_2), \ldots, h_{b_d,\Delta}(x_d))$$

Shifted partition of space

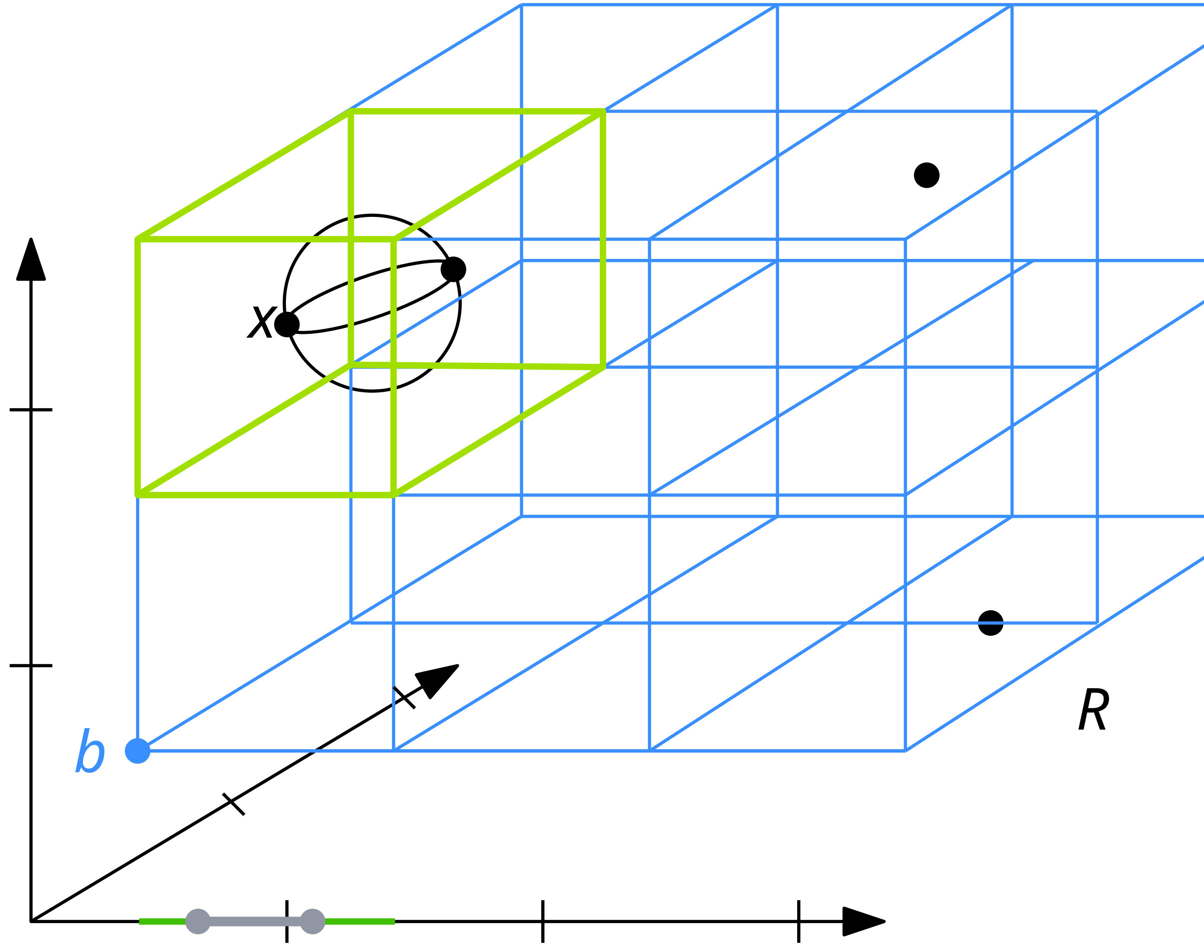
Now let P be a point set in $\mathbb{R}^d$ and $b = (b_1, \dots, b_d)$ uniformly randomly choosen from the hypercube $[0, \Delta]^d$. Consider the (shifted) grid $G^d(b, \Delta)$ with origin in b and sidelength Δ.

Lemma: Let B be a ball in $\mathbb{R}^d$ with Radius r (or an axis-parallel hypercube with sidelength $2r$). The probability that B is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.

Proof:

project B onto the i^{th} coordinate $\rightarrow$ interval B_i of length $2r$ and shifted 1-dim grid $G^1(b_i, \Delta)$.

B in a single cell if all B_i in a shifted interval.



$$h_{b,\Delta}(x) = (h_{b_1,\Delta}(x_1), h_{b_2,\Delta}(x_2), \dots, h_{b_d,\Delta}(x_d))$$

Shifted partition of space

Now let P be a point set in $\mathbb{R}^d$ and $b = (b_1, \dots, b_d)$ uniformly randomly choosen from the hypercube $[0, \Delta]^d$. Consider the (shifted) grid $G^d(b, \Delta)$ with origin in b and sidelength Δ.

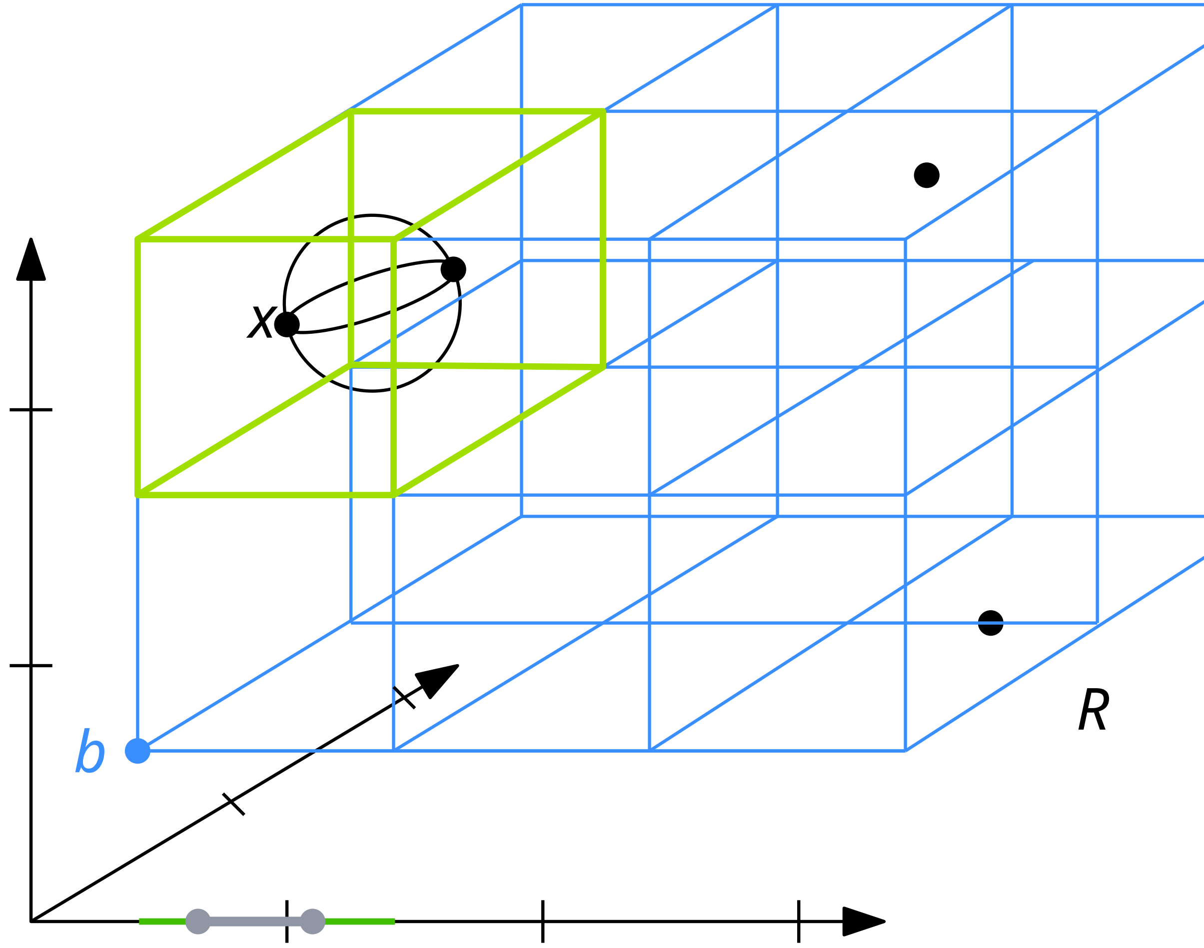
Lemma: Let B be a ball in $\mathbb{R}^d$ with Radius r (or an axis-parallel hypercube with sidelength $2r$). The probability that B is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.

Proof:

project B onto the i^{th} coordinate $\rightarrow$ interval B_i of length $2r$ and shifted 1-dim grid $G^1(b_i, \Delta)$.

B in a single cell if all B_i in a shifted interval.

event E_i: B_i not in a shifted interval.



$$h_{b,\Delta}(x) = (h_{b_1,\Delta}(x_1), h_{b_2,\Delta}(x_2), \dots, h_{b_d,\Delta}(x_d))$$

Shifted partition of space

Now let P be a point set in $\mathbb{R}^d$ and $b = (b_1, \dots, b_d)$ uniformly randomly choosen from the hypercube $[0, \Delta]^d$. Consider the (shifted) grid $G^d(b, \Delta)$ with origin in b and sidelength Δ.

Lemma: Let B be a ball in $\mathbb{R}^d$ with Radius r (or an axis-parallel hypercube with sidelength $2r$). The probability that B is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.

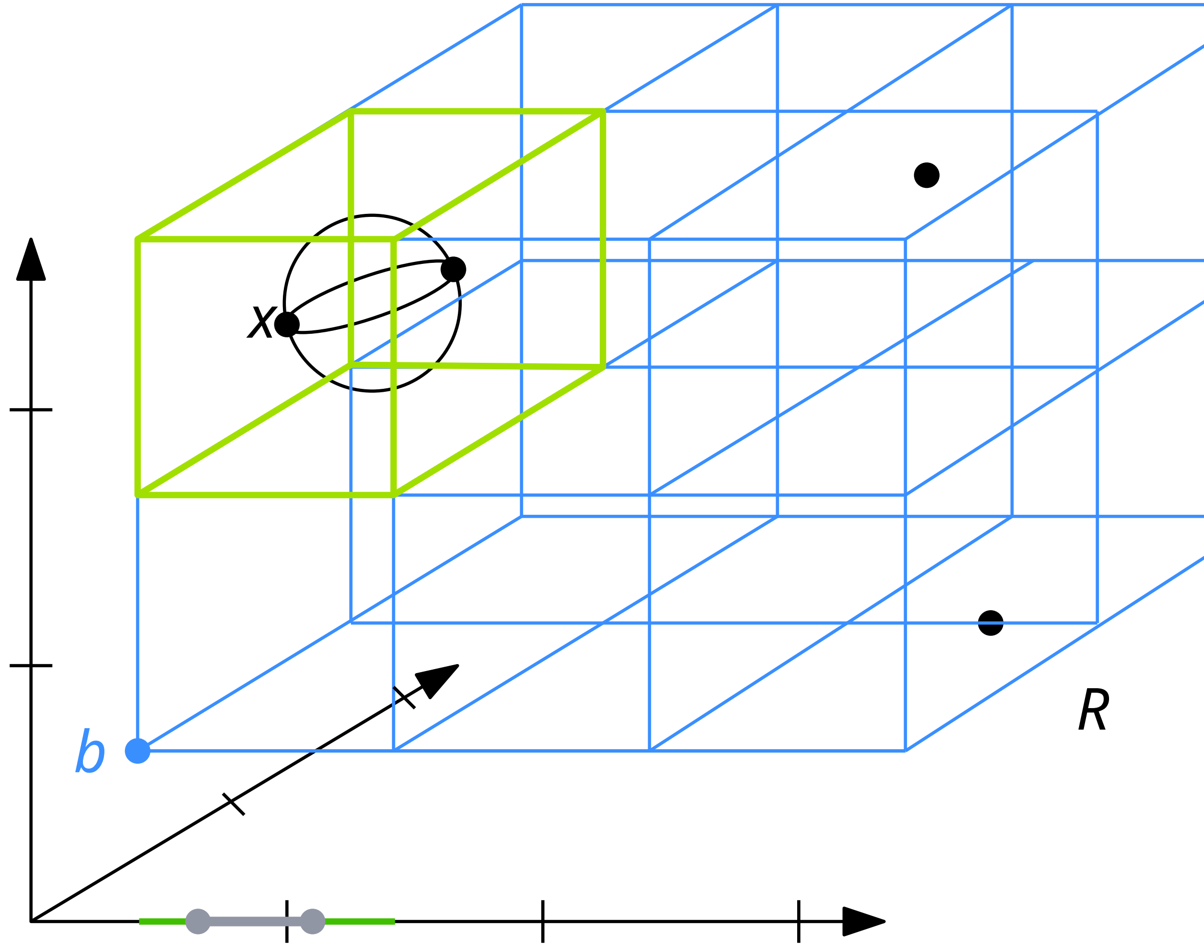
Proof:

project B onto the i^{th} coordinate $\rightarrow$ interval B_i of length $2r$ and shifted 1-dim grid $G^1(b_i, \Delta)$.

B in a single cell if all B_i in a shifted interval.

event E_i: B_i not in a shifted interval.

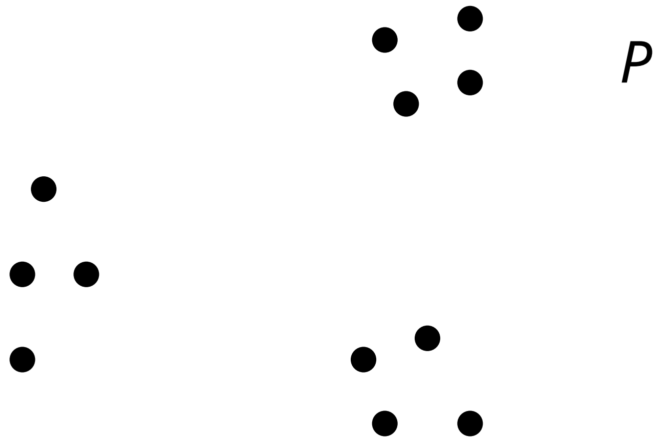
bound: $\mathbb{P}\left[\cup_{i=1}^d E_i\right] \leq \sum_{i=1}^d \mathbb{P}[E_i] \leq 2dr/\Delta$



$h_{b,\Delta}(x) = (h_{b_1,\Delta}(x_1), h_{b_2,\Delta}(x_2), \dots, h_{b_d,\Delta}(x_d))$

Covering with unit disks

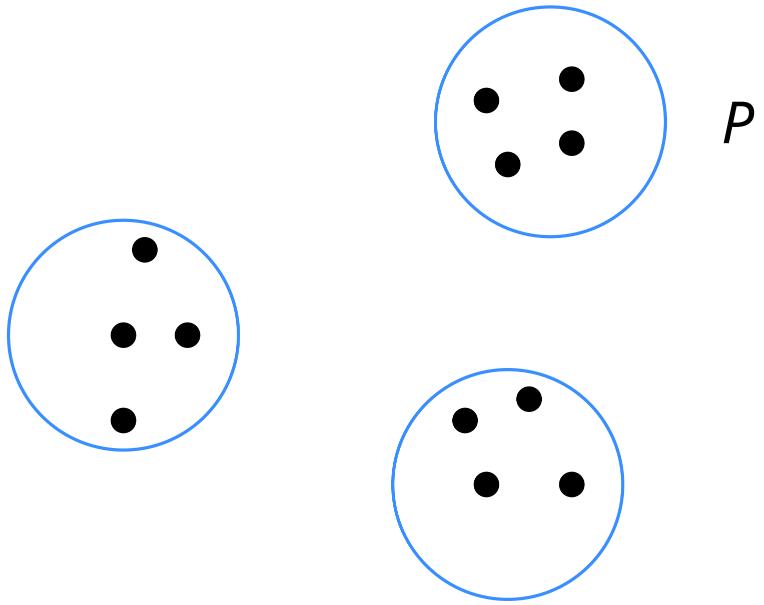
Goal: We want to find a minimal unit disk cover of a point set *P* of *n* points in the plane



Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

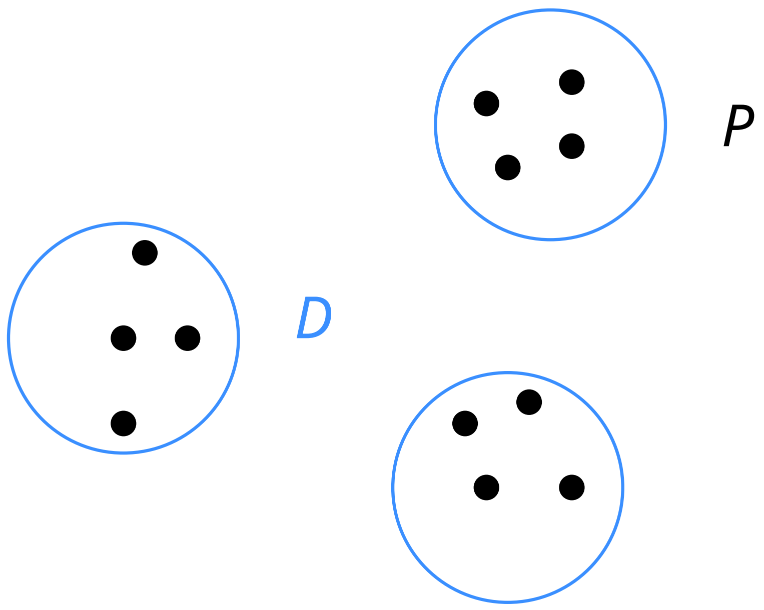


Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

Consider a unit disk D which covers $Q \subseteq P$



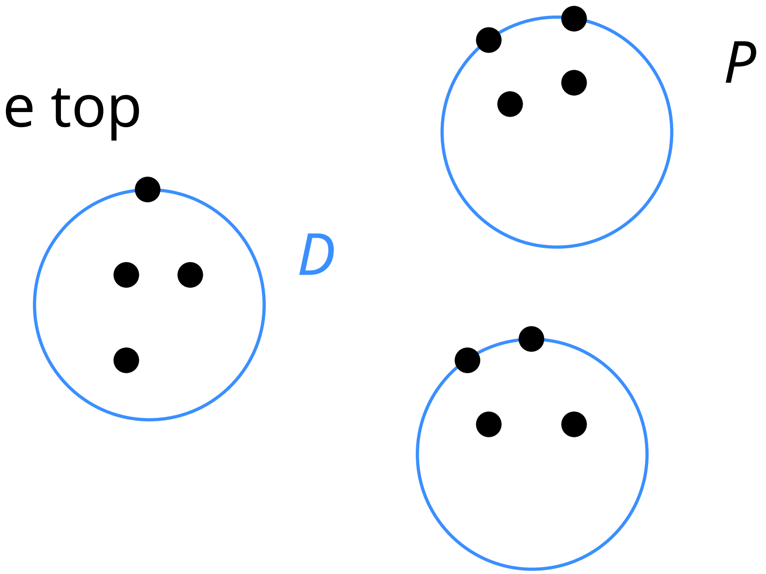
Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

Consider a unit disk D which covers $Q \subseteq P$

We can move D such that two points lie on its boundary or one top



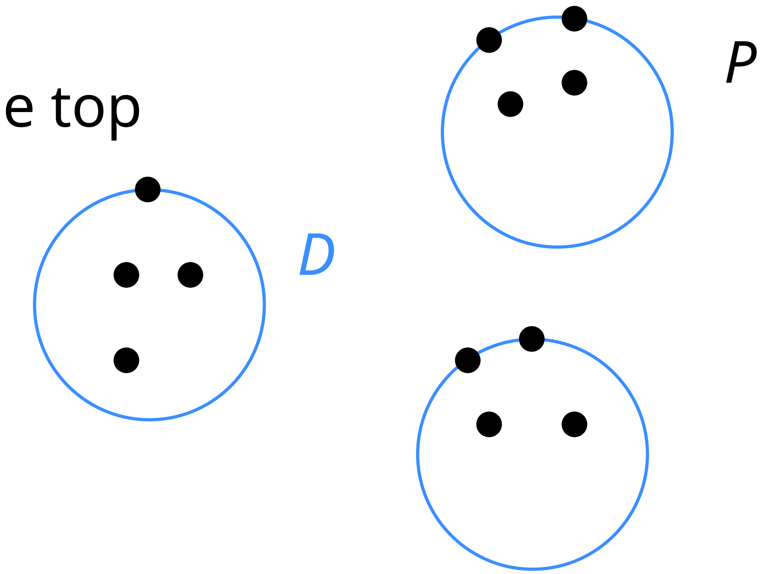
Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

Consider a unit disk D which covers $Q \subseteq P$

We can move D such that two points lie on its boundary or one top



Remark:

- Each pair of points p, q in P defines (at most) two *canonical* unit disks if $||p - q|| \leq 2$.

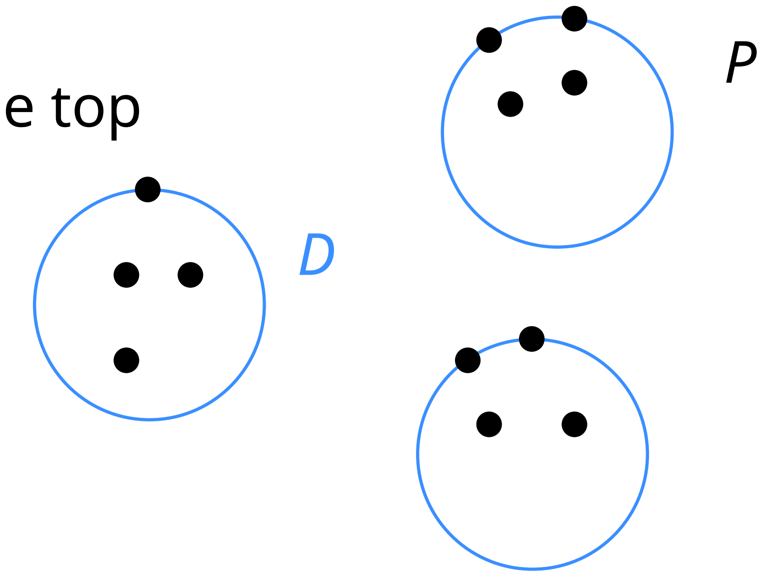
Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

Consider a unit disk D which covers $Q \subseteq P$

We can move D such that two points lie on its boundary or one top



Remark:

- Each pair of points p, q in P defines (at most) two *canonical* unit disks if $||p - q|| \leq 2$.
- The same set $Q \subseteq P$ can be covered by more than one canonical unit disk.

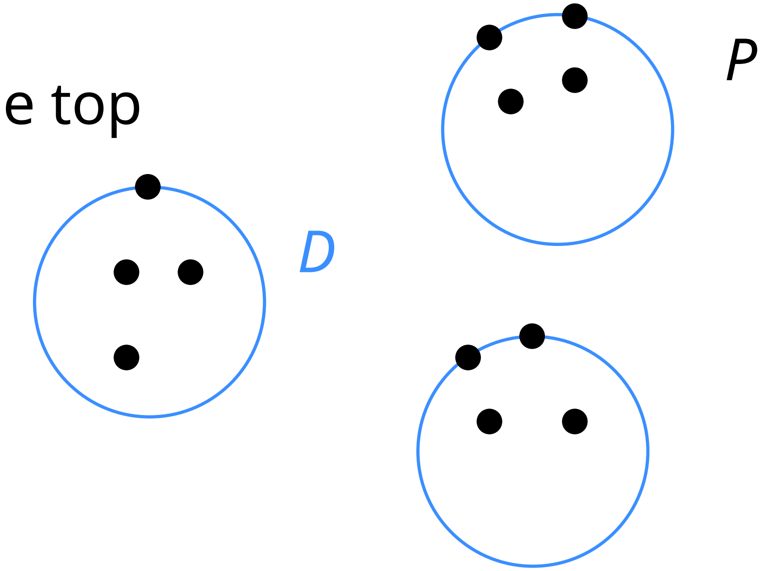
Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

Consider a unit disk D which covers $Q \subseteq P$

We can move D such that two points lie on its boundary or one top



Remark:

- Each pair of points p, q in P defines (at most) two *canonical* unit disks if $||p - q|| \leq 2$.
- The same set $Q \subseteq P$ can be covered by more than one canonical unit disk.

Hence there are at most $2\binom{n}{2} + n \leq n^2$ canonical unit disks; wlog we cover only with these.

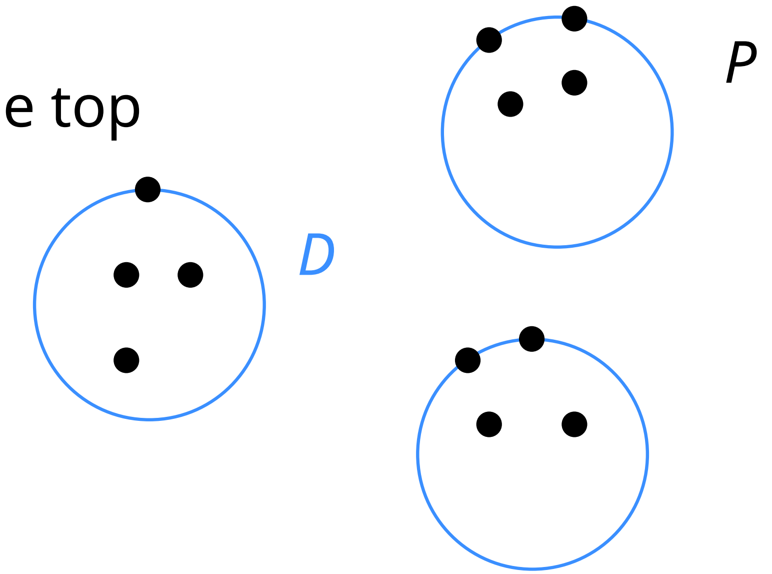
Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

Consider a unit disk D which covers $Q \subseteq P$

We can move D such that two points lie on its boundary or one top



Remark:

- Each pair of points p, q in P defines (at most) two *canonical* unit disks if $||p - q|| \leq 2$.
- The same set $Q \subseteq P$ can be covered by more than one canonical unit disk.

Hence there are at most $2\binom{n}{2} + n \leq n^2$ canonical unit disks; wlog we cover only with these.

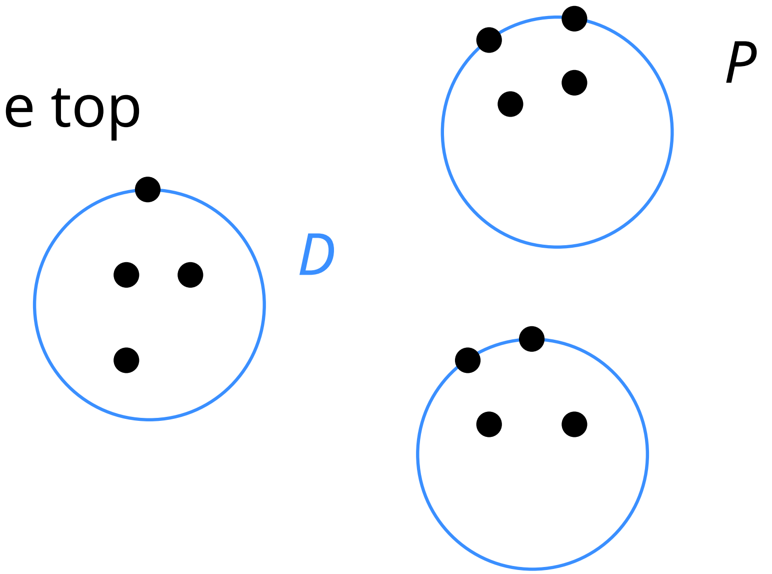
We try all at most n^{2k} many $k-$ tuples of these; each we test in $O(nk)$ time.

Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

Consider a unit disk D which covers $Q \subseteq P$
We can move D such that two points lie on its boundary or one top



Remark:

- Each pair of points p, q in P defines (at most) two *canonical* unit disks if $||p - q|| \leq 2$.
- The same set $Q \subseteq P$ can be covered by more than one canonical unit disk.

Hence there are at most $2\binom{n}{2} + n \leq n^2$ canonical unit disks; wlog we cover only with these.

We try all at most n^{2k} many $k-$ tuples of these; each we test in $O(nk)$ time.

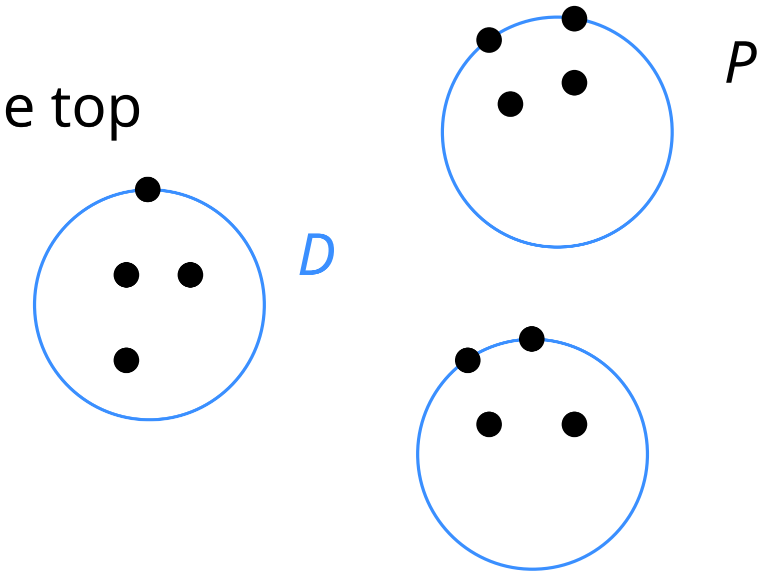
Lemma: For n points in $\mathbb{R}^2$, we can determine in $O(kn^{2k+1})$ time if a k unit disk cover exists.

Covering with unit disks

Goal: We want to find a minimal unit disk cover of a point set P of n points in the plane

First we can test if a cover of size k exists in time $O(kn^{2k+1})$

Consider a unit disk D which covers $Q \subseteq P$
We can move D such that two points lie on its boundary or one top



Remark:

- Each pair of points p, q in P defines (at most) two *canonical* unit disks if $||p - q|| \leq 2$.
- The same set $Q \subseteq P$ can be covered by more than one canonical unit disk.

Hence there are at most $2\binom{n}{2} + n \leq n^2$ canonical unit disks; wlog we cover only with these.

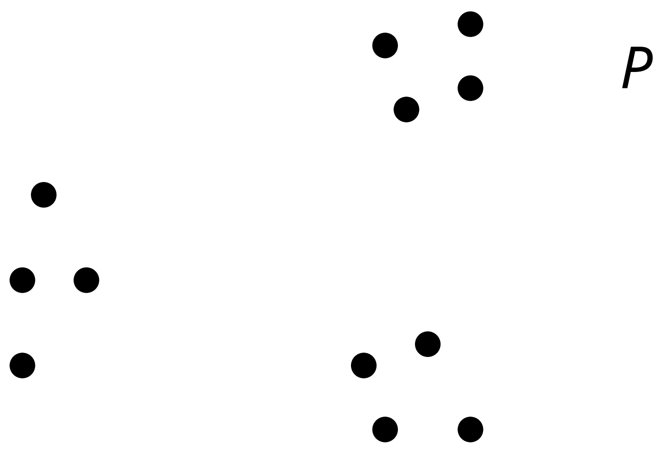
We try all at most n^{2k} many $k-$ tuples of these; each we test in $O(nk)$ time.

Lemma: For n points in $\mathbb{R}^2$, we can determine in $O(kn^{2k+1})$ time if a k unit disk cover exists.

but k can be linear in n

Faster approximate covering with unit disks

Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$



Faster approximate covering with unit disks

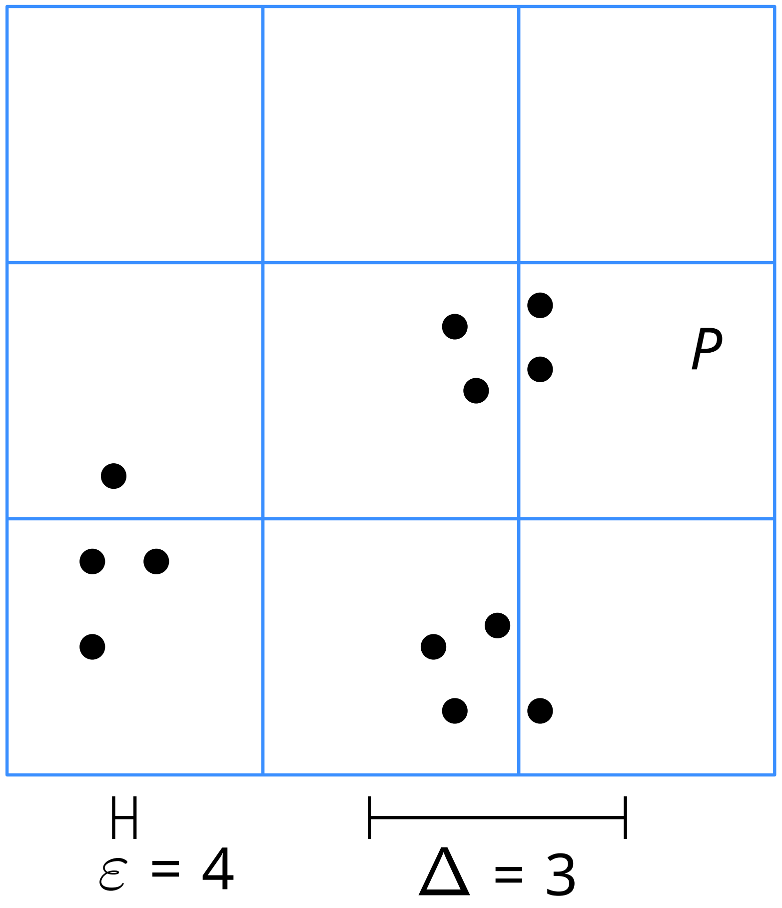
Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

    compute minimal # unit disks containing all points in cell

    using slow algorithm



Faster approximate covering with unit disks

Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

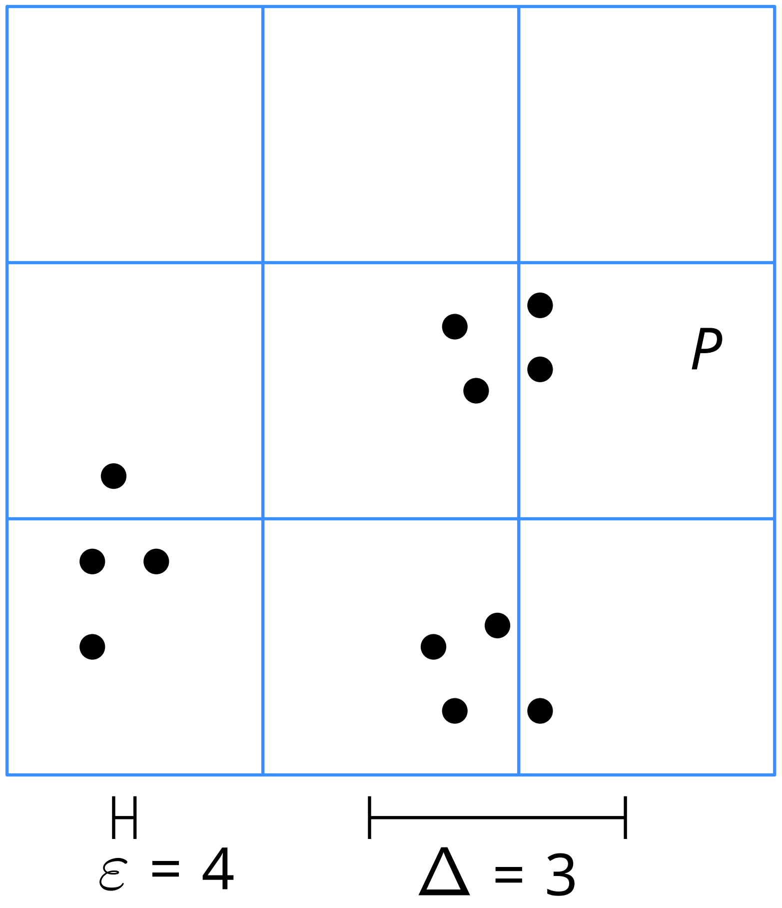
- compute all grid cells containing points in P
- for each non-empty grid cell

    compute minimal # unit disks containing all points in cell

    using slow algorithm

Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$



Faster approximate covering with unit disks

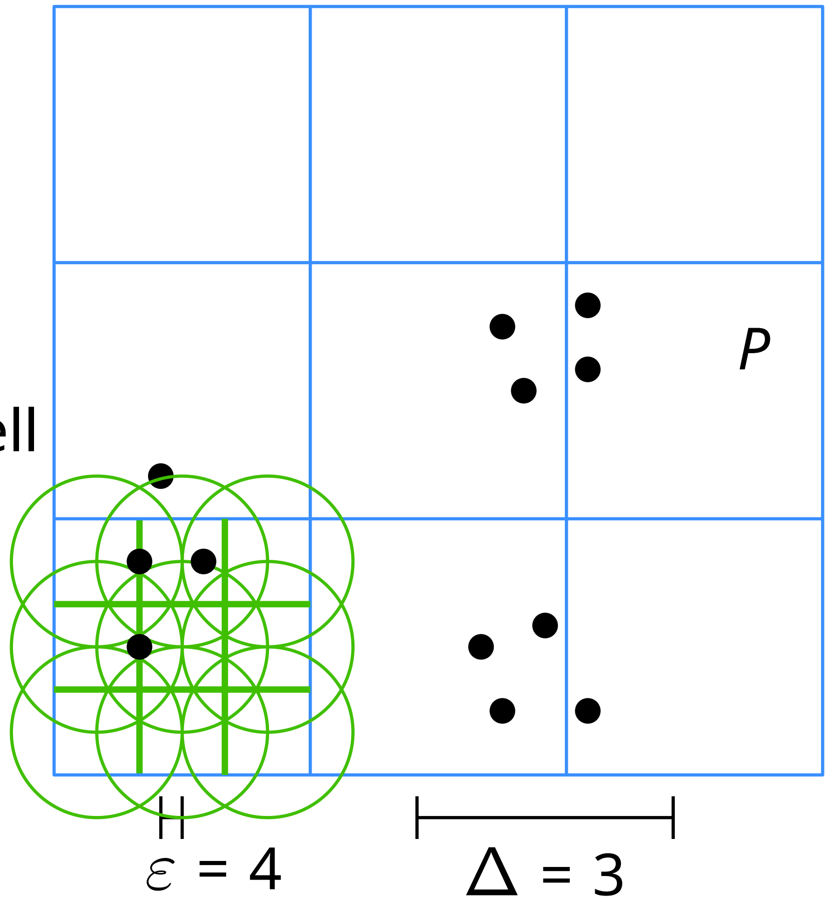
Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

  compute minimal # unit disks containing all points in cell

  using slow algorithm



Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$

  using hashing and the fact that each grid cell can be covered by $(\Delta + 1)^2 = O(1/\varepsilon^2)$ many unit disks; hence for at most n cells we can compute this in $O(Mn^{2M+2}) = n^{O(1/\varepsilon^2)}$ time

Faster approximate covering with unit disks

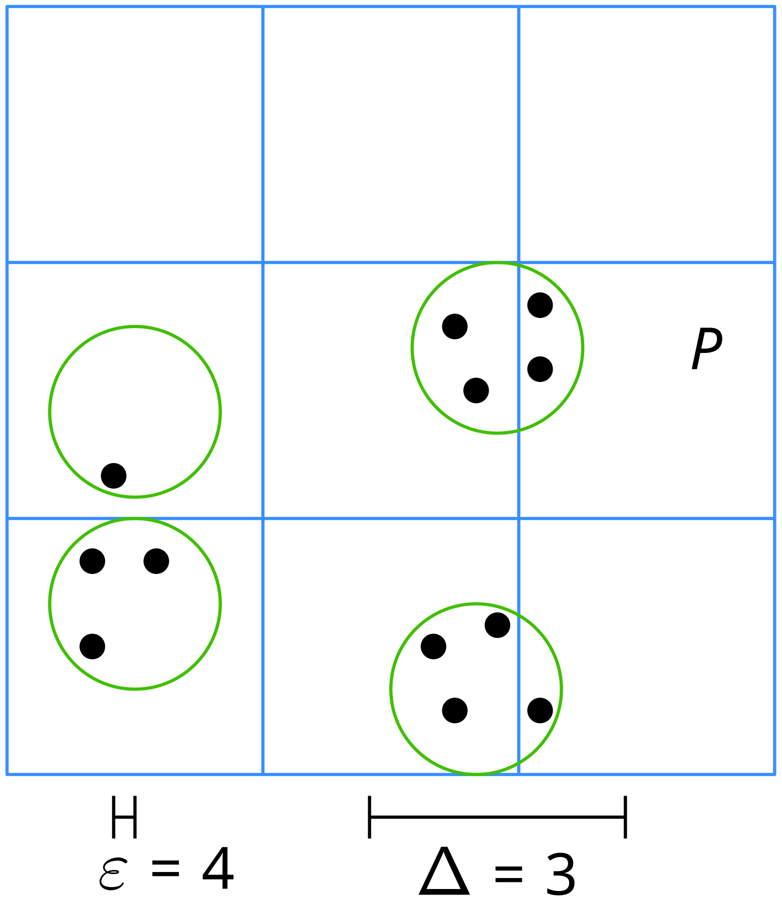
Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

  compute minimal # unit disks containing all points in cell

  using slow algorithm



Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$
- at most $(1 + \varepsilon)opt$ disks are computed in expectation

Faster approximate covering with unit disks

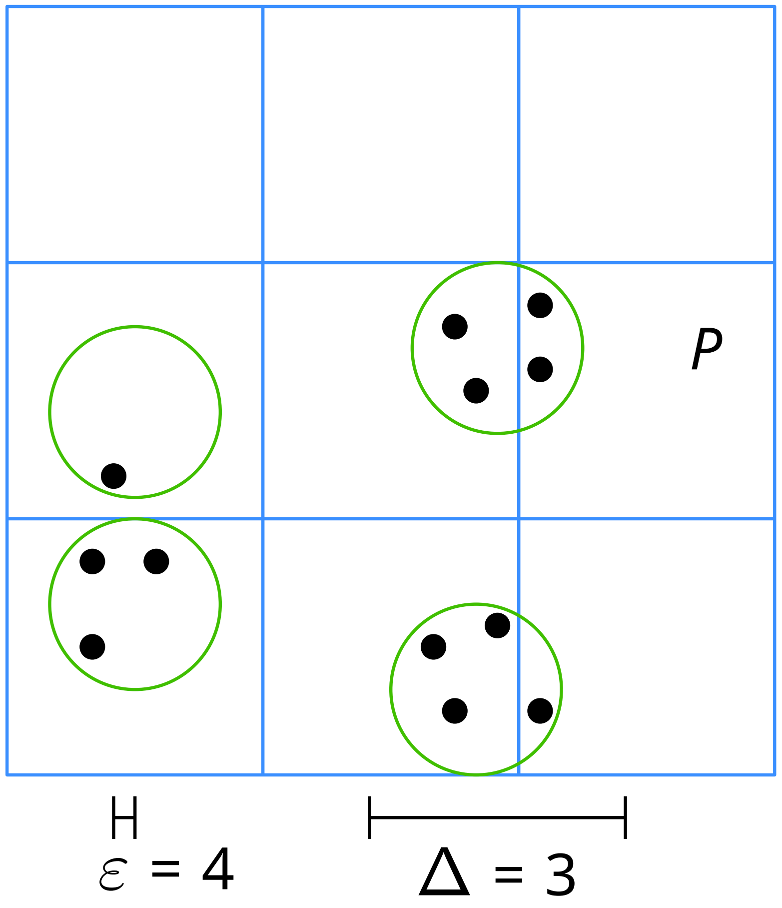
Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

  compute minimal # unit disks containing all points in cell

  using slow algorithm



Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$
- at most $(1 + \varepsilon)opt$ disks are computed in expectation

  $F = \{D_1, \dots, D_{opt}\}$: optimal solution

  We construct a valid solution G from F that the algorithm finds.

Faster approximate covering with unit disks

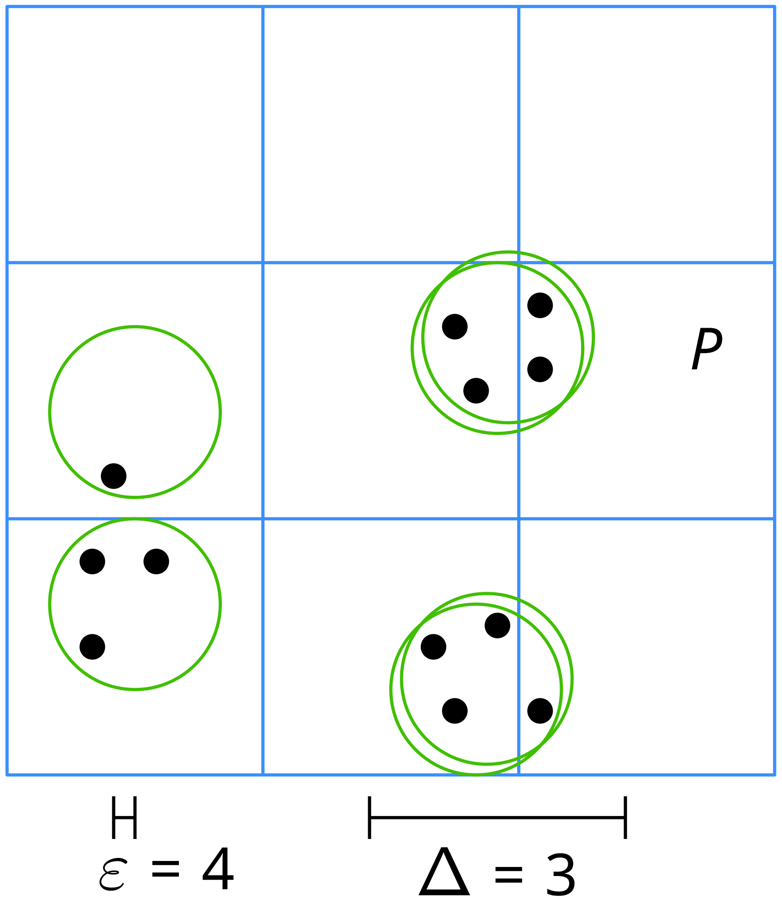
Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

  compute minimal # unit disks containing all points in cell

  using slow algorithm



Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$
- at most $(1+\varepsilon)opt$ disks are computed in expectation

  $F = \{D_1, \dots, D_{opt}\}$: optimal solution

  We construct a valid solution G from F that the algorithm finds.

  For each grid cell C: F_C the disks in F that intersect C. Let $G = \cup_C F_C$ (a multiset).

Faster approximate covering with unit disks

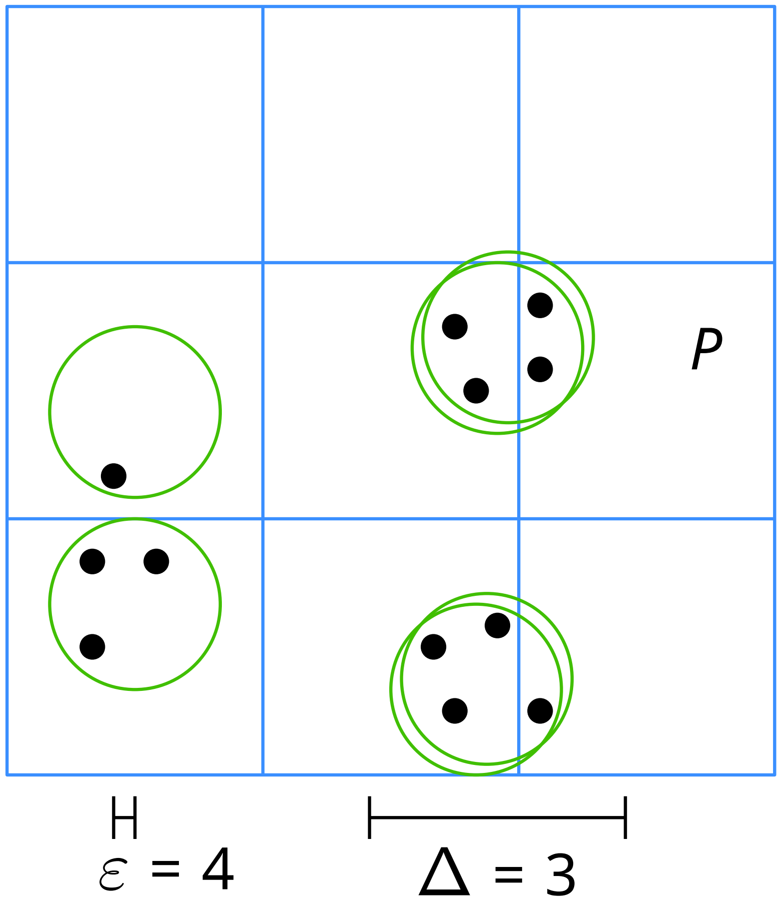
Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

  compute minimal # unit disks containing all points in cell

  using slow algorithm



Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$
- at most $(1+\varepsilon)opt$ disks are computed in expectation

  $F = \{D_1, \ldots, D_{opt}\}$: optimal solution

  We construct a valid solution G from F that the algorithm finds.

  For each grid cell C: F_C the disks in F that intersect C. Let $G = \cup_C F_C$ (a multiset).

  For each cell C the algorithm returns at most $|F_C|$ disks.

Faster approximate covering with unit disks

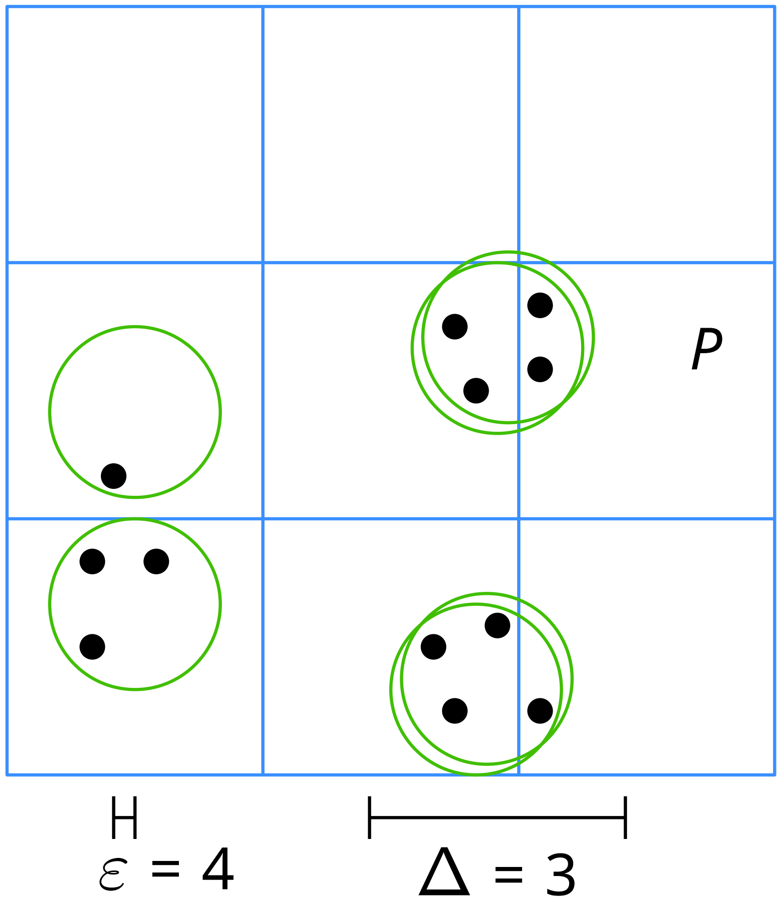
Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

  compute minimal # unit disks containing all points in cell

  using slow algorithm



Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$
- at most $(1+\varepsilon)opt$ disks are computed in expectation

  $F = \{D_1, \ldots, D_{opt}\}$: optimal solution

  We construct a valid solution G from F that the algorithm finds.

  For each grid cell C: F_C the disks in F that intersect C. Let $G = \cup_C F_C$ (a multiset).

  For each cell C the algorithm returns at most $|F_C|$ disks.

  For $\varepsilon < 12$ each disk D in F intersects ≤ 4 cells, thus appears at most 4 times in G.

Faster approximate covering with unit disks

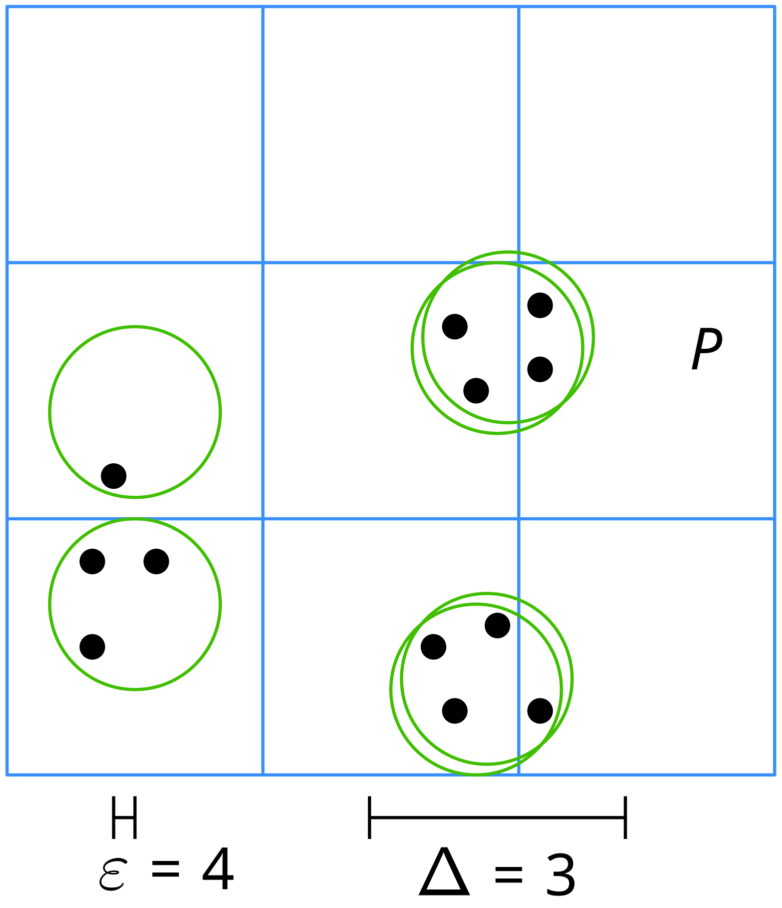
Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$

Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

  compute minimal # unit disks containing all points in cell

  using slow algorithm



Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$
- at most $(1 + \varepsilon)opt$ disks are computed in expectation

  $F = \{D_1, \ldots, D_{opt}\}$: optimal solution

  We construct a valid solution G from F that the algorithm finds.

  For each grid cell C: F_C the disks in F that intersect C. Let $G = \cup_C F_C$ (a multiset).

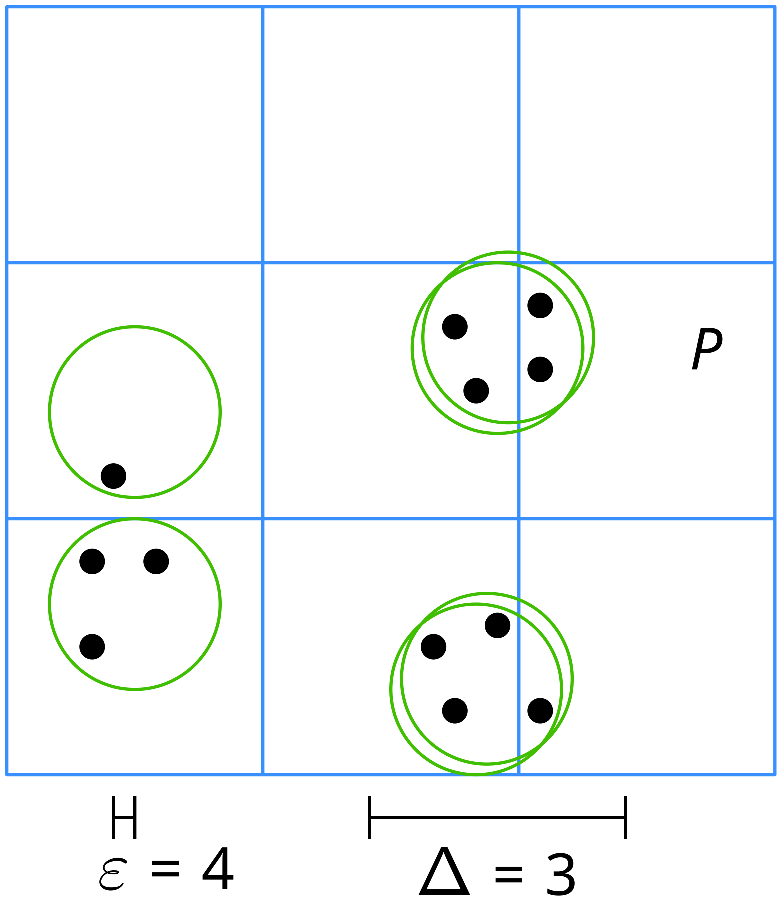
  For each cell C the algorithm returns at most $|F_C|$ disks.

  For $\varepsilon < 12$ each disk D in F intersects ≤ 4 cells, thus appears at most 4 times in G.

  Disk D_i in F appears more than once in $G \Leftrightarrow D_i$ not in one cell; (X_i := indicator variable of this event)

Faster approximate covering with unit disks

Let $\Delta = 12/\varepsilon$ and consider shifted grid $G^2(b, \Delta)$



Algorithm

- compute all grid cells containing points in P
- for each non-empty grid cell

  compute minimal # unit disks containing all points in cell

  using slow algorithm

Analysis:

- the running time is $n^{O(1/\varepsilon^2)}$
- at most $(1 + \varepsilon)opt$ disks are computed in expectation

  $F = \{D_1, \dots, D_{opt}\}$: optimal solution

  We construct a valid solution G from F that the algorithm finds.

  For each grid cell C: F_C the disks in F that intersect C. Let $G = \cup_C F_C$ (a multiset).

  For each cell C the algorithm returns at most $|F_C|$ disks.

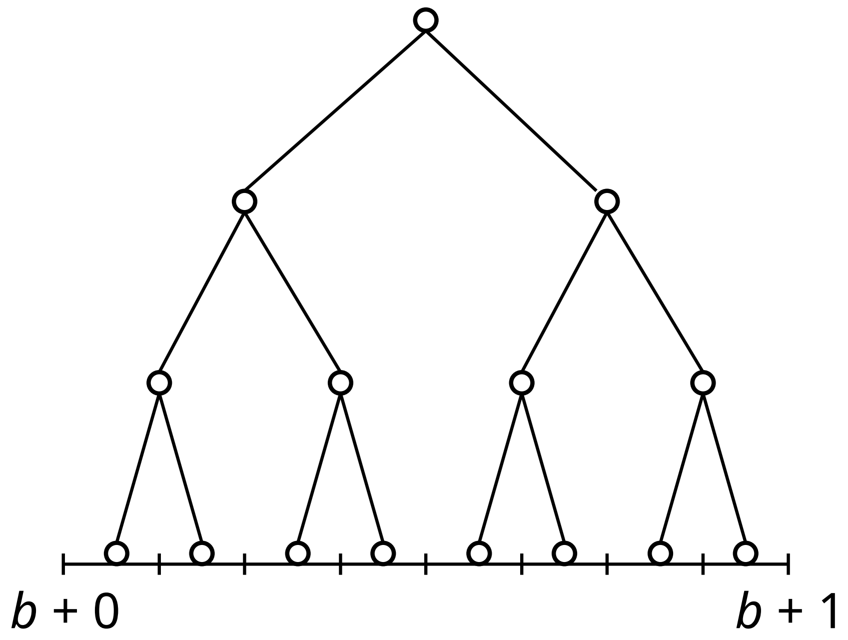
  For $\varepsilon < 12$ each disk D in F intersects ≤ 4 cells, thus appears at most 4 times in G.

  Disk D_i in F appears more than once in $G \Leftrightarrow D_i$ not in one cell; (X_i := indicator variable of this event)

  $\mathbb{E}\left[|G|\right] \leq \mathbb{E}\left[opt + \sum_{i=1}^{opt} 3X_i\right] \leq opt + \sum_{i=1}^{opt} 3\mathbb{E}\left[X_i\right] \leq opt + \sum_{i=1}^{opt} 3\frac{4}{\Delta} = (1 + \frac{12}{\Delta})opt = (1 + \varepsilon)opt$

Shifting Quadtrees in 1 dimension

Given point set P of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]$. Draw $b \in \left[0, \frac{1}{2}\right]$ uniformly at random.
Consider 1-dim Quadtree T on P with root interval $b + [0, 1]$

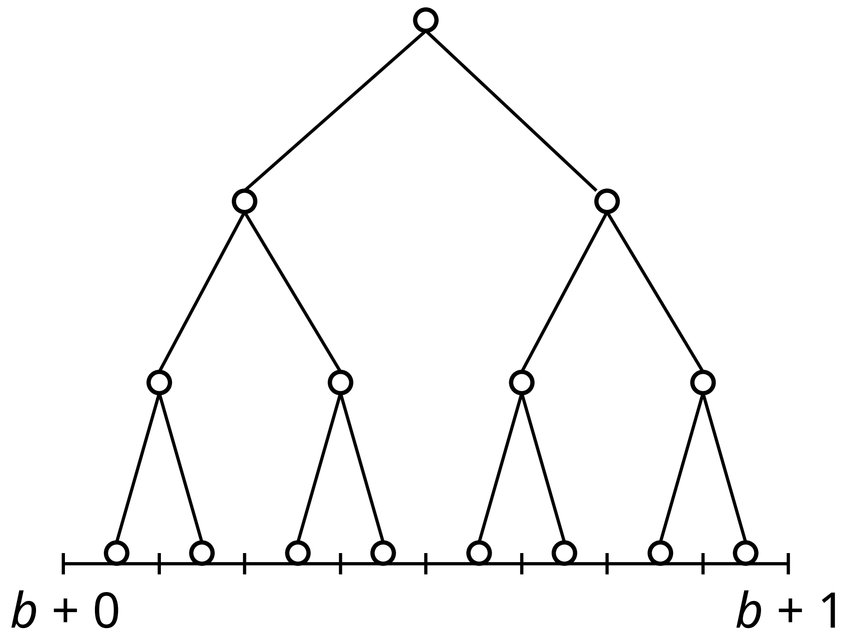


Shifting Quadtrees in 1 dimension

Given point set P of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]$. Draw $b \in \left[0, \frac{1}{2}\right]$ uniformly at random.

Consider 1-dim Quadtree T on P with root interval $b + [0, 1]$

For $\alpha, \beta \in P$ let $\mathbb{L}_b(\alpha, \beta) = 1 - bit_\Delta(\alpha - b, \beta - b) = level(lca(\alpha, \beta) \text{ in } T)$

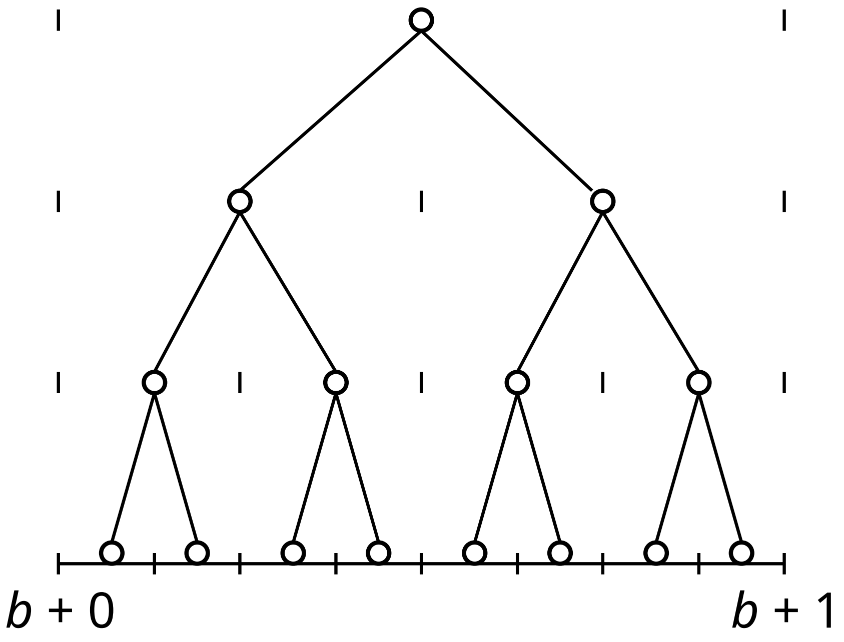


Shifting Quadtrees in 1 dimension

Given point set P of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]$. Draw $b \in \left[0, \frac{1}{2}\right]$ uniformly at random.
Consider 1-dim Quadtree T on P with root interval $b + [0, 1]$

For $\alpha, \beta \in P$ let $\mathbb{L}_b(\alpha, \beta) = 1 - bit_\Delta(\alpha - b, \beta - b) = level(lca(\alpha, \beta) \text{ in } T)$

note that $\mathbb{L}_b(\alpha, \beta)$ only depends on α, β, b and can be precomputed



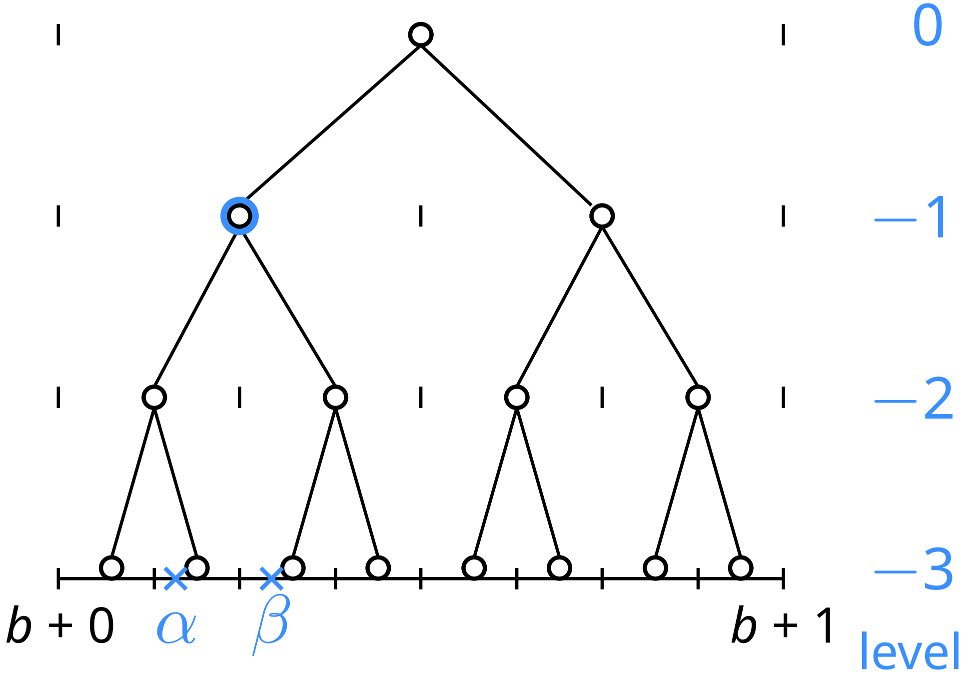
Shifting Quadtrees in 1 dimension

Given point set P of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]$. Draw $b \in \left[0, \frac{1}{2}\right]$ uniformly at random.
Consider 1-dim Quadtree T on P with root interval $b + [0, 1]$

For $\alpha, \beta \in P$ let $\mathbb{L}_b(\alpha, \beta) = 1 - bit_\Delta(\alpha - b, \beta - b) = level(lca(\alpha, \beta) \text{ in } T)$

note that $\mathbb{L}_b(\alpha, \beta)$ only depends on α, β, b and can be precomputed

Lemma: Let $\alpha, \beta \in \left[\frac{1}{2}, \frac{3}{4}\right]$ and $b \in \left[0, \frac{1}{2}\right]$.
For $t \in \mathbb{N}$ holds $\mathbb{P}\left[\mathbb{L}_b(\alpha, \beta) > \log_2 |\alpha - \beta| + t\right] \leq \frac{4}{2^t}$



$\mathbb{L}_b(\alpha, \beta) = -1$

Shifting Quadtrees in 1 dimension

Given point set P of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]$. Draw $b \in \left[0, \frac{1}{2}\right]$ uniformly at random.
Consider 1-dim Quadtree T on P with root interval $b + [0, 1]$

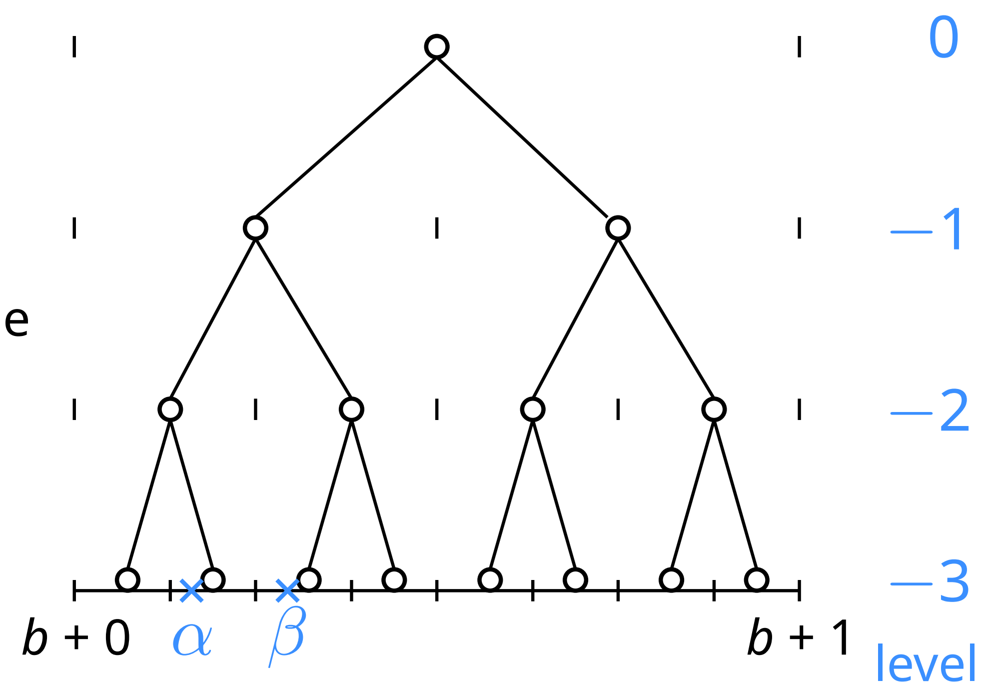
For $\alpha, \beta \in P$ let $\mathbb{L}_b(\alpha, \beta) = 1 - bit_\Delta(\alpha - b, \beta - b) = level(lca(\alpha, \beta) \text{ in } T)$

note that $\mathbb{L}_b(\alpha, \beta)$ only depends on α, β, b and can be precomputed

Lemma: Let $\alpha, \beta \in \left[\frac{1}{2}, \frac{3}{4}\right]$ and $b \in \left[0, \frac{1}{2}\right]$.
For $t \in \mathbb{N}$ holds $\mathbb{P}\left[\mathbb{L}_b(\alpha, \beta) > \log_2 |\alpha - \beta| + t\right] \leq \frac{4}{2^t}$

Proof:

Let $M = \lfloor \log_2 |\alpha - \beta| \rfloor$ and consider shifted partition of real line with side length $\Delta_{M+i} = 2^{M+i}$ and shift b.



Shifting Quadtrees in 1 dimension

Given point set P of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]$. Draw $b \in \left[0, \frac{1}{2}\right]$ uniformly at random.
Consider 1-dim Quadtree T on P with root interval $b + [0, 1]$

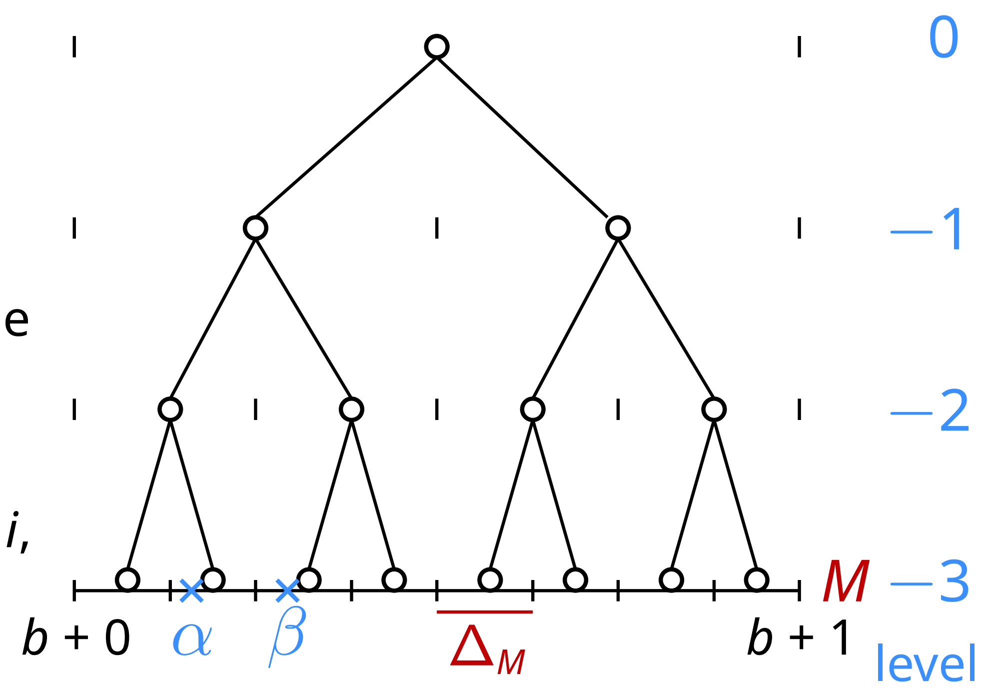
For $\alpha, \beta \in P$ let $\mathbb{L}_b(\alpha, \beta) = 1 - bit_\Delta(\alpha - b, \beta - b) = level(lca(\alpha, \beta) \text{ in } T)$

note that $\mathbb{L}_b(\alpha, \beta)$ only depends on α, β, b and can be precomputed

Lemma: Let $\alpha, \beta \in \left[\frac{1}{2}, \frac{3}{4}\right]$ and $b \in \left[0, \frac{1}{2}\right]$.
For $t \in \mathbb{N}$ holds $\mathbb{P}\left[\mathbb{L}_b(\alpha, \beta) > \log_2 |\alpha - \beta| + t\right] \leq \frac{4}{2^t}$

Proof:

Let $M = \lfloor \log_2 |\alpha - \beta| \rfloor$ and consider shifted partition of real line with side length $\Delta_{M+i} = 2^{M+i}$ and shift b.
Let $X_{M+i} = \mathbb{1}_{\alpha, \beta \text{ in different intervals}}$.
If $\mathbb{L}_b(\alpha, \beta) = M + i$, then α, β lie in the same interval at level $M + i$, but not $M + i - 1$. By previous lemma $\mathbb{P}[X_{M+i} = 1] \leq \frac{|\alpha - \beta|}{\Delta_{M+i}}$.



Shifting Quadtrees in 1 dimension

Given point set P of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]$. Draw $b \in \left[0, \frac{1}{2}\right]$ uniformly at random.
Consider 1-dim Quadtree T on P with root interval $b + [0, 1]$

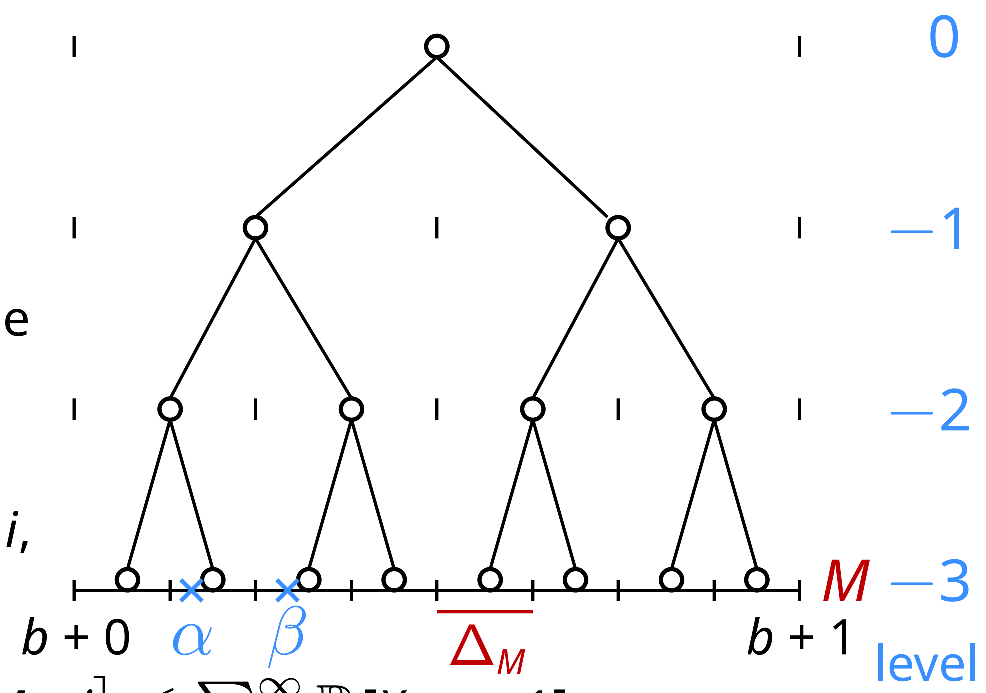
For $\alpha, \beta \in P$ let $\mathbb{L}_b(\alpha, \beta) = 1 - bit_\Delta(\alpha - b, \beta - b) = level(lca(\alpha, \beta) \text{ in } T)$

note that $\mathbb{L}_b(\alpha, \beta)$ only depends on α, β, b and can be precomputed

Lemma: Let $\alpha, \beta \in \left[\frac{1}{2}, \frac{3}{4}\right]$ and $b \in \left[0, \frac{1}{2}\right]$.
For $t \in \mathbb{N}$ holds $\mathbb{P}\left[\mathbb{L}_b(\alpha, \beta) > \log_2 |\alpha - \beta| + t\right] \leq \frac{4}{2^t}$

Proof:

Let $M = \lfloor \log_2 |\alpha - \beta| \rfloor$ and consider shifted partition of real line with side length $\Delta_{M+i} = 2^{M+i}$ and shift b.
Let $X_{M+i} = \mathbb{1}_{\alpha, \beta \text{ in different intervals}}$.
If $\mathbb{L}_b(\alpha, \beta) = M + i$, then α, β lie in the same interval at level $M + i$, but not $M + i - 1$. By previous lemma $\mathbb{P}[X_{M+i} = 1] \leq \frac{|\alpha - \beta|}{\Delta_{M+i}}$.



Hence $\mathbb{P}\left[\mathbb{L}_b(\alpha, \beta) > \log_2 |\alpha - \beta| + t\right] \leq \sum_{i=1+t}^{\infty} \mathbb{P}\left[\mathbb{L}_b(\alpha, \beta) = M + i\right] \leq \sum_{i=t}^{\infty} \mathbb{P}[X_{M+i} = 1]$
$\leq \sum_{i=t}^{\infty} \frac{|\alpha - \beta|}{\Delta_{M+i}} \leq \sum_{i=t}^{\infty} 2^{1-i} \leq 2^{2-t}$

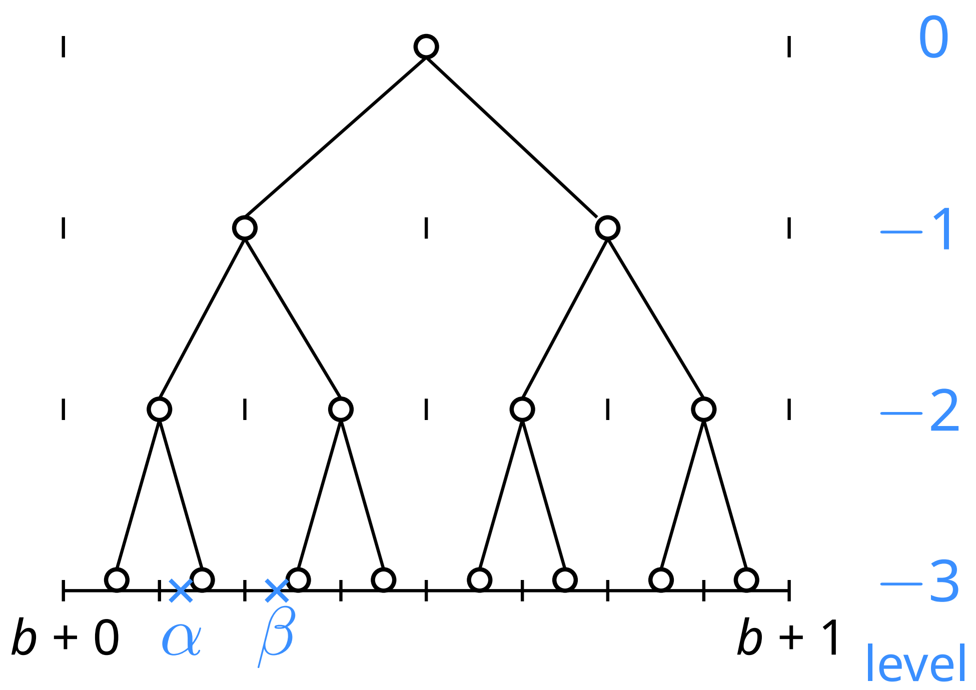
Shifting Quadtrees in 1 dimension

Given point set P of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]$. Draw $b \in \left[0, \frac{1}{2}\right]$ uniformly at random.
Consider 1-dim Quadtree T on P with root interval $b + [0, 1]$

For $\alpha, \beta \in P$ let $\mathbb{L}_b(\alpha, \beta) = 1 - bit_\Delta(\alpha - b, \beta - b) = level(lca(\alpha, \beta) \text{ in } T)$

note that $\mathbb{L}_b(\alpha, \beta)$ only depends on α, β, b and can be precomputed

Lemma: Let $\alpha, \beta \in \left[\frac{1}{2}, \frac{3}{4}\right]$ and $b \in \left[0, \frac{1}{2}\right]$.
For $t \in \mathbb{N}$ holds $\mathbb{P}\left[\mathbb{L}_b(\alpha, \beta) > \log_2|\alpha - \beta| + t\right] \leq \frac{4}{2^t}$



Corollary: Let $\alpha, \beta \in \left[\frac{1}{2}, \frac{3}{4}\right]$ and $b \in \left[0, \frac{1}{2}\right]$.
For $c > 1$ holds $\mathbb{P}\left[\mathbb{L}_b(\alpha, \beta) > \log_2|\alpha - \beta| + c \log n\right] \leq \frac{4}{n^c}$ where $|P| = n$.

Shifting Quadtrees in higher dimensions

Now let P be a set of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]^d$ and b in $\left[0, \frac{1}{2}\right]^d$.

Consider the shifted compressed quadtree T of P with $b + [0, 1]^d$ as root cell.

Shifting Quadtrees in higher dimensions

Now let P be a set of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]^d$ and b in $\left[0, \frac{1}{2}\right]^d$.

Consider the shifted compressed quadtree T of P with $b + [0, 1]^d$ as root cell.

As before, for $p, q \in P$ consider $lca(p, q)$ in T.

Note that T is the combination of 1dim Quadtrees $T_1, \dots, T_d$ in each coordinate.

Hence $\mathbb{L}_b(p, q) = \max_{i=1}^{d} \mathbb{L}_{b_i}(p_i, q_i)$ and is again independent of all other points in P.

Shifting Quadtrees in higher dimensions

Now let P be a set of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]^d$ and b in $\left[0, \frac{1}{2}\right]^d$.

Consider the shifted compressed quadtree T of P with $b + [0, 1]^d$ as root cell.

As before, for $p, q \in P$ consider $lca(p, q)$ in T.
Note that T is the combination of 1dim Quadtrees $T_1, \ldots, T_d$ in each coordinate.
Hence $\mathbb{L}_b(p, q) = \max_{i=1}^{d} \mathbb{L}_{b_i}(p_i, q_i)$ and is again independent of all other points in P.

We consider $\mathbb{L}_b(p, q)$ as random variable and use
Lemma
For $t > 0$ holds $\mathbb{P}\left[\mathbb{L}_b(p, q) > \log_2 ||p - q|| + t\right] \leq \frac{4d}{2^t}$.

Low quality ANN-Search

Now we want to use shifted quadtrees to quickly answer ANN-queries in $\mathbb{R}^d$.
That is, we want to preprocess a set P of n points in $\mathbb{R}^d$, so that for query point q we can quickly find $p \in P$, s.t. $||q - p|| \leq \tau\, d(q, P)$ where $d(q, P) = min_{p \in P} ||q - p||$.

Low quality ANN-Search

Now we want to use shifted quadtrees to quickly answer ANN-queries in $\mathbb{R}^d$.
That is, we want to preprocess a set P of n points in $\mathbb{R}^d$, so that for query point q we can quickly find $p \in P$, s.t. $||q - p|| \leq \tau\, d(q, P)$ where $d(q, P) = min_{p \in P}||q - p||$.

Data structure: The shifted quadtree T of P, i.e., for P a set of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]^d$ and b in $\left[0, \frac{1}{2}\right]^d$, we use the shifted compressed quadtree T of P with $b + [0, 1]^d$ as root cell.

Low quality ANN-Search

Now we want to use shifted quadtrees to quickly answer ANN-queries in $\mathbb{R}^d$.
That is, we want to preprocess a set P of n points in $\mathbb{R}^d$, so that for query point q we can quickly find $p \in P$, s.t. $||q - p|| \leq \tau\, d(q, P)$ where $d(q, P) = min_{p \in P} ||q - p||$.

Data structure: The shifted quadtree T of P, i.e., for P a set of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]^d$ and b in $\left[0, \frac{1}{2}\right]^d$, we use the shifted compressed quadtree T of P with $b + [0, 1]^d$ as root cell.

For each node v of T choose a representative point rep_v in P_v.

Low quality ANN-Search

Now we want to use shifted quadtrees to quickly answer ANN-queries in $\mathbb{R}^d$.
That is, we want to preprocess a set P of n points in $\mathbb{R}^d$, so that for query point q we can quickly find $p \in P$, s.t. $||q - p|| \leq \tau\, d(q, P)$ where $d(q, P) = min_{p \in P} ||q - p||$.

Data structure: The shifted quadtree T of P, i.e., for P a set of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]^d$ and b in $\left[0, \frac{1}{2}\right]^d$, we use the shifted compressed quadtree T of P with $b + [0, 1]^d$ as root cell.

For each node v of T choose a representative point rep_v in P_v.

Query: For $q \in \left[\frac{1}{2}, \frac{3}{4}\right]^d$ let v be the lowest node in T s.t. q in the region of v.
If rep_v is defined (i.e. $P_v \neq \emptyset$), return it; otherwise return $rep_{par(v)}$.

Low quality ANN-Search

Now we want to use shifted quadtrees to quickly answer ANN-queries in $\mathbb{R}^d$.
That is, we want to preprocess a set P of n points in $\mathbb{R}^d$, so that for query point q we can quickly find $p \in P$, s.t. $||q - p|| \leq \tau\, d(q, P)$ where $d(q, P) = min_{p \in P} ||q - p||$.

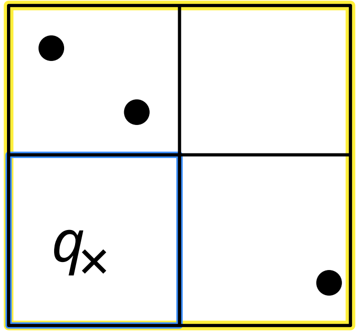
Data structure: The shifted quadtree T of P, i.e., for P a set of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]^d$ and b in $\left[0, \frac{1}{2}\right]^d$, we use the shifted compressed quadtree T of P with $b + [0, 1]^d$ as root cell.

For each node v of T choose a representative point rep_v in P_v.

Query: For $q \in \left[\frac{1}{2}, \frac{3}{4}\right]^d$ let v be the lowest node in T s.t. q in the region of v.
If rep_v is defined (i.e. $P_v \neq \emptyset$), return it; otherwise return $rep_{par(v)}$.

Analysis:

1. If v is a non-empty leaf, then rep_v is returned
2. If v is an empty leaf, then $rep_{par(v)}$ is returned
3. If v is a compressed node, i.e. its region an annulus, we return rep_v



Low quality ANN-Search

Now we want to use shifted quadtrees to quickly answer ANN-queries in $\mathbb{R}^d$.
That is, we want to preprocess a set P of n points in $\mathbb{R}^d$, so that for query point q we can quickly find $p \in P$, s.t. $||q - p|| \leq \tau\, d(q, P)$ where $d(q, P) = min_{p \in P} ||q - p||$.

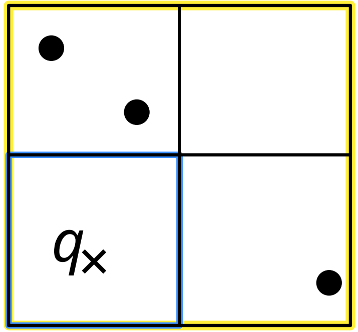
Data structure: The shifted quadtree T of P, i.e., for P a set of n points in $\left[\frac{1}{2}, \frac{3}{4}\right]^d$ and b in $\left[0, \frac{1}{2}\right]^d$, we use the shifted compressed quadtree T of P with $b + [0, 1]^d$ as root cell.

For each node v of T choose a representative point rep_v in P_v.

Query: For $q \in \left[\frac{1}{2}, \frac{3}{4}\right]^d$ let v be the lowest node in T s.t. q in the region of v.
If rep_v is defined (i.e. $P_v \neq \emptyset$), return it; otherwise return $rep_{par(v)}$.

Analysis:

1. If v is a non-empty leaf, then rep_v is returned
2. If v is an empty leaf, then $rep_{par(v)}$ is returned
3. If v is a compressed node, i.e. its region an annulus, we return rep_v



In 1. and 3. $||q - p|| \leq diam(v)$ and in 2. $||q - p|| \leq 2diam(v)$

Low quality ANN-Search

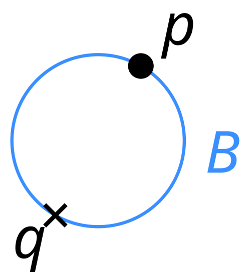
Lemma: For $\tau > 1$ and query point q, a τ-approximate NN is returned with probability at least $(1 - 4d^{3/2})/\tau$.

Low quality ANN-Search

Lemma: For $\tau > 1$ and query point q, a τ-approximate NN is returned with probability at least $(1 - 4d^{3/2})/\tau$.

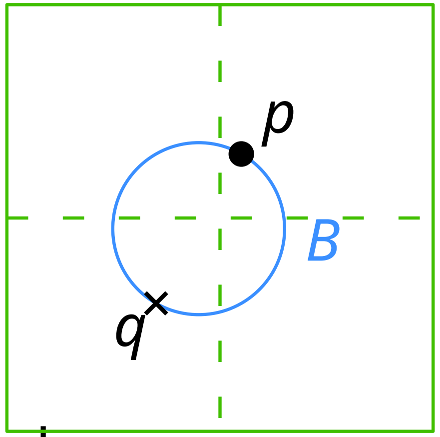
Proof:

Let p be NN of q in P; consider ball B defined by p, q.



Low quality ANN-Search

Lemma: For $\tau > 1$ and query point q, a τ-approximate NN is returned with probability at least $(1 - 4d^{3/2})/\tau$.



Proof:

Let p be NN of q in P; consider ball B defined by p, q.

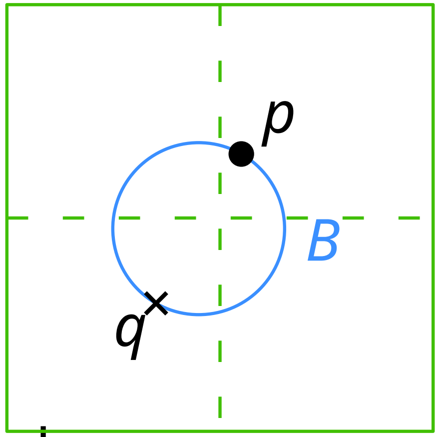
Let u be the lowest node in T that fully contains B.

The query returns either u or one of its descendants.

Hence an ANN at distance at most $2 diam(v) \leq 2 diam(u)$ is returned.

Low quality ANN-Search

Lemma: For $\tau > 1$ and query point q, a τ-approximate NN is returned with probability at least $(1 - 4d^{3/2})/\tau$.



Proof:

Let p be NN of q in P; consider ball B defined by p, q.

Let u be the lowest node in T that fully contains B.

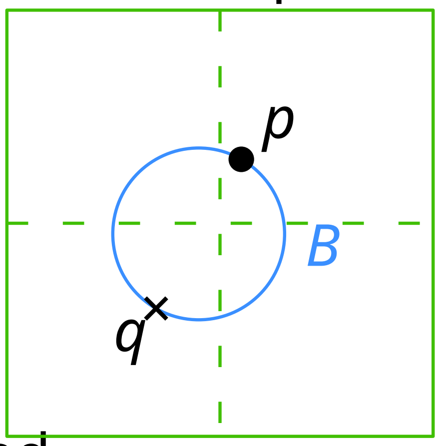
The query returns either u or one of its descendants.

Hence an ANN at distance at most $2diam(v) \leq 2diam(u)$ is returned.

Now let $\ell = ||p - q||$; by a previous lemma, $\mathbb{P}[B \text{ lies in a cell at level } i] \geq 1 - \frac{d\ell}{2^i}$ $(\star)$

Low quality ANN-Search

Lemma: For $\tau > 1$ and query point q, a τ-approximate NN is returned with probability at least $(1 - 4d^{3/2})/\tau$.



Proof:

Let p be NN of q in P; consider ball B defined by p, q.

Let u be the lowest node in T that fully contains B.

The query returns either u or one of its descendants.

Hence an ANN at distance at most $2diam(v) \leq 2diam(u)$ is returned.

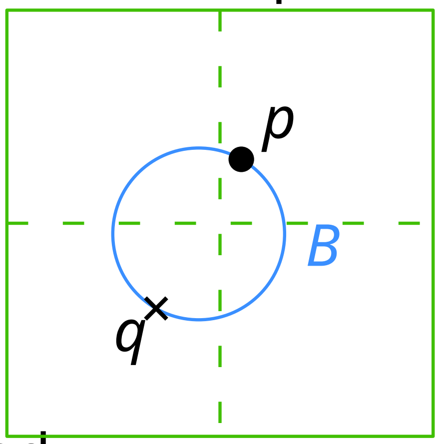
Now let $\ell = ||p - q||$; by a previous lemma, $\mathbb{P}[B \text{ lies in a cell at level } i] \geq 1 - \frac{d\ell}{2^i}$ $(\star)$

If B lies completely in a cell, let $\square$ be this cell.

The ANN returned has distance $\leq 2diam(\square) \leq 2\sqrt{d}2^i$ hence quality $2\sqrt{d}2^i/\ell$

Low quality ANN-Search

Lemma: For $\tau > 1$ and query point q, a τ-approximate NN is returned with probability at least $(1 - 4d^{3/2})/\tau$.



Proof:

Let p be NN of q in P; consider ball B defined by p, q.

Let u be the lowest node in T that fully contains B.

The query returns either u or one of its descendants.

Hence an ANN at distance at most $2diam(v) \leq 2diam(u)$ is returned.

Now let $\ell = ||p - q||$; by a previous lemma, $\mathbb{P}[B \text{ lies in a cell at level } i] \geq 1 - \frac{d\ell}{2^i}$ $(\star)$

If B lies completely in a cell, let $\square$ be this cell.

The ANN returned has distance $\leq 2diam(\square) \leq 2\sqrt{d}2^i$ hence quality $2\sqrt{d}2^i/\ell$

And it holds $2\sqrt{d}2^i/\ell \leq \tau \Leftrightarrow i \leq \log_2 \frac{\ell\tau}{2\sqrt{d}}$

Set $i := \lfloor \log_2(\frac{\ell\tau}{2\sqrt{d}}) \rfloor$ then it follows with $(\star)$ that an τ-ANN is returned with probability at least $1 - \frac{d\ell}{2^i} \geq 1 - \frac{4d^{3/2}}{\tau}$

Summary

shifting grids $\rightarrow$ approximate disk cover

shifting quadtrees $\rightarrow$ approximate nearest neighbor query

Summary

shifting grids $\rightarrow$ approximate disk cover

The probability that a ball B of radius r is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.

shifting quadtrees $\rightarrow$ approximate nearest neighbor query

Summary

shifting grids $\rightarrow$ approximate disk cover

The probability that a ball B of radius r is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.

algorithm design: Solve problem per cell; bound error of B not being in one cell using probability

shifting quadtrees $\rightarrow$ approximate nearest neighbor query

Summary

shifting grids $\to$ approximate disk cover

The probability that a ball B of radius r is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.

algorithm design: Solve problem per cell; bound error of B not being in one cell using probability

shifting quadtrees $\to$ approximate nearest neighbor query

For $t > 0$ holds $\mathbb{P}\left[\mathbb{L}_b(p, q) > \log_2 ||p - q|| + t\right] \leq \frac{4d}{2^t}$.

Summary

shifting grids $\rightarrow$ approximate disk cover

The probability that a ball B of radius r is not in a single cell of $G^d(b, \Delta)$ is at most $min\left(\frac{2dr}{\Delta}, 1\right)$.

algorithm design: Solve problem per cell; bound error of B not being in one cell using probability

shifting quadtrees $\rightarrow$ approximate nearest neighbor query

For $t > 0$ holds $\mathbb{P}\left[\mathbb{L}_b(p, q) > \log_2 ||p - q|| + t\right] \leq \frac{4d}{2^t}$.

With high probability p and q in same cell at level $\log_2 ||p - q|| + c \log n$