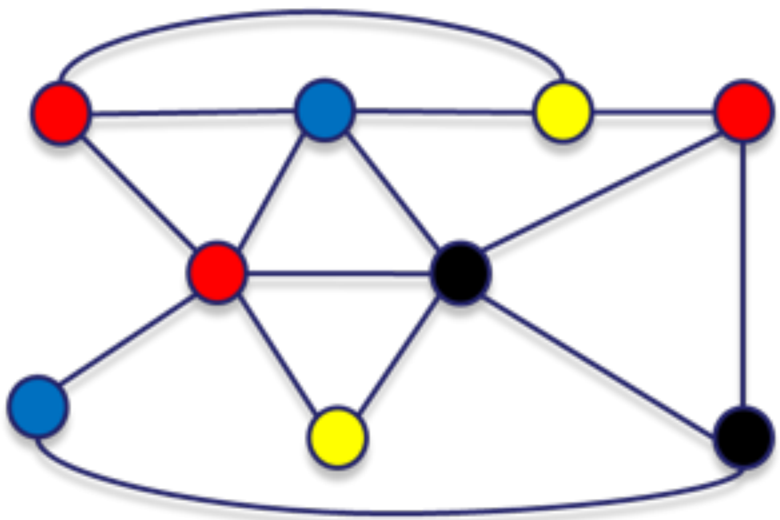
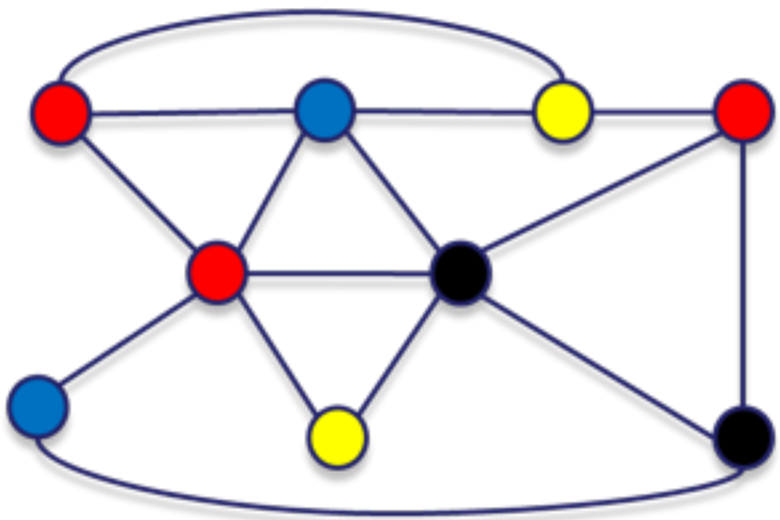
Distance Oracles for Vertex-Colored Graphs

Oren Weimann
Weizmann Institute



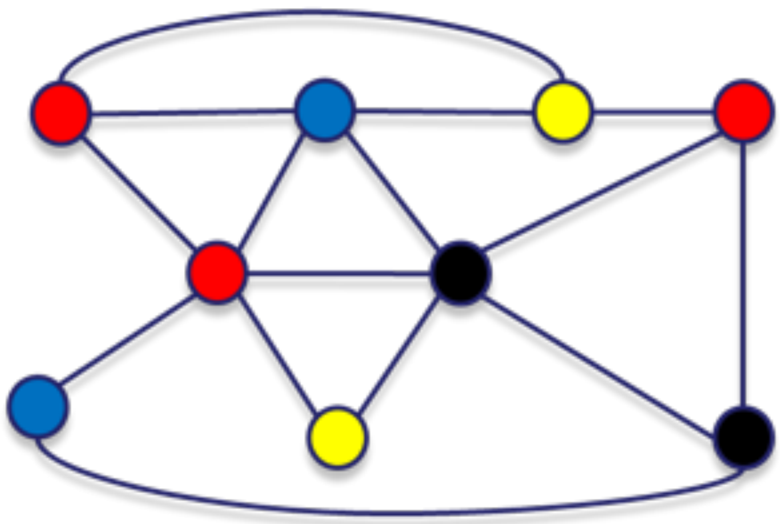
Joint work with:
Danny Hermelin, Avivit Levy, and Raphael Yuster

Vertex-Colored Networks



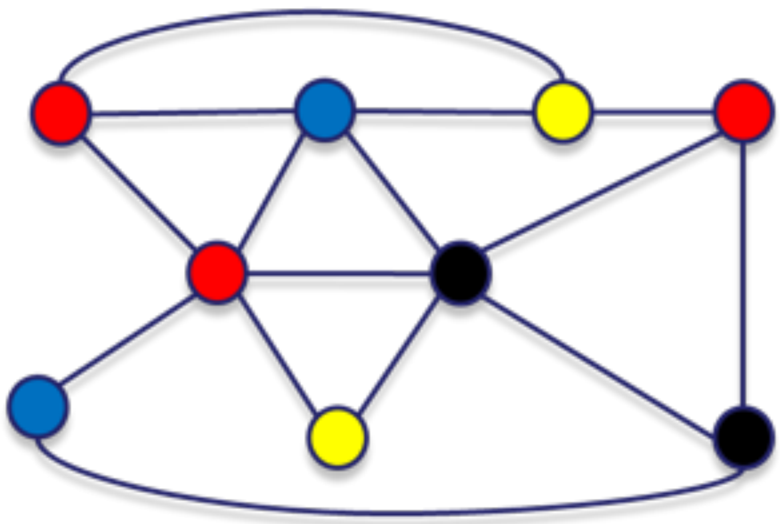
Vertex-Colored Networks

- A graph G where each vertex v has a color $c(v)$.



Vertex-Colored Networks

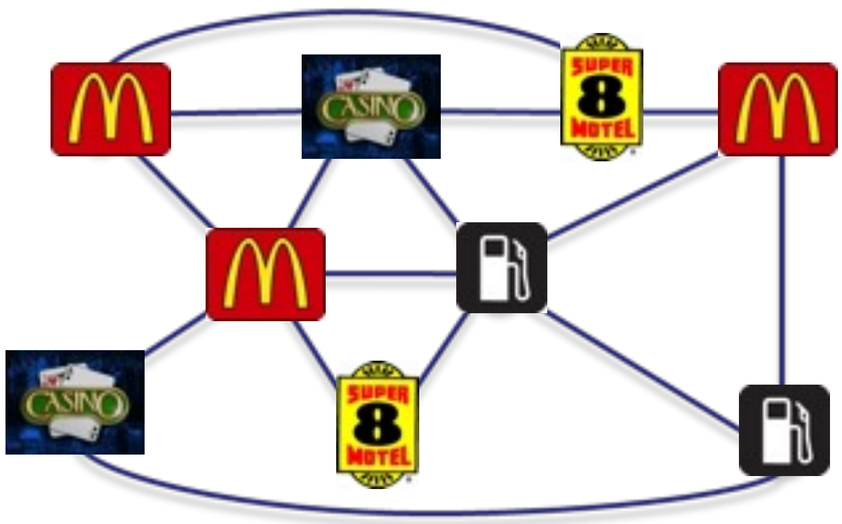
- A graph G where each vertex v has a color $c(v)$.
- Colors indicate functionality of a node.



Vertex-Colored Networks

- A graph G where each vertex v has a color $c(v)$.
- Colors indicate functionality of a node.

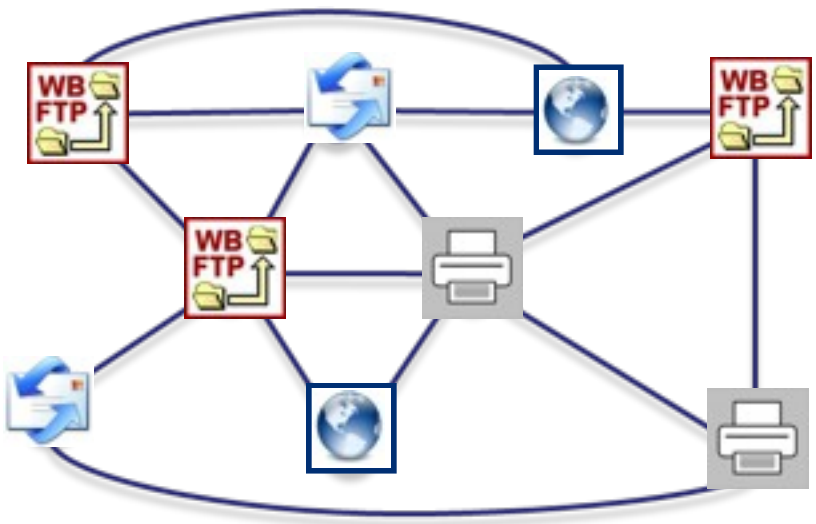
road network



Vertex-Colored Networks

- A graph G where each vertex v has a color $c(v)$.
- Colors indicate functionality of a node.

computer
network

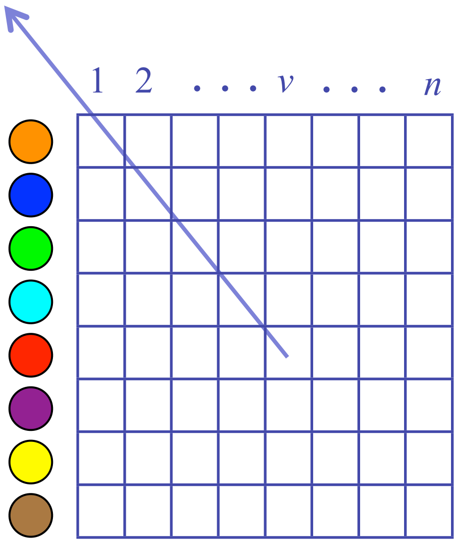


Vertex-Color Distance Oracles

- Data Structure for queries $\delta(v, ●)$.
  - $\delta(v, ●)$ = vertex-to-*functionality* *distance.*

Vertex-Color Distance Oracles

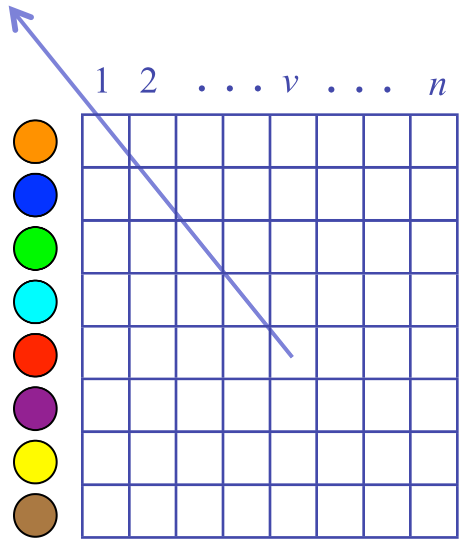
- Data Structure for queries $\delta(v,$ ●$)$.
  - $\delta(v,$ ●$)$ = vertex-to-*functionality* *distance.*
- $\Theta(nc)$ space: an $n \times c$ matrix storing all $\delta(v,$ ●$)$.



Vertex-Color Distance Oracles

- Data Structure for queries $\delta(v,$ ●$)$.
  - $\delta(v,$ ●$)$ = vertex-to-*functionality* *distance.*
- $\Theta(nc)$ space: an $n \times c$ matrix storing all $\delta(v,$ ●$)$.

  1. $\Omega(nc)$ space is too much.

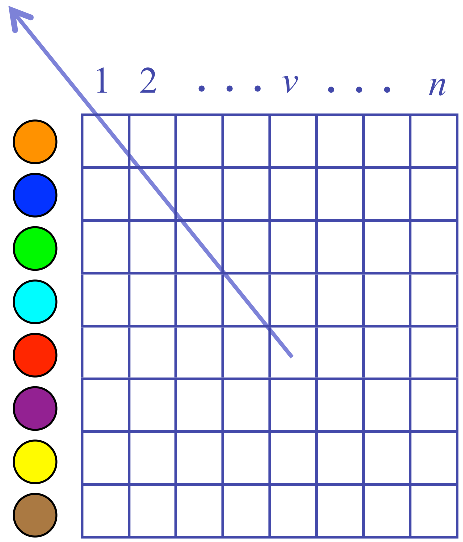


Dijkstra's algorithm
after contracting color

Vertex-Color Distance Oracles

- Data Structure for queries $\delta(v,$ ●$)$.
  - $\delta(v,$ ●$)$ = vertex-to-*functionality* *distance.*
- $\Theta(nc)$ space: an $n \times c$ matrix storing all $\delta(v,$ ●$)$.

  1. $\Omega(nc)$ space is too much.
  2. Changing color requires $\Omega(n)$ time.



Dijkstra's algorithm after contracting color

Vertex-Color Distance Oracles

- Data Structure for queries $\delta(v,$●$)$.
  – $\delta(v,$●$)$ = vertex-to-*functionality distance.*

- $\Theta(nc)$ space: an $n \times c$ matrix storing all $\delta(v,$●$)$.

  1. $\Omega(nc)$ space is too much.
  2. Changing color requires $\Omega(n)$ time.

Dijkstra's algorithm after contracting color

Vertex-Color Distance Oracles

- Data Structure for queries $\delta(v,$●$)$.
  - $\delta(v,$●$)$ = vertex-to-*functionality* *distance.*

- $\Theta(nc)$ space: an $n \times c$ matrix storing all $\delta(v,$●$)$.

  1. $\Omega(nc)$ space is too much.
  2. Changing color requires $\Omega(n)$ time.

- $o(nl)$ space means imprecise answers.

Dijkstra's algorithm
after contracting color

Vertex-Color Distance Oracles

- Data Structure for queries $\delta(v,$●$)$.
  - $\delta(v,$●$)$ = vertex-to-*functionality distance.*
- $\Theta(nc)$ space: an $n \times c$ matrix storing all $\delta(v,$●$)$.

  1. $\Omega(nc)$ space is too much.
  2. Changing color requires $\Omega(n)$ time.

- $o(nl)$ space means imprecise answers.
  - *stretch* α if outputs $d(v,$●$)$ s.t.

Dijkstra's algorithm
after contracting color

Vertex-Color Distance Oracles

- Data Structure for queries $\delta(v,\bullet)$.
  - $\delta(v,\bullet)$ = vertex-to-*functionality* *distance.*

- $\Theta(nc)$ space: an $n \times c$ matrix storing all $\delta(v,\bullet)$.

  1. $\Omega(nc)$ space is too much.
  2. Changing color requires $\Omega(n)$ time.

Dijkstra's algorithm after contracting color

- $o(nl)$ space means imprecise answers.
  - *stretch* α if outputs $d(v,\bullet)$ s.t.

$$\delta(v,\bullet) \le d(v,\bullet) \le \alpha\cdot\delta(v,\bullet)$$

Vertex-Vertex Distance Oracles

Vertex-Vertex Distance Oracles

- Benchmark result by Thorup-Zwick [JACM'02]:
  - $O(kn^{1+1/k})$ space.
  - $(2k\text{-}1)$ stretch queries in $O(k)$ time.
  - Optimal under a graph-theoretic conjecture of Erdős

Vertex-Vertex Distance Oracles

- Benchmark result by Thorup-Zwick [JACM'02]:
  - $O(kn^{1+1/k})$ space.
  - $(2k\text{-}1)$ stretch queries in $O(k)$ time.
  - Optimal under a graph-theoretic conjecture of Erdős

- Problems adapting the Thorup-Zwick oracles:

Vertex-Vertex Distance Oracles

- Benchmark result by Thorup-Zwick [JACM'02]:
  - $O(kn^{1+1/k})$ space.
  - $(2k\text{-}1)$ stretch queries in $O(k)$ time.
  - Optimal under a graph-theoretic conjecture of Erdős

- Problems adapting the Thorup-Zwick oracles:
  - Query algorithm relies on knowing both source and destination,

Vertex-Vertex Distance Oracles

- Benchmark result by Thorup-Zwick [JACM'02]:
  - $O(kn^{1+1/k})$ space.
  - $(2k\text{-}1)$ stretch queries in $O(k)$ time.
  - Optimal under a graph-theoretic conjecture of Erdős

- Problems adapting the Thorup-Zwick oracles:
  - Query algorithm relies on knowing both source and destination,
  - Space bound independent of c (consider $c = polylog(n)$)

Vertex-Vertex Distance Oracles

- Benchmark result by Thorup-Zwick [JACM'02]:
  - $O(kn^{1+1/k})$ space.
  - $(2k\text{-}1)$ stretch queries in $O(k)$ time.
  - Optimal under a graph-theoretic conjecture of Erdős

- Problems adapting the Thorup-Zwick oracles:
  - Query algorithm relies on knowing both source and destination,
  - Space bound independent of c (consider $c = polylog(n)$)
  - Doesn't support changing colors

Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k$-$5)$-stretch vertex-color oracles.*

Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k$-$5)$-stretch vertex-color oracles.*

- Space bound independent of c.

Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k-5)$-stretch vertex-color oracles.*

- Space bound independent of c.
- Does not support changing colors.

Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k-5)$-stretch vertex-color oracles.*

- Space bound independent of c.
- Does not support changing colors.

Theorem: *$O(knc^{1/k})$-space (2^k-1)-stretch vertex-color oracles.*

Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: $O(kn^{1+1/k})$-*space* $(4k\text{-}5)$-*stretch vertex-color oracles.*

- Space bound independent of c.
- Does not support changing colors.

Theorem: $O(knc^{1/k})$-*space* $(2^k\text{-}1)$-*stretch vertex-color oracles.*

- Still does not support changing colors.

Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k\text{-}5)$-stretch vertex-color oracles.*

- Space bound independent of c.
- Does not support changing colors.

Theorem: *$O(knc^{1/k})$-space $(2^k\text{-}1)$-stretch vertex-color oracles.*

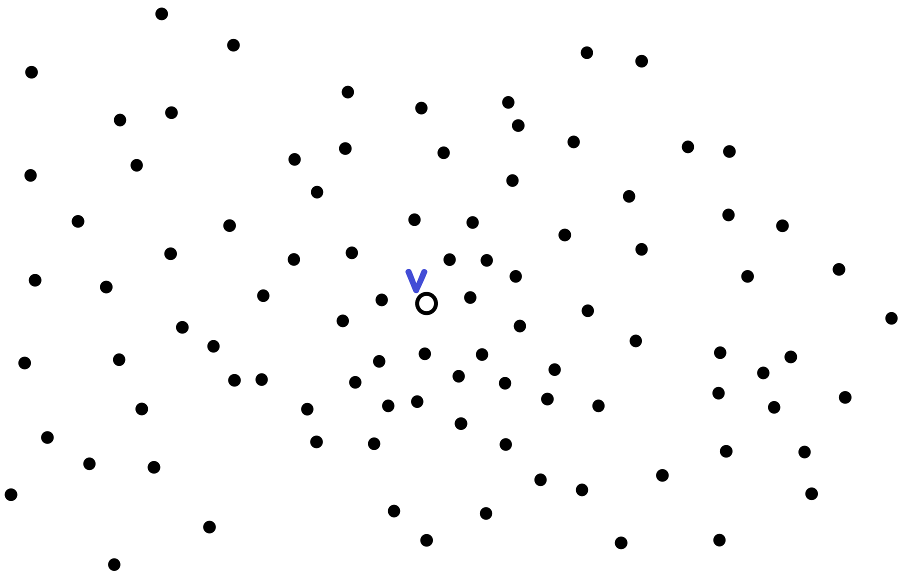
- Still does not support changing colors.

Theorem: *$O(kn^{1+1/k})$-space $(3^{k-1}\text{-}1)$-stretch vertex-color oracles allowing changing colors in $O(kn^{1/k}\lg n)$ time.*

Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k$-$5)$-stretch vertex-color oracles.*

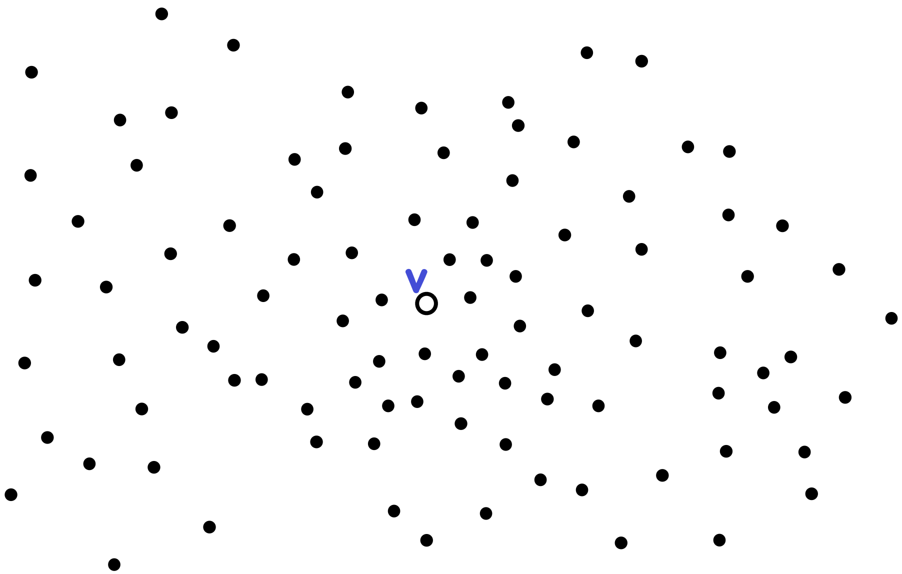


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k\text{-}5)$-stretch vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$

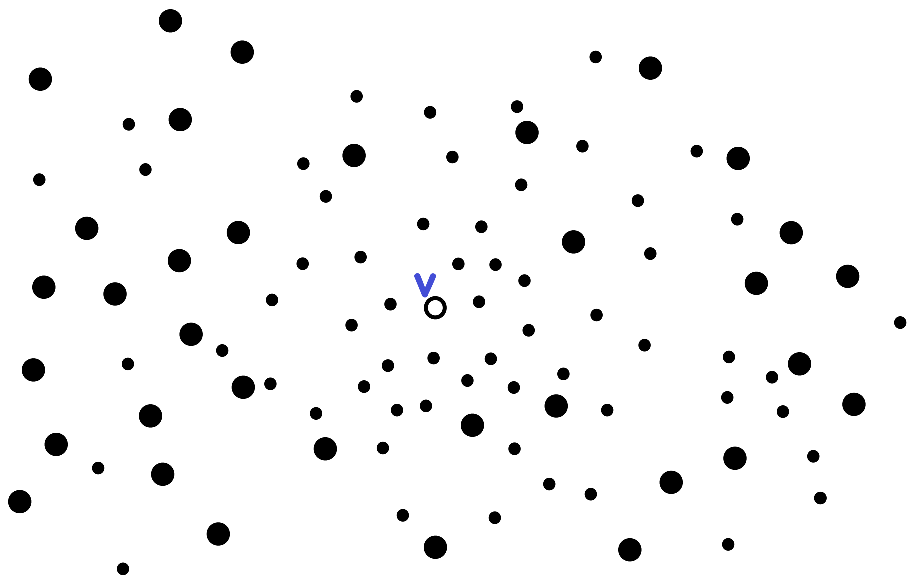


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k$-$5)$-stretch vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$

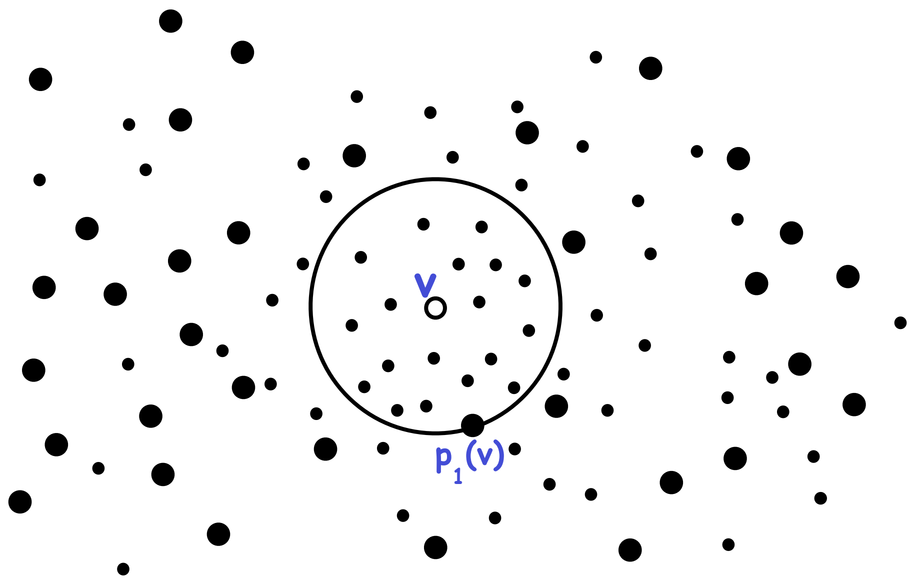


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k\text{-}5)$-stretch vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$

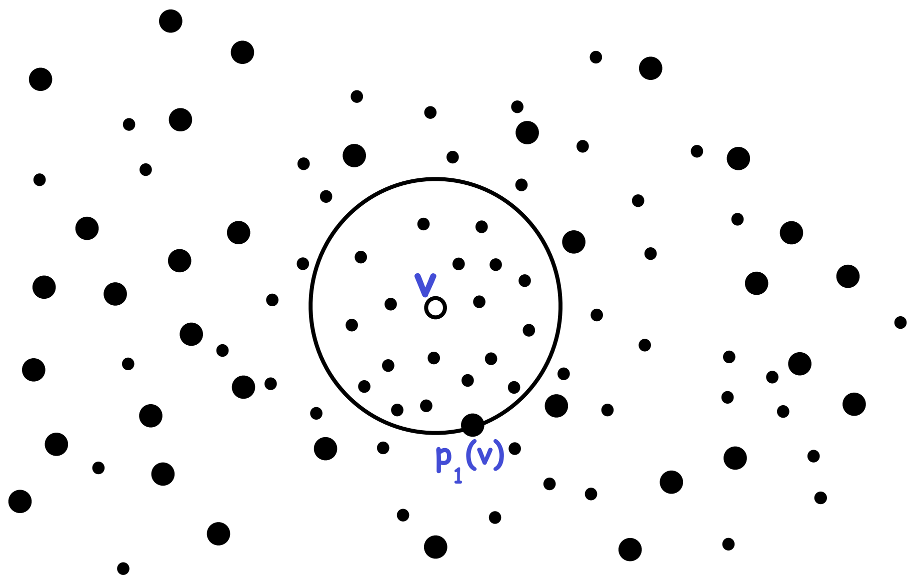


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k$-$5)$-stretch vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

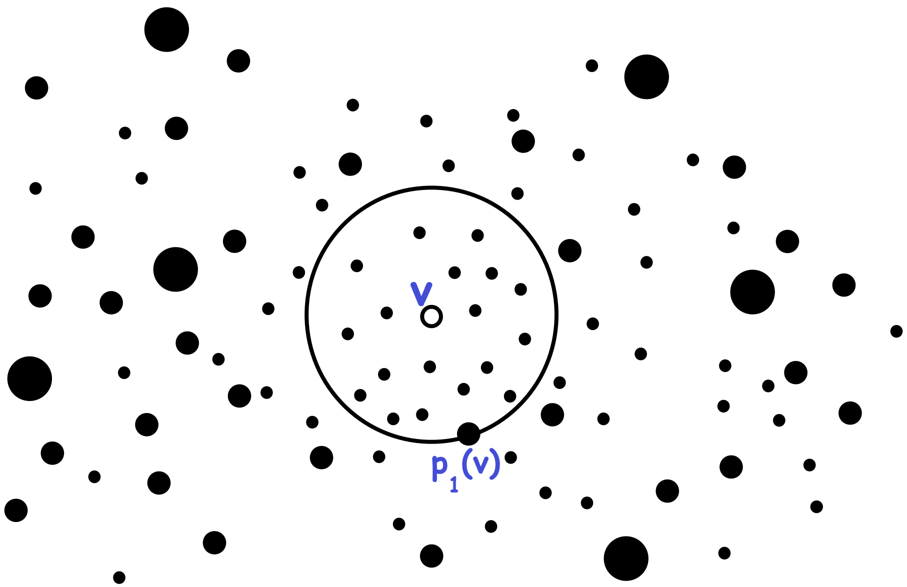


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k\text{-}5)$-stretch vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

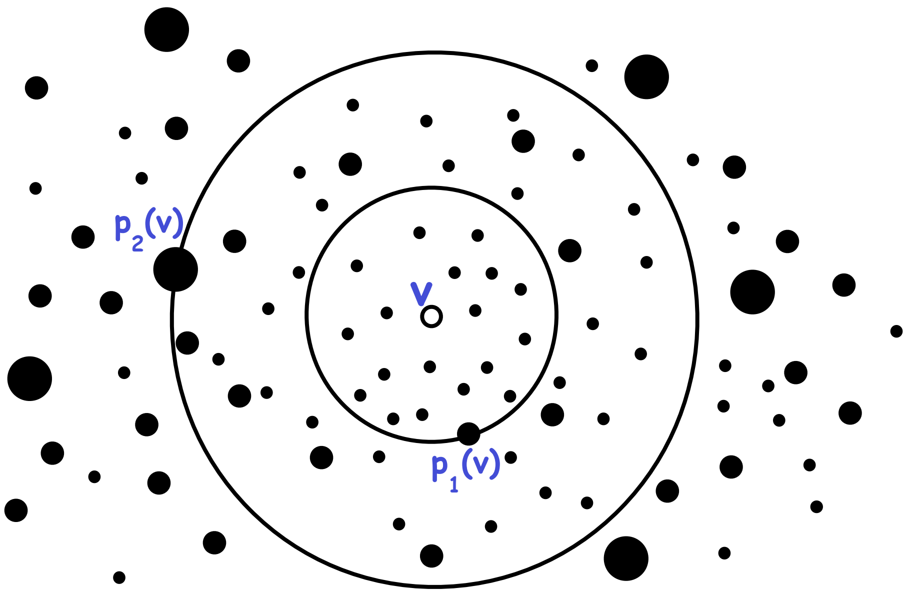


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k\text{-}5)$-stretch vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

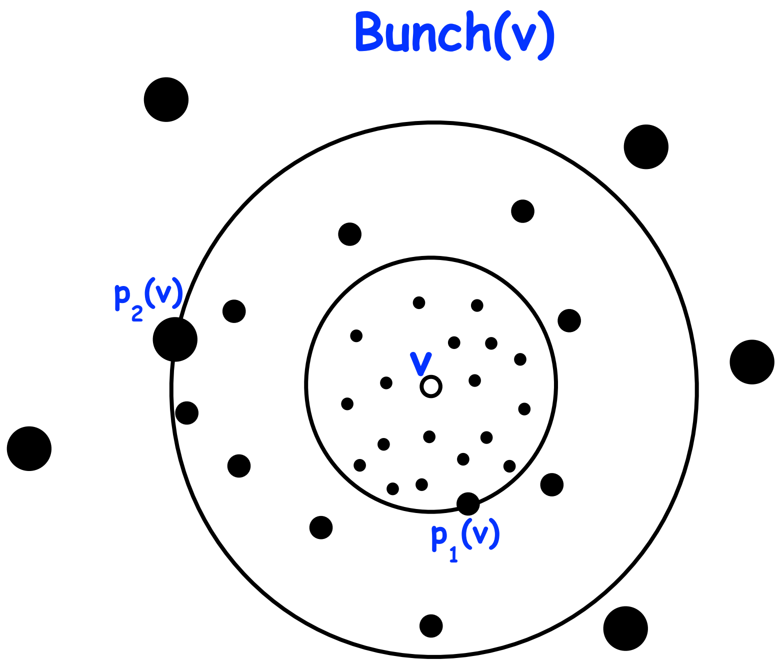


Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space $(4k$-$5)$-stretch vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

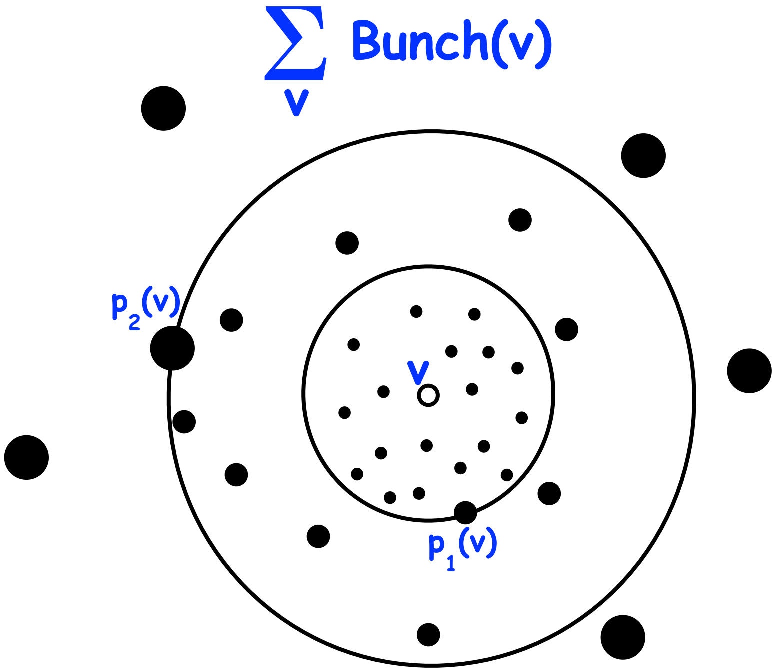


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: ***$O(kn^{1+1/k})$-space*** $(4k\text{-}5)$-*stretch vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

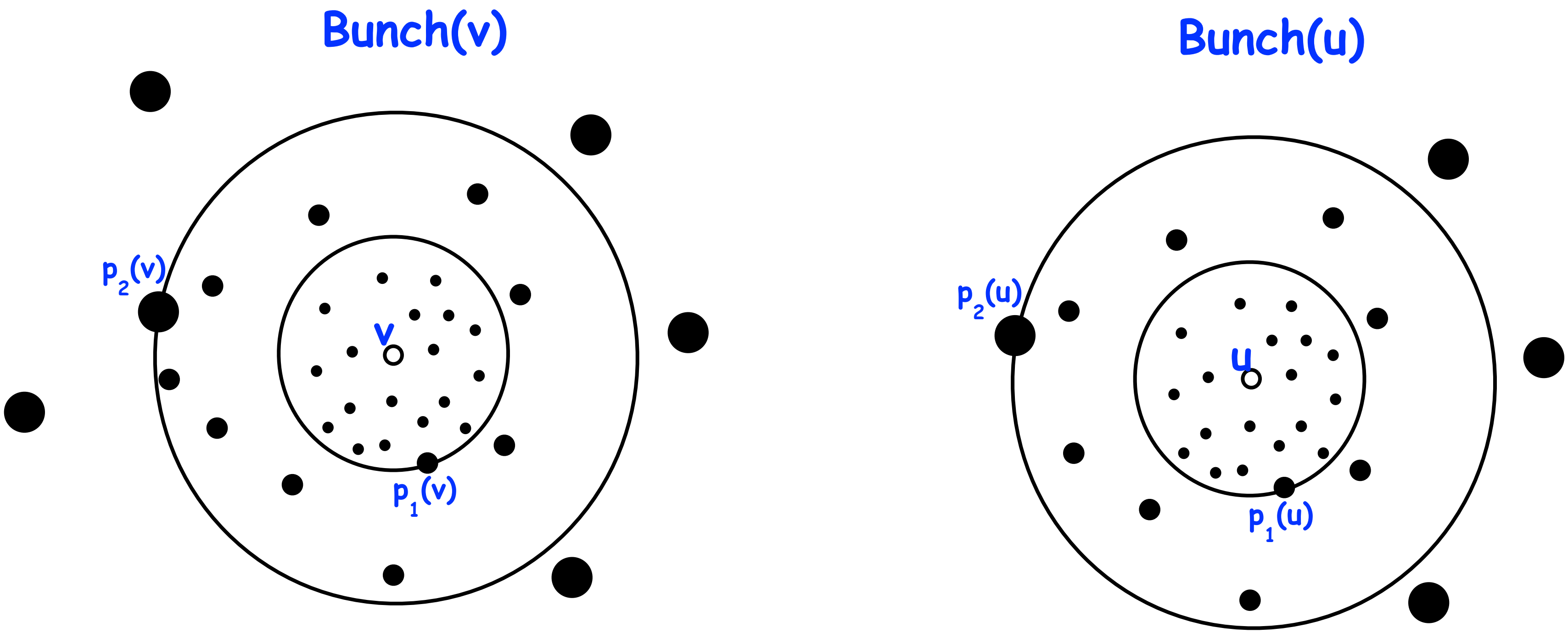


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space* ***(4k-5)-stretch*** *vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

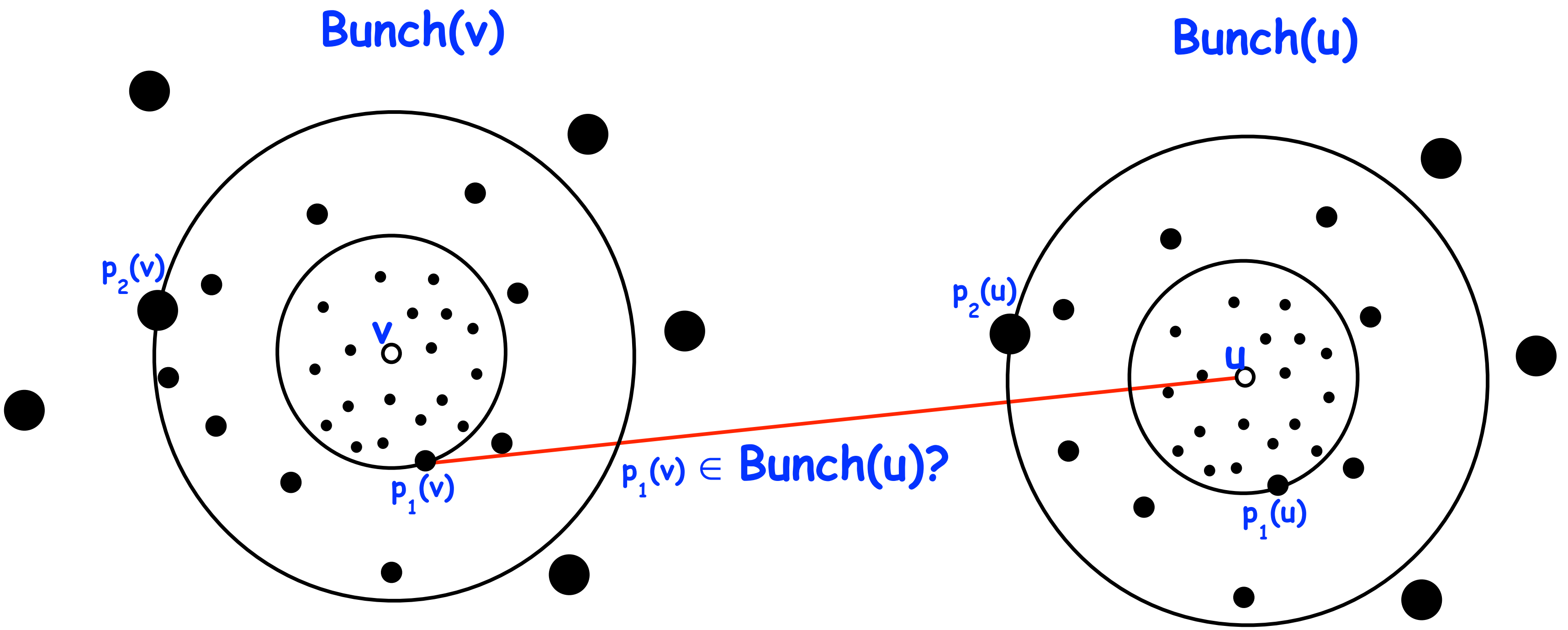


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space* ***(4k-5)-stretch*** *vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

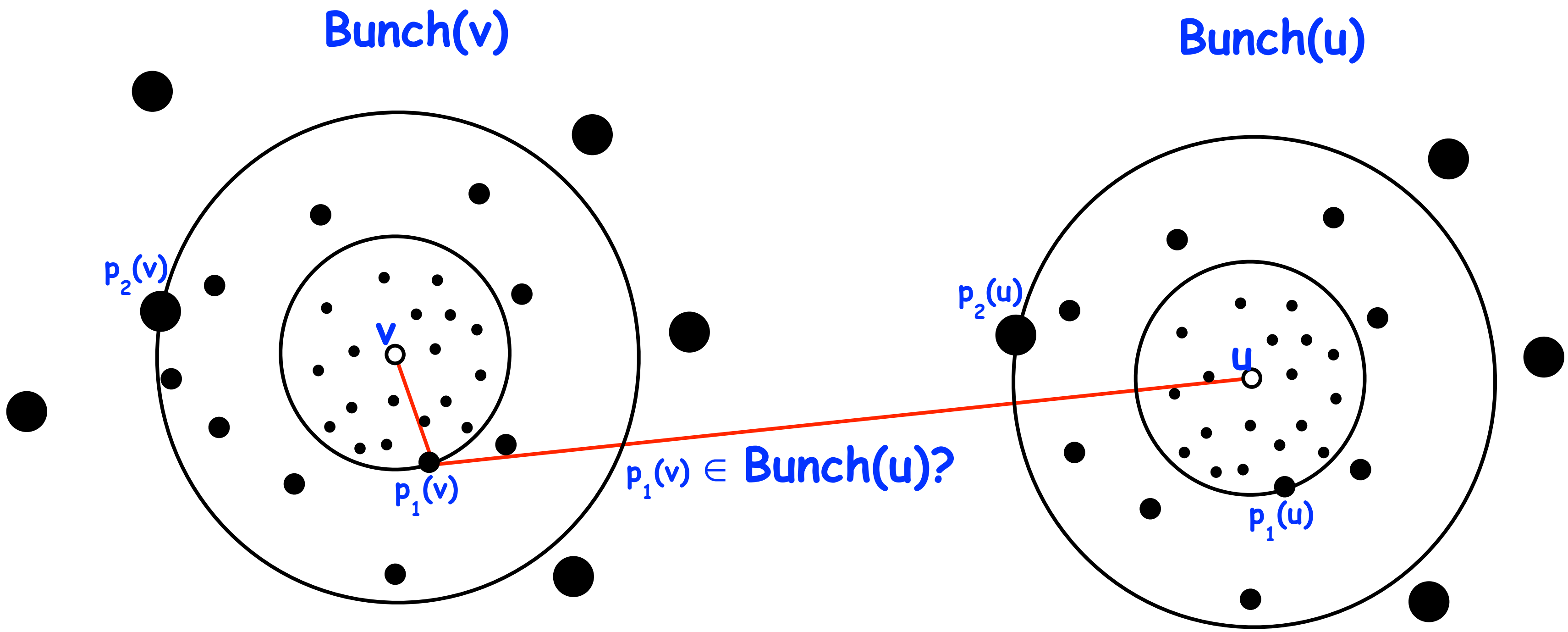


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space* ***(4k-5)-stretch*** *vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

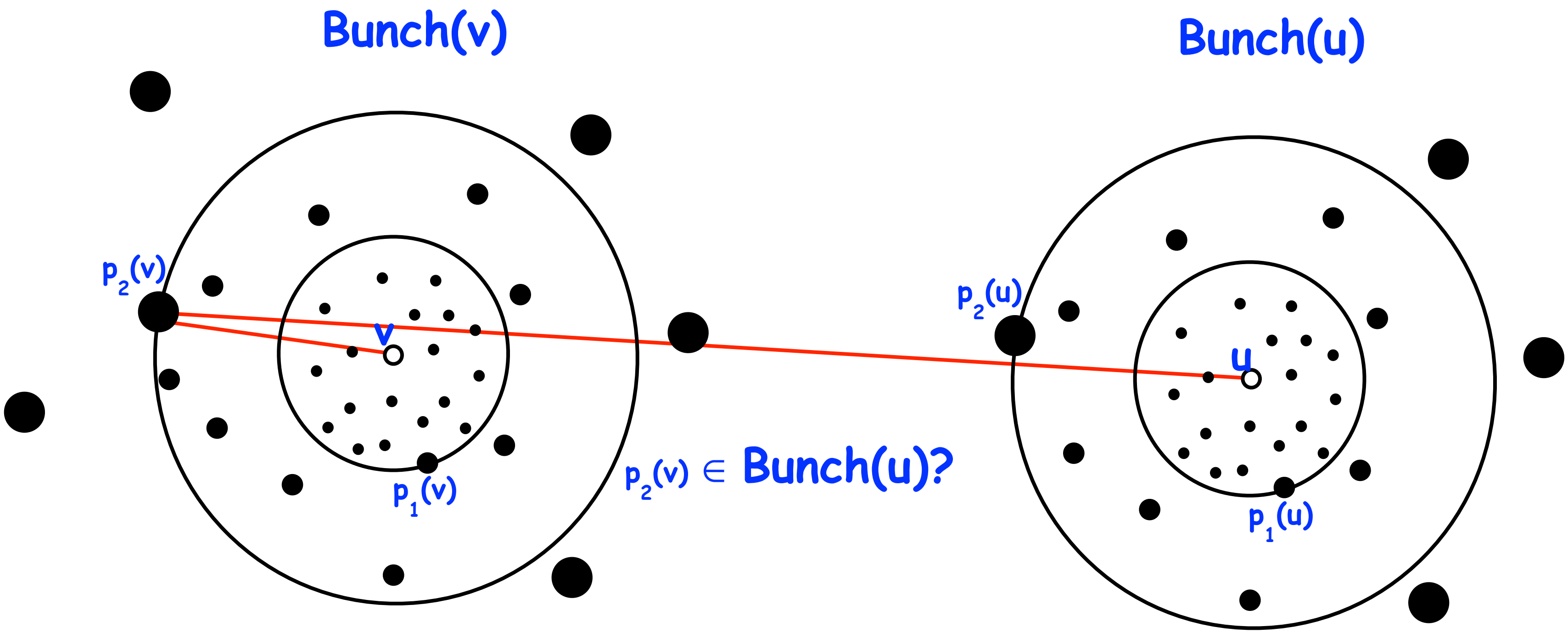


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space* ***(4k-5)-stretch*** *vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

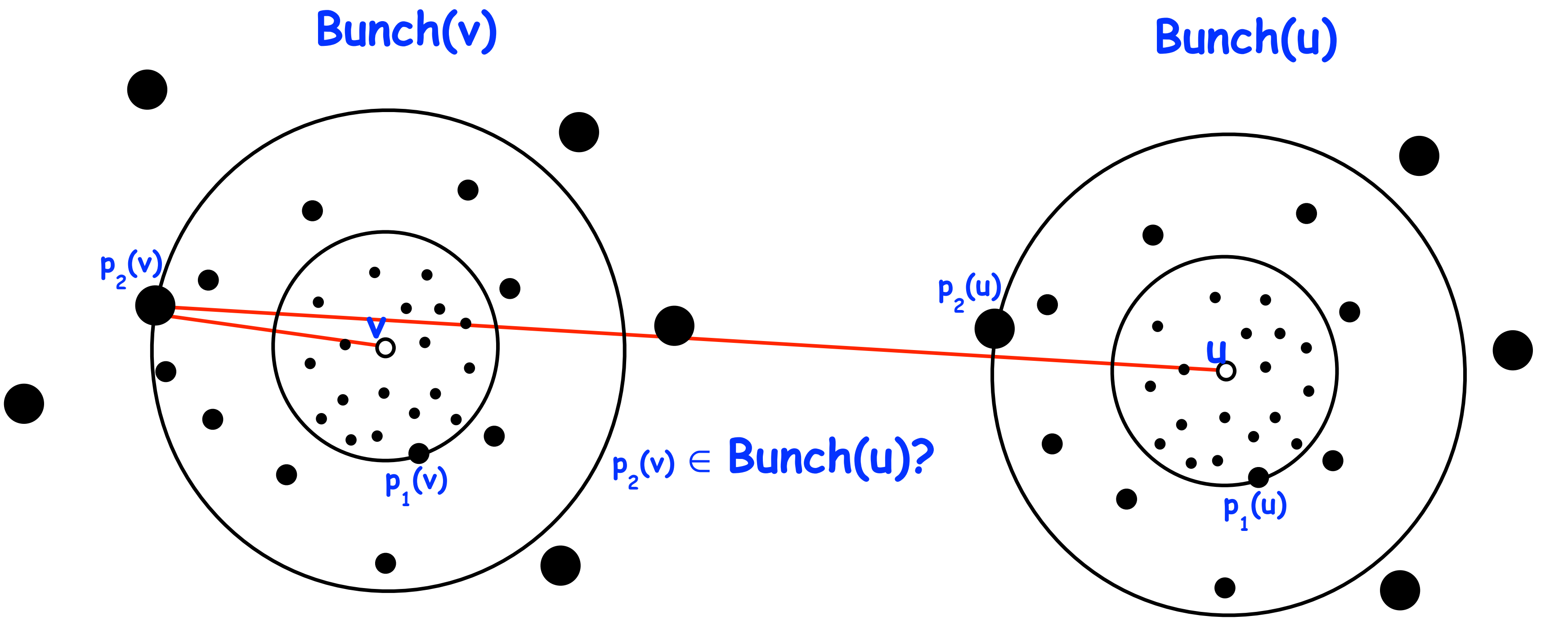


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space* ***(2k-1)-stretch*** *vertex-color oracles.*

- Select *pivots* uniformly at random with probability $n^{-1/k}$
- Select *pivots* from pivots... with probability $n^{-1/k}$

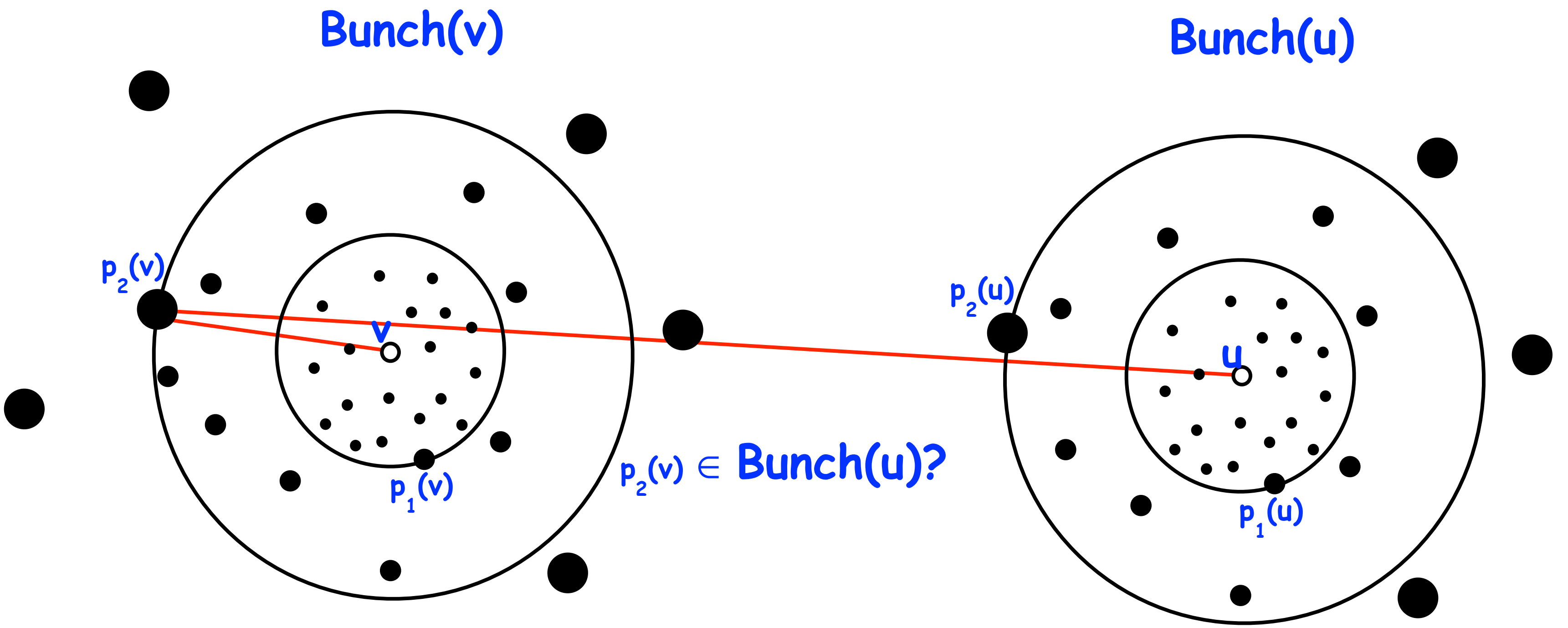


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space **(4k-5)-stretch** vertex-color oracles.*

1. we don't know identity of **u** so advance in one side only

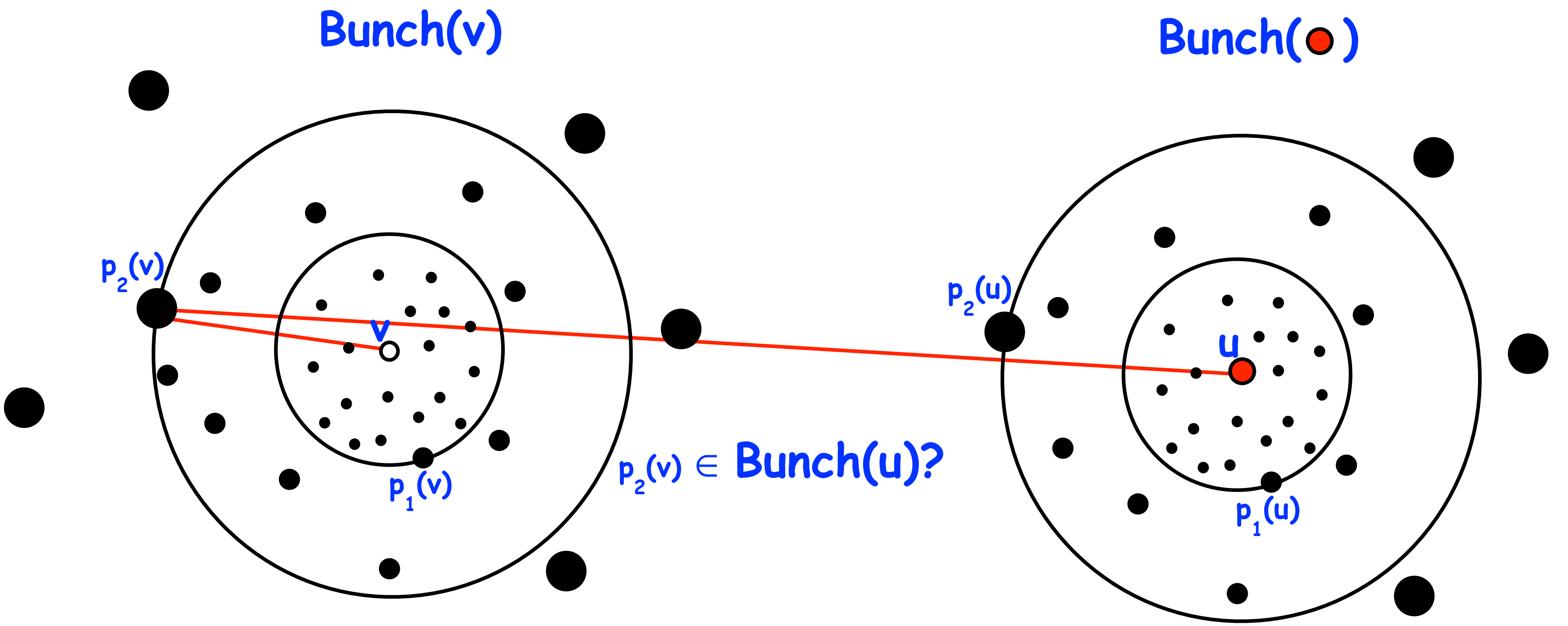


Vertex-Color Distance Oracles

- Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space* ***(4k-5)-stretch*** *vertex-color oracles.*

1. we don't know identity of **u** so advance in one side only
2. Bunches for colors

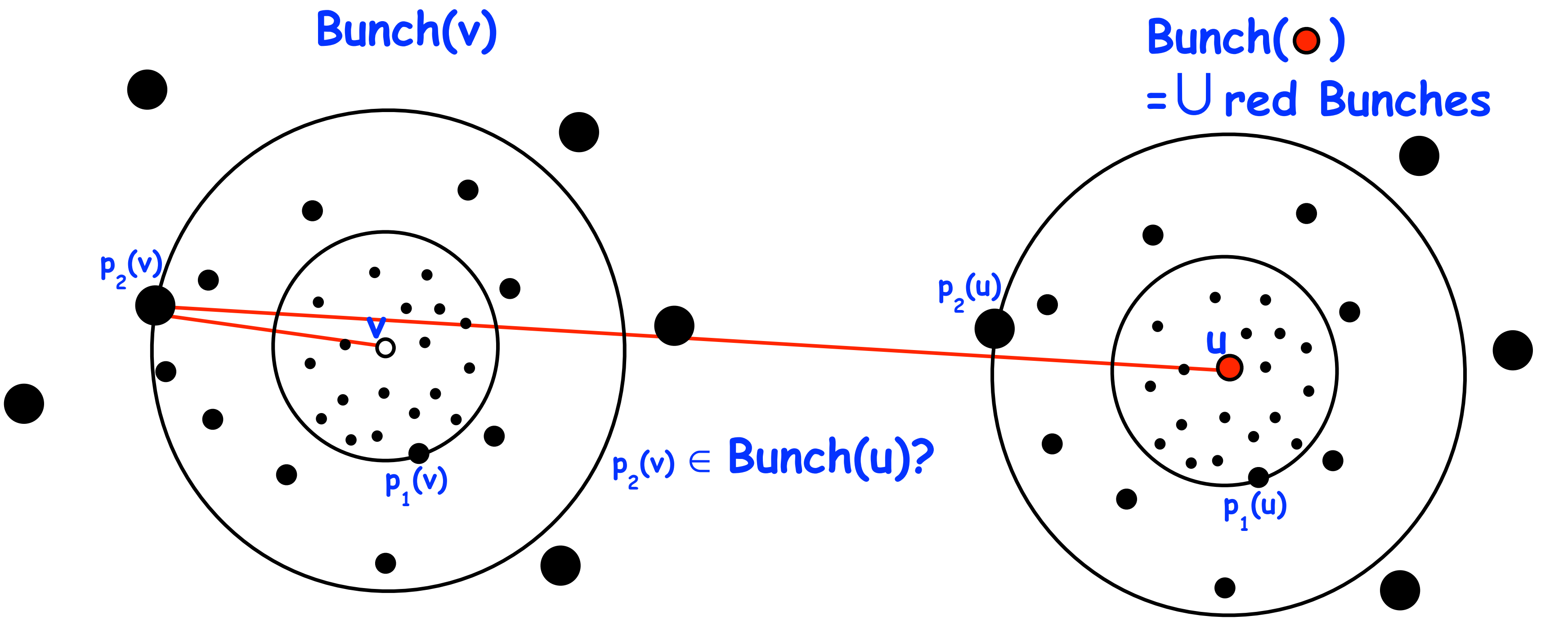


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

Theorem: *$O(kn^{1+1/k})$-space* ***(4k-5)-stretch*** *vertex-color oracles.*

1. we don't know identity of **u** so advance in one side only
2. Bunches for colors

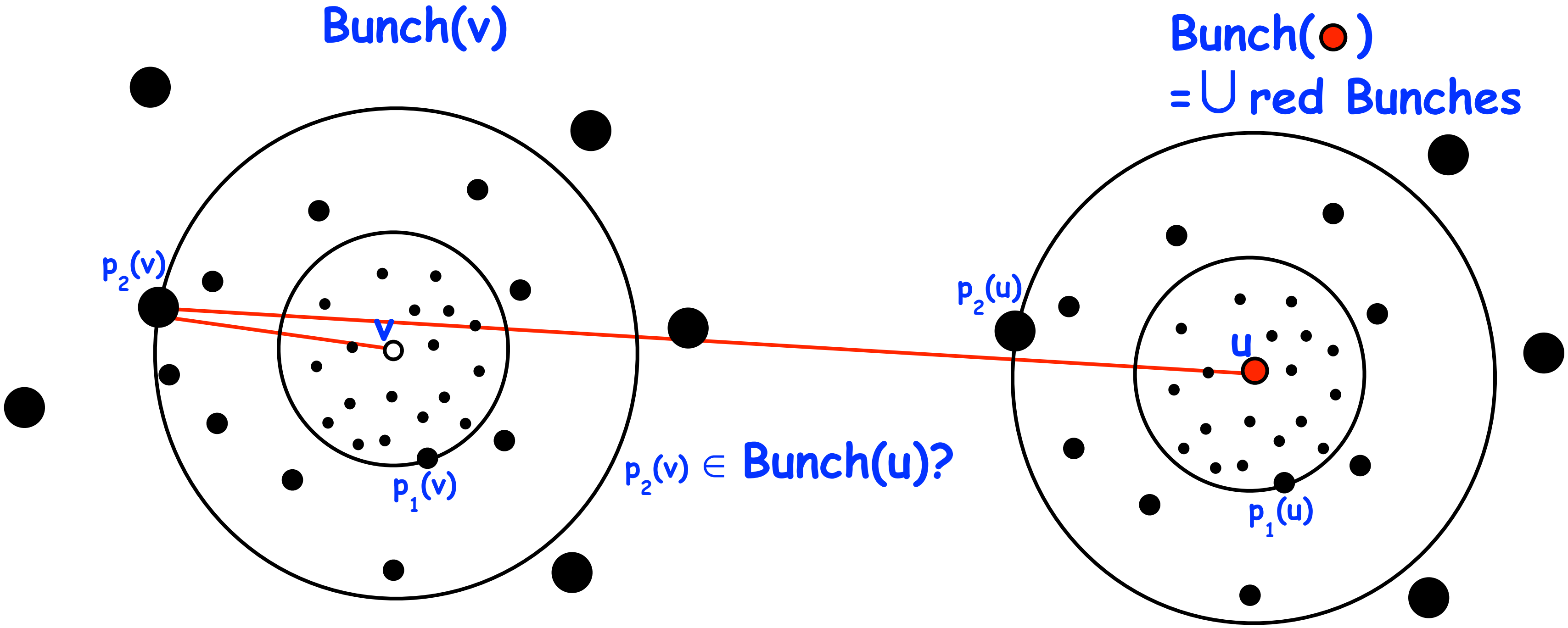


Vertex-Color Distance Oracles

➢ Adaption of the Thorup-Zwick oracles

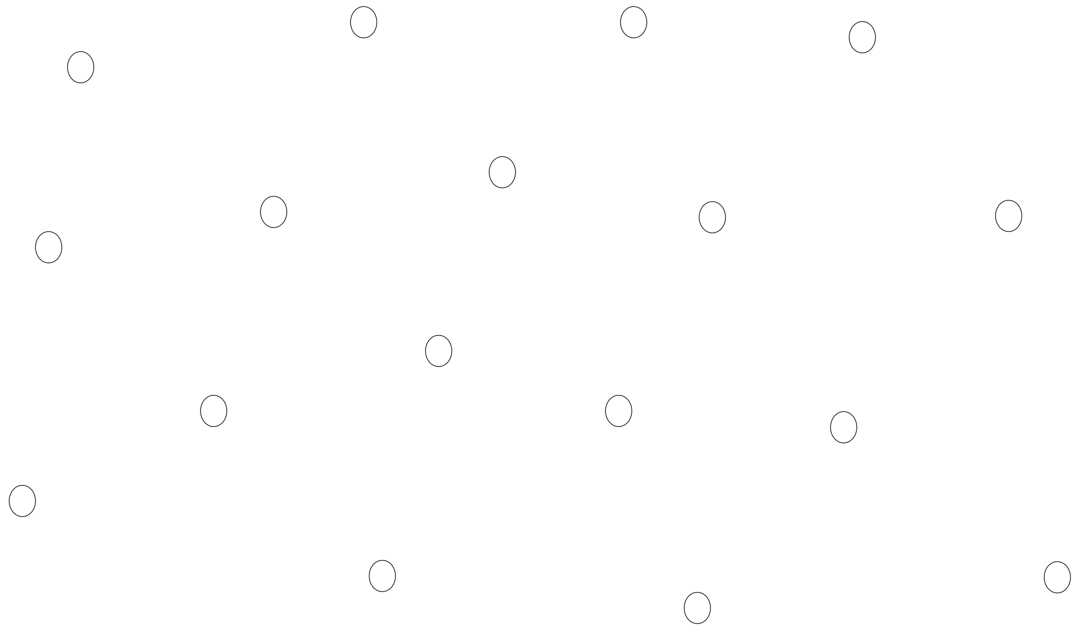
Theorem: *$O(kn^{1+1/k})$-space* ***(4k-5)-stretch*** *vertex-color oracles.*

1. we don't know identity of **u** so advance in one side only
2. Bunches for colors
3. check all $p_i(v)$



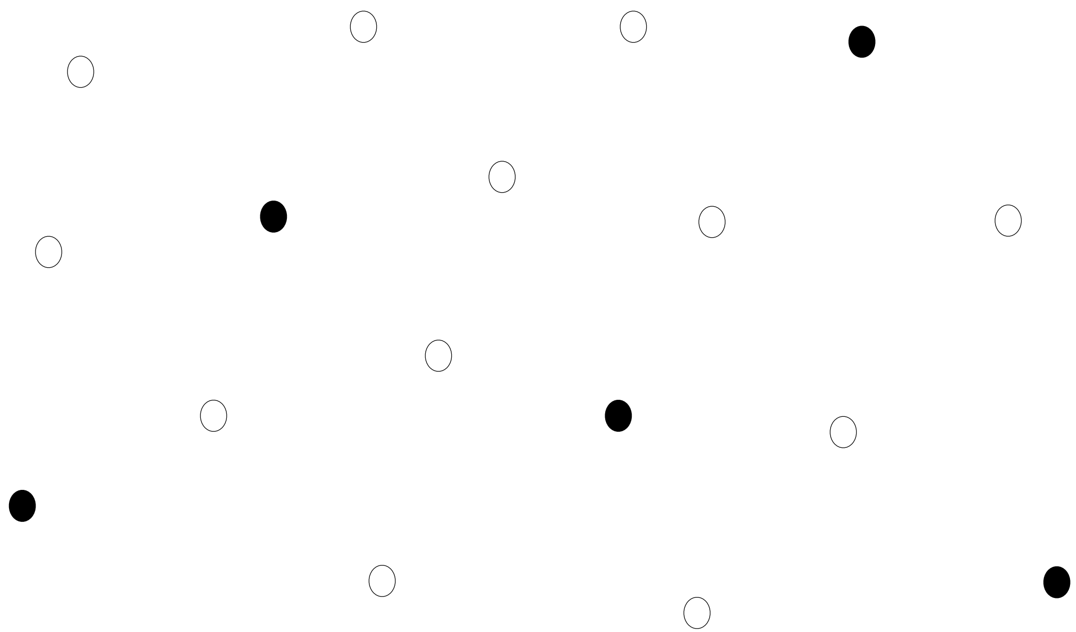
An $O(nc^{1/2})$-space 3-stretch Oracle

- Select *routers* uniformly at random with prob. $c^{-1/2}$.



An $O(nc^{1/2})$-space 3-stretch Oracle

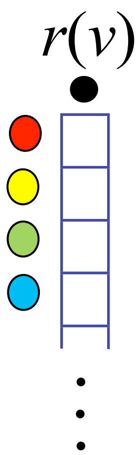
- Select *routers* uniformly at random with prob. $c^{-1/2}$.

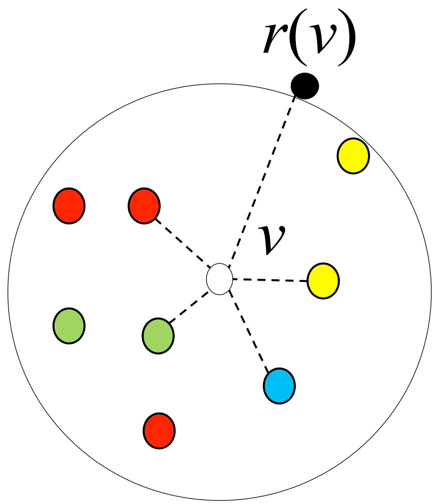


An $O(nc^{1/2})$-space 3-stretch Oracle

- Select *routers* uniformly at random with prob. $c^{-1/2}$.
- Store all distances
  - $\delta(r,\lambda)$ for every router r and color λ.
  - $\delta(v,r(v))$ from every vertex v to its closest router $r(v)$.
  - $\delta(v,\lambda)$ from every vertex v to every color λ with $\delta(v,\lambda) < \delta(v,r(v))$.

The ball $B(v)$ of v





An O($nc^{1/2}$)-space 3-stretch Oracle

An $O(nc^{1/2})$-space 3-stretch Oracle

- Expected space required:

An $O(nc^{1/2})$-space 3-stretch Oracle

- Expected space required:
  - total size of all routers tables = $nc^{1/2}$

An $O(nc^{1/2})$-space 3-stretch Oracle

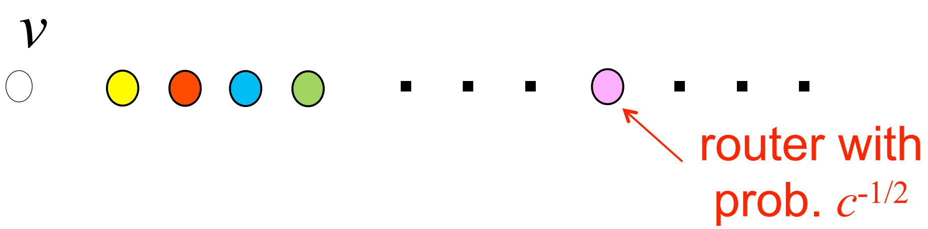
- Expected space required:
  - total size of all routers tables = $nc^{1/2}$
  - size of any ball $B(v) \leq c^{1/2}$

An O($nc^{1/2}$)-space 3-stretch Oracle

- Expected space required:
  - total size of all routers tables = $nc^{1/2}$
  - size of any ball $B(v) \leq c^{1/2}$
    - Sort all colors according to their distances from v:

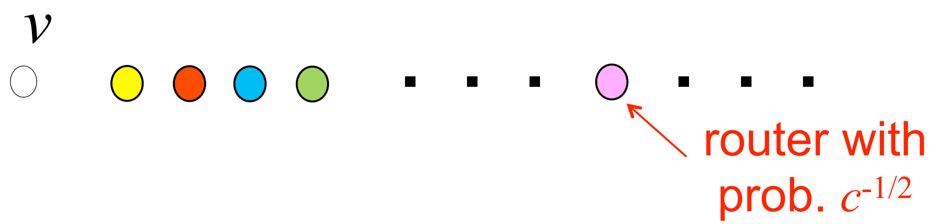
An O($nc^{1/2}$)-space 3-stretch Oracle

- Expected space required:
  - total size of all routers tables = $nc^{1/2}$
  - size of any ball $B(v) \le c^{1/2}$
    - Sort all colors according to their distances from v:



An $O(nc^{1/2})$-space 3-stretch Oracle

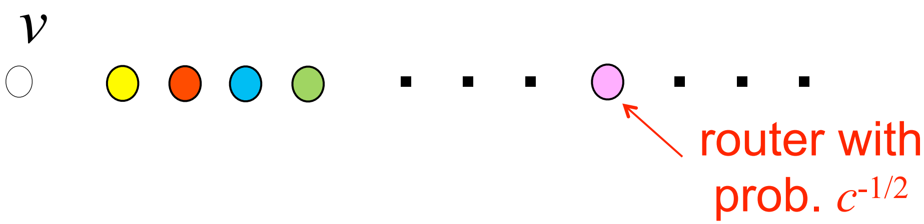
- Expected space required:
  - total size of all routers tables = $nc^{1/2}$
  - size of any ball $B(v) \leq c^{1/2}$
    - Sort all colors according to their distances from v:



    - $|B(v)|$ bounded by the location of the first router on this list.

An $O(nc^{1/2})$-space 3-stretch Oracle

- Expected space required:
  - total size of all routers tables = $nc^{1/2}$
  - size of any ball $B(v) \le c^{1/2}$
    - Sort all colors according to their distances from v:



    - $|B(v)|$ bounded by the location of the first router on this list.
  - total size of all ball tables $\le nc^{1/2}$

An $O(nc^{1/2})$-space <u>3-stretch</u> Oracle

An $O(nc^{1/2})$-space 3-stretch Oracle

- On query (v, λ):

An $O(nc^{1/2})$-space 3-stretch Oracle

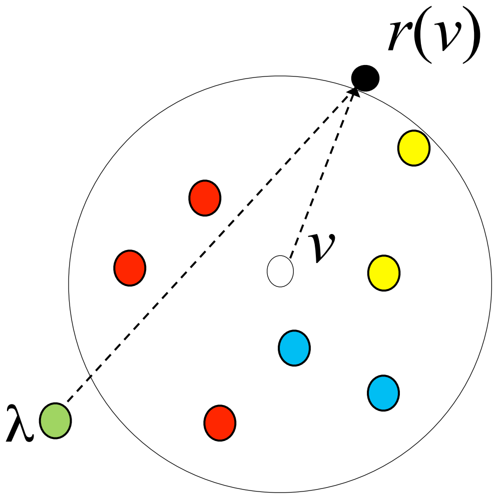
- On query (v,λ):
  - If $\lambda \in B(v)$ then return $\delta(v,\lambda)$. (stretch 1)

An $O(nc^{1/2})$-space 3-stretch Oracle

- On query (v,λ):
  - If $\lambda \in B(v)$ then return $\delta(v,\lambda)$. (stretch 1)

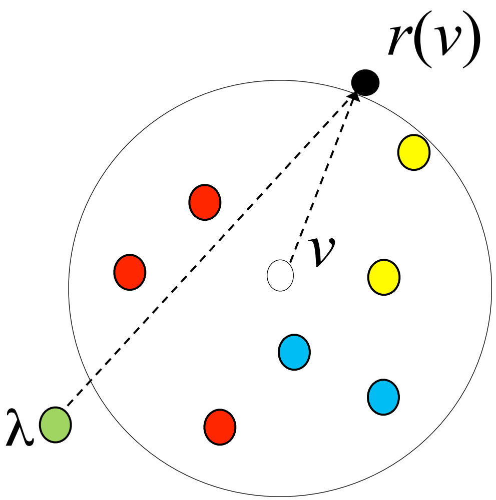
An $O(nc^{1/2})$-space 3-stretch Oracle

- On query (v,λ):
  - If $\lambda \in B(v)$ then return $\delta(v,\lambda)$. (stretch 1)
  - If $\lambda \notin B(v)$ then return $\delta(v,r(v)) + \delta(r(v),\lambda)$. (stretch 3)



An $O(nc^{1/2})$-space 3-stretch Oracle

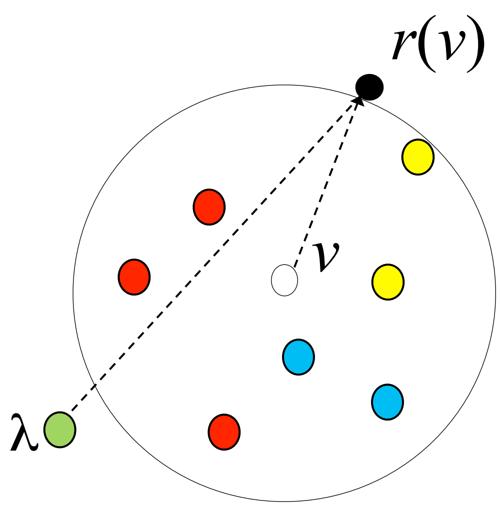
- On query (v,λ):
  - If $\lambda \in B(v)$ then return $\delta(v,\lambda)$. (stretch 1)
  - If $\lambda \notin B(v)$ then return $\delta(v,r(v)) + \delta(r(v),\lambda)$. (stretch 3)



1. $\lambda \notin B(v) \Rightarrow \delta(v,r(v)) \leq \delta(v,\lambda)$

An $O(nc^{1/2})$-space <u>3-stretch</u> Oracle

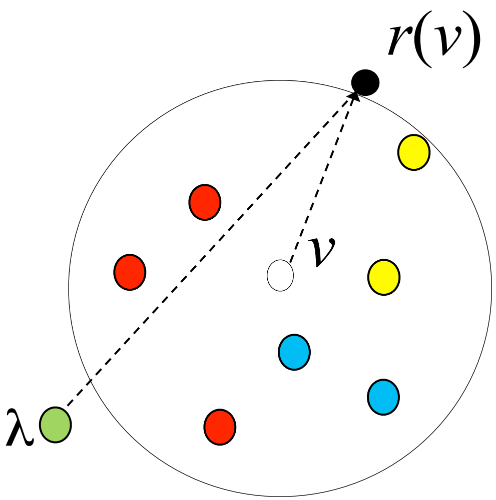
- On query (v,λ):
  - If $\lambda \in B(v)$ then return $\delta(v,\lambda)$. (stretch 1)
  - If $\lambda \notin B(v)$ then return $\delta(v,r(v)) + \delta(r(v),\lambda)$. (stretch 3)



1. $\lambda \notin B(v) \Rightarrow \delta(v,r(v)) \leq \delta(v,\lambda)$
2. $\delta(r(v),\lambda) \leq \delta(v,r(v)) + \delta(v,\lambda)$

An $O(nc^{1/2})$-space 3-stretch Oracle

- On query (v,λ):
  - If $\lambda \in B(v)$ then return $\delta(v,\lambda)$. (stretch 1)
  - If $\lambda \notin B(v)$ then return $\delta(v,r(v)) + \delta(r(v),\lambda)$. (stretch 3)

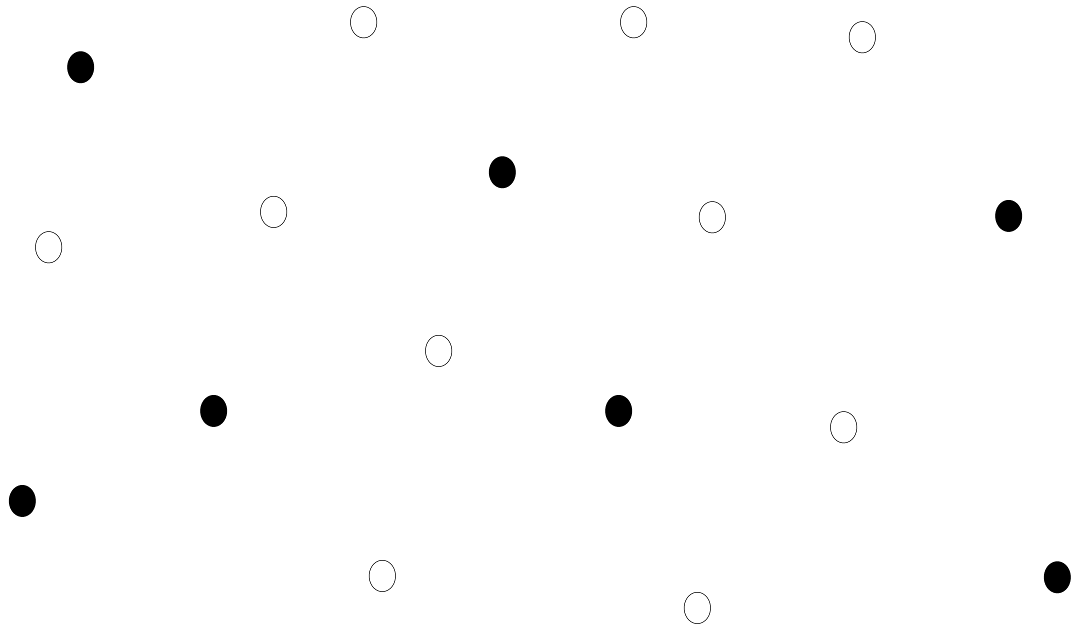


1. $\lambda \notin B(v) \Rightarrow \delta(v,r(v)) \leq \delta(v,\lambda)$
2. $\delta(r(v),\lambda) \leq \delta(v,r(v)) + \delta(v,\lambda)$

⇓

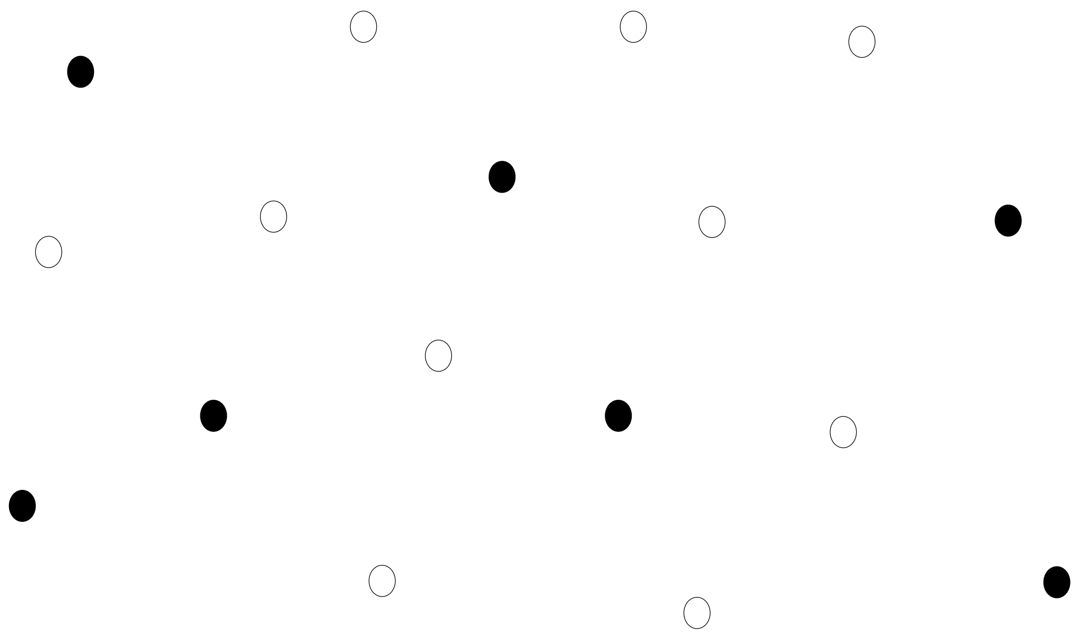
$$\delta(v,r(v)) + \delta(r(v),\lambda) \leq 3\delta(v,\lambda)$$

$O(knc^{1/k})$-space (2^k-1)-stretch Oracles



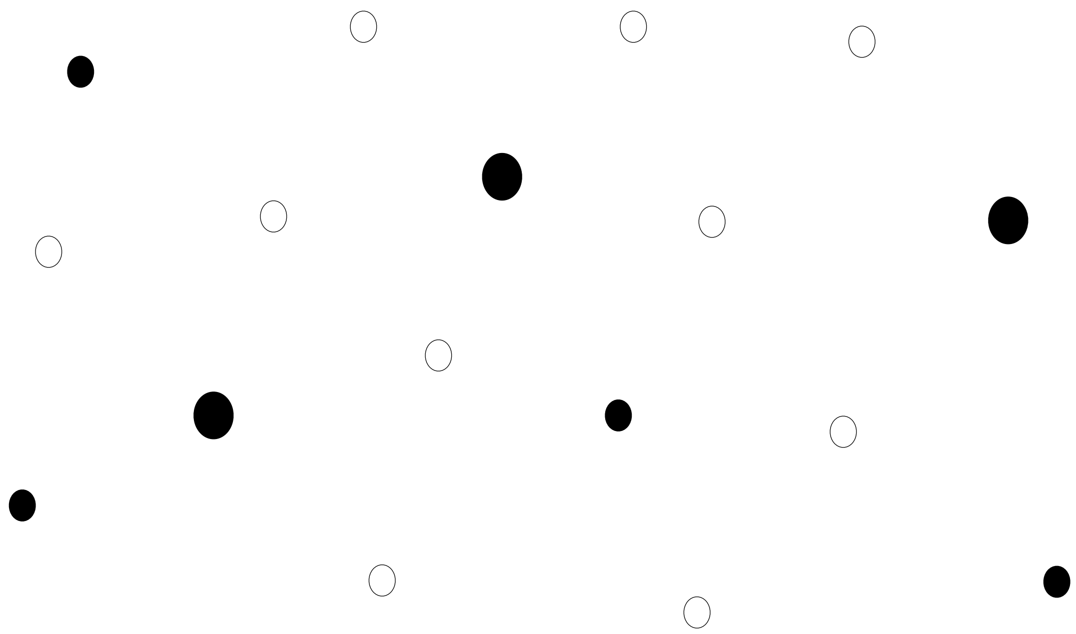
O(knc$^{1/k}$)-space (2^k-1)-stretch Oracles

- Select *routers* for the routers with prob. $c^{-1/k}$.



$O(knc^{1/k})$-space (2^k-1)-stretch Oracles

- Select *routers* for the routers with prob. $c^{-1/k}$.
  - and select routers for the routers ...



$O(knc^{1/k})$-space (2^k-1)-stretch Oracles

- Select *routers* for the routers with prob. $c^{-1/k}$.
  - and select routers for the routers ...

$O(knc^{1/k})$-space (2^k-1)-stretch Oracles

- Select *routers* for the routers with prob. $c^{-1/k}$.
  - and select routers for the routers ...
- k-1 levels of routers.

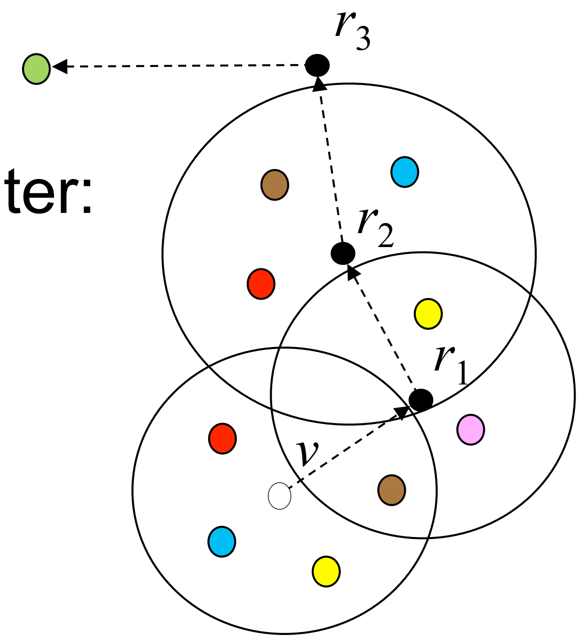
$O(knc^{1/k})$-space $(2^k\text{-}1)$-stretch Oracles

- Select *routers* for the routers with prob. $c^{-1/k}$.
  - and select routers for the routers ...
- k-1 levels of routers.

– Total size of router tables $O(knc^{-1/k})$.

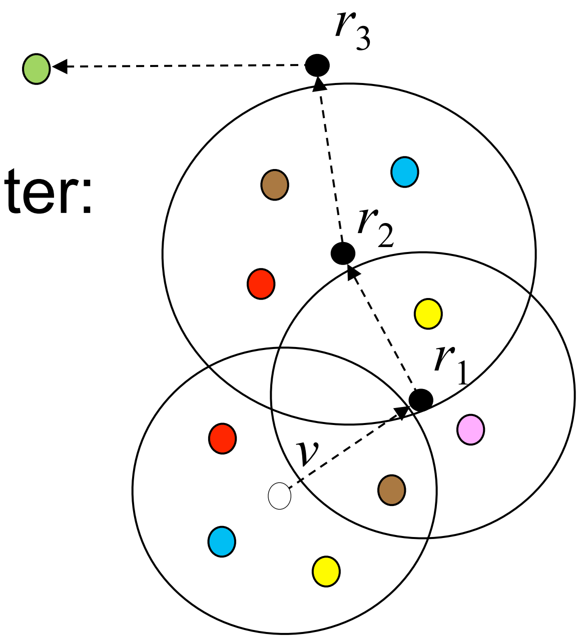
$O(knc^{1/k})$-space (2^k-1)-stretch Oracles

- Select *routers* for the routers with prob. $c^{-1/k}$.
  - and select routers for the routers ...
- k-1 levels of routers.
  - Total size of router tables $O(knc^{-1/k})$.
- Query algorithm hops from router to router:



$O(knc^{1/k})$-space $(2^k\text{-}1)$-stretch Oracles

- Select *routers* for the routers with prob. $c^{-1/k}$.
  - and select routers for the routers ...
- k-1 levels of routers.
  - Total size of router tables $O(knc^{-1/k})$.
- Query algorithm hops from router to router:
  - Stretch increases to 2^k-1.



Changing Colors

- ➢ Maintain balls using heaps.

Changing Colors

- Maintain balls using heaps.
  - Heap per color, sorted by distance from ball center.

Changing Colors

- Maintain balls using heaps.
  - Heap per color, sorted by distance from ball center.

- Requires selecting routers with probability depending on n.
  - $O(kn^{1+1/k})$ space instead of $O(knc^{1/k})$.

Changing Colors

- Maintain balls using heaps.
  - Heap per color, sorted by distance from ball center.

- Requires selecting routers with probability depending on n.
  - $O(kn^{1+1/k})$ space instead of $O(knc^{1/k})$.

- On color change of v:
  - Update two heaps in each ball that contains v.

Changing Colors

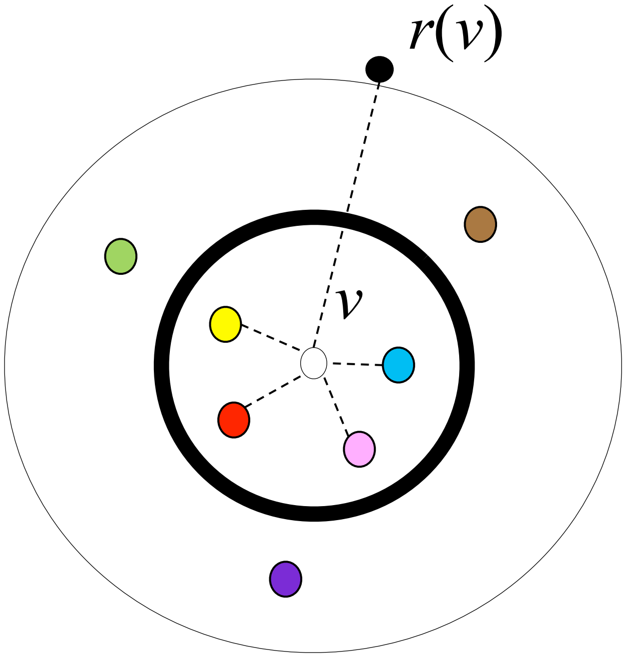
Changing Colors

- Problem: v can belong to a lot of balls.

Changing Colors

- Problem: v can belong to a lot of balls.
- Solution: use half-balls.

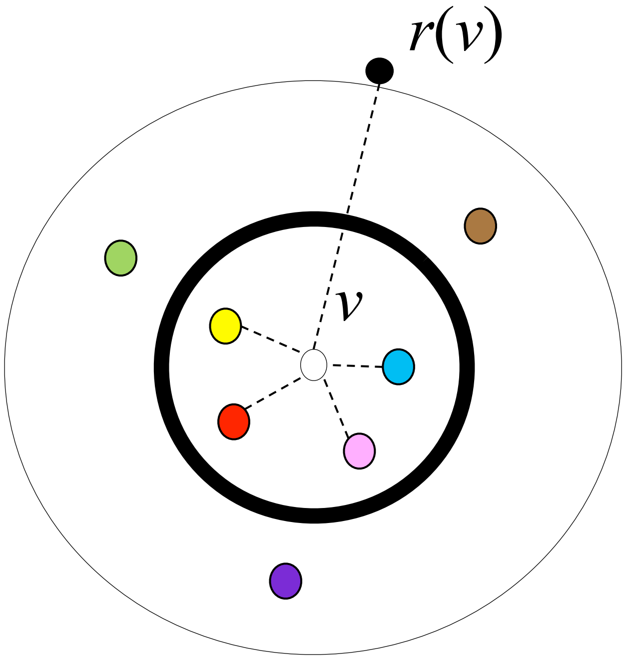
$$B^{1/2}(v) := \{\, u \mid \delta(v,u) < \tfrac{1}{2} \cdot \delta(v, r(v)) \,\}$$



Changing Colors

- Problem: v can belong to a lot of balls.
- Solution: use half-balls.

$$B^{1/2}(v) := \{ u \mid \delta(v,u) < 1/2 \cdot \delta(v,r(v)) \}$$



this causes the stretch of the queries to increase

Changing Colors

Changing Colors

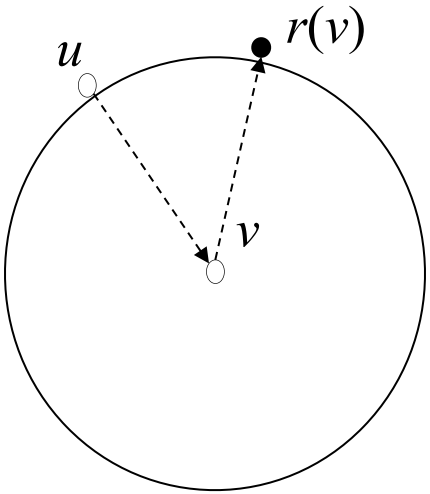
- <u>Observation:</u> $v \in B^{1/2}(u) \Rightarrow u \in B(v)$.

Changing Colors

- <u>Observation:</u> $v \in B^{1/2}(u) \Rightarrow u \in B(v)$.
- <u>Equivalently:</u> $u \notin B(v) \Rightarrow v \notin B^{1/2}(u)$.

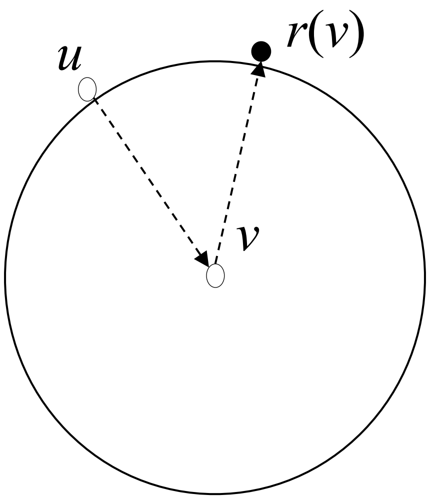
Changing Colors

- Observation: $v \in B^{1/2}(u) \Rightarrow u \in B(v)$.
- Equivalently: $u \notin B(v) \Rightarrow v \notin B^{1/2}(u)$.



Changing Colors

- Observation: $v \in B^{1/2}(u) \Rightarrow u \in B(v)$.
- Equivalently: $u \notin B(v) \Rightarrow v \notin B^{1/2}(u)$.
- v is in at most $kn^{1/k}$ balls.



Open questions

Open questions

1. $O(kn^{1+1/k})$-space $(2k\text{-}1)$-stretch ?

Open questions

1. $O(kn^{1+1/k})$-space $(2k\text{-}1)$-stretch ?

2. $O(knc^{1/k})$-space $\mathrm{poly}(k)$-stretch ?

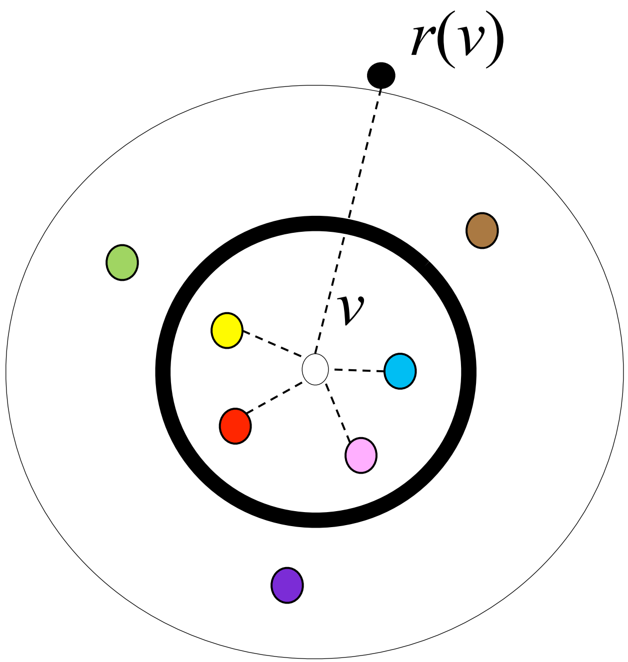
Open questions

1. $O(kn^{1+1/k})$-space $(2k\text{-}1)$-stretch ?

2. $O(knc^{1/k})$-space poly(k)-stretch ?

3. $O(knc^{1/k})$-space with changing colors ?

Thank you!



Thank you!