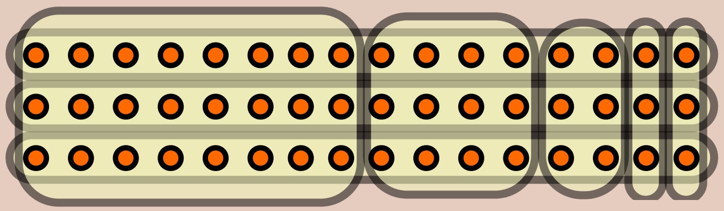


Linear-Work **Parallel Set Cover** and Variants Using MaNIS

Kanat Tangwongsan

Carnegie Mellon University

(Joint work with Guy Blelloch and Richard Peng)

Min Set Cover

Min Set Cover

Max Cover

Min Set Cover

Max Cover

Min-Sum Set Cover

Min Set Cover

Max Cover

Min-Sum Set Cover

Facility Location

Min Set Cover

Max Cover

Min-Sum Set Cover

Facility Location

Asymmetric *k*-Center

Min Set Cover

Max Cover

Min-Sum Set Cover

Facility Location

Asymmetric *k*-Center

Min Set Cover

Max Cover

Min-Sum Set Cover

Facility Location

Asymmetric *k*-Center

```
while (not done)
    pick highest utility option
```

simple greedy solution

Min Set Cover

Max Cover

Min-Sum Set Cover

Facility Location

Asymmetric *k*-Center

```
while (not done)
    pick highest utility option
```

simple greedy solution

inherently sequential

Min Set Cover

while (not done)

This Talk: MaNIS: Maximal Nearly Independent Set

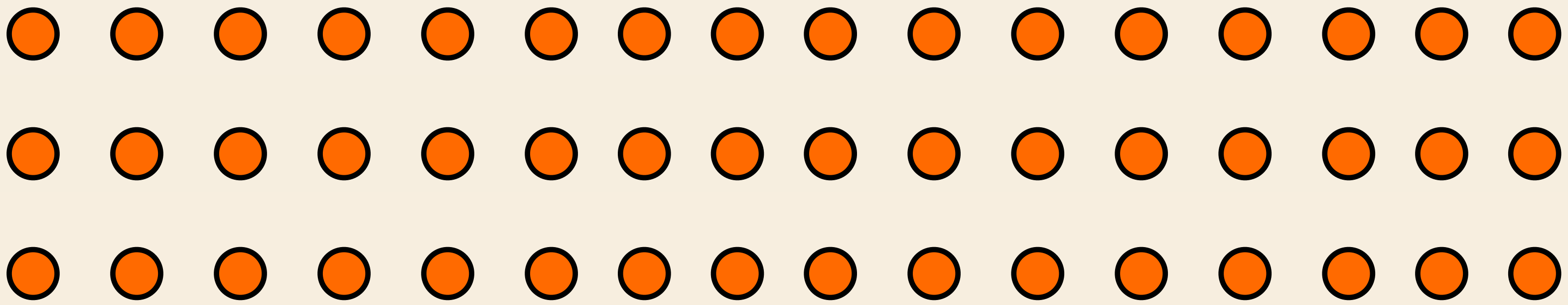
techniques for parallel approximate greedy algorithms

How to select a *maximal* collection of *nearly* non-overlapping sets
in linear work and polylog depth?

Asymmetric *k*-Center

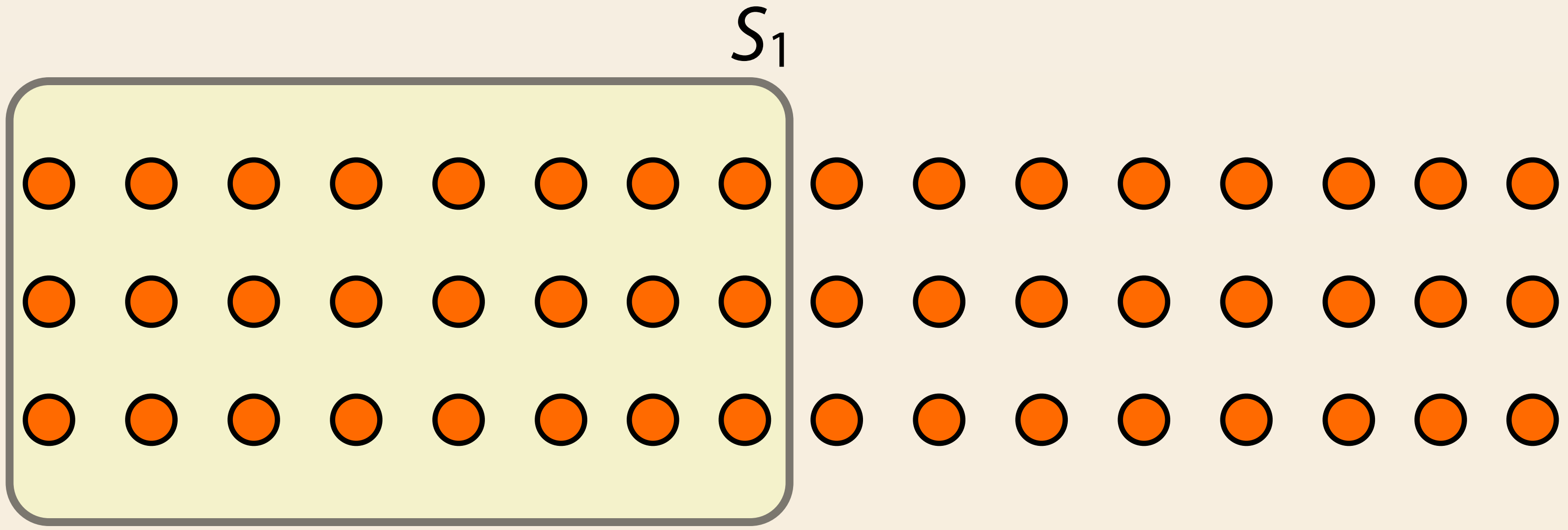
(Unweighted) Set Cover

Instance: elements and sets covering them



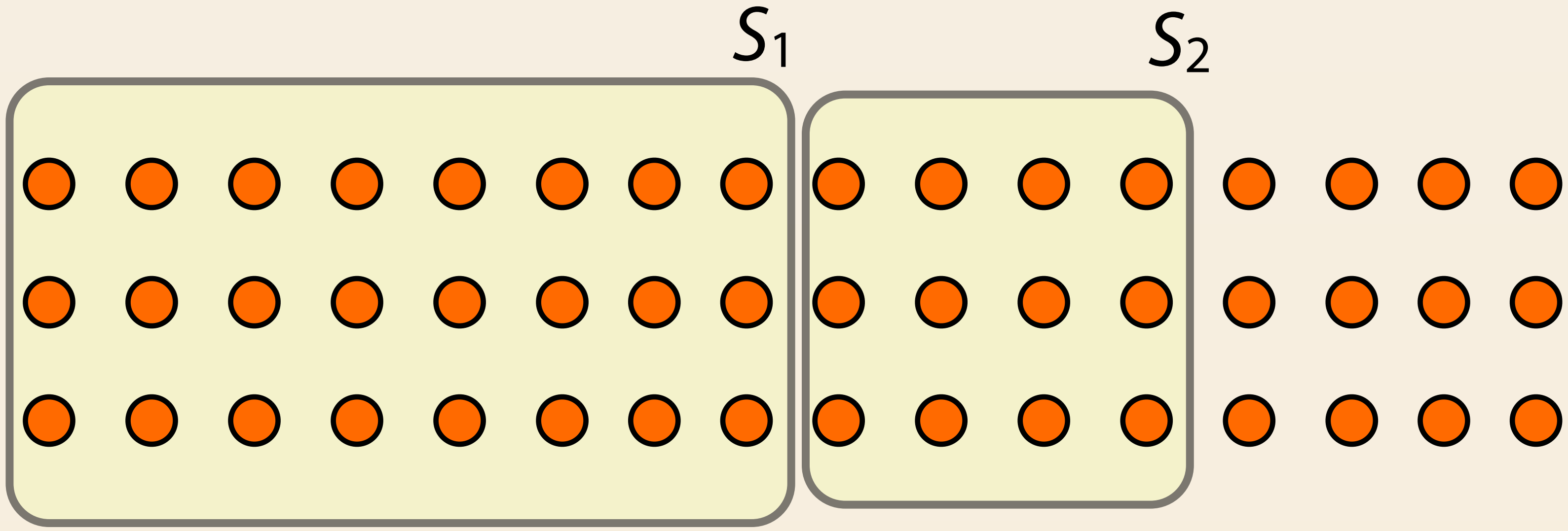
(Unweighted) Set Cover

Instance: elements and sets covering them



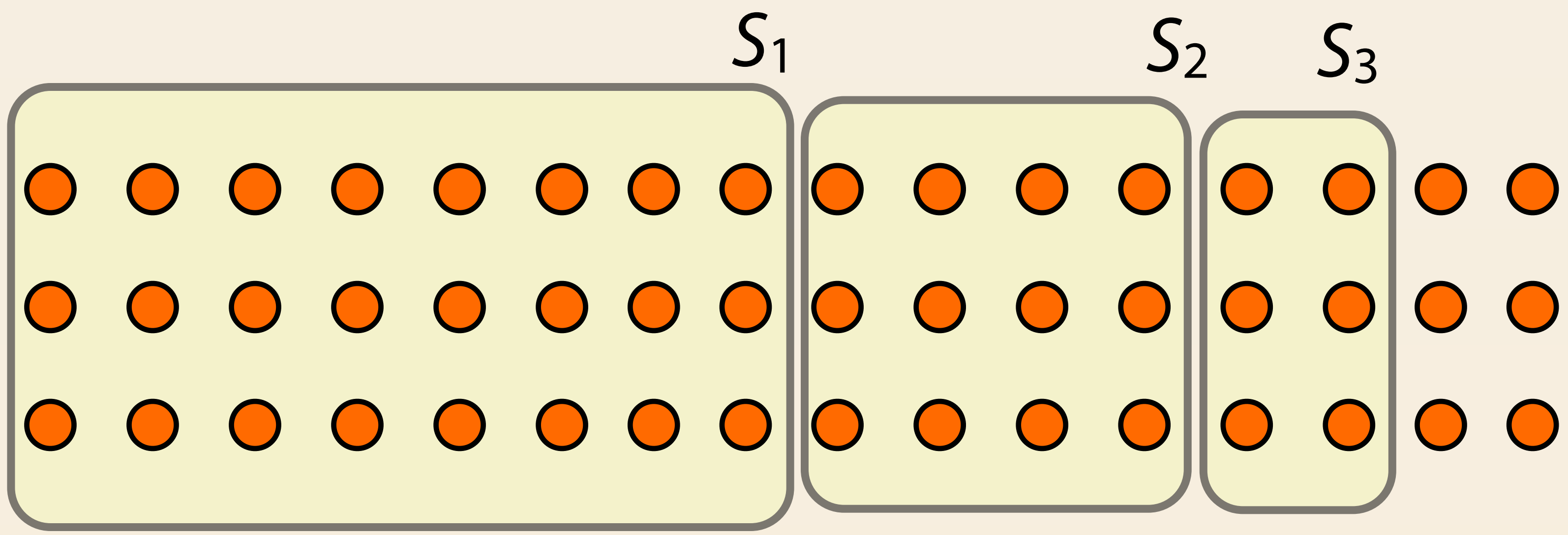
(Unweighted) Set Cover

Instance: elements and sets covering them



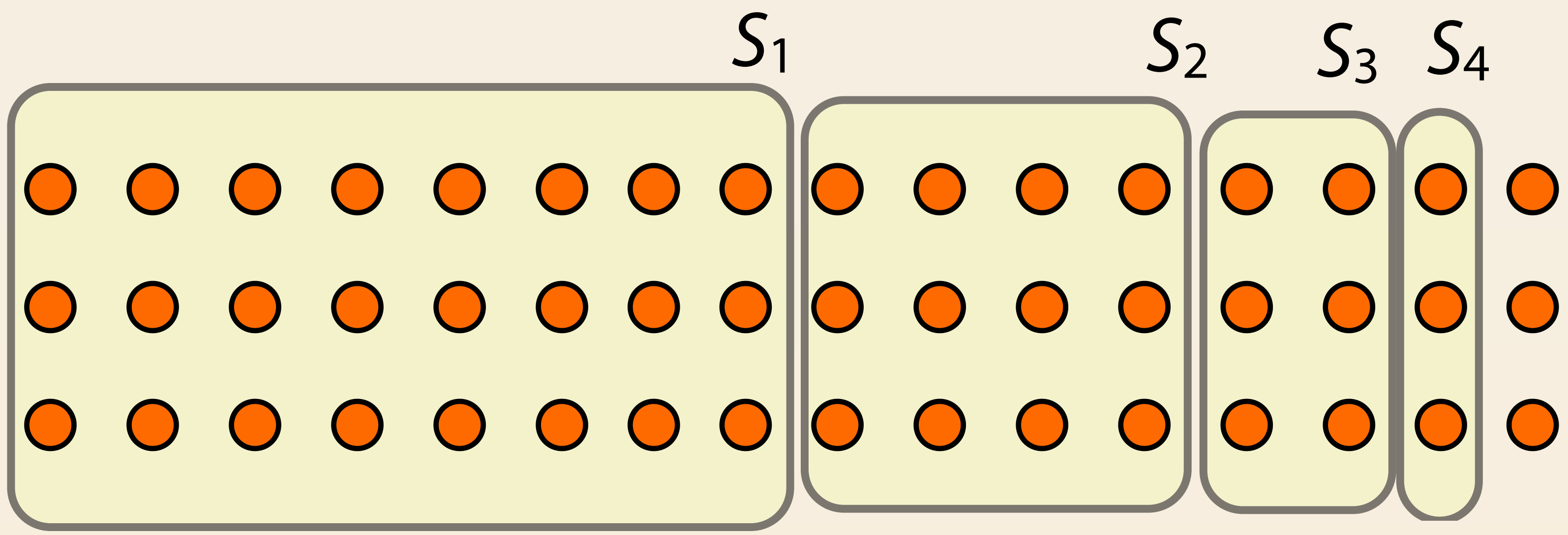
(Unweighted) Set Cover

Instance: elements and sets covering them



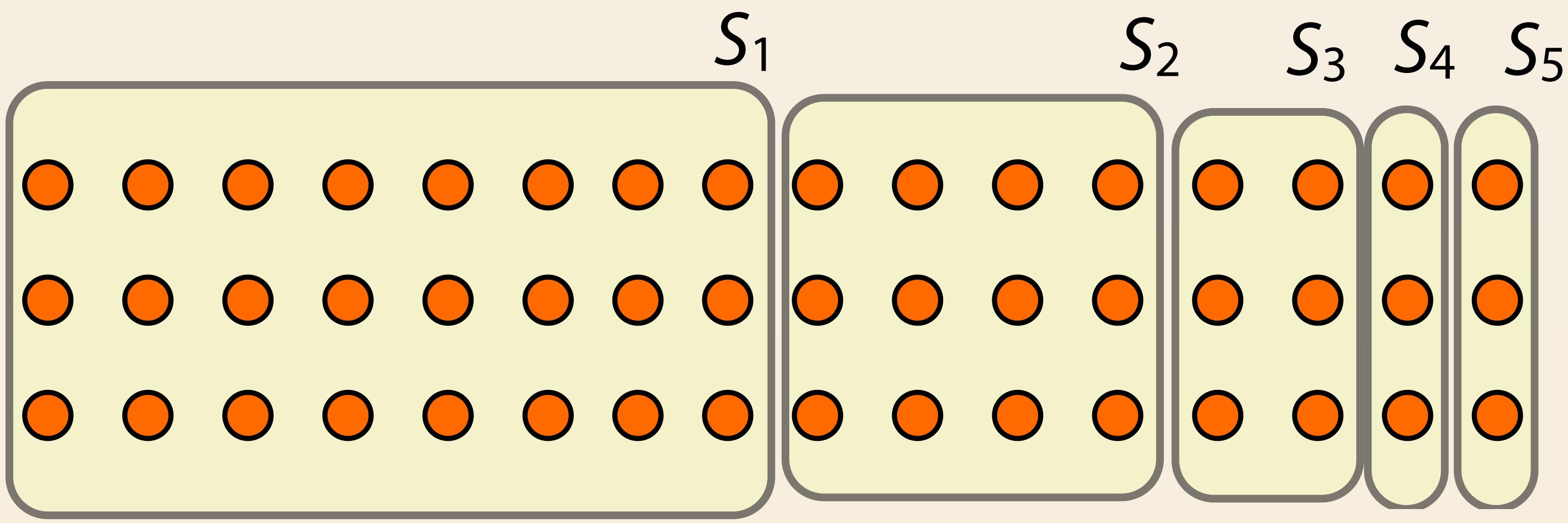
(Unweighted) Set Cover

Instance: elements and sets covering them



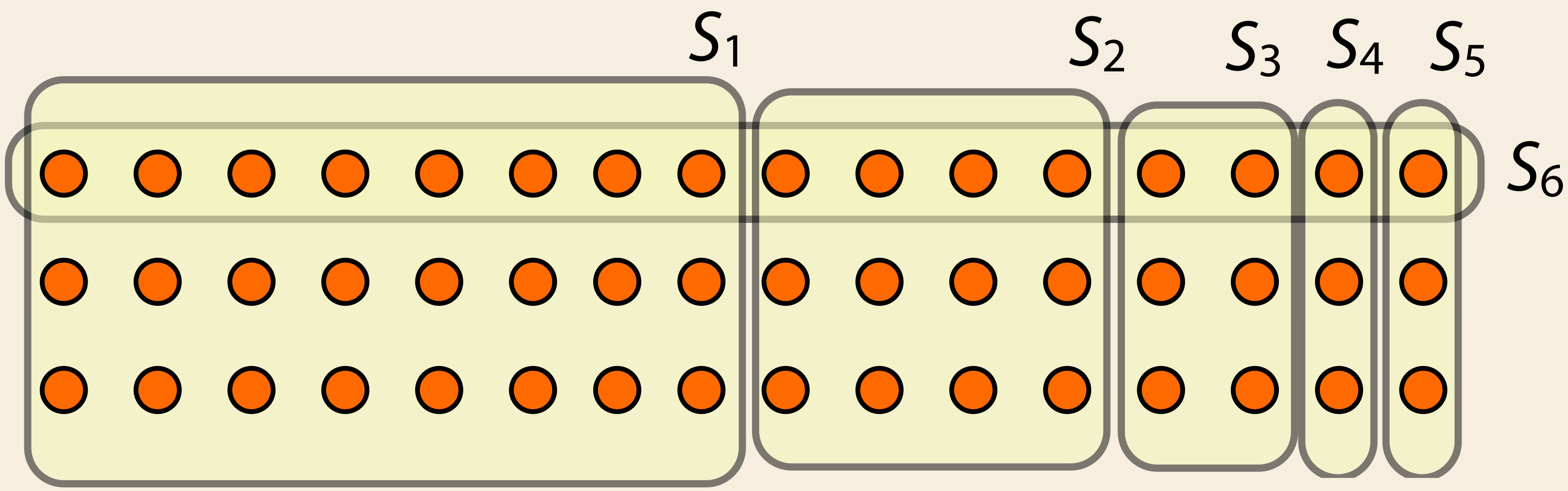
(Unweighted) Set Cover

Instance: elements and sets covering them



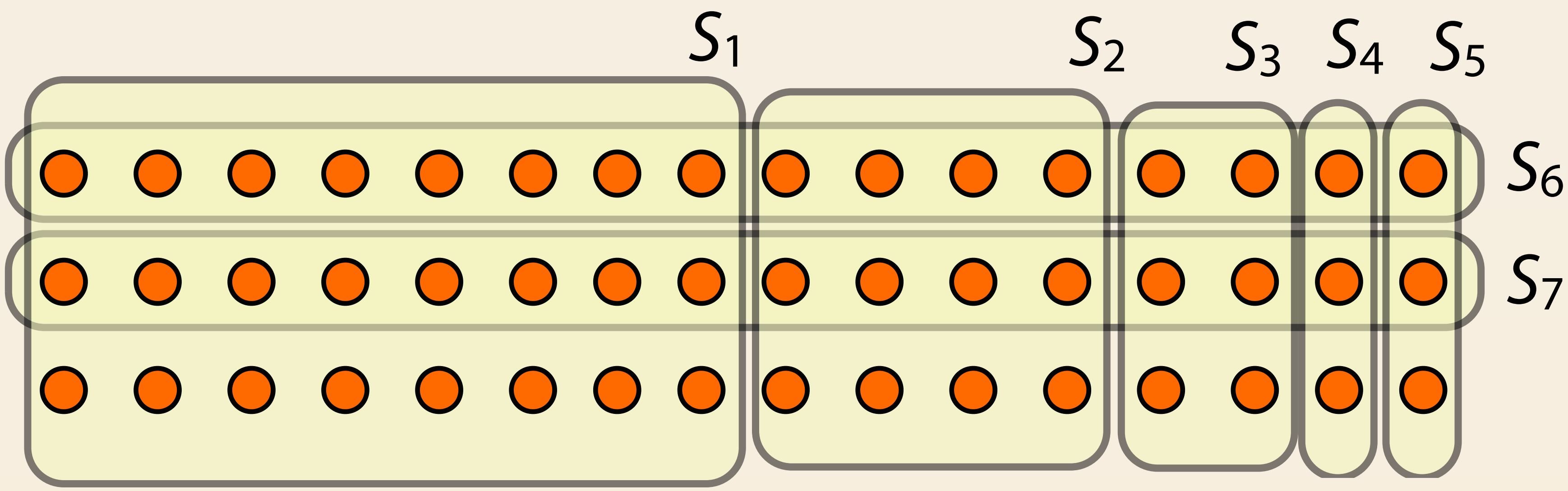
(Unweighted) Set Cover

Instance: elements and sets covering them



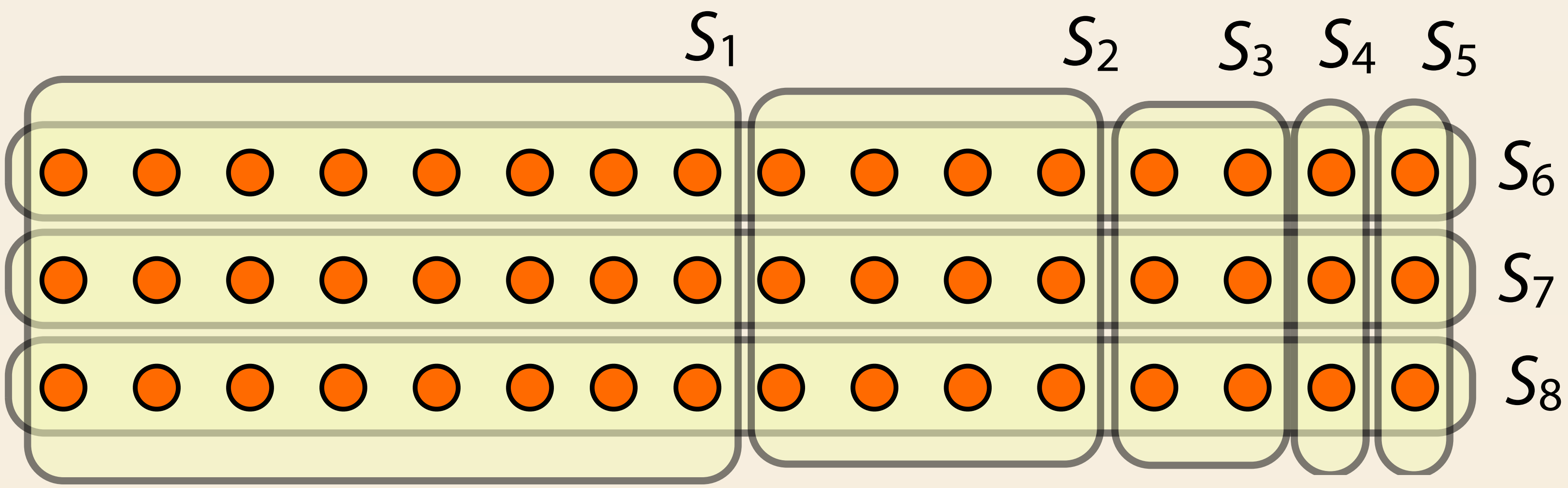
(Unweighted) Set Cover

Instance: elements and sets covering them



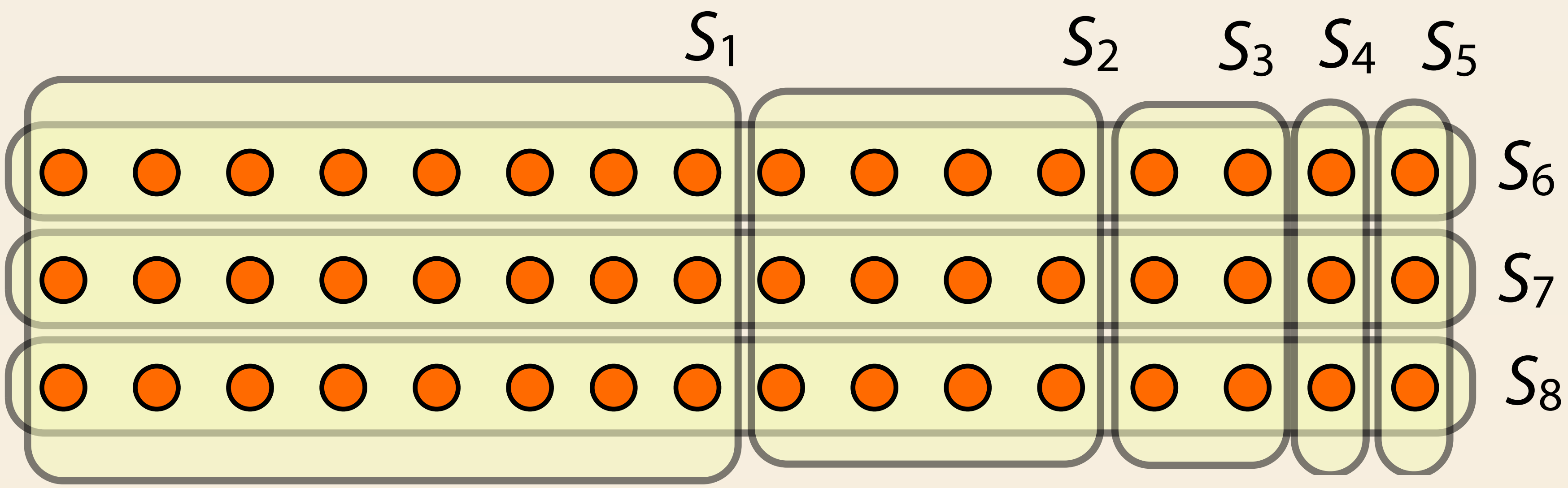
(Unweighted) Set Cover

Instance: elements and sets covering them



(Unweighted) Set Cover

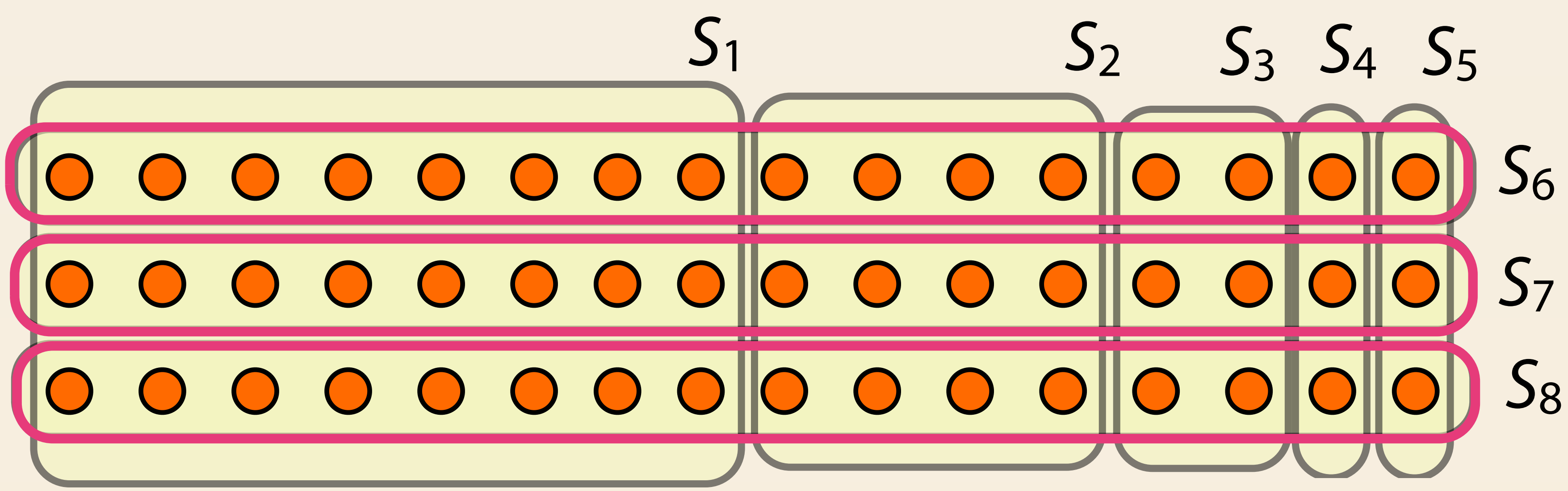
Instance: elements and sets covering them



Task: cover all elements using fewest sets

(Unweighted) Set Cover

Instance: elements and sets covering them



Task: cover all elements using fewest sets

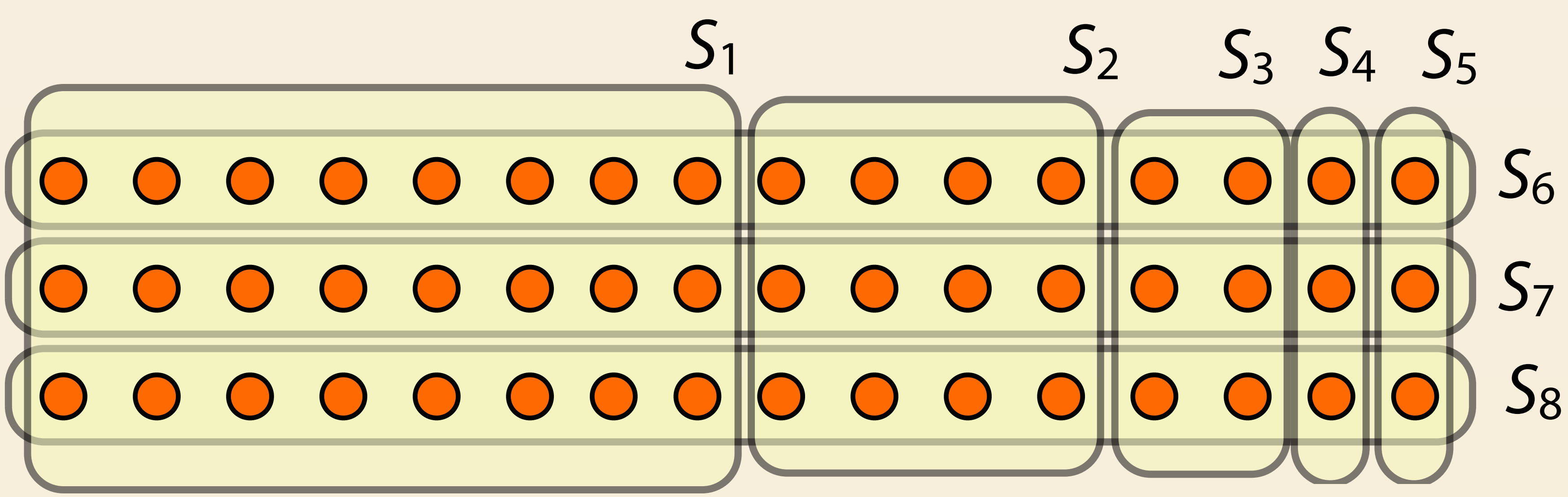
Greedy Set Cover

[Johnson'74, Chvatal'79]

For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered



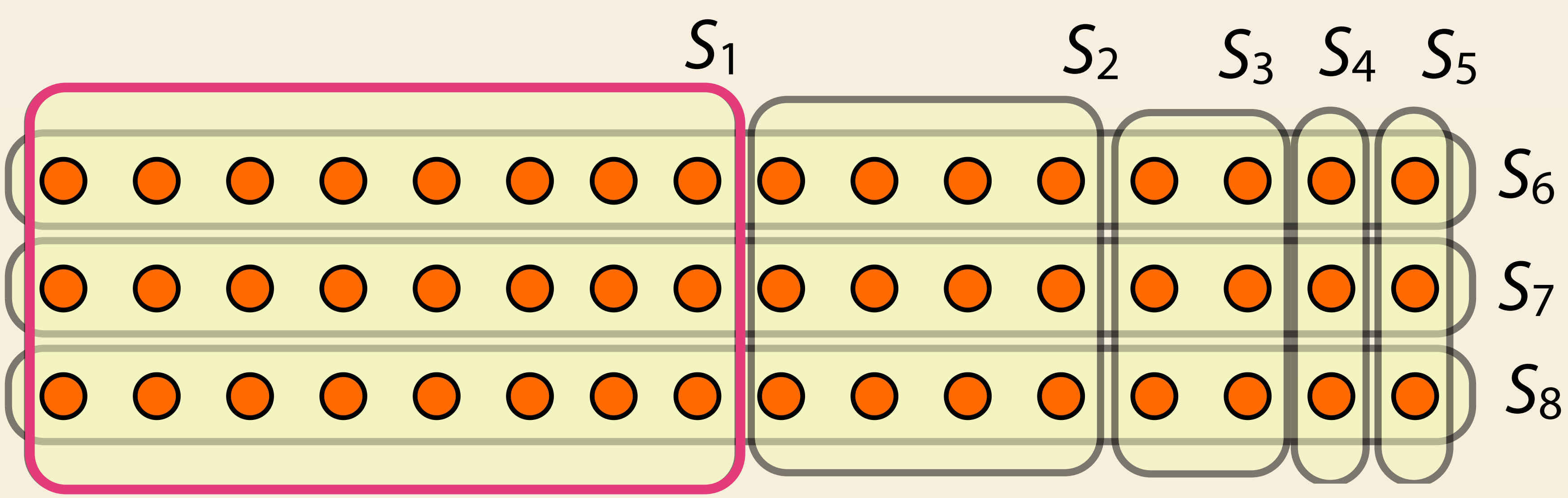
Greedy Set Cover

[Johnson'74, Chvatal'79]

For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered



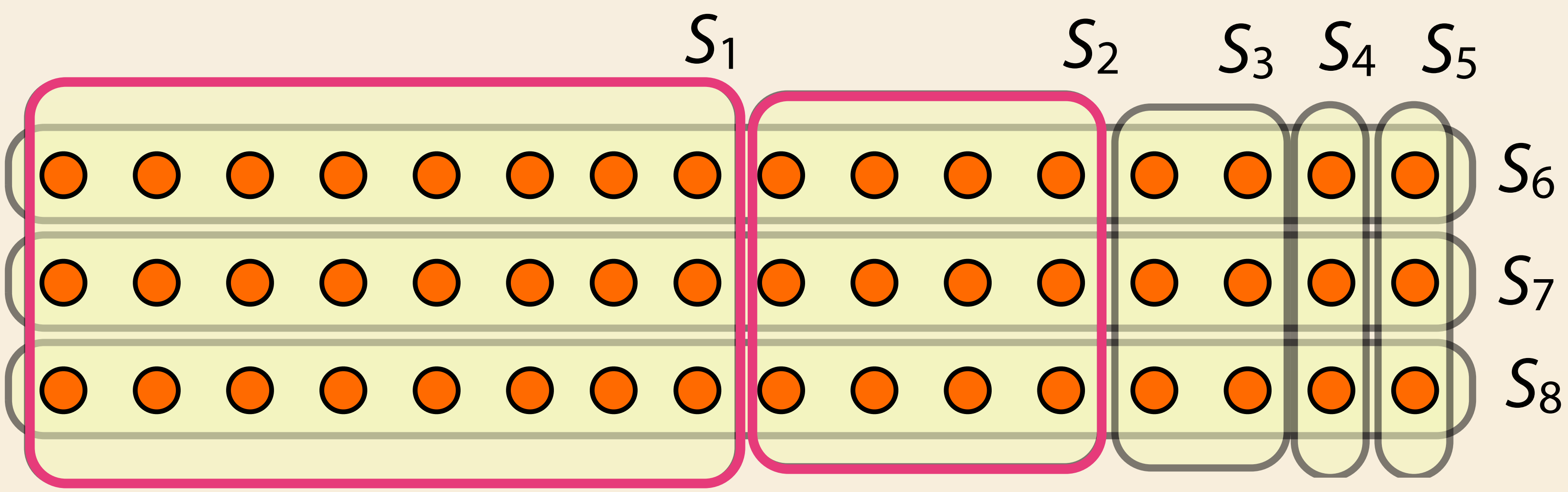
Greedy Set Cover

[Johnson'74, Chvatal'79]

For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered



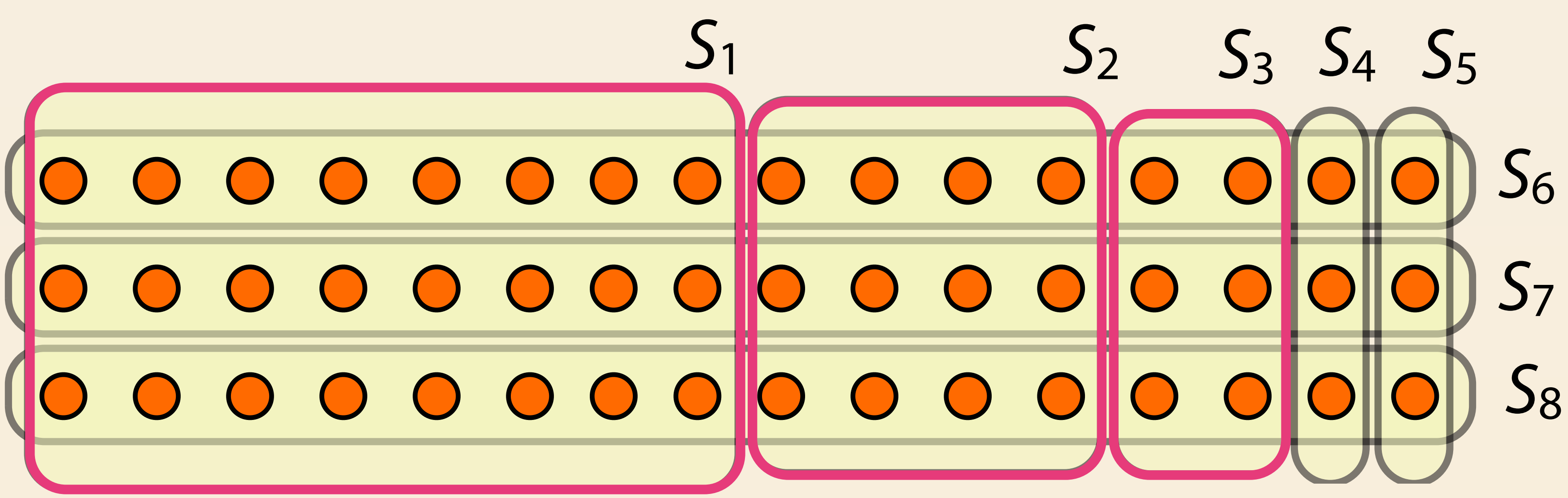
Greedy Set Cover

[Johnson'74, Chvatal'79]

For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered



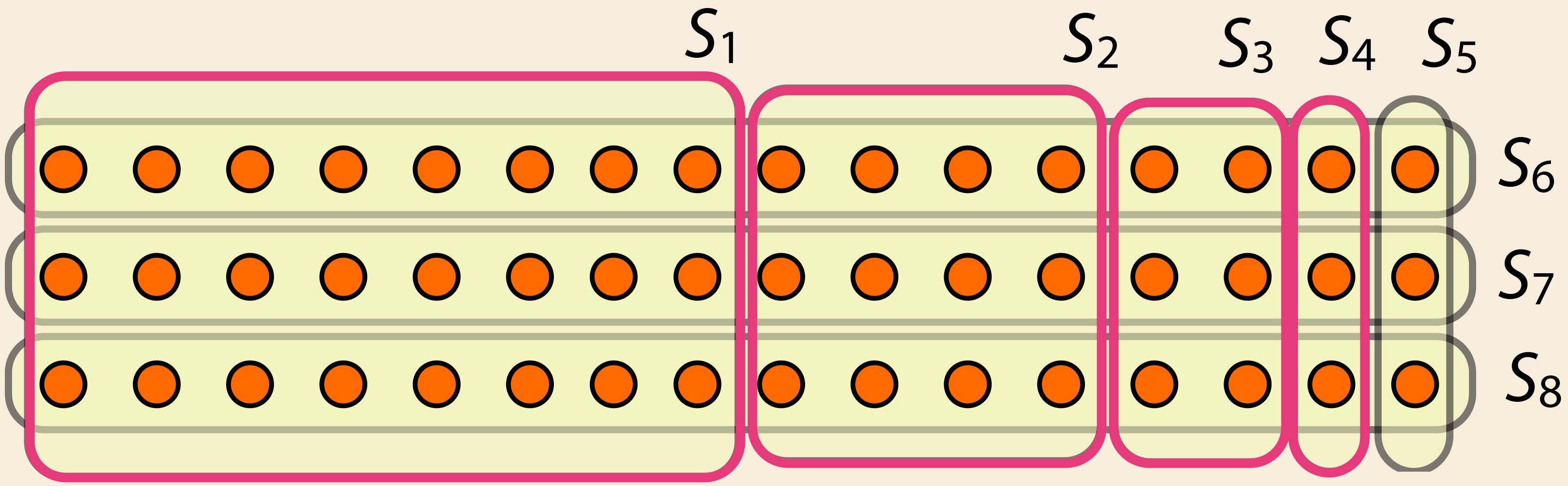
Greedy Set Cover

[Johnson'74, Chvatal'79]

For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered



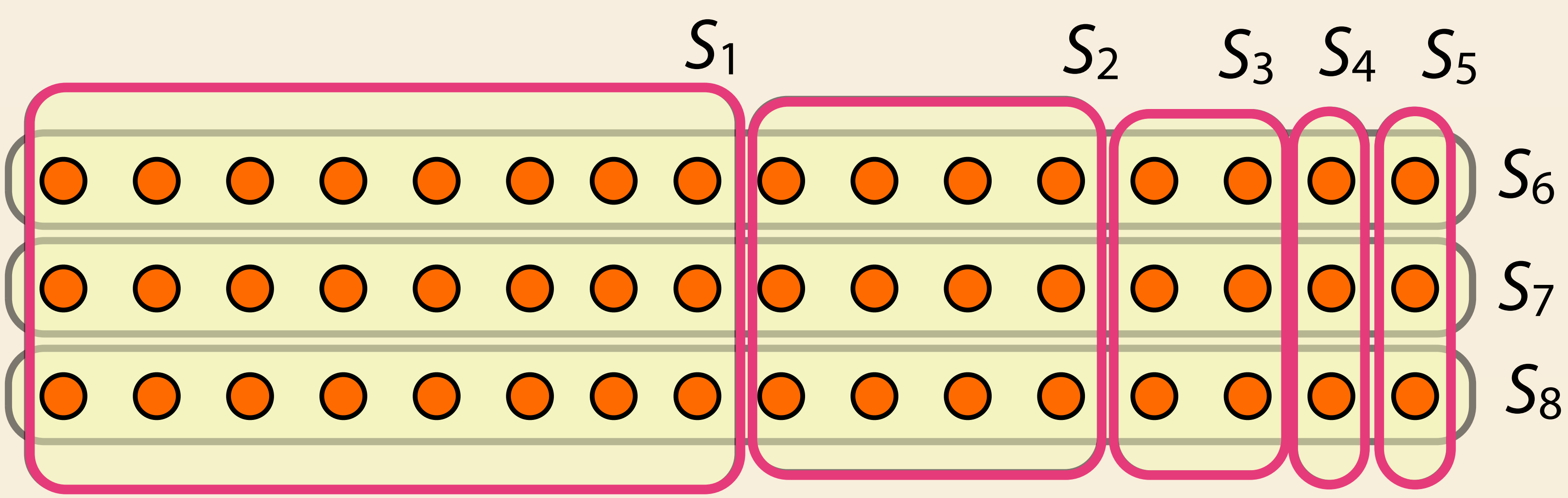
Greedy Set Cover

[Johnson'74, Chvatal'79]

For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered



Greedy Set Cover

[Johnson'74, Chvatal'79]

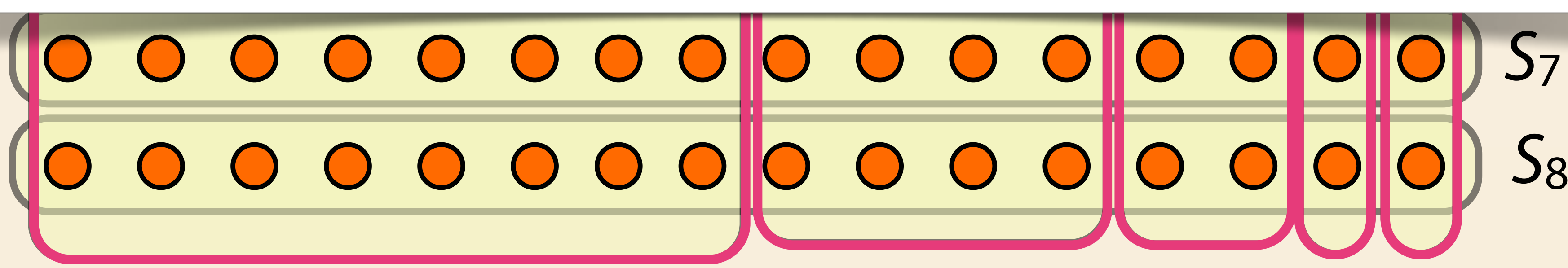
For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered

Thm: Greedy is a $(\ln n + 1)$-approximation.

[Johnson'74, Chvatal'79]



Greedy Set Cover

[Johnson'74, Chvatal'79]

For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered

Thm: Greedy is a ($\ln n + 1$)-approximation.

[Johnson'74, Chvatal'79]

Thm: Can't beat greedy unless P = NP.

[Raz-Safra'97, Feige'98,Alon et al.'06]

Parallelizing Greedy Set Cover

Parallelizing Greedy Set Cover

Thm: Faithfully greedy set cover is P-complete.
[Bongiovanni et al.'95, Blelloch et al.'11]

Parallelizing Greedy Set Cover

Thm: Faithfully greedy set cover is P-complete.
[Bongiovanni et al.'95, Blelloch et al.'11]

m = sum of set sizes, n = # of elements

Parallelizing Greedy Set Cover

Thm: Faithfully greedy set cover is P-complete.
[Bongiovanni et al.'95, Blelloch et al.'11]

m = sum of set sizes, n = # of elements

Berger, Rompel, and Shor'94

- $(1+\varepsilon)(1+\ln n)$-approx, RNC $O(m\log^4 m)$-work

Rajagopalan and Vazirani'98

- $(2+\varepsilon)(1+\ln n)$-approx, RNC $O(m\log^2 m)$-work

Parallelizing Greedy Set Cover

Our Result for Set Cover:

$(1+\varepsilon)(1+\ln n)$-approximation, $O(m)$-work, $O(\log^3 m)$-depth

m = sum of set sizes, n = # of elements

Berger, Rompel, and Shor'94

- $(1+\varepsilon)(1+\ln n)$-approx, RNC $O(m\log^4 m)$-work

Rajagopalan and Vazirani'98

- $(2+\varepsilon)(1+\ln n)$-approx, RNC $O(m\log^2 m)$-work

Parallelizing Greedy Set Cover

Our Result for Set Cover:

$(1+\varepsilon)(1+\ln n)$-approximation, $O(m)$-work, $O(\log^3 m)$-depth

m = sum of set sizes, n = # of elements

Our Main Result:

- Formulation of MaNIS
- $O(|E|)$-work, $O(\log^2 |E|)$-depth algorithm for MaNIS

Idea: Bulk Processing

handling multiple sets simultaneously

For $t = 0, 1, \ldots$ **until** elts all covered

Pick the set that covers the most **new** elements (say X_t)

X_t = new elements covered

Idea: Bulk Processing

handling multiple sets simultaneously

For $t = 0, 1, \ldots$ **until** elts all covered

~~Pick the set that covers the most **new** elements (say X_t)~~

X_t = new elements covered

Idea: Bulk Processing

handling multiple sets simultaneously

For $t = 0, 1, \ldots$ **until** elts all covered

~~Pick the set that covers the most **new** elements (say X_t)~~

X_t = new elements covered

Identify **all** sets that cover at least $(1-\varepsilon)|X_t|$
and bulk-process them a way that mimics greedy

Idea: Bulk Processing

handling multiple sets simultaneously

For $t = 0, 1, \ldots$ **until** elts all covered

~~Pick the set that covers the most **new** elements (say X_t)~~

X_t = new elements covered

Identify **all** sets that cover at least $(1-\varepsilon)|X_t|$
and bulk-process them a way that mimics greedy

Observation:

If $|X_{t+1}| \le (1 - \varepsilon)|X_t|$, then # of rounds is $O(\log_{1+\varepsilon} n)$.

Bulk processing these very best sets

Given sets covering roughly $|X_t|$ (i.e., between $(1-\varepsilon)|X_t|$ and $|X_t|$)

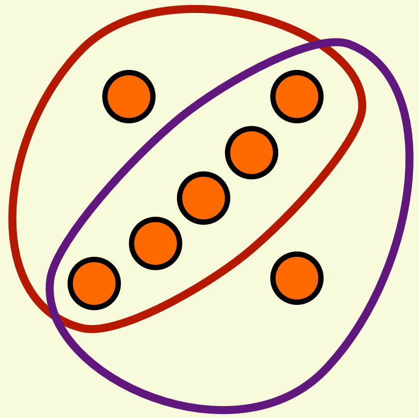
Want: each set chosen to cover roughly $|X_t|$ *new* elts.

Bulk processing these very best sets

Given sets covering roughly $|X_t|$ (i.e., between $(1-\varepsilon)|X_t|$ and $|X_t|$)

Want: each set chosen to cover roughly $|X_t|$ *new* elts.

Example 1: lots of sharing



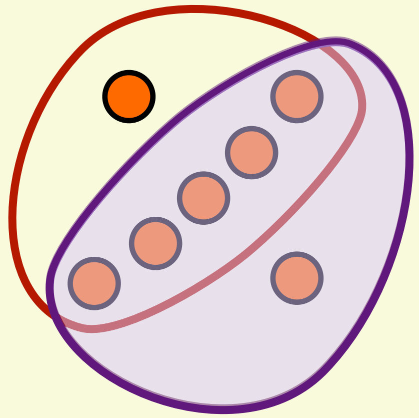
intended behavior: choose one

Bulk processing these very best sets

Given sets covering roughly $|X_t|$ (i.e., between $(1-\varepsilon)|X_t|$ and $|X_t|$)

Want: each set chosen to cover roughly $|X_t|$ *new* elts.

Example 1: lots of sharing



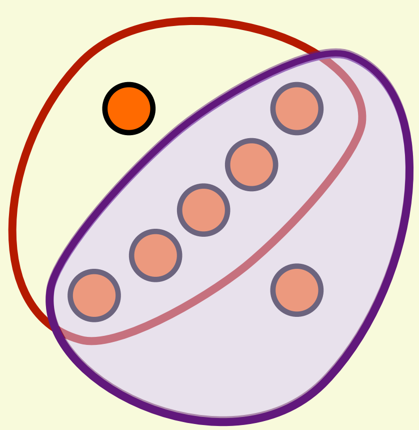
intended behavior: choose one

Bulk processing these very best sets

Given sets covering roughly $|X_t|$ (i.e., between $(1-\varepsilon)|X_t|$ and $|X_t|$)

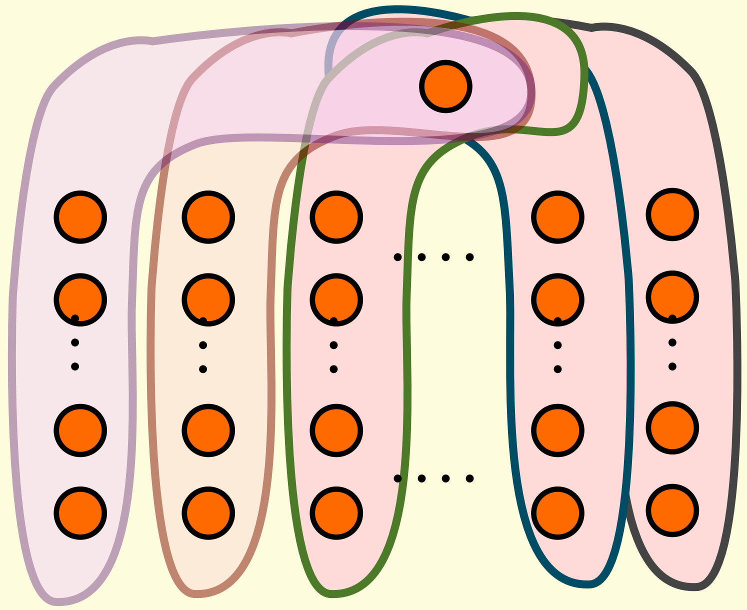
Want: each set chosen to cover roughly $|X_t|$ *new* elts.

Example 1: lots of sharing



intended behavior: choose one

Example 2: little sharing



intended behavior: choose all

Bulk processing these very best sets

(cont'd)

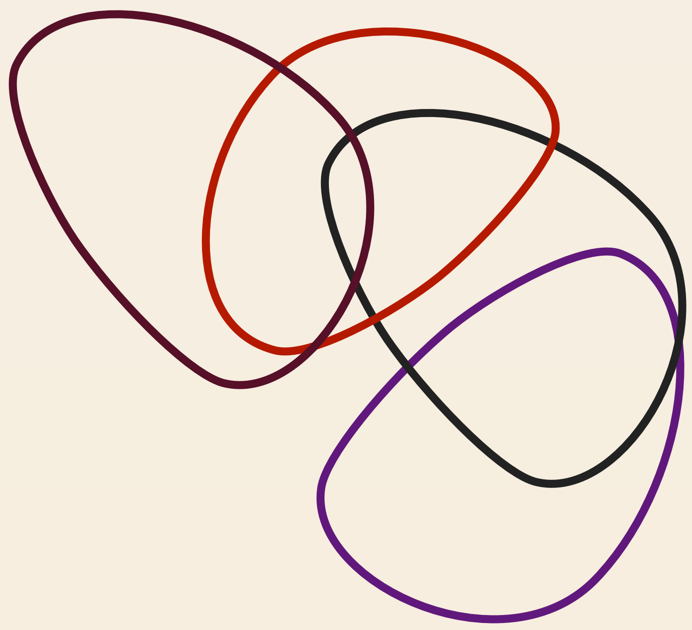
Bulk processing these very best sets

(cont'd)

1 **PICK:**
if has a small-overlapping sequence

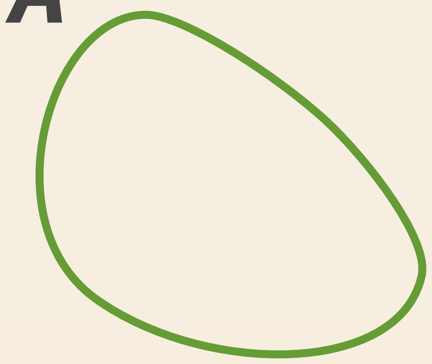
Bulk processing these very best sets

(cont'd)



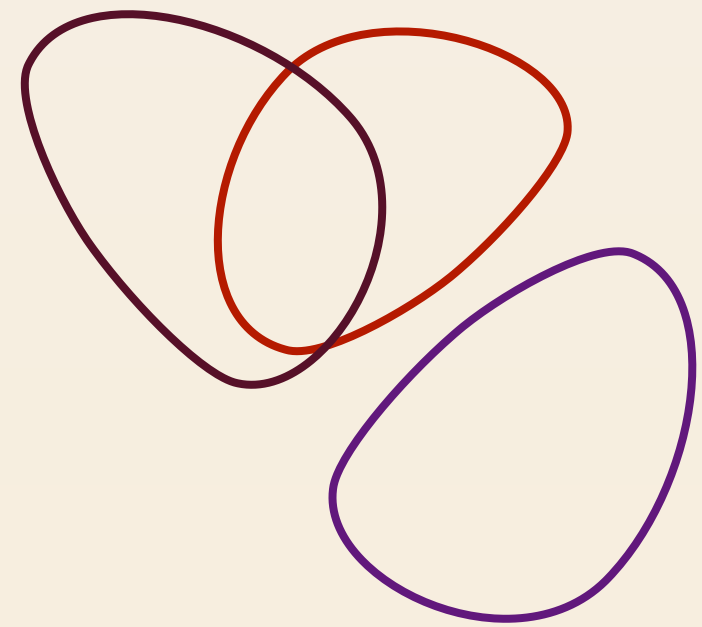
1 **PICK:**
if has a small-overlapping sequence

A



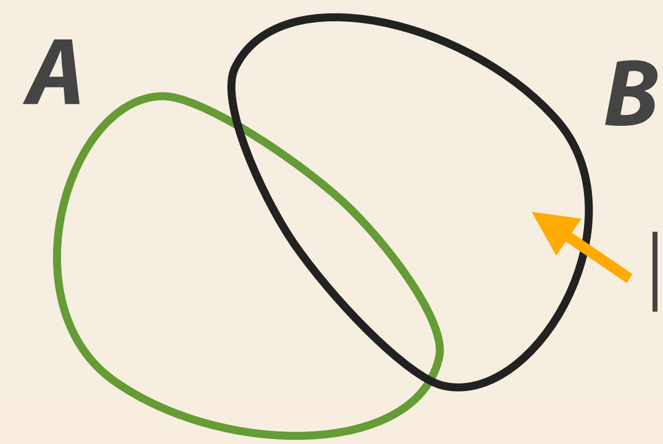
Bulk processing these very best sets

(cont'd)



1 **PICK:**

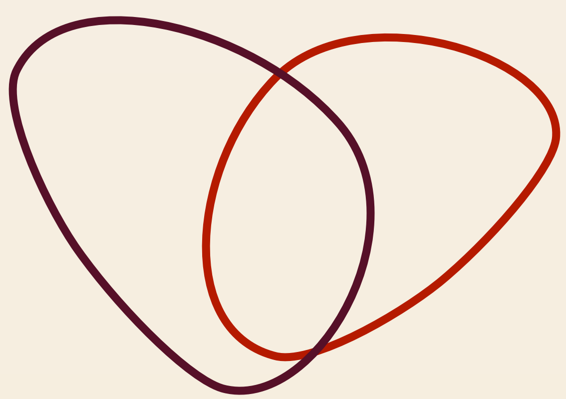
if has a small-overlapping sequence



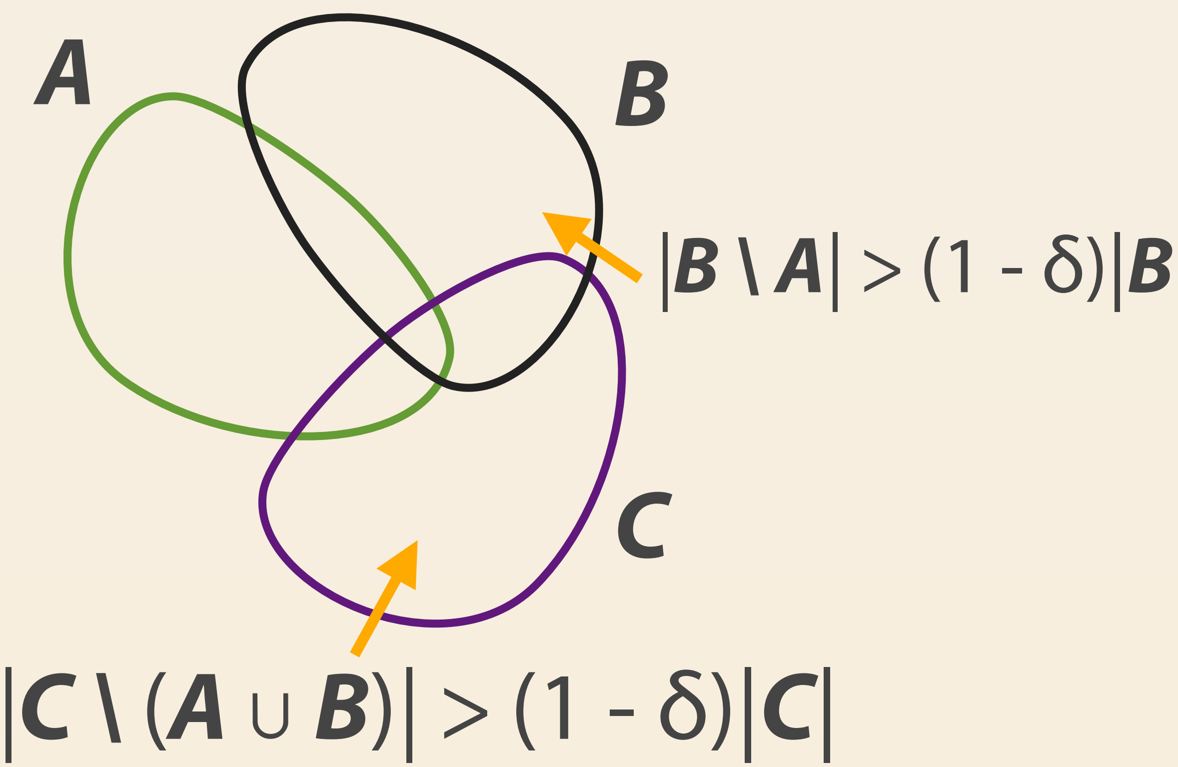
$|\boldsymbol{B} \setminus \boldsymbol{A}| > (1 - \delta)|\boldsymbol{B}|$

Bulk processing these very best sets

(cont'd)



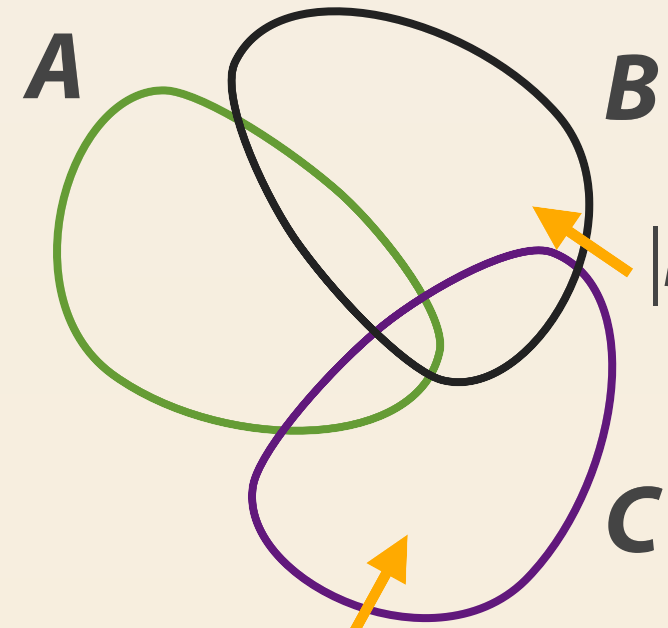
1 **PICK:**
if has a small-overlapping sequence



Bulk processing these very best sets

(cont'd)

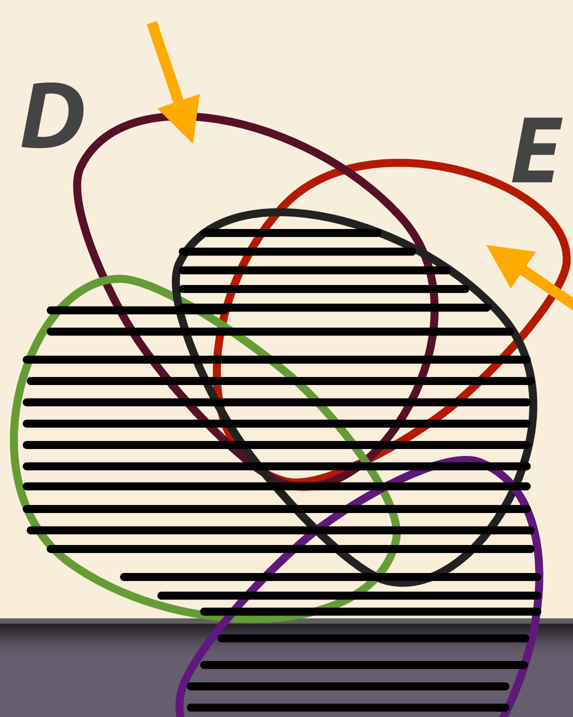
1 **PICK:**
if has a small-overlapping sequence



$|B \setminus A| > (1 - \delta)|B|$

$|C \setminus (A \cup B)| > (1 - \delta)|C|$

$|D \setminus (A \cup B \cup C)| < (1 - \varepsilon)|D|$



2 **REJECT:** if retains too few elts

$|E \setminus (A \cup B \cup C)| < (1 - \varepsilon)|E|$

MaNIS: Maximal Nearly Independent Set

formalizing our intuitions

Input: **SETS** = collection of sets

For δ ≥ ε, (ε, δ)-**MaNIS** is ***J* = <*S*$_1$, ... , *S*$_k$>** ⊆ **SETS such that**

MaNIS: Maximal Nearly Independent Set

formalizing our intuitions

Input: **SETS** = collection of sets

For δ ≥ ε, (ε, δ)**-MaNIS** **is** ***J* = <*S*1, ... , *S*k>** ⊆ **SETS such that**

1 **Nearly Independent.**

$$|S_i \setminus (\cup_{j<i} S_j)| \geq (1 - \delta)|S_i|$$

MaNIS: Maximal Nearly Independent Set

formalizing our intuitions

Input: **SETS** = collection of sets

For δ ≥ ε, (ε, δ)-**MaNIS** **is** ***J* = <*S*$_1$, ... , *S*$_k$>** ⊆ **SETS such that**

1 **Nearly Independent.**

Only "slightly" worse than greedy

$$|S_i \setminus (\cup_{j<i} S_j)| \geq (1-\delta)|S_i|$$

MaNIS: Maximal Nearly Independent Set

formalizing our intuitions

Input: **SETS** = collection of sets

For δ ≥ ε, (ε, δ)-**MaNIS** is ***J* = <*S*$_1$, ... , *S*$_k$>** ⊆ **SETS such that**

1 **Nearly Independent.**

Only "slightly" worse than greedy

$$|S_i \setminus (\cup_{j<i} S_j)| \geq (1 - \delta)|S_i|$$

2 **Maximal.** Each $R_i \notin J$ *satisfies*

covered = $S_1 \cup S_2 \cup \cdots \cup S_k$

$|R_i \setminus$ **covered**$| < (1 - \varepsilon)|R_i|$

MaNIS: Observations and variants

For $\delta \geq \varepsilon$, (ε, δ)-**MaNIS** **is** ***J* = <*S*$_1$, … , *S*$_k$>** $\subseteq$ **SETS such that**

1 **Nearly Independent.**

$$|S_i \setminus (\cup_{j<i} S_j)| \geq (1-\delta)|S_i|$$

2 **Maximal.** Each $R_i \notin J$ *satisfies*

covered = $S_1 \cup S_2 \cup \cdots \cup S_k$

$|R_i \setminus \textbf{covered}| < (1-\varepsilon)|R_i|$

MaNIS: Observations and variants

For δ ≥ ε, (ε, δ)-**MaNIS** **is** ***J* = <*S*₁, … , *S*ₖ>** ⊆ **SETS such that**

1 **Nearly Independent.**

$$|S_i \setminus (\cup_{j<i} S_j)| \geq (1-\delta)|S_i|$$

2 **Maximal.** Each $R_i \notin J$ *satisfies*

covered = $S_1 \cup S_2 \cup \cdots \cup S_k$

$|R_i \setminus \textbf{covered}| < (1-\varepsilon)|R_i|$

MaNIS needs not be unique

MaNIS: Observations and variants

For $\delta \geq \varepsilon$, (ε, δ)-**MaNIS** is ***J* = <*S*$_1$, ... , *S*$_k$>** $\subseteq$ **SETS such that**

1 **Nearly Independent.**

$$|S_i \setminus (\cup_{j<i} S_j)| \geq (1-\delta)|S_i|$$

2 **Maximal.** Each $R_i \notin J$ *satisfies*

covered $= S_1 \cup S_2 \cup \cdots \cup S_k$

$|R_i \setminus \textbf{covered}| < (1-\varepsilon)|R_i|$

MaNIS needs not be unique

small overlaps *on average*

$$|\cup_i S_i| \geq (1-\delta) \sum_i |S_i|$$

MaNIS: Observations and variants

For $\delta \geq \varepsilon$, (ε, δ)**-MaNIS** is ***J* = <*S*$_1$, ... , *S*$_k$>** $\subseteq$ **SETS such that**

1 **Nearly Independent.**

$$|S_i \setminus (\cup_{j<i} S_j)| \geq (1-\delta)|S_i|$$

2 **Maximal.** Each $R_i \notin J$ *satisfies*

covered $= S_1 \cup S_2 \cup \cdots \cup S_k$

$|R_i \setminus$ **covered**$| < (1-\varepsilon)|R_i|$

MaNIS needs not be unique

$\delta = \varepsilon = 0$ is maximal set packing

small overlaps *on average*

$$|\cup_i S_i| \geq (1-\delta) \sum_i |S_i|$$

MaNIS: Observations and variants

For δ ≥ ε, (ε, δ)-**MaNIS** is ***J* = <*S*$_1$, … , *S*$_k$>** ⊆ **SETS such that**

1 **Nearly Independent.**

$$|S_i \setminus (\cup_{j<i} S_j)| \geq (1-\delta)|S_i|$$

2 **Maximal.** Each $R_i \notin J$ *satisfies*

covered = $S_1 \cup S_2 \cup \cdots \cup S_k$

$|R_i \setminus$ **covered**$| < (1-\varepsilon)|R_i|$

MaNIS needs not be unique

δ = ε = 0 is maximal set packing

small overlaps *on average*

$$|\cup_i S_i| \geq (1-\delta)\sum_i |S_i|$$

simple $O(|E|)$ sequential algo

How to Compute MaNIS?

Implicit in algorithms from previous work

For $\delta \geq \varepsilon$:

How to Compute MaNIS?

Implicit in algorithms from previous work

For $\delta \geq \varepsilon$:

Berger, Rompel, and Shor'94
(also Chierichetti, Kumar, and Tomkins'10)

RNC $O(|E| \log^3 |E|)$-work

(ε, δ)-MaNIS
on average

How to Compute MaNIS?

Implicit in algorithms from previous work

For $\delta \geq \varepsilon$:

Berger, Rompel, and Shor'94
(also Chierichetti, Kumar, and Tomkins'10)

RNC $O(|E| \log^3 |E|)$-work

(ε, δ)-MaNIS
on average
super-linear work

How to Compute MaNIS?

Implicit in algorithms from previous work

For $\delta \geq \varepsilon$:

Berger, Rompel, and Shor'94
(also Chierichetti, Kumar, and Tomkins'10)

(ε, δ)-MaNIS *on average*

RNC $O(|E| \log^3 |E|)$-work

super-linear work

Rajagopalan and Vazirani'98

(ε, δ)-MaNIS

RNC $O(|E| \log |E|)$-work

How to Compute MaNIS?

Implicit in algorithms from previous work

For $\delta \geq \varepsilon$:

Berger, Rompel, and Shor'94
(also Chierichetti, Kumar, and Tomkins'10)

RNC $O(|E| \log^3 |E|)$-work

(ε, δ)-MaNIS
on average
super-linear work

Rajagopalan and Vazirani'98

RNC $O(|E| \log |E|)$-work

(ε, δ)-MaNIS
$\delta > 1/2$

How to Compute MaNIS?

Implicit in algorithms from previous work

For $\delta \geq \varepsilon$:

Berger, Rompel, and Shor'94
(also Chierichetti, Kumar, and Tomkins'10)

(ε, δ)-MaNIS

on average

RNC $O(|E| \log^3 |E|)$-work

super-linear work

Rajagopalan and Vazirani'98

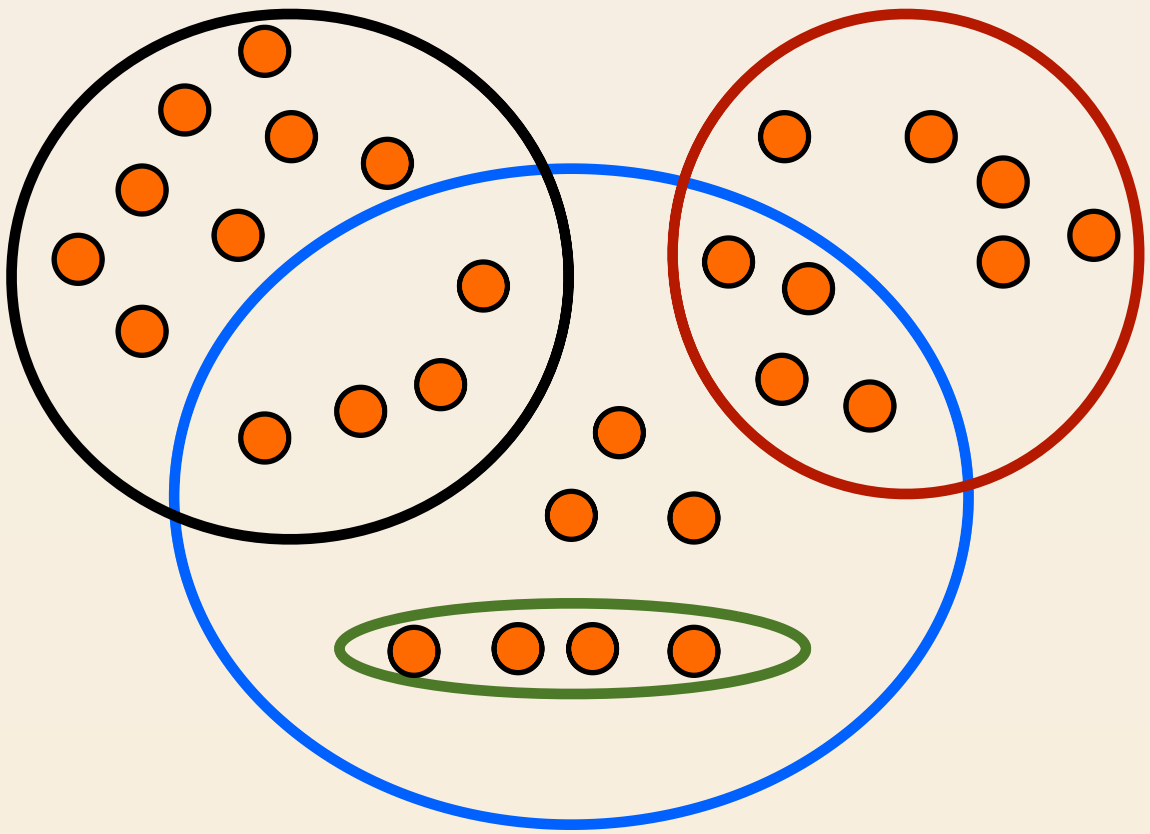
(ε, δ)-MaNIS

$\delta > 1/2$

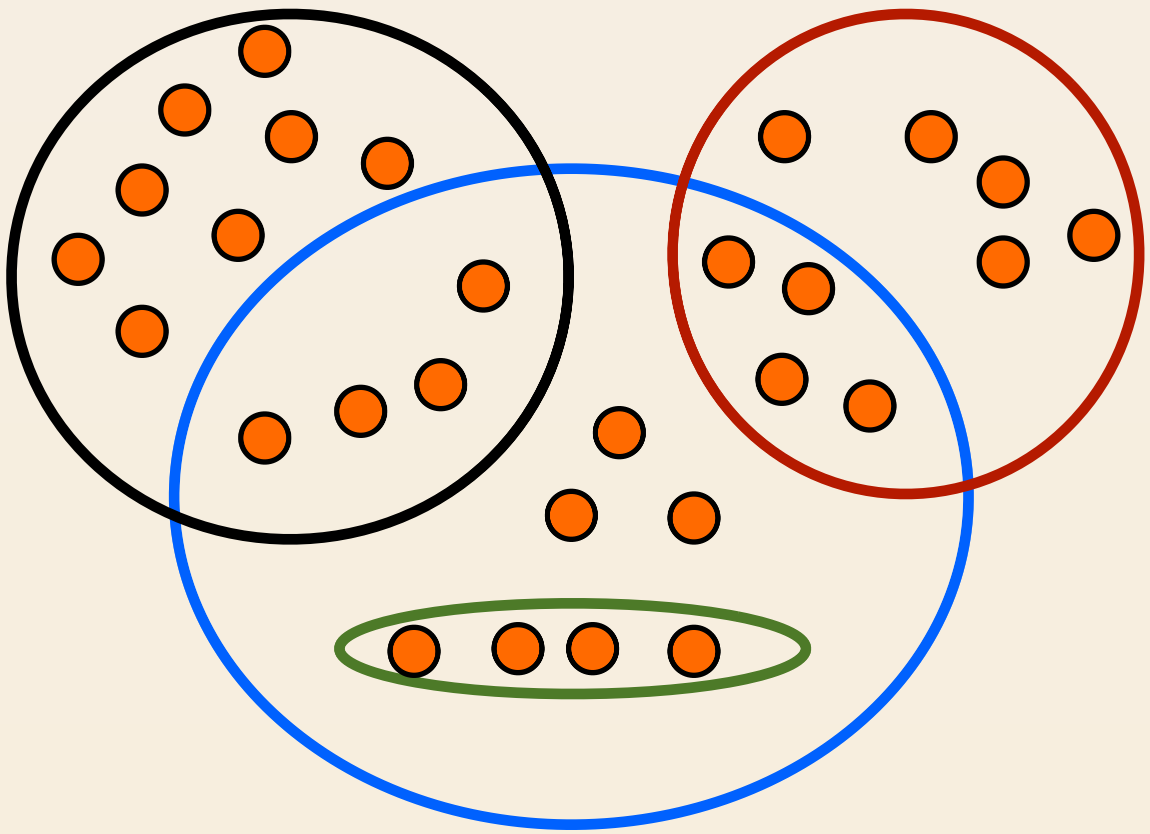
RNC $O(|E| \log |E|)$-work

This Work: simple RNC linear work (ε, δ)-MaNIS

A Simple Linear-Work MaNIS

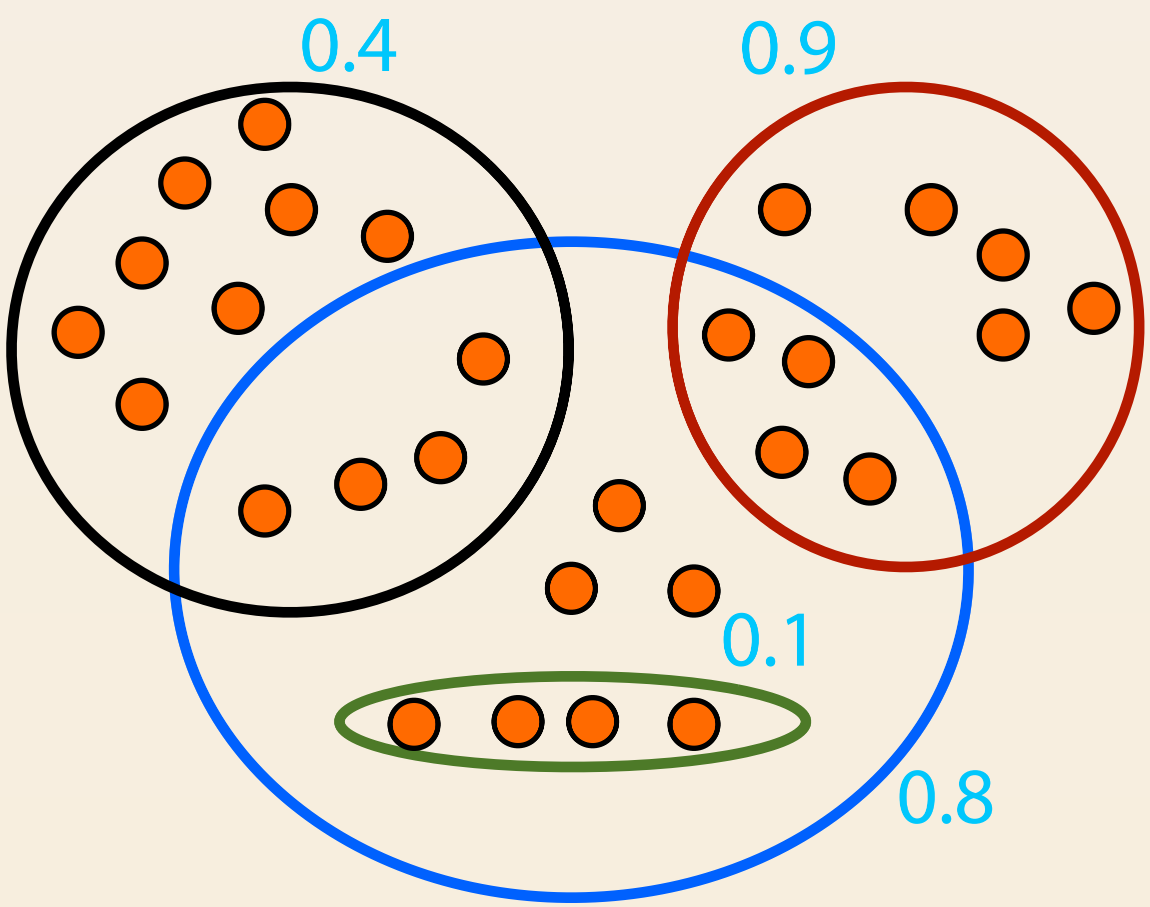


A Simple Linear-Work MaNIS



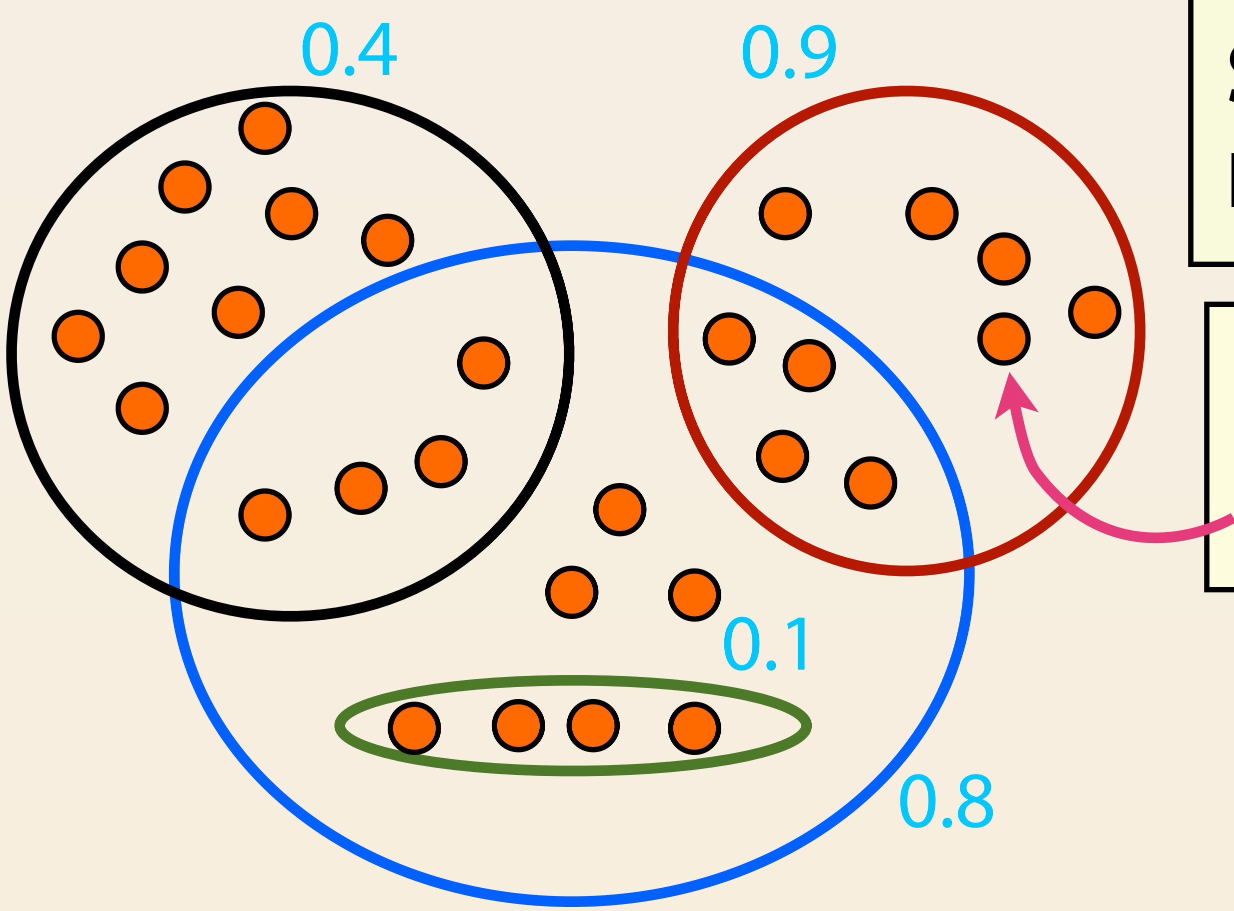
Step 1:
Each set picks a random val ∈ [0, 1]

A Simple Linear-Work MaNIS



Step 1:
Each set picks a random val ∈ [0, 1]

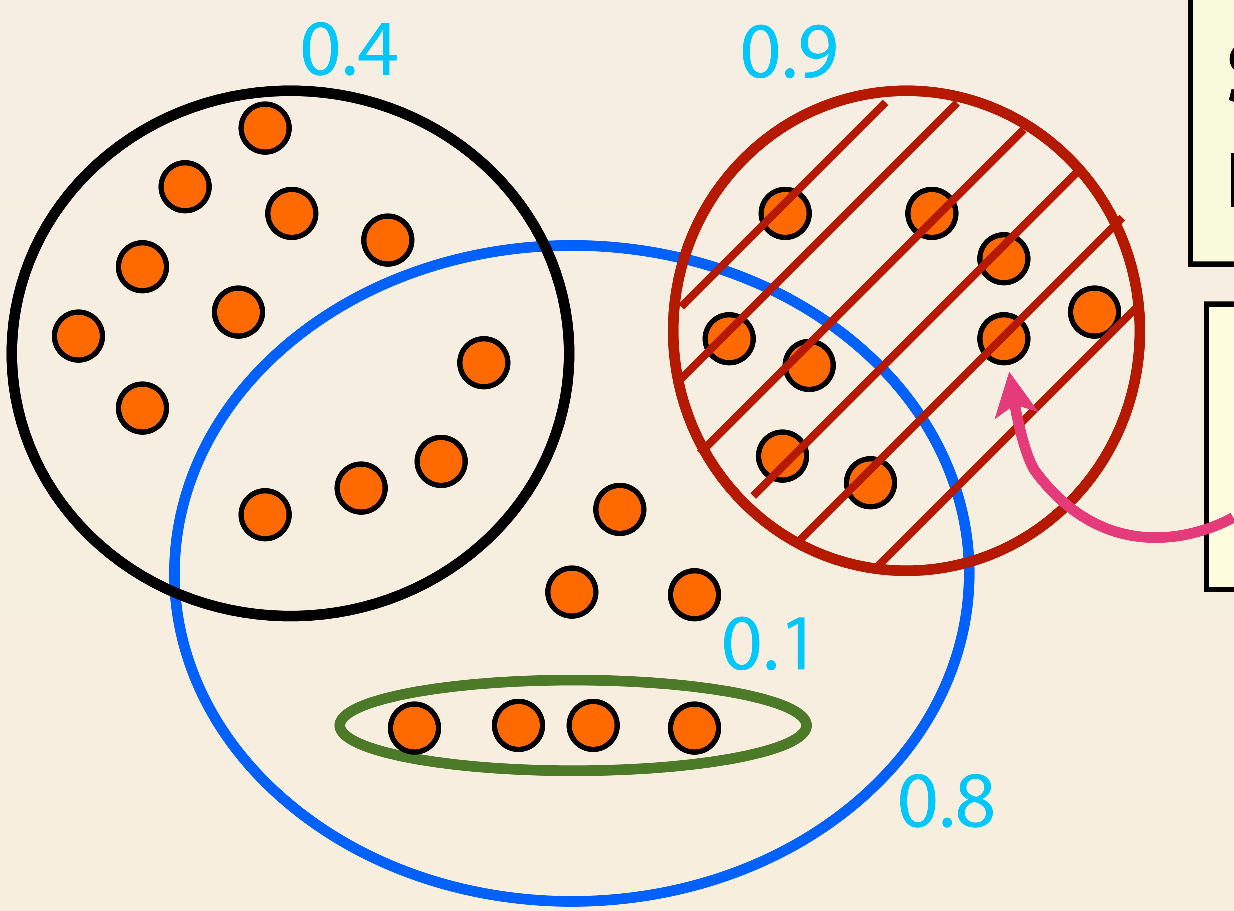
A Simple Linear-Work MaNIS



Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(S) : elt ∈ S }

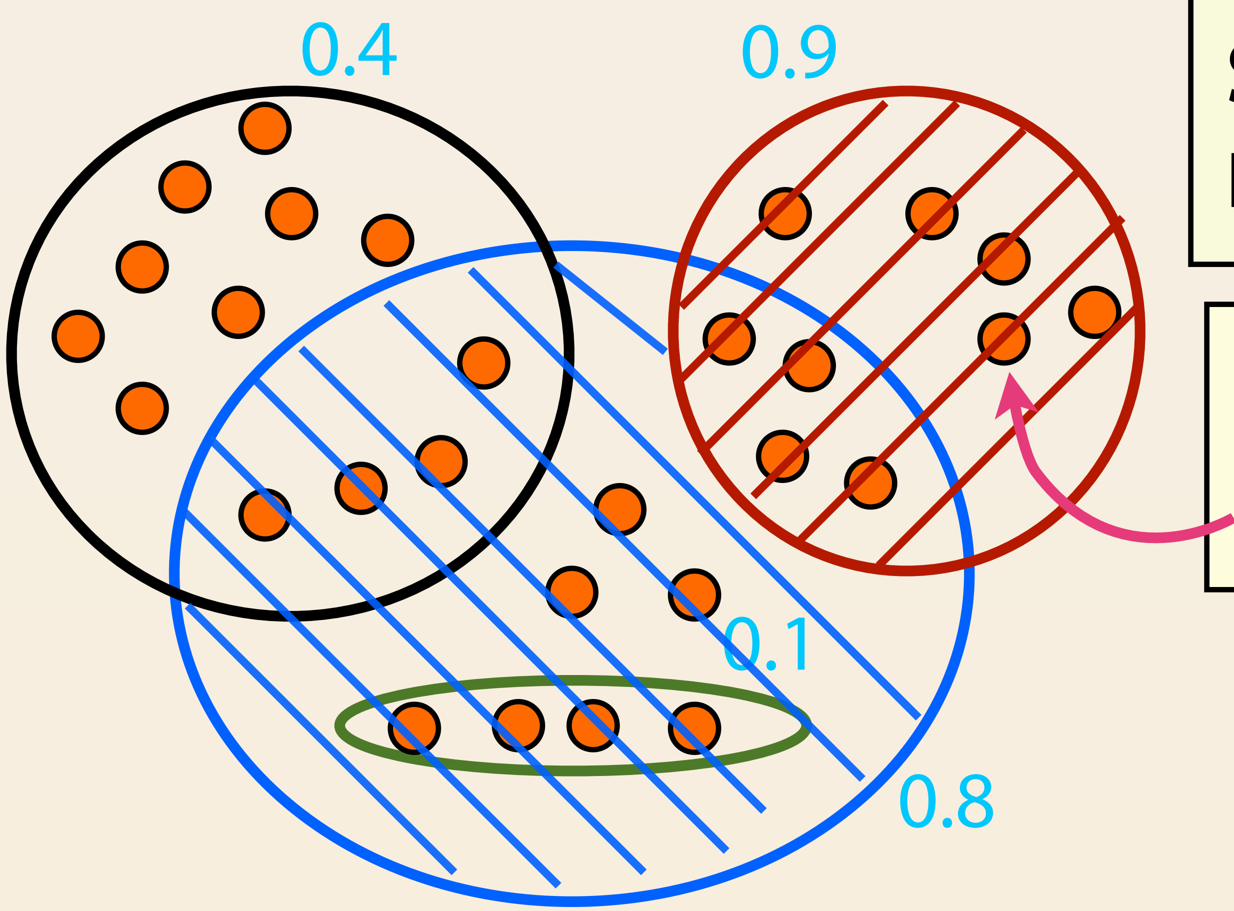
A Simple Linear-Work MaNIS



Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(S) : elt ∈ S }

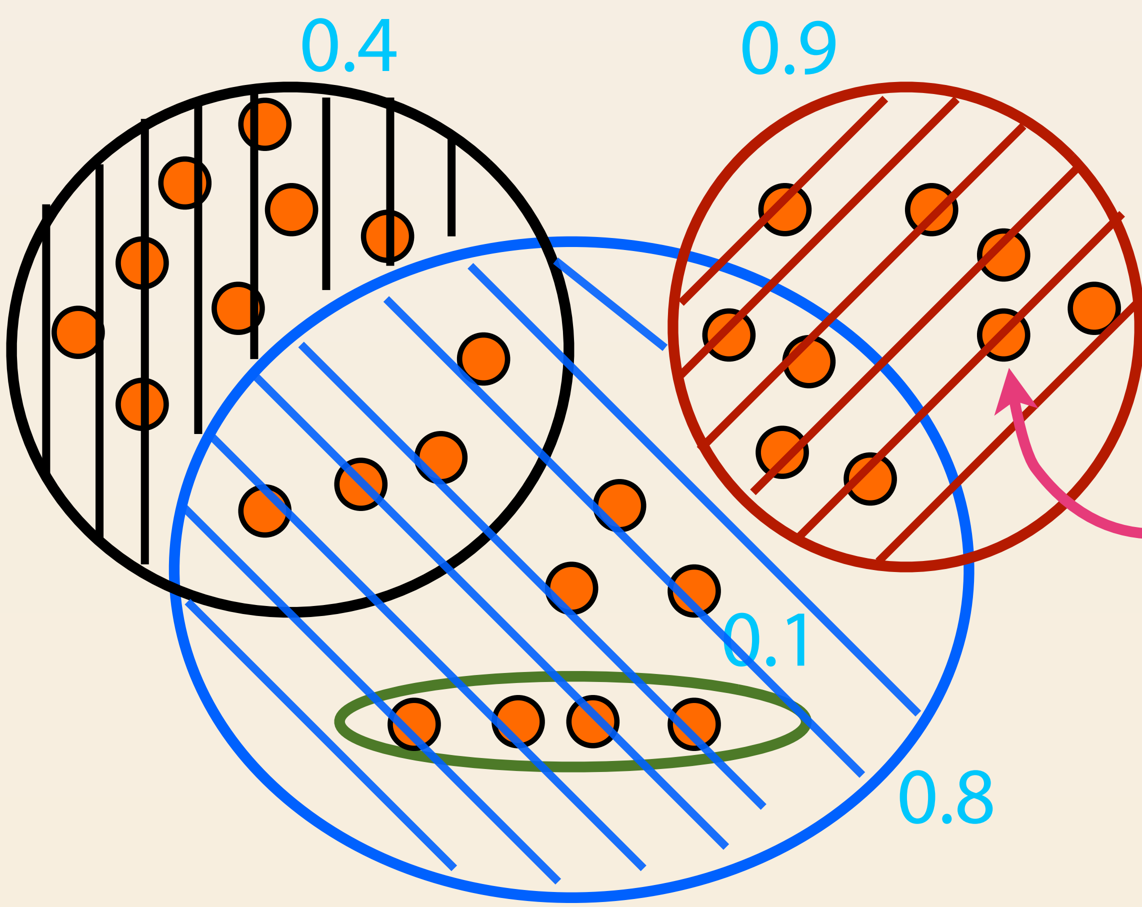
A Simple Linear-Work MaNIS



Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(*S*) : elt ∈ *S* }

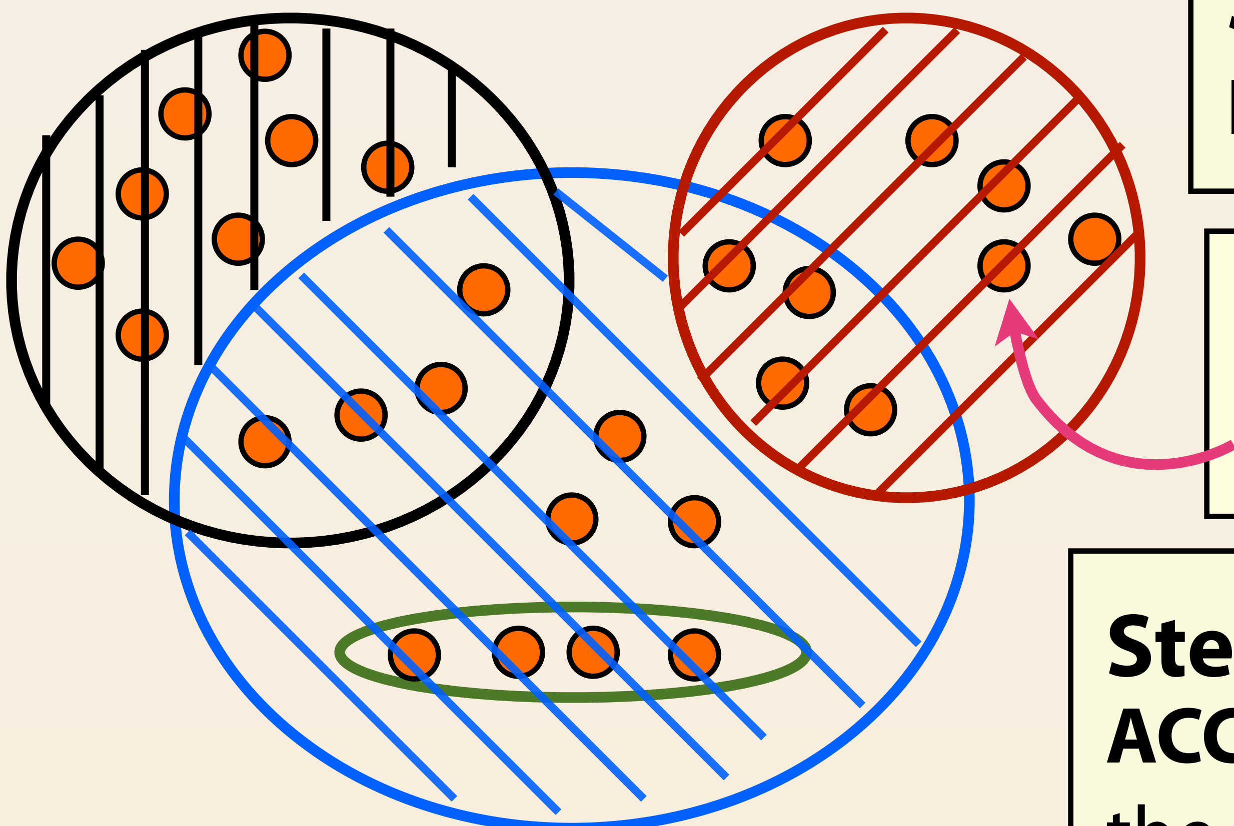
A Simple Linear-Work MaNIS



Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(S) : elt ∈ S }

A Simple Linear-Work MaNIS

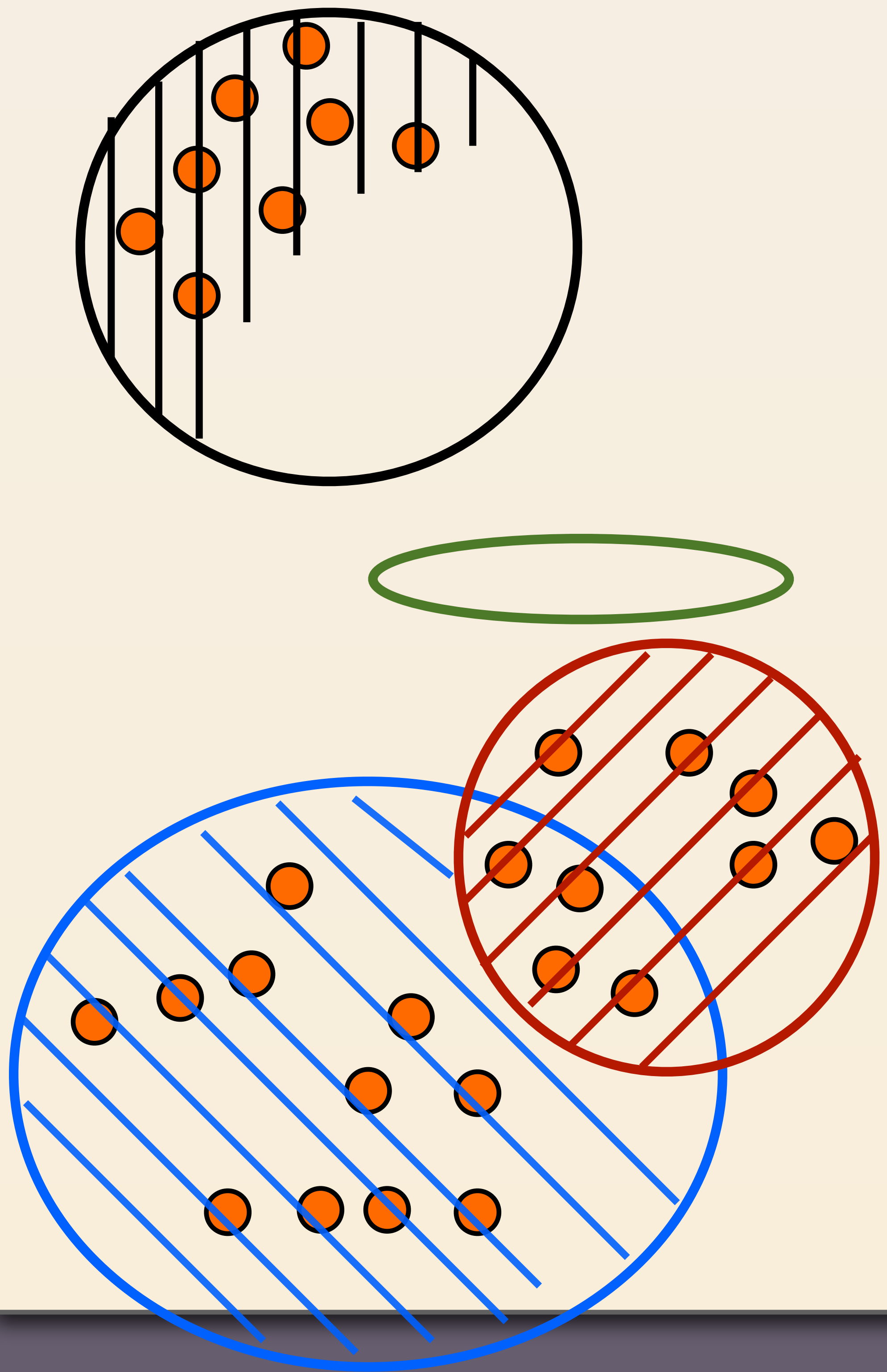


Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(*S*) : elt ∈ *S* }

Step 3:
ACCEPT *S* if > (1-δ) fraction of *S* has the same val as *S*.

A Simple Linear-Work MaNIS

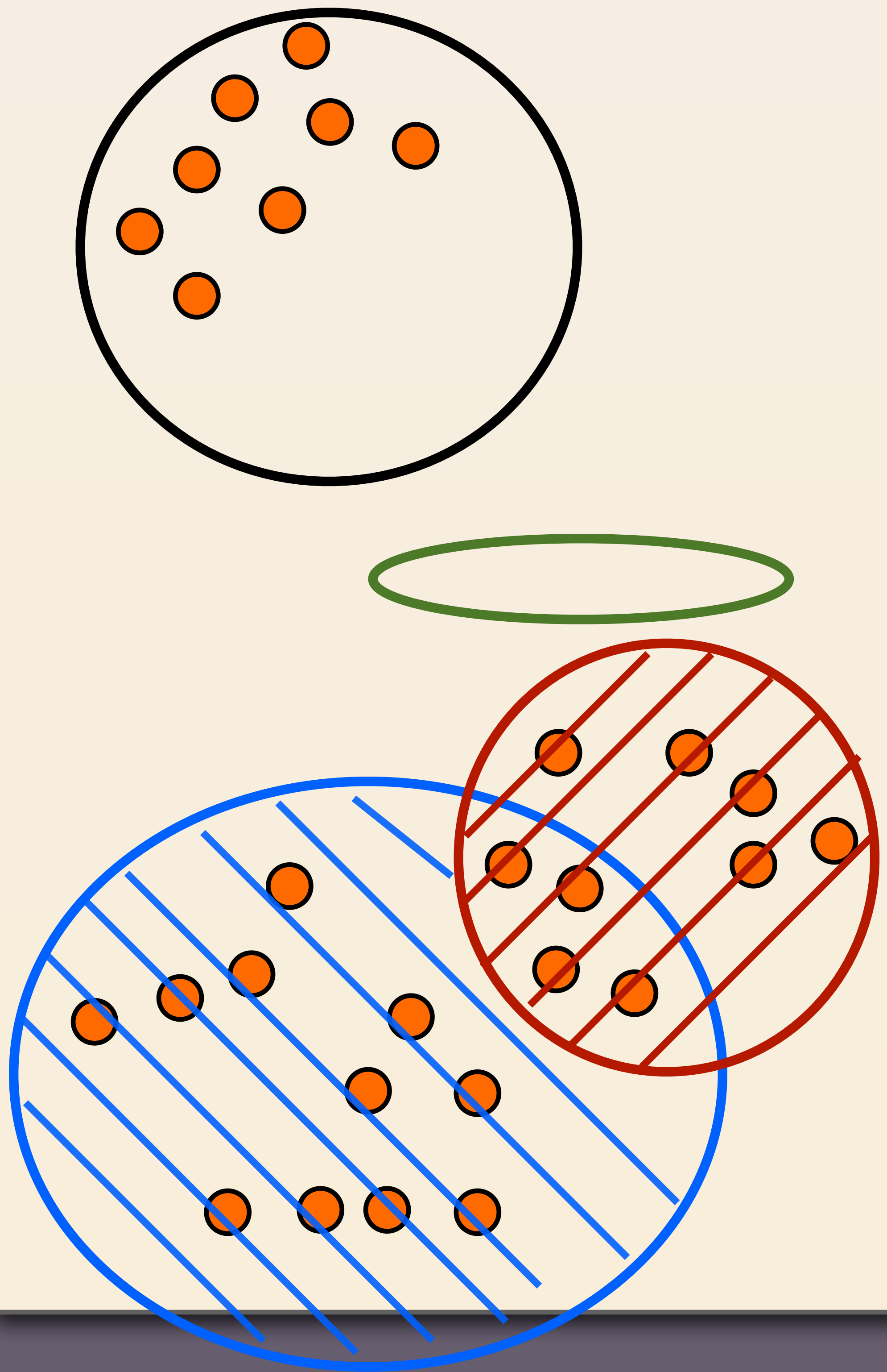


Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(*S*) : elt ∈ *S* }

Step 3:
ACCEPT *S* if > (1-δ) fraction of *S* has the same val as *S*.

A Simple Linear-Work MaNIS



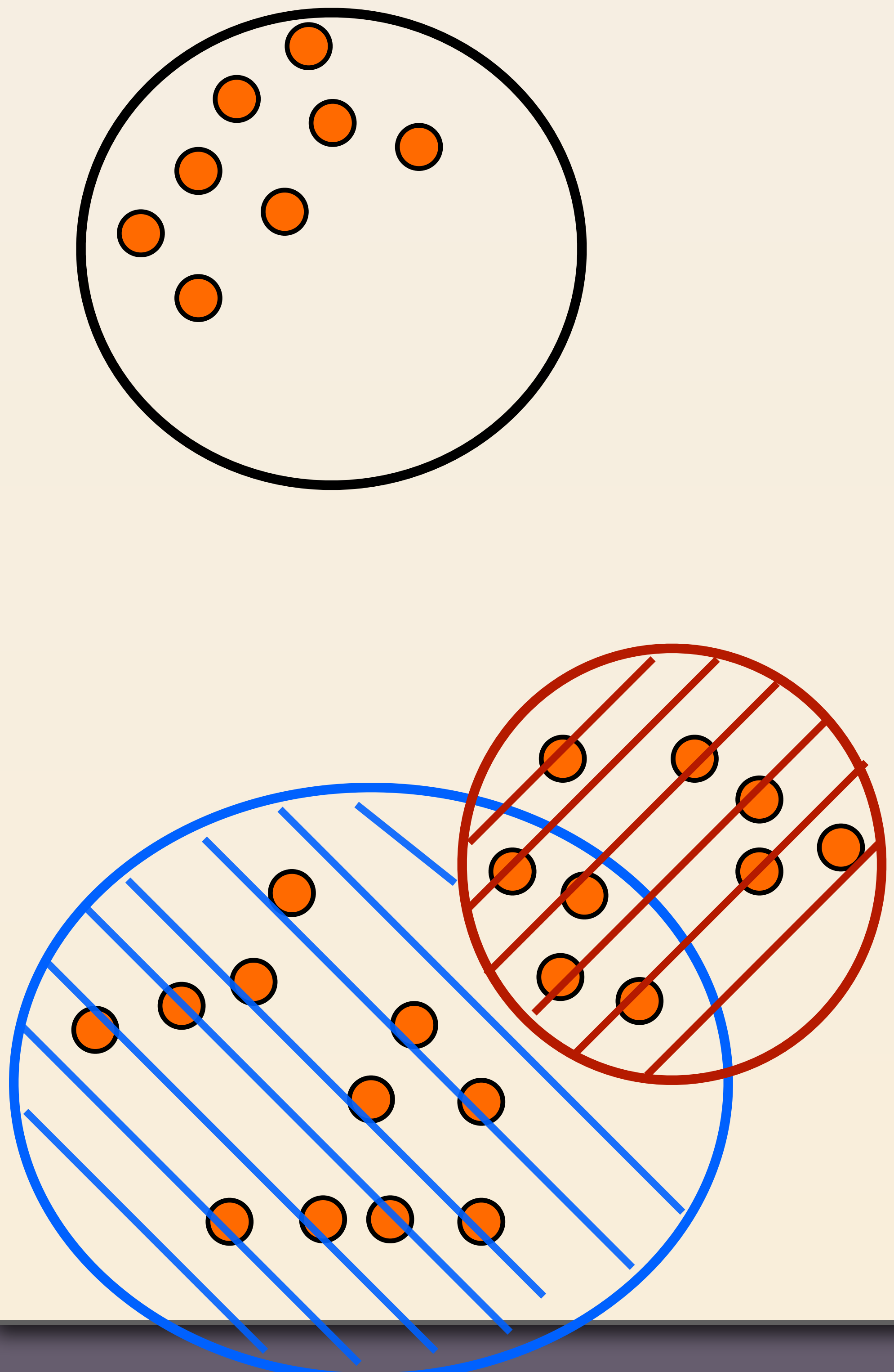
Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(*S*) : elt ∈ *S* }

Step 3:
ACCEPT *S* if > (1-δ) fraction of *S* has the same val as *S*.

Step 4:
REJECT *S* if < (1-ε) frac of *S* remains

A Simple Linear-Work MaNIS



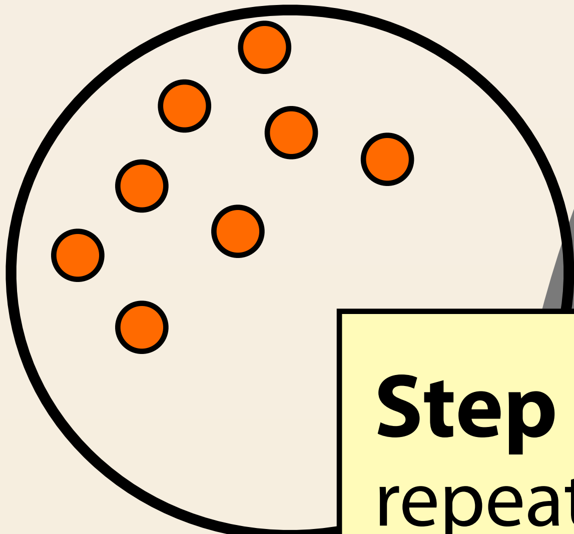
Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(*S*) : elt ∈ *S* }

Step 3:
ACCEPT *S* if > (1-δ) fraction of *S* has the same val as *S*.

Step 4:
REJECT *S* if < (1-ε) frac of *S* remains

A Simple Linear-Work MaNIS



Step 1:
Each set picks a random val ∈ [0, 1]

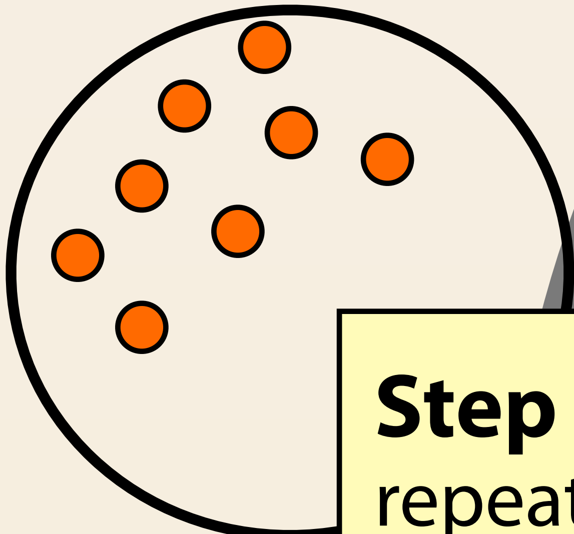
Step 2:
val(elt) = max{ val(*S*) : elt ∈ *S* }

Step 3:
ACCEPT *S* if > (1-δ) fraction of *S* has the same val as *S*.

Step 4:
REJECT *S* if < (1-ε) frac of *S* remains

Step 5:
repeat until empty

A Simple Linear-Work MaNIS



Step 1:
Each set picks a random val ∈ [0, 1]

Step 2:
val(elt) = max{ val(*S*) : elt ∈ *S* }

Step 3:
ACCEPT *S* if > (1-δ) fraction of *S* has the same val as *S*.

Step 4:
REJECT *S* if < (1-ε) frac of *S* remains

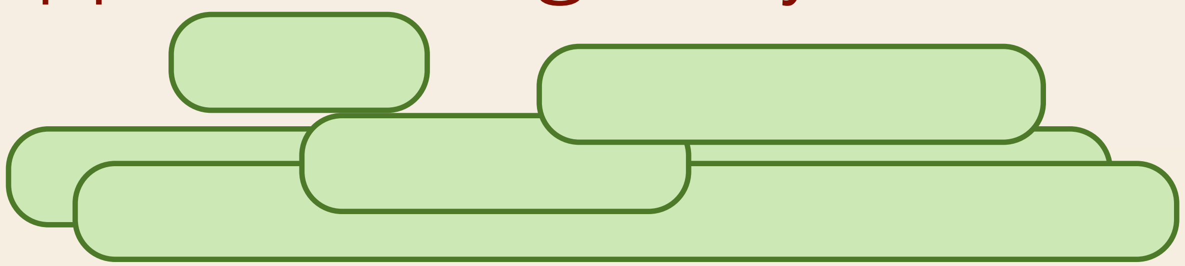
Step 5:
repeat until empty

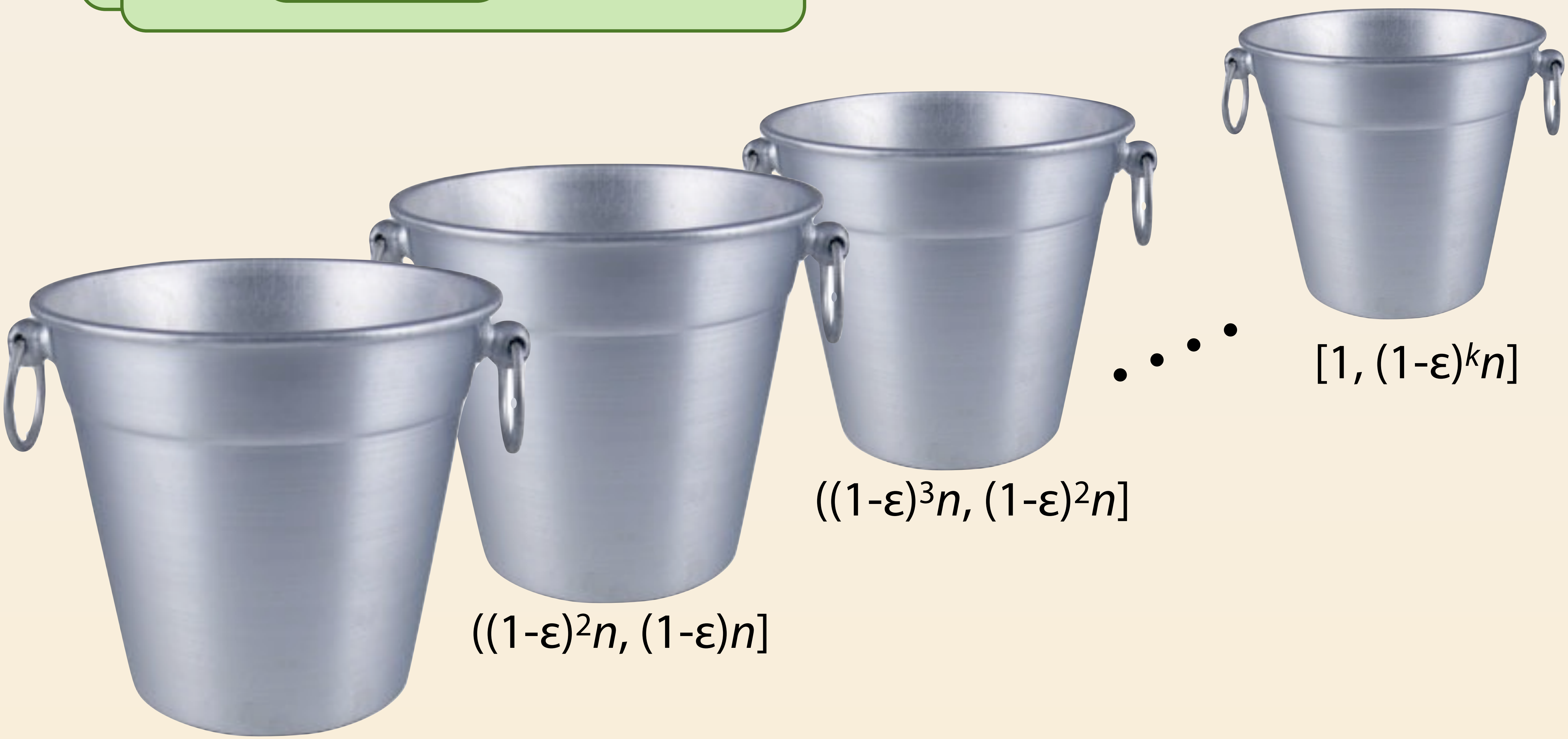
Theorem:

$O(|E|)$-work, $O(\log^2 |E|)$-depth algorithm for (ε, δ)-MaNIS

Parallel Greedy Set Cover

approximate greedy via bucketing





$((1-\varepsilon)n, n]$

$((1-\varepsilon)^2 n, (1-\varepsilon)n]$

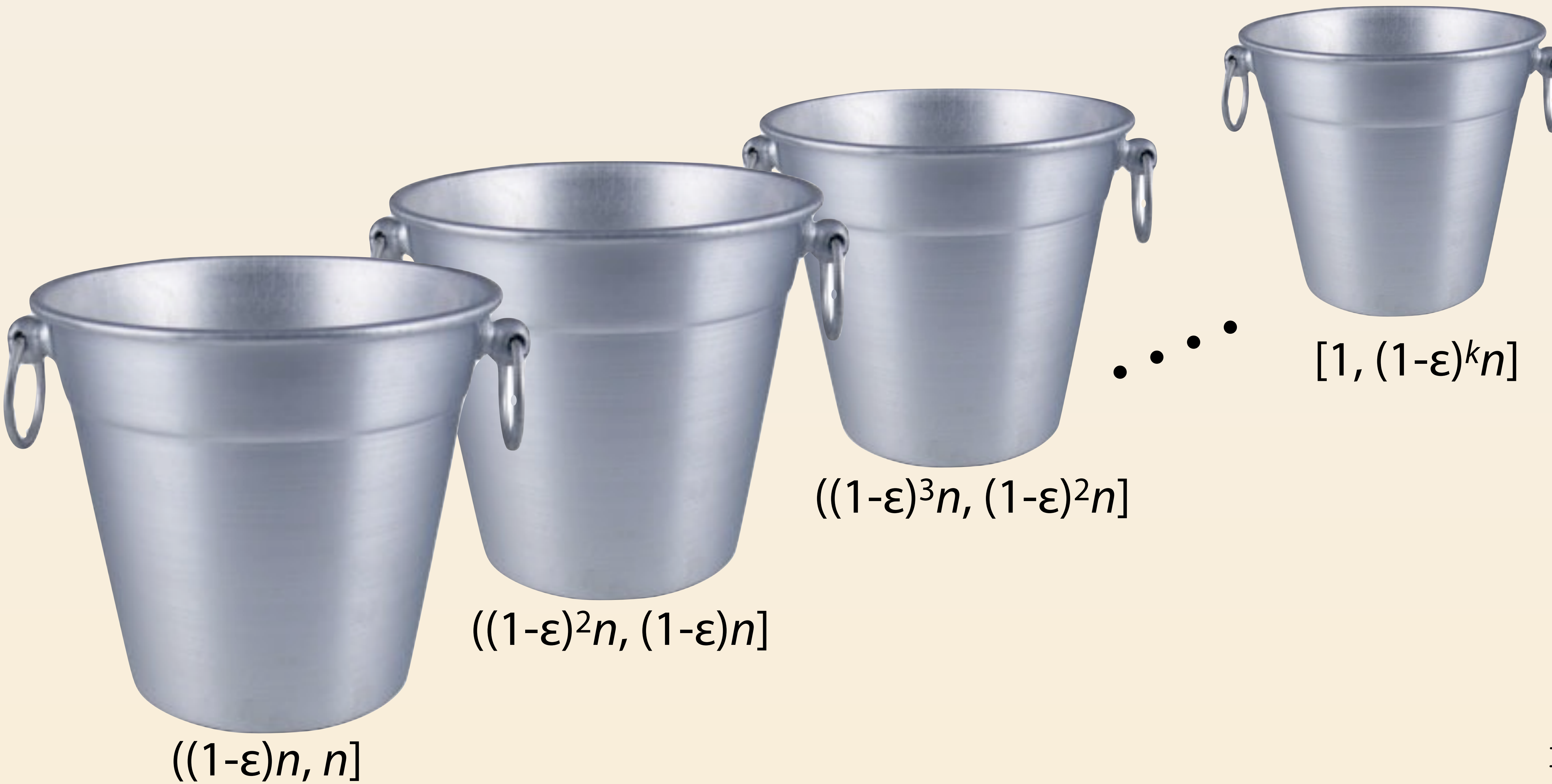
$((1-\varepsilon)^3 n, (1-\varepsilon)^2 n]$

. . . .

$[1, (1-\varepsilon)^k n]$

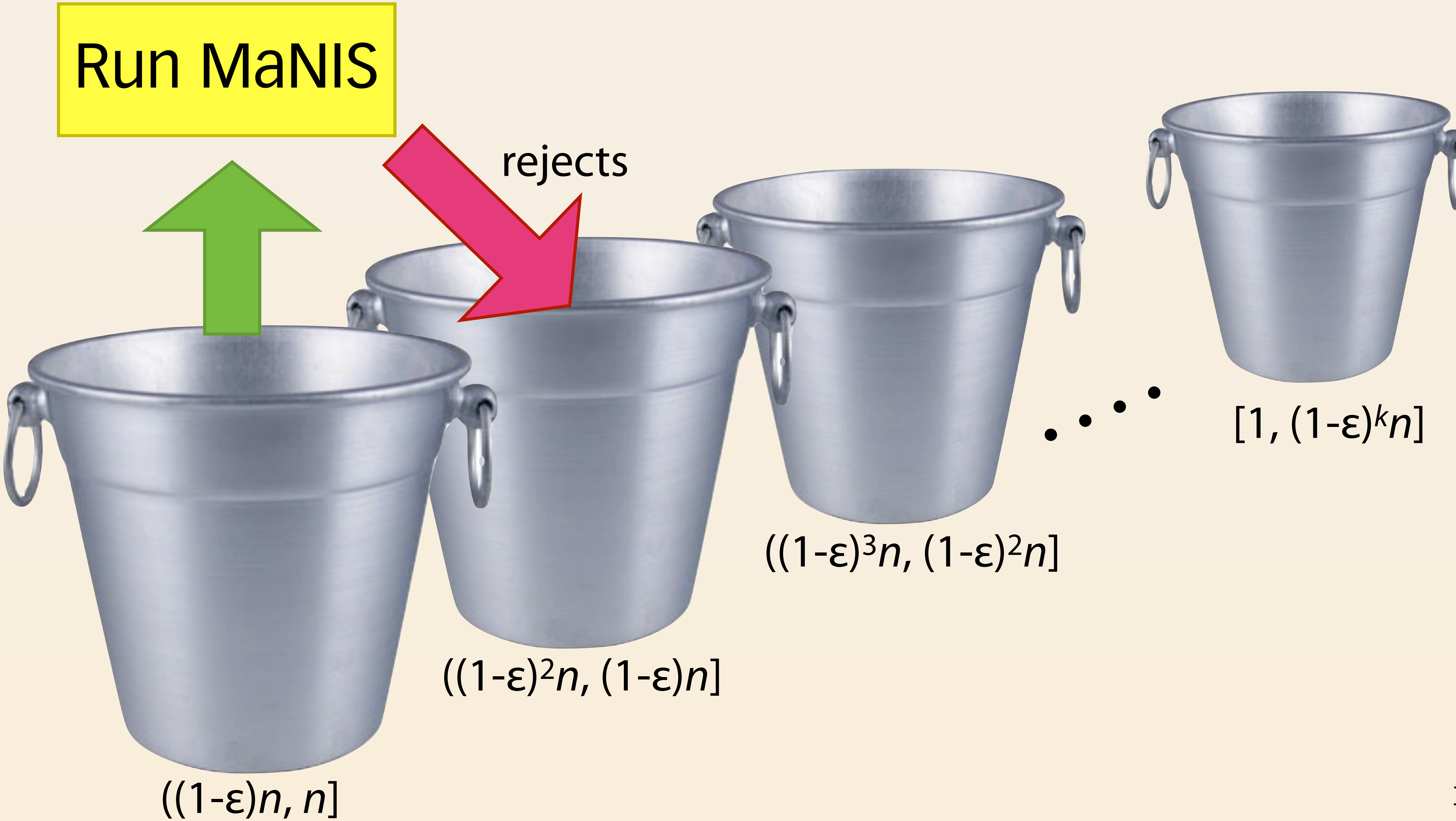
Parallel Greedy Set Cover

approximate greedy via bucketing



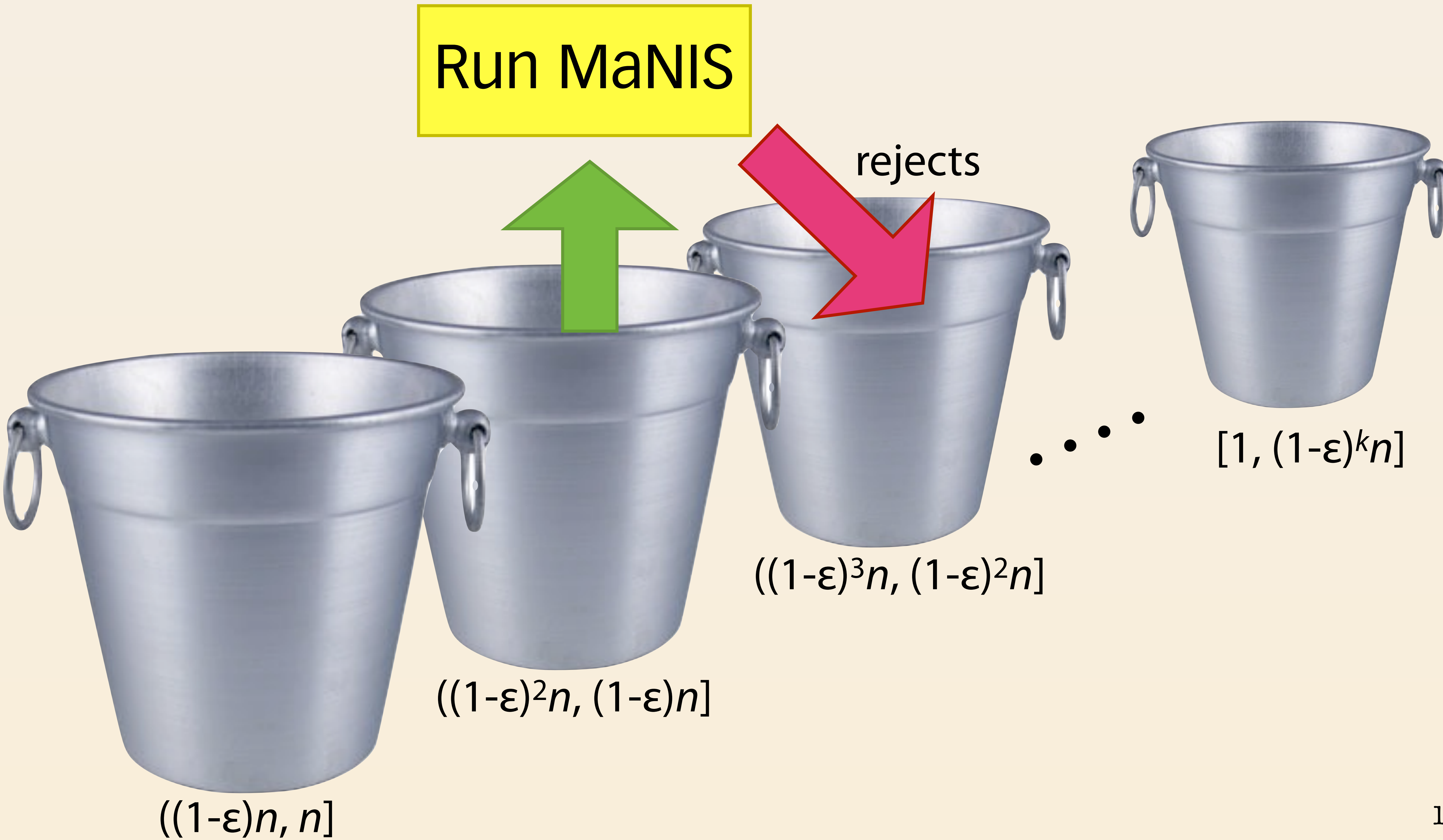
Parallel Greedy Set Cover

approximate greedy via bucketing



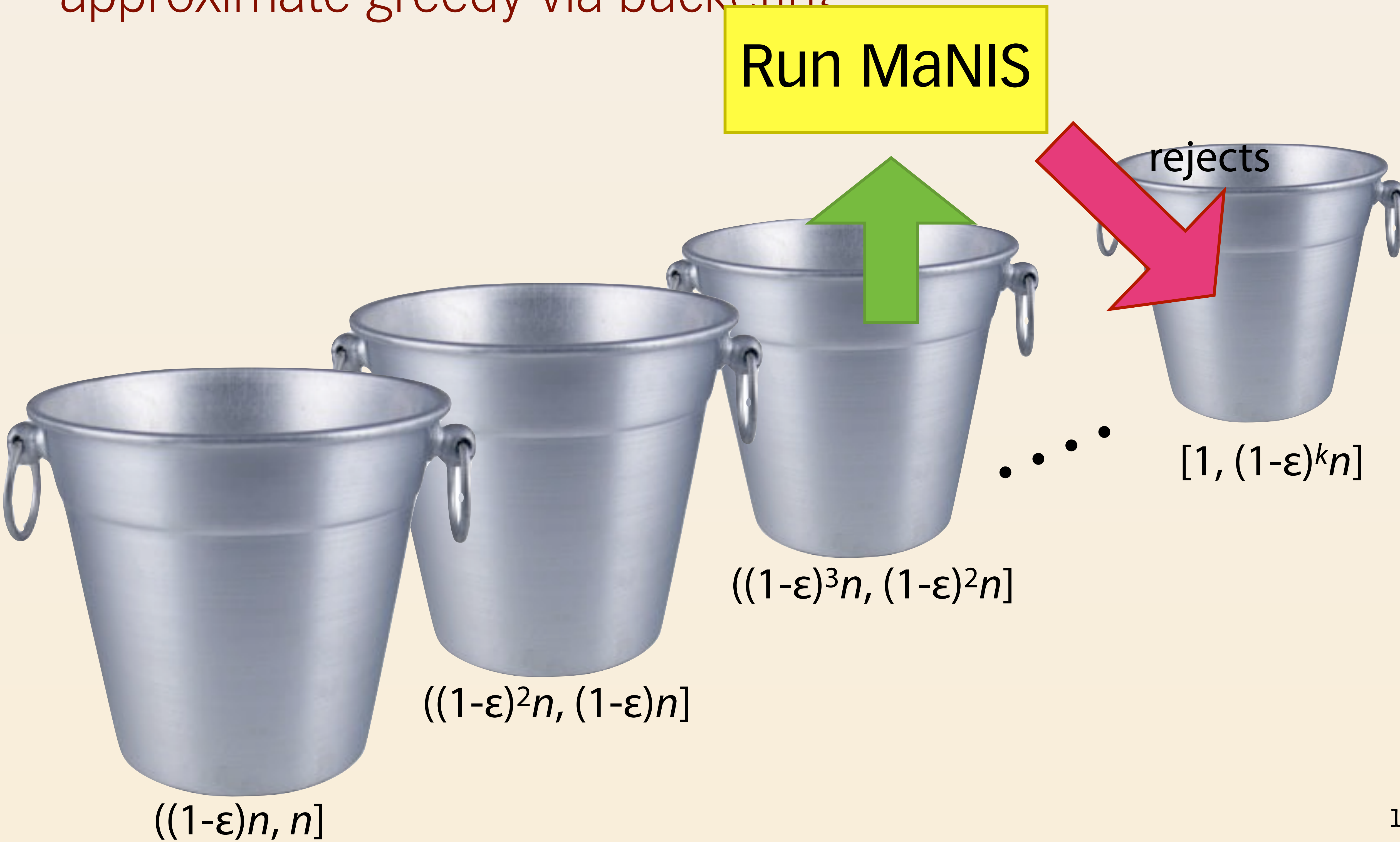
Parallel Greedy Set Cover

approximate greedy via bucketing



Parallel Greedy Set Cover

approximate greedy via bucketing



Parallel Greedy Set Cover
(cont'd)

Initial bucketing: linear work

Iterating through buckets:

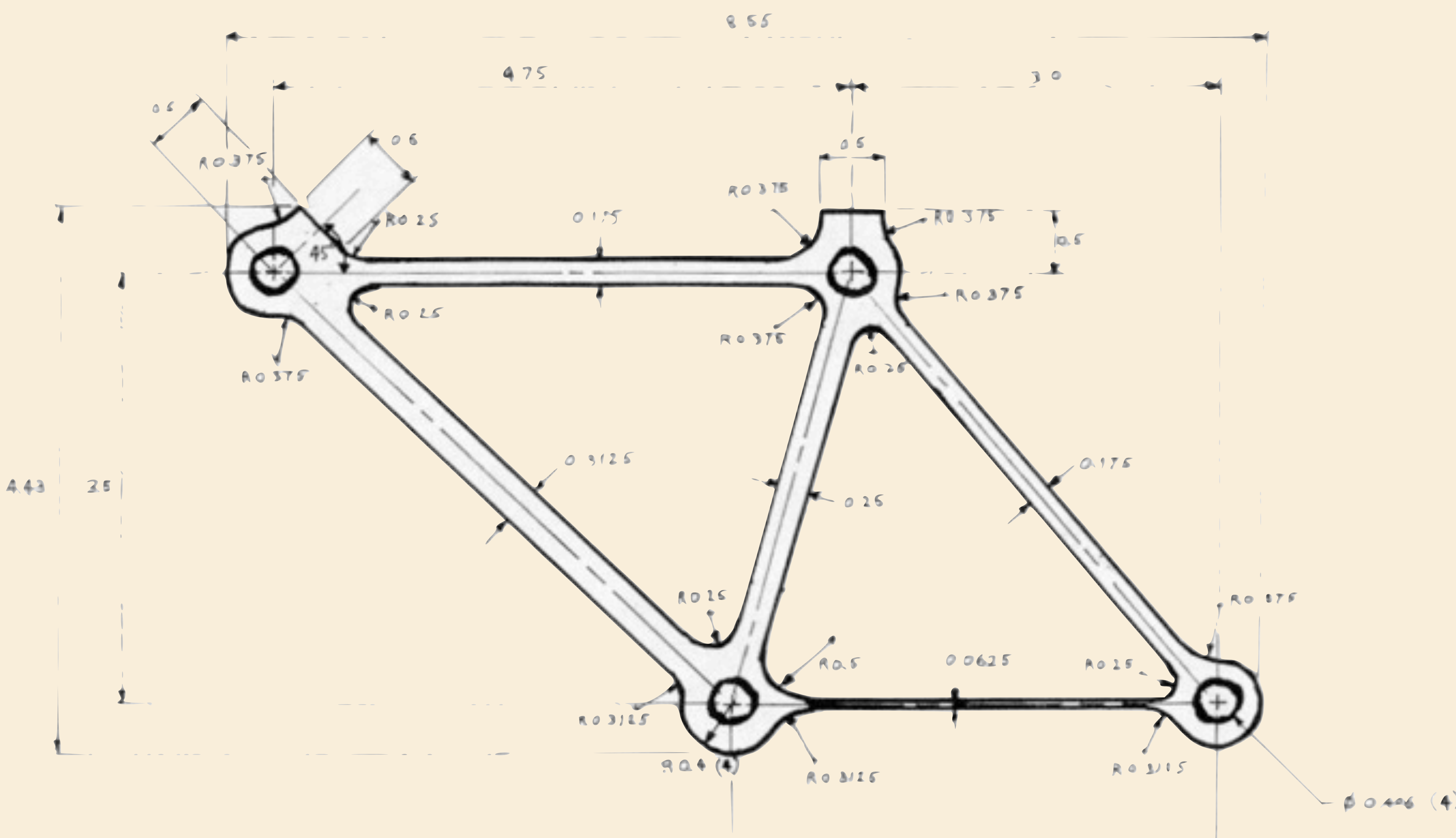
MaNIS is $O(m) = O(\text{sum of set sizes in that bucket})$

When a set changes buckets, size shrinks by ε

Total work: Linear in sum of set sizes

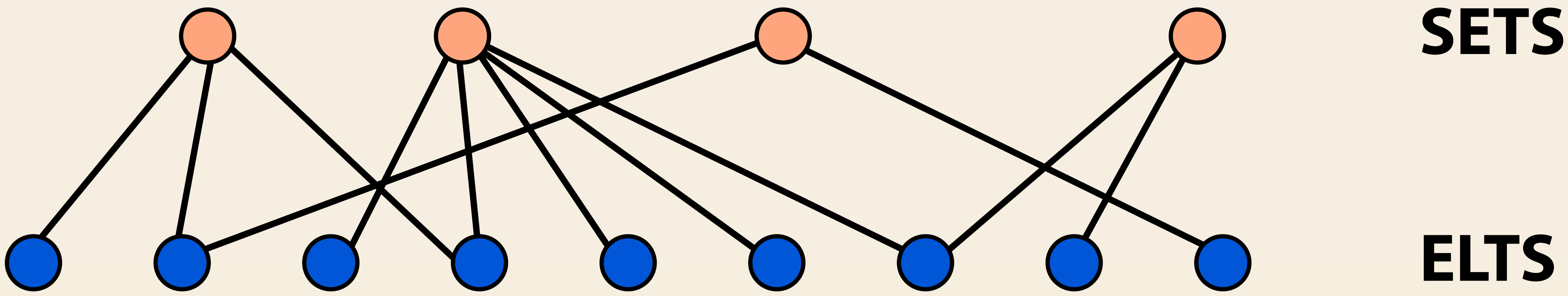
Now for a proof sketch….

$O(|E|)$-work, $O(\log^2 |E|)$-depth



Proof's Overview

a bipartite graph view:



Key Lemma:

Each iteration of MaNIS removes a constant fraction of the edges.

Key Lemma:

Each iteration of MaNIS removes a constant fraction of the edges.

For each set *S*, W_S = *"# of edges S responsible for deleting"*

Key Lemma:

Each iteration of MaNIS removes a constant fraction of the edges.

For each set *S*, *W_S = "# of edges S responsible for deleting"*

if S not accepted: $W_S = 0$

Key Lemma:

Each iteration of MaNIS removes a constant fraction of the edges.

For each set *S*, W_S = *"# of edges S responsible for deleting"*

if S not accepted: $W_S = 0$

if S accepted:

W_S = *sum of degrees of S's elts with same val as S*

Key Lemma:

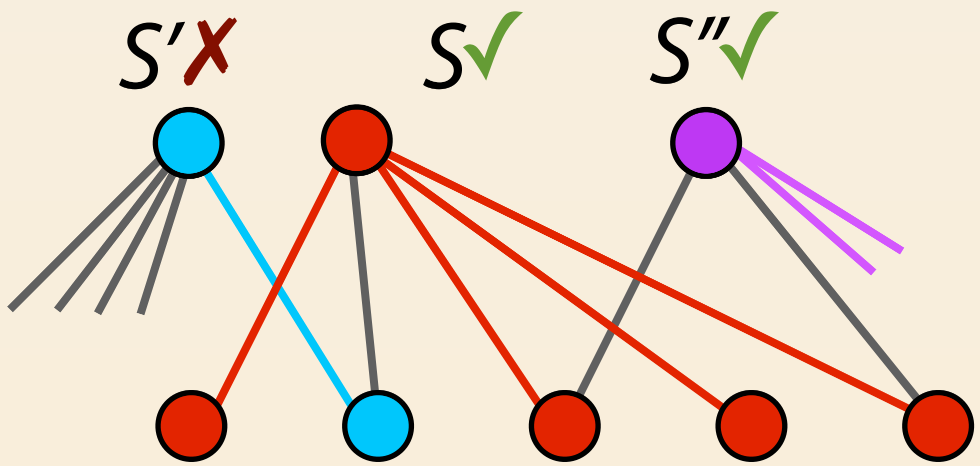
Each iteration of MaNIS removes a constant fraction of the edges.

For each set *S*, W_S = *"# of edges S responsible for deleting"*

if S not accepted: $W_S = 0$

if S accepted:

W_S = *sum of degrees of S's elts with same val as S*



$W_{S'} = 0$

$W_S = 1 + 0 + 2 + 1 + 2$

Key Lemma:

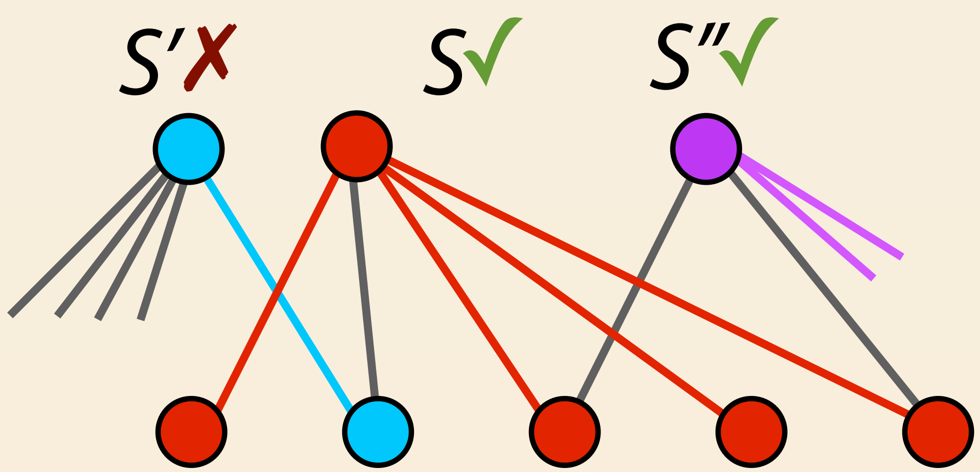
Each iteration of MaNIS removes a constant fraction of the edges.

Observation:

edges removed ≥ $\sum_{S \in \textbf{SETS}} W_S$

if S accepted:

W_S = sum of degrees of S's elts with same val as S



$W_{S'} = 0$

$W_S = 1 + 0 + 2 + 1 + 2$

Key Lemma:

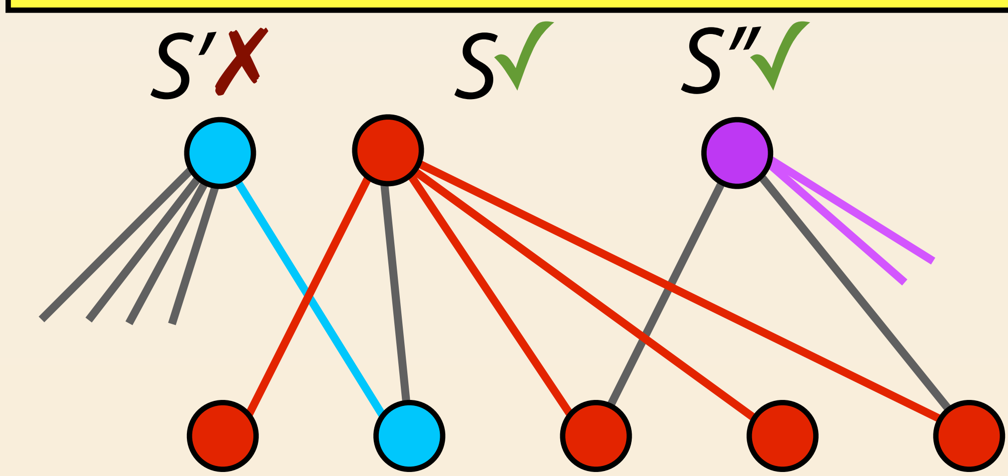
Each iteration of MaNIS removes a constant fraction of the edges.

Observation:

edges removed ≥ $\sum_{S \in \textbf{SETS}} W_S$

Technical Claim:

$$\mathbb{E}[W_S] \geq c \cdot \deg(S)$$

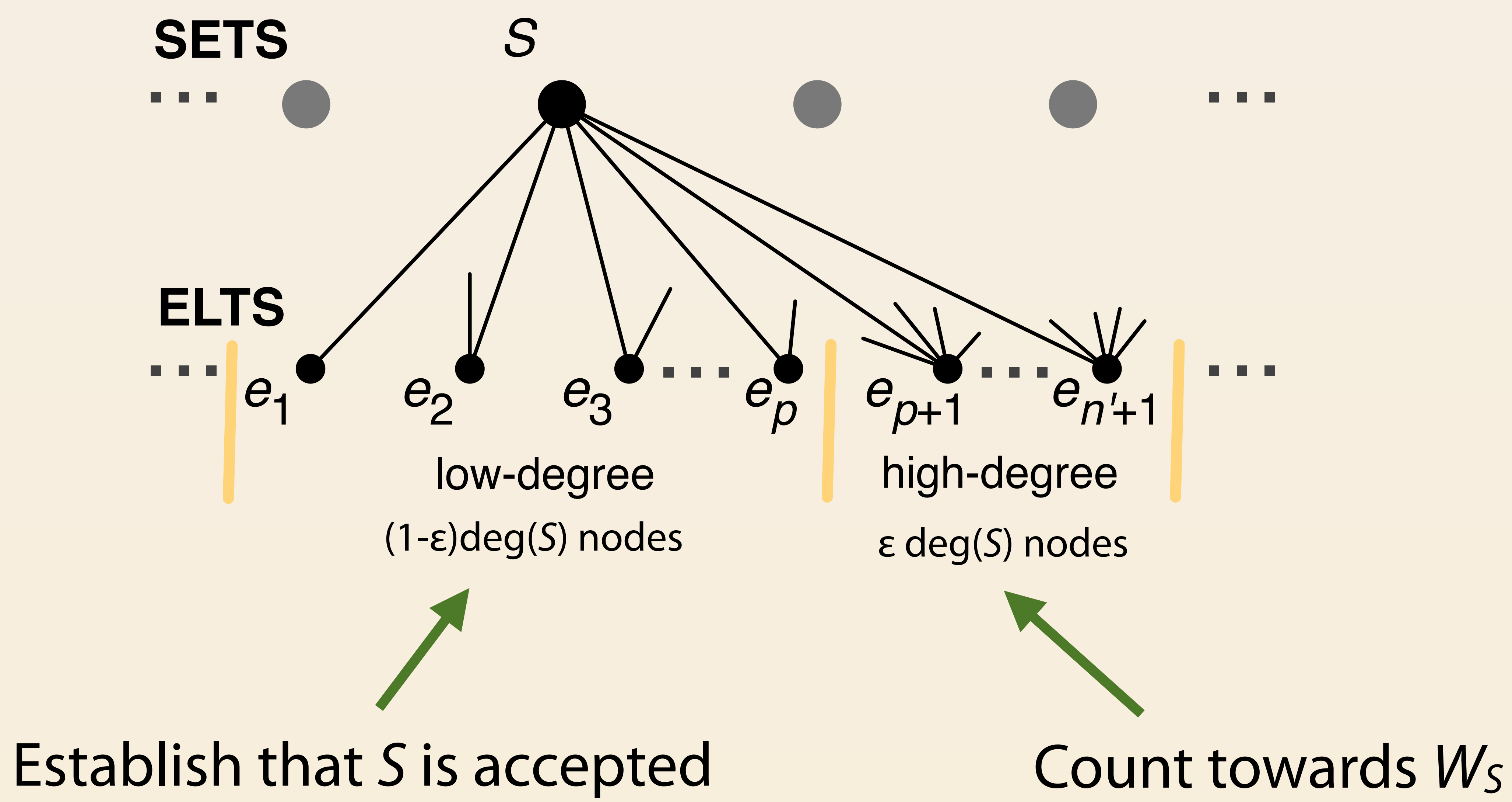


$W_{S'} = 0$

$W_S = 1 + 0 + 2 + 1 + 2$

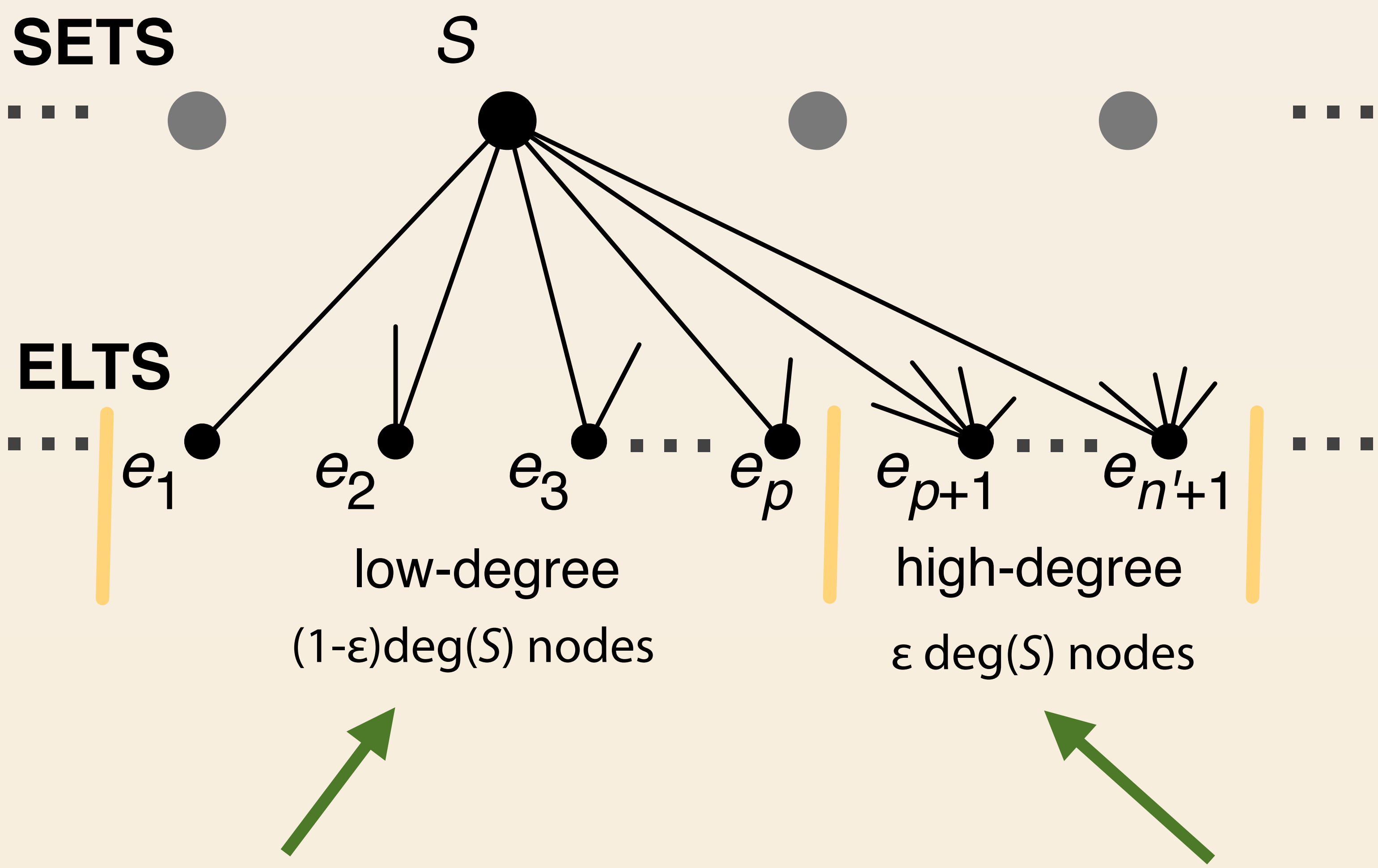
Technical Claim:

$$\mathbb{E}[W_S] \geq c \cdot \deg(S)$$



Technical Claim:

$$\mathbb{E}[W_S] \geq c \cdot \deg(S)$$



Establish that S is accepted

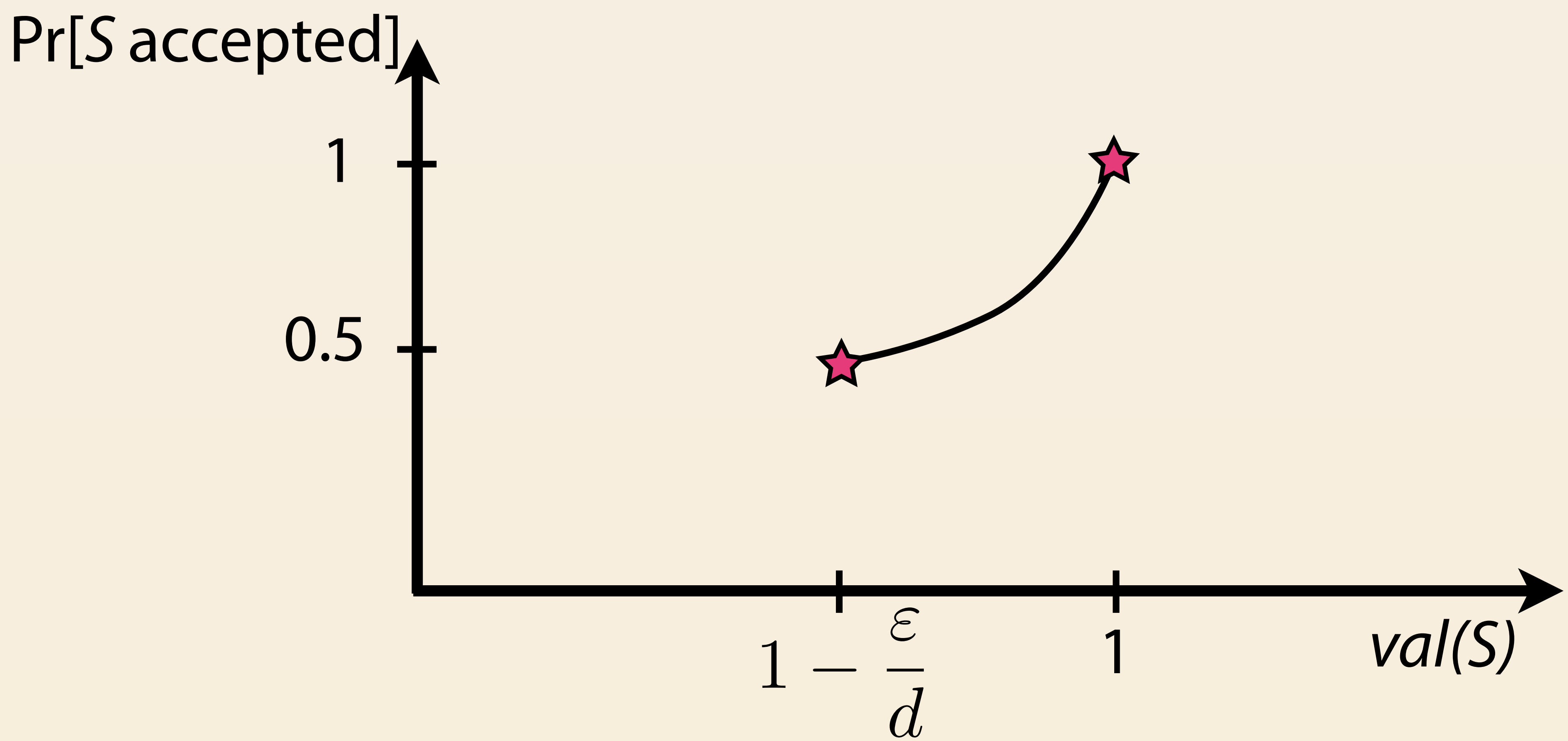
Count towards W_S

Suppose every node in the low degree group has degree $\leq d$, and every node in the high degree group has degree $> d$.

Technical Claim:

$$\mathbb{E}[W_S] \geq c \cdot \deg(S)$$

Use low-degree neighbors of *S* to establish that *S* is picked



Claim: If *val(S)* ≥ 1 - ε/*d*, then Pr[*S* accepted] ≥ 1/2

Technical Claim:

$$\mathbb{E}[W_S] \geq c \cdot \deg(S)$$

Claim: If *val(S)* ≥ 1 - ε/*d*, then Pr[*S* accepted] ≥ 1/2

Use high-degree neighbors to account for W_S

Claim: For a high-degree node *e*, if *val(S)* ≥ 1 - ε/deg(*e*) and *S* is accepted, then Pr[*e has the same value as S*] ≥ 1 - ε

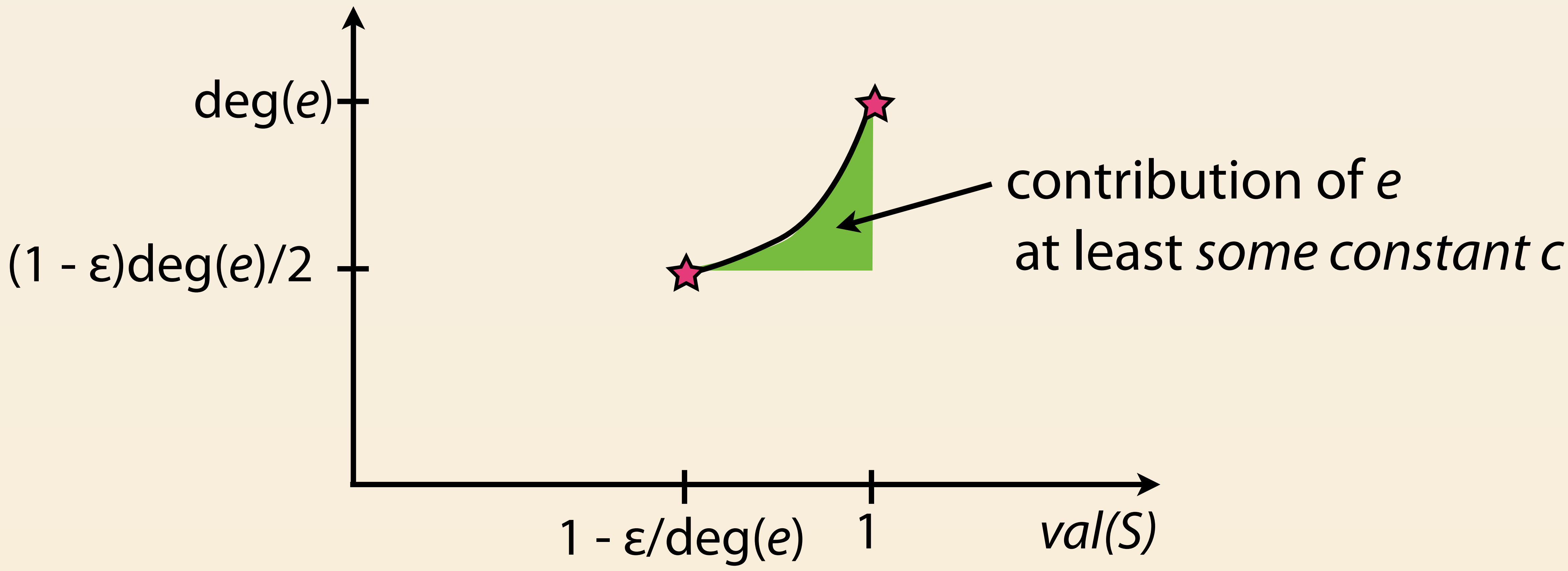
Technical Claim:

$$\mathbb{E}[W_S] \geq c \cdot \deg(S)$$

Claim: If *val(S)* ≥ 1 - ε/*d*, then Pr[*S* accepted] ≥ 1/2

Use high-degree neighbors to account for W_S

Claim: For a high-degree node *e*, if *val(S)* ≥ 1 - ε/deg(*e*) and *S* is accepted, then Pr[*e has the same value as S*] ≥ 1 - ε



RNC Algorithms via MaNIS

RNC Algorithms via MaNIS

Min Set Cover

Max Cover

Min-Sum Set Cover

Facility Location

Asymmetric *k*-Center

RNC Algorithms via MaNIS

Min Set Cover

Max Cover

Min-Sum Set Cover

Facility Location

Asymmetric *k*-Center

RNC Algorithms via MaNIS

Min Set Cover
(1+ε)(1+ln *n*)-approx, linear work

Max Cover
(1 - 1/*e* - ε)-approx, linear work

Min-Sum Set Cover
(4+ε)-approx, linear work

Facility Location
(1.861+ε)-approx, O(*n*log *n*)-work

Asymmetric *k*-Center
O(log* *n*)-approx, work-efficient

RNC Algorithms via MaNIS

Min Set Cover
(1+ε)(1+ln *n*)-approx, linear work

Max Cover
(1 - 1/*e* - ε)-approx, linear work

Min-Sum Set Cover
(4+ε)-approx, linear work

Facility Location
(1.861+ε)-approx, O(*n*log *n*)-work

Asymmetric *k*-Center
O(log* *n*)-approx, work-efficient

Common Tool: (ε,δ)-MaNIS in $O(|E|)$-work, $O(\log^2 |E|)$-depth

RNC Algorithms via MaNIS

Min Set Cover
(1+ε)(1+ln *n*)-approx, linear work

Max Cover
(1 - 1/*e* - ε)-approx, linear work

Min-Sum Set Cover
(4+ε)-approx, linear work

Facility Location
(1.861+ε)-approx, O(*n*log *n*)-work

Asymmetric *k*-Center
O(log* *n*)-approx, work-efficient

Thank you!

Common Tool: (ε,δ)-MaNIS in $O(|E|)$-work, $O(\log^2 |E|)$-depth