Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

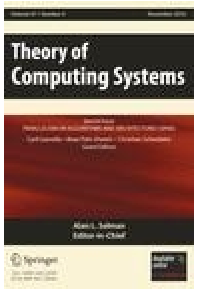
Tell Me Who I Am: An Interactive Recommendation System

N. Alon, B. Awerbuch, Y. Azar, B. Patt-Shamir

Richard Huber

Publication

- Theory of Computing Systems
  - Volume 45
  - August 2009



- Tel Aviv University, Israel
- John Hopkins University, Baltimore, USA

Experiment

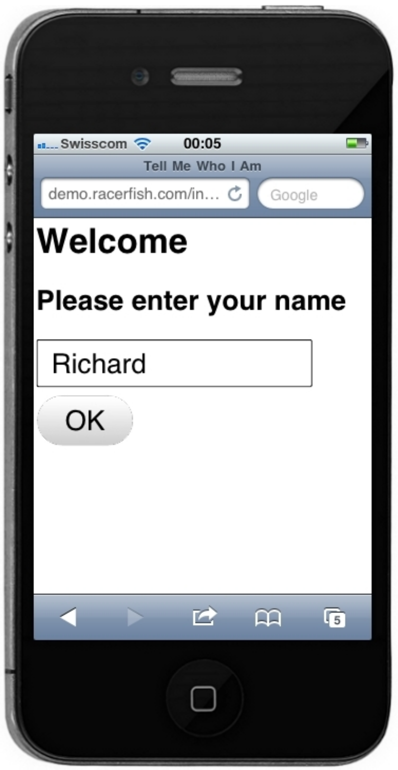
- Travel in a foreign country
- Unknown language
- Learn to know the night life subculture
- Not allowed to talk to each other

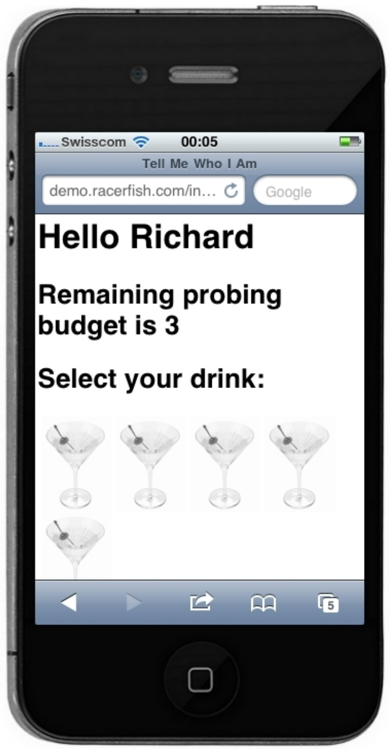
Experiment

- Problem:
  - 5 typical drinks
  - money for 3 drinks
- Waitress asks whether you liked the drink
- Idea: Human preferences correlate

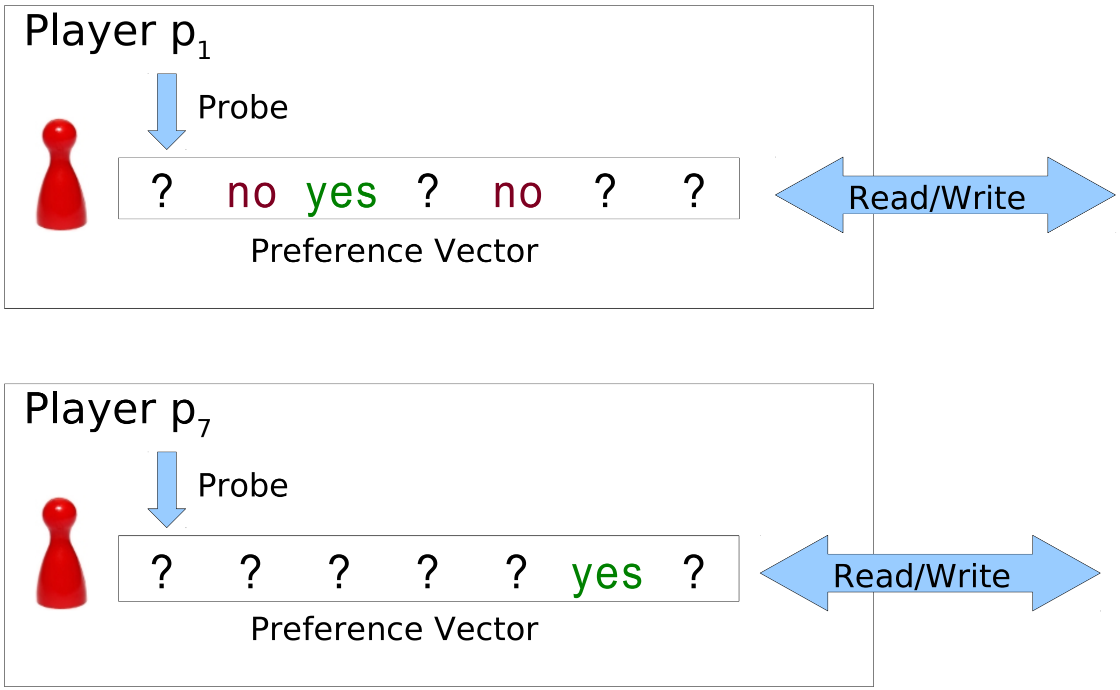
Experiment

- http://demo.racerfish.com





Players and Billboard



Shared Billboard

	O_1	O_2	O_3	O_4	O_5	O_6	O_7
p_1	?	no	yes	?	no	?	?
p_2	yes	?	no	?	?	?	no
p_3	?	yes	?	yes	yes	?	no
p_4	no	no	?	no	?	?	yes
p_5	?	no	no	?	yes	?	?
p_6	?	yes	?	?	?	no	no
p_7	?	?	?	?	?	yes	?

How can a player find out his preferences with only a few probes?

Statement of the Problem

- n players and m objects
- each player has an unknown yes/no grade for each object
- Parallel rounds: in each round each player
  - reads the shared billboard
  - probes one object
  - writes the result of the probe on the billboard
- For each player: output a vector as close as possible to that player's original preference vector

Statement of the Problem (Formal)

- Input:
  - A set P of n players and a set O of m objects
  - A vector $v(p) \in \{yes, no\}^m$ for each player p
- Output:
  - An estimate vector $w(p) \in \{yes, no\}^m$ for each player p
- Goal:
  - Minimize $dist(v(p), w(p))$ for each player p

    $dist(x, y)$ is the Hamming distance
  - Minimize the number of probes

Input Characteristic

- **Diameter** of a subset $A \subset P$

$$D(A) = max\{dist(v(p), v(q)) | p, q \in A\}$$

- (α, D) **-typical set**: Subset $A \subset P$ with

$$|A| \geqslant \alpha n, \quad 0 \leqslant \alpha \leqslant 1$$

$$D(A) \leqslant D, \quad D \geqslant 0$$

Approximation Quality

- **Discrepancy** of a subset $A \subset P$

$$\Delta(A) = max\{dist(w(p), v(p)) | p \in A\}$$

- **Stretch** of a subset $A \subset P$

$$\rho(A) = \frac{\Delta(A)}{D(A)}$$

The CHOOSE_CLOSEST Problem

- Input
  - A set V of preference Vectors with $|V|=k$
  - A player p with (initially unknown) preference vector $v(p)$
- Output
  - A vector $w_{min} \in V$ such that

$$dist(w_{min}, v(p)) \leqslant dist(w, v(p)) \quad , w \in V$$

		Object 1	Object 2	Object 3
Player p		yes	yes	no
V	v_1	yes	no	no
	v_2	yes	no	yes
	v_3	no	yes	yes

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The SELECT Algorithm

- Solves an adapted version of the CHOOSE_CLOSEST problem
- Adaptions:
  - Additional input D
  - There is a vector $w \in V$ such that $dist(w, v(p)) \leqslant D$

The SELECT Algorithm

D=1		X(V)						
		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	?	?	?	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The SELECT Algorithm

D=1		X(V)						
		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	?	?	?	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Reapeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The SELECT Algorithm

D=1		X(V) Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	no	?	?	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

The SELECT Algorithm

D=1	X(V)						
	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p	?	no	?	?	?	?	?
V v_1	yes	no	yes	no	no	yes	yes
V v_2	yes	no	no	yes	yes	no	no
V v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

The SELECT Algorithm

D=1		X(V)						
		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	no	?	?	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

  1a) Let X(V) be the set of Objects on which some two vectors in V differ.

  1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

  1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

The SELECT Algorithm

D=1		X(V)						
		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	no	no	?	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The SELECT Algorithm

D=1		X(V)						
		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	no	no	?	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

The SELECT Algorithm

D=1		X(V) Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	no	no	?	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The SELECT Algorithm

D=1		X(V)						
		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	no	no	yes	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

The SELECT Algorithm

D=1		X(V)						
		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	no	no	yes	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The SELECT Algorithm

D=1		X(V)						
		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
Player p		?	no	no	yes	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

1) Repeat

1a) Let X(V) be the set of Objects on which some two vectors in V differ.

1b) Execute Probe on the first coordinate in X(V) that has not been probed yet.

1c) Remove from V any vector with more than D disagreements with v(p).

Until all coordinates in X(V) are probed or X(V) is empty.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The SELECT Algorithm

D=1		Object 1	Object 2	Object 3 (Y)	Object 4	Object 5	Object 6	Object 7
Player p		?	no	no	yes	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

2) Let Y be the set of objects probed by p. Output the vector closest to v(p) regarding only the objects in Y.

The SELECT Algorithm

D=1		Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Object 7
			Y					
Player p		?	no	no	yes	?	?	?
V	v_1	yes	no	yes	no	no	yes	yes
	v_2	yes	no	no	yes	yes	no	no
	v_3	yes	yes	no	yes	yes	no	no

2) Let Y be the set of objects probed by p. Output the vector closest to v(p) regarding only the objects in Y.

The SELECT Algorithm: Correctness

- Any vector removed from V is at distance more than D from v(p).
- All distinguishing coordinates of the remaining vectors were probed.
- Distance to v(p) exactly known up to a common additive term.

The SELECT Algorithm: Cost

- Each probe exposes at least one disagreement.
- No vector remains in V after finding D+1 disagreements
- After k(D+1) probes, no vector remains in V
  (k is the number of Vectors in V)
- Total cost upper bounded by k(D+1)

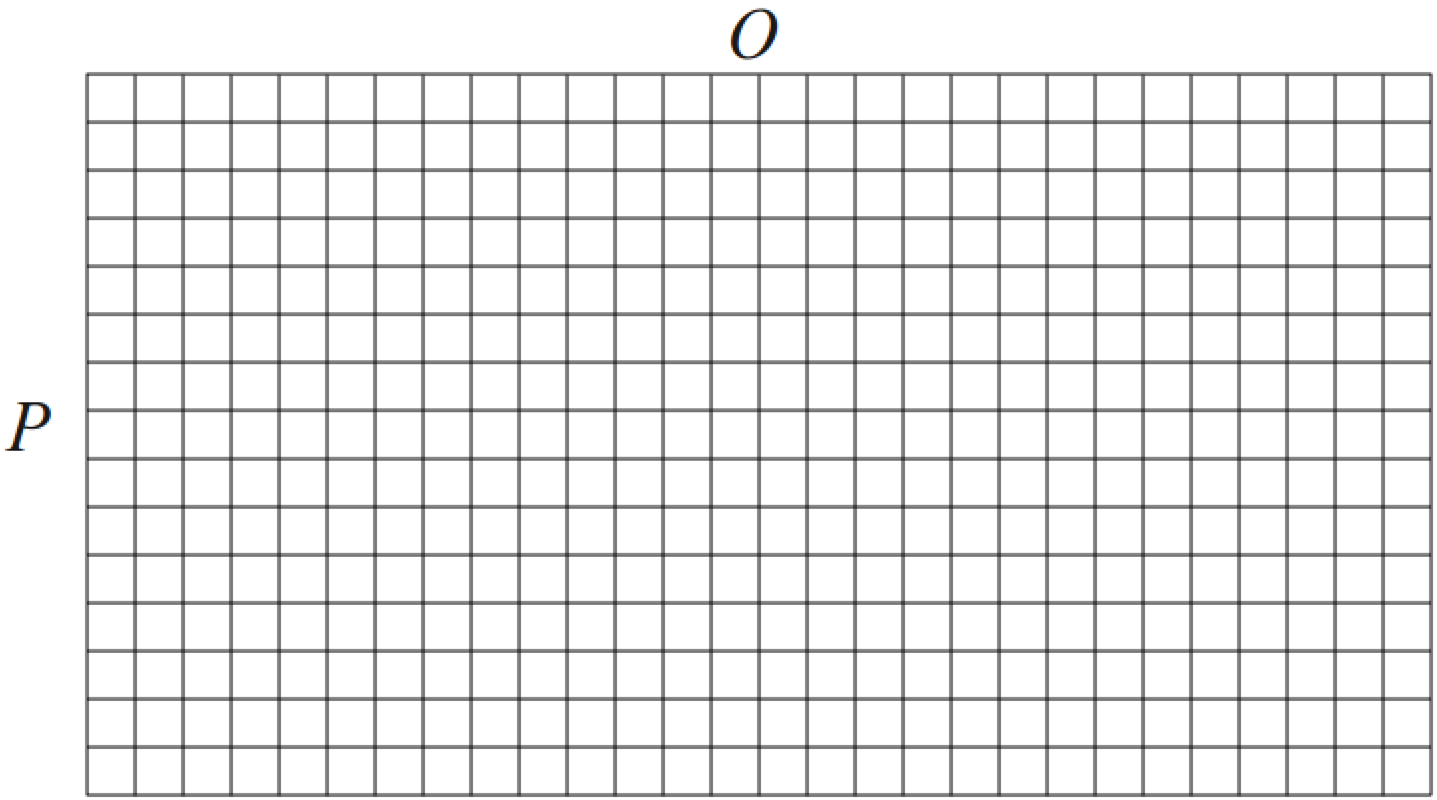
The ZERO_RADIUS Algorithm

- Input:
  - A set of players P and a set of objects O
  - Parameter $\alpha, \quad 0 \leqslant \alpha \leqslant 1$
- Output:
  - The correct vector for all players in a $(\alpha, 0)$-typical set
- Fails with probability $n^{-\Omega(1)}$
- Terminates after $O\left(\frac{\log(n)}{\alpha}\right)$ probes

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

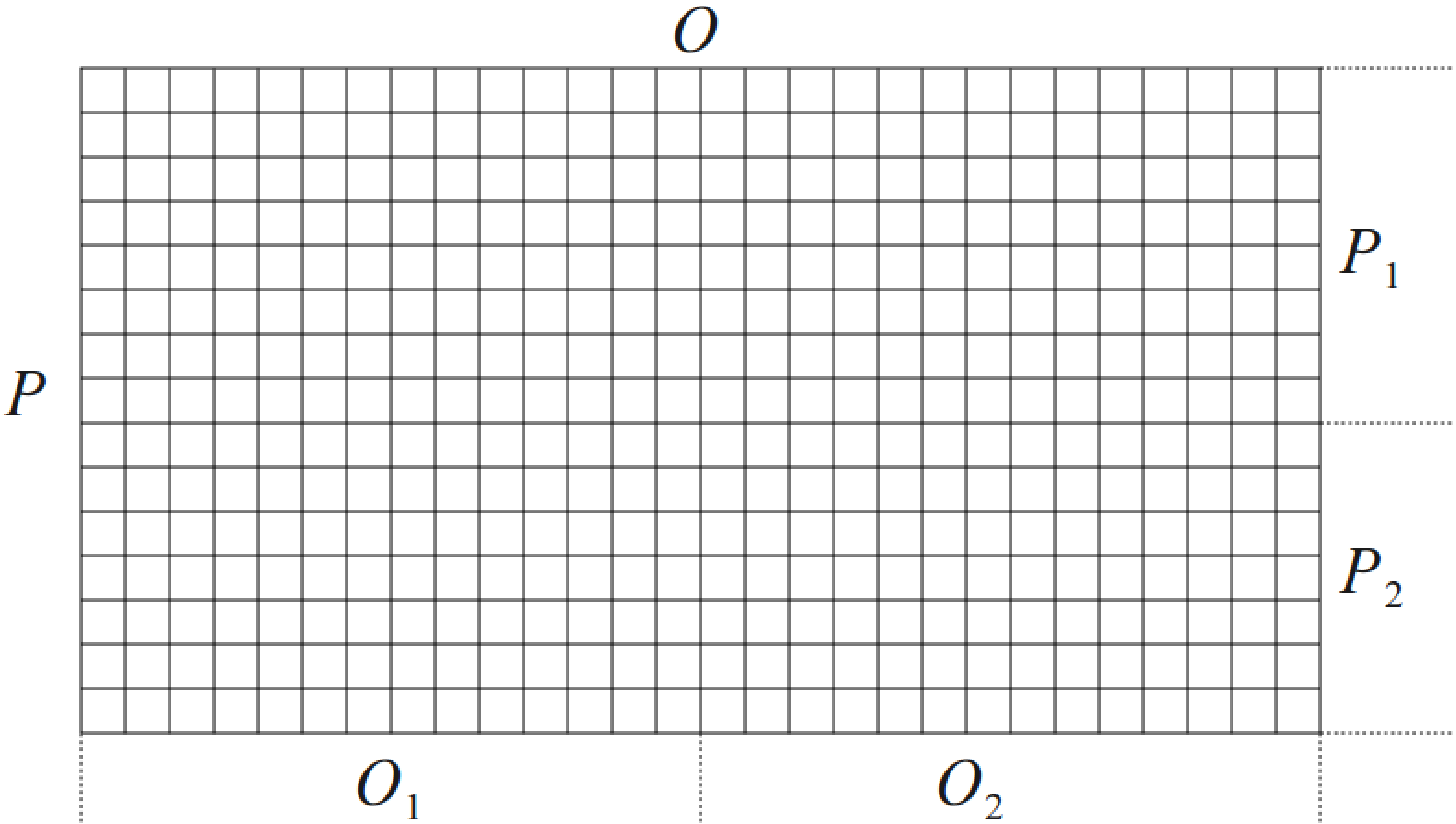
The ZERO_RADIUS Algorithm

1) If $min(|P|, |O|) \leqslant \frac{c \ln n}{\alpha}$ probe all objects and return



The ZERO_RADIUS Algorithm

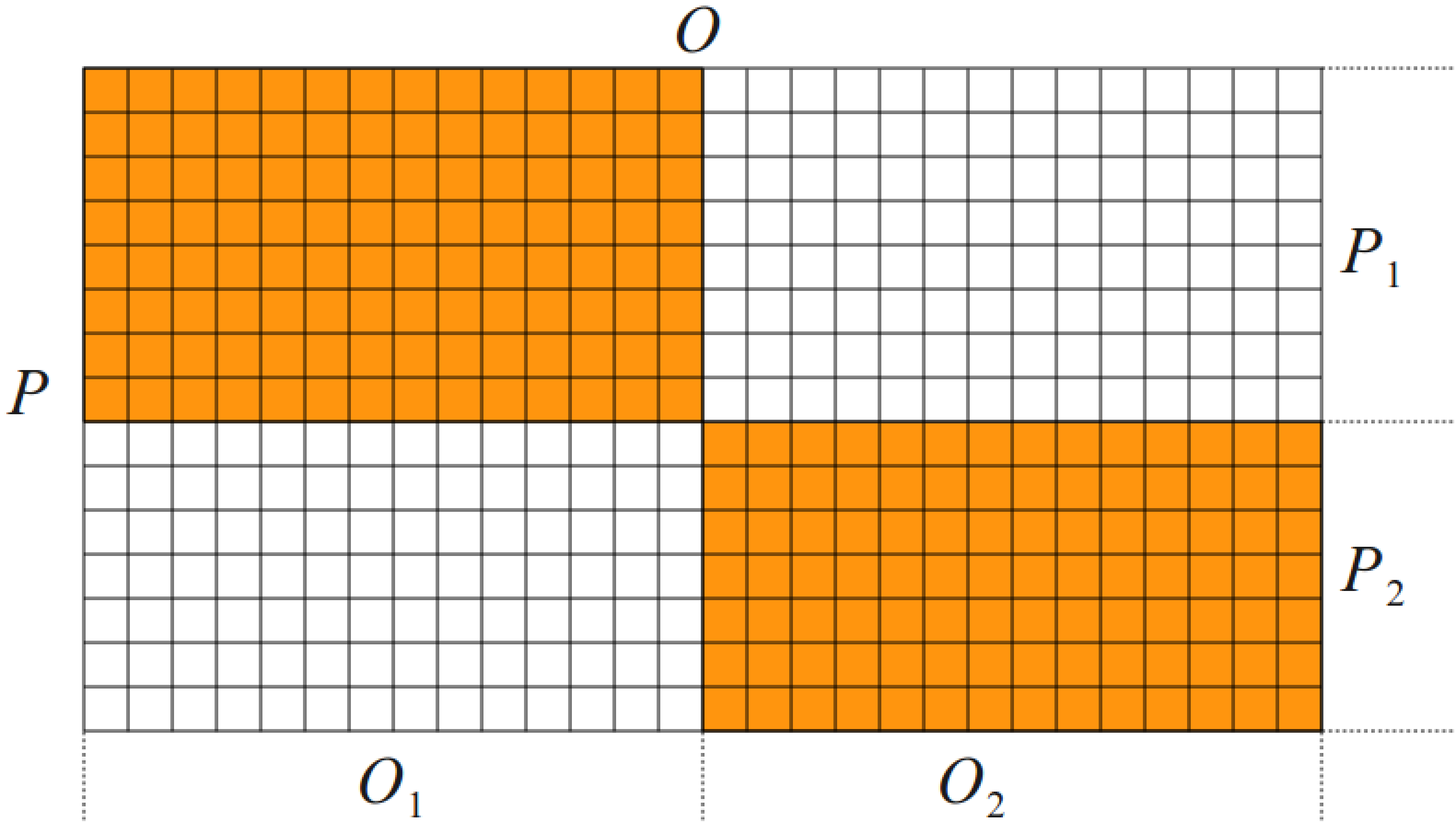
2) Partition randomly $P = P_1 \cup P_2$ and $O = O_1 \cup O_2$



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

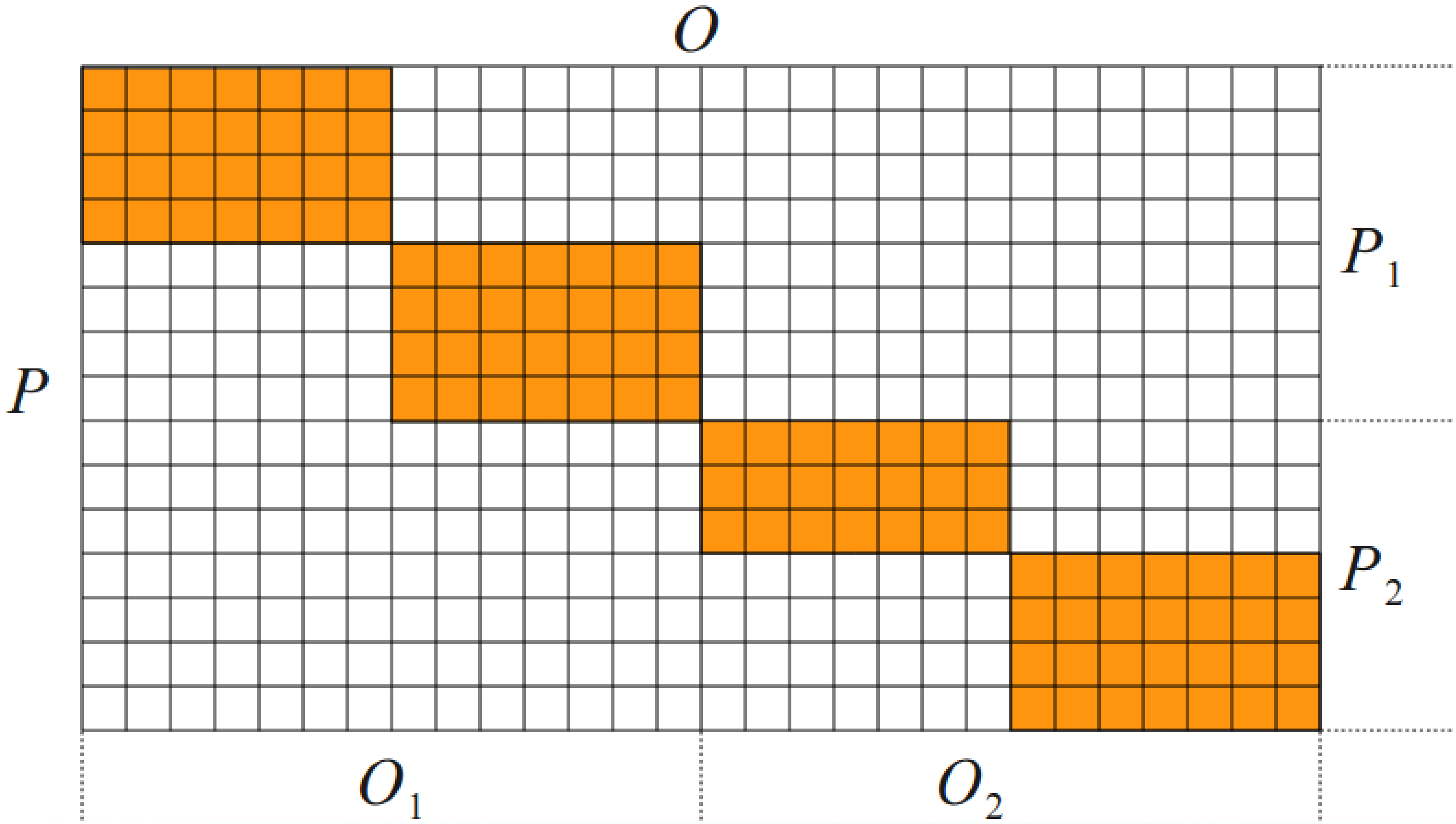
The ZERO_RADIUS Algorithm

3) Recursively execute ZERO_RADIUS for the yellow areas



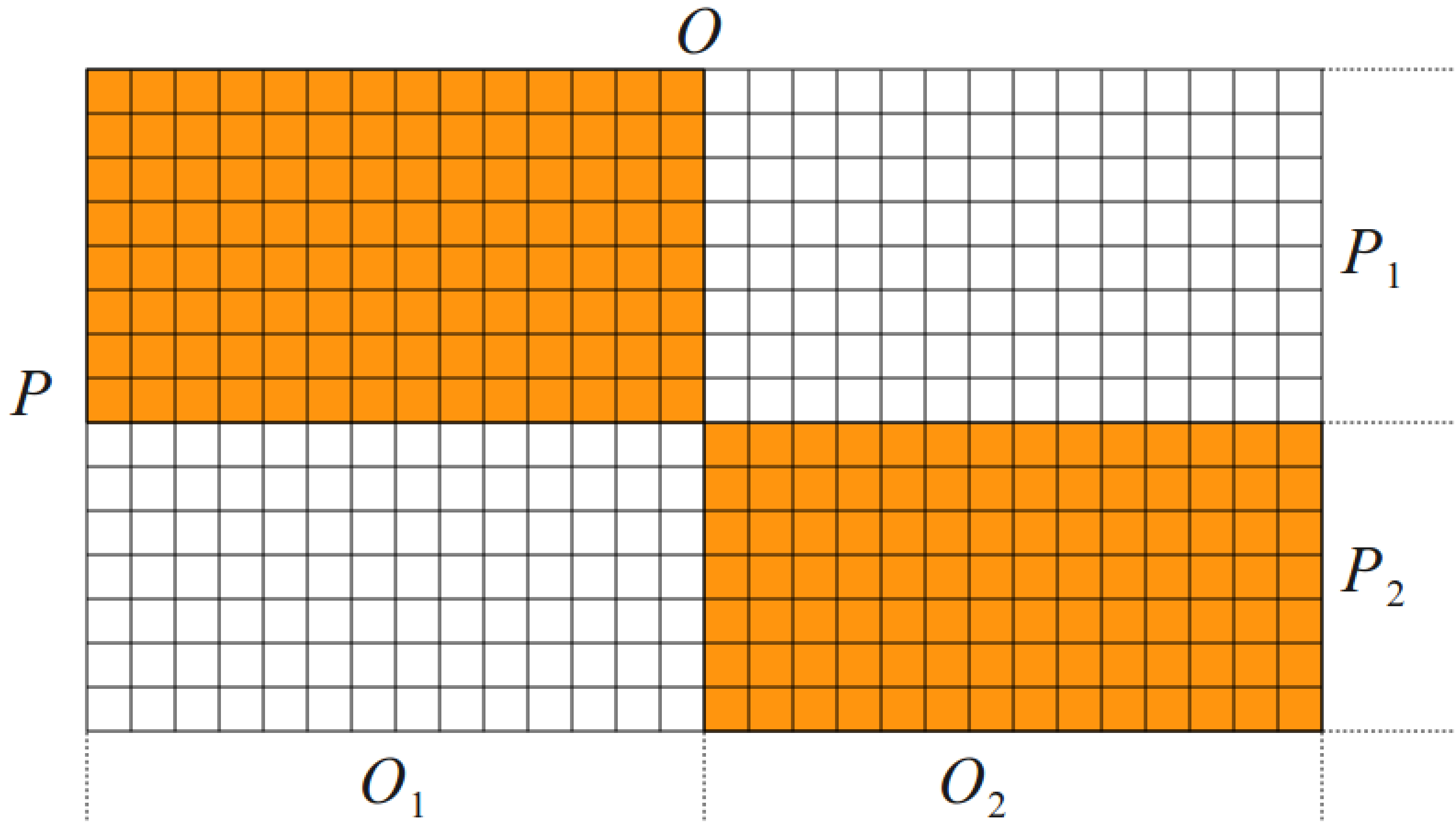
The ZERO_RADIUS Algorithm

3) Recursively execute ZERO_RADIUS for the yellow areas



The ZERO_RADIUS Algorithm

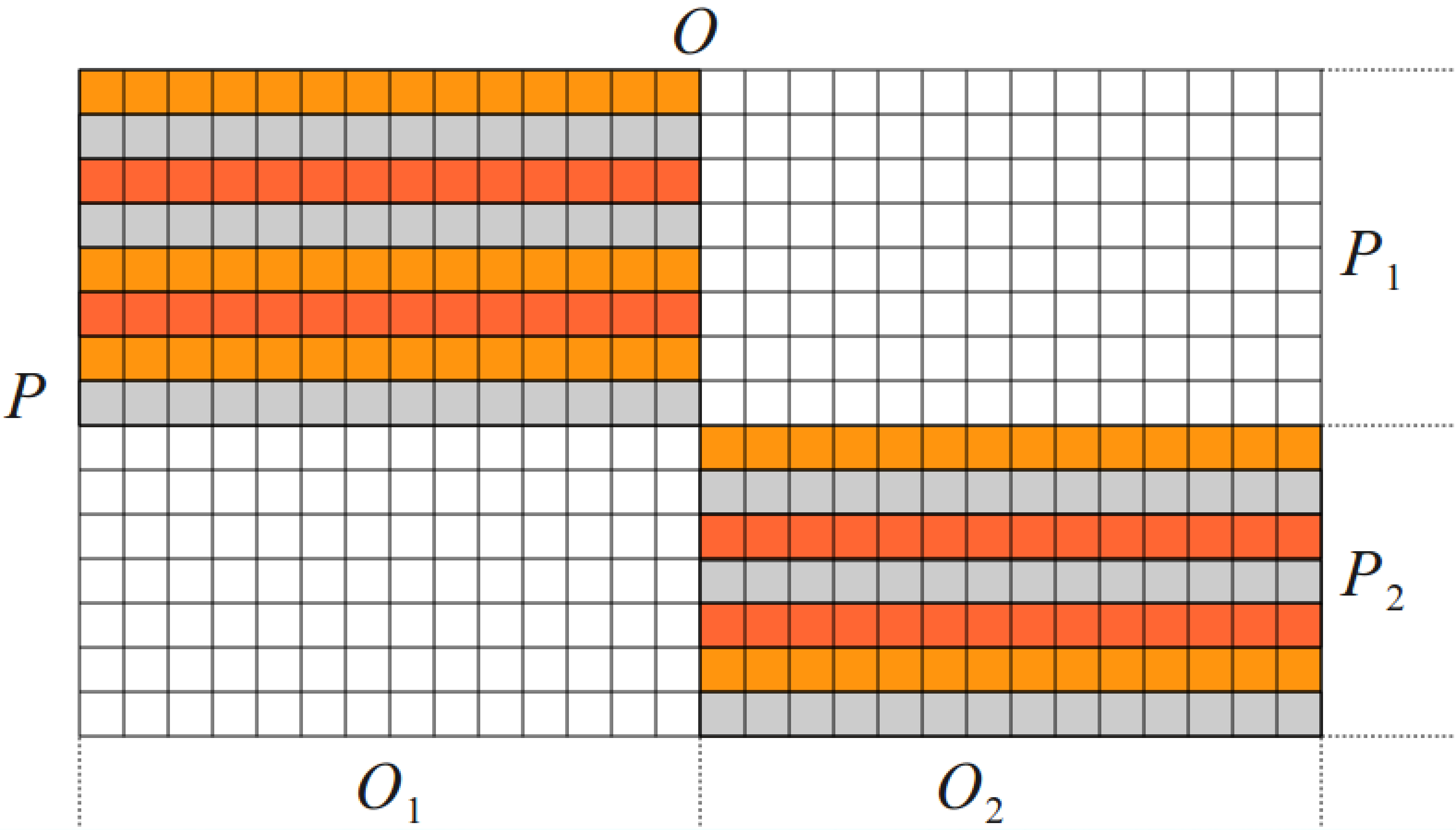
3) Recursively execute ZERO_RADIUS for the yellow areas



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The ZERO_RADIUS Algorithm

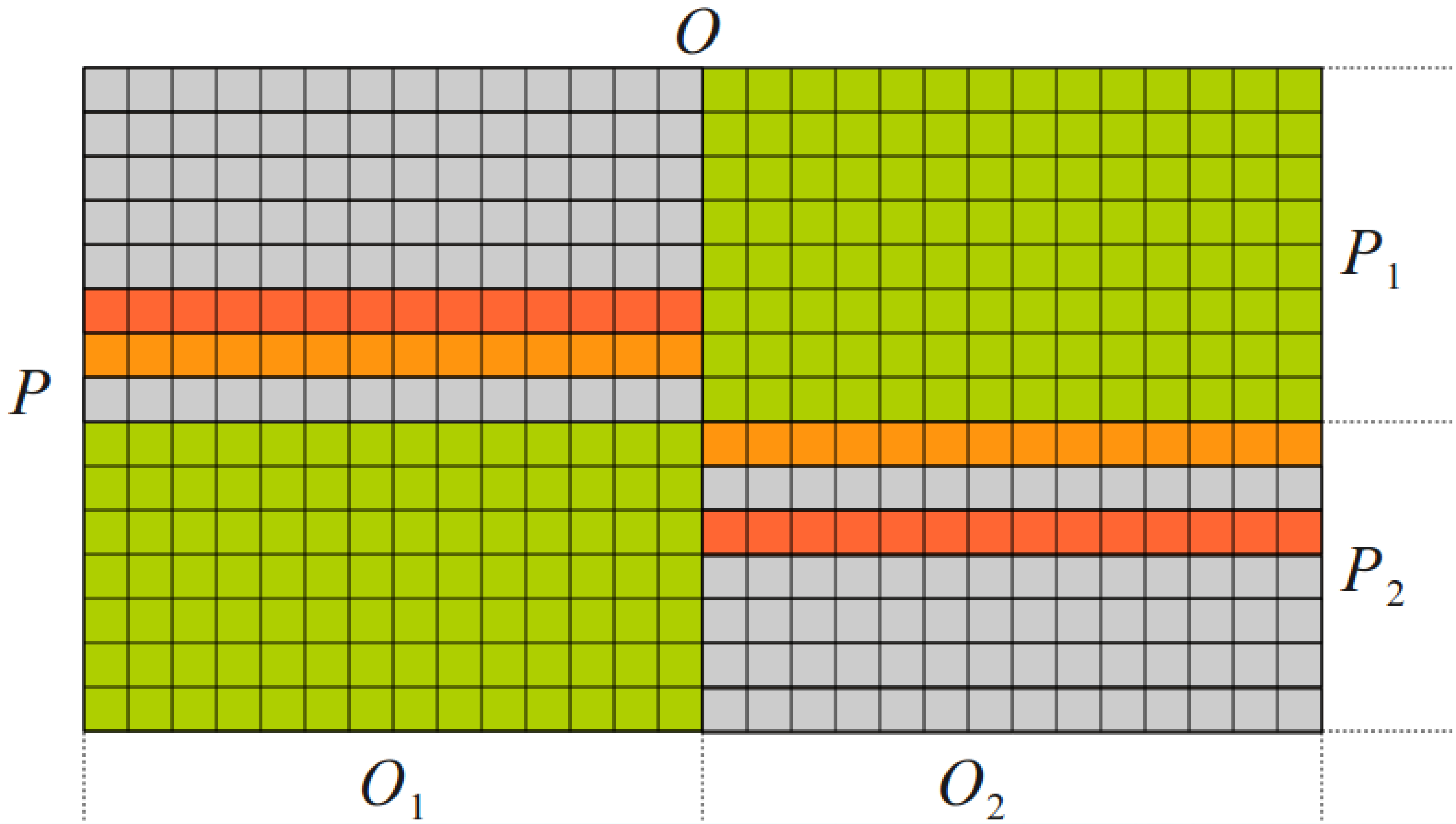
4) Consider only vectors, which are returned by a $\alpha/2$ fraction of the players.



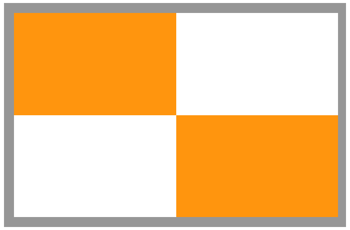
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The ZERO_RADIUS Algorithm

5) Execute SELECT for the green areas with the $\alpha/2$ remaining orange vectors as input and D=0

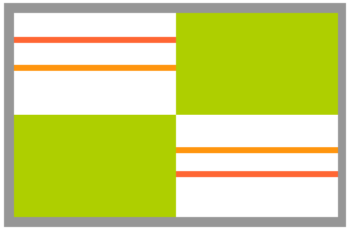


ZERO_RADIUS: Cost Analysis



- Step 1) Probing whole sub-area
  - Executed at most once by each player
  - How many objects probed by each player?
    - Recursive halving maintains $|O| \approx |P| \cdot m/n$
    - $n < m$:
      - Recursion stops when $|P| = O(\log n/\alpha)$
      - Player probes $O(m/n \cdot \log n/\alpha)$ objects
    - $n \geq m$:
      - Recursion stops when $|O| = O(\log n/\alpha)$
      - Player probes $O(\log n/\alpha)$ objects
  - Total cost of step 1) per player is $O(\lceil m/n \rceil \log n/\alpha)$

ZERO_RADIUS: Cost Analysis



- Step 5) (call to SELECT)
  - Call SELECT with $\mathcal{O}(1/\alpha)$ candidates and D=0
  - Recursion depth upper bounded by $\mathcal{O}(\log n)$
  - Total cost per player upper bounded by $\mathcal{O}(\log n/\alpha)$
- ZERO_RADIUS terminates after

$$\mathcal{O}\left(\left\lceil \frac{m}{n} \right\rceil \frac{\log n}{\alpha}\right) + \mathcal{O}\left(\frac{\log n}{\alpha}\right) = \mathcal{O}\left(\left\lceil \frac{m}{n} \right\rceil \frac{\log n}{\alpha}\right)$$

probes

Summary

- SELECT
  - Find closest of k vectors within distance D
  - $k(D+1)$
- ZERO_RADIUS
  - Find correct preference vector for players in $(\alpha, 0)$-typical sets
  - $\mathcal{O}(\lceil m/n \rceil \log n/\alpha)$

The SMALL_RADIUS Algorithm

- Input
  - Parameter $\alpha,\ 0 \leqslant \alpha \leqslant 1$
  - Parameter $D = O(\log n)$
- Output
  - An estimate vector $w(p)$ for every player p which is a member of a (α, D)-typical set A with

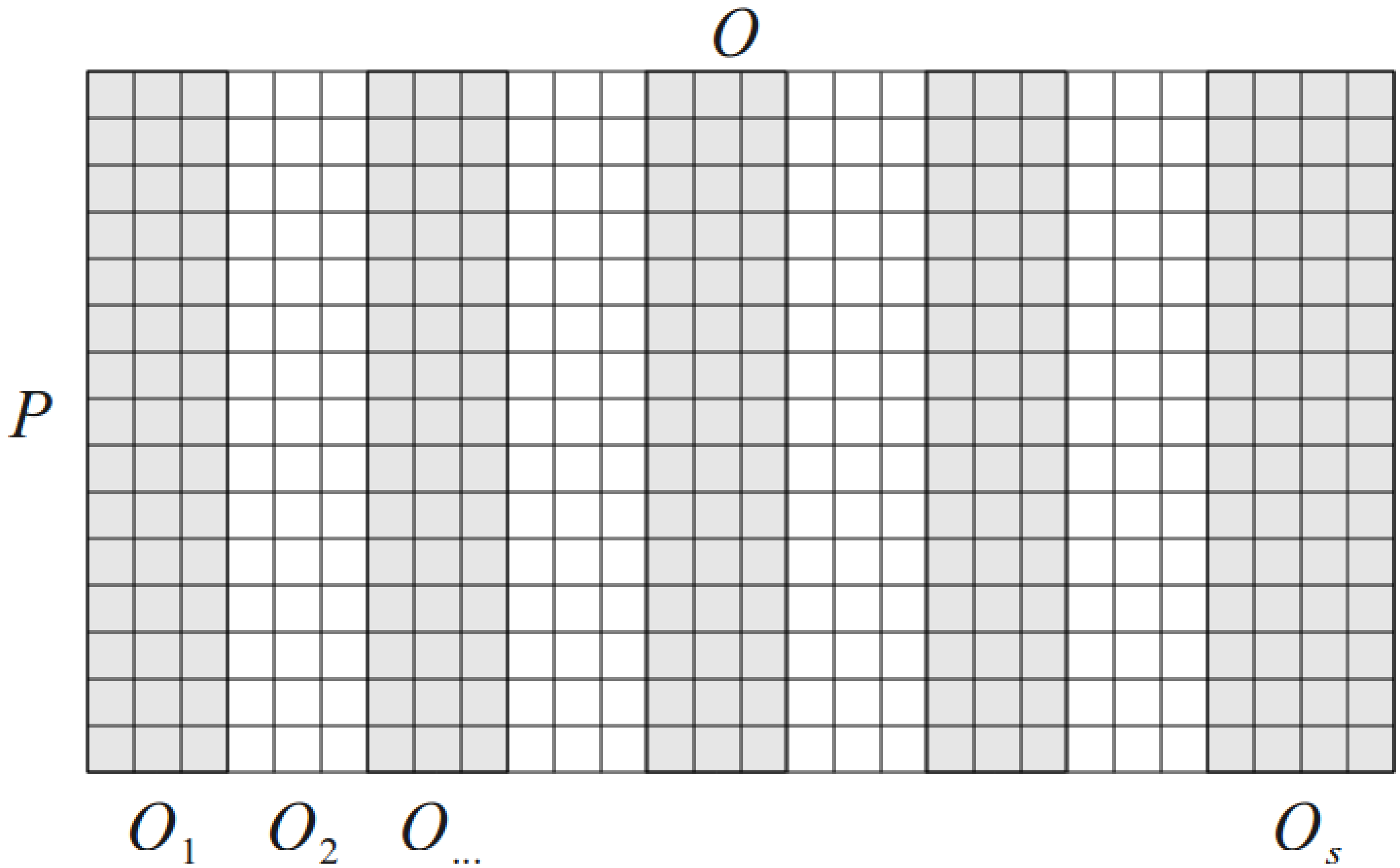
$$dist(w(p), v(p)) \leqslant 5\mathrm{D},\quad p \in A$$

$$\Rightarrow \Delta(A) \leqslant 5\mathrm{D}$$

$$\Rightarrow \rho(A) \leqslant 5$$

The SMALL_RADIUS Algorithm

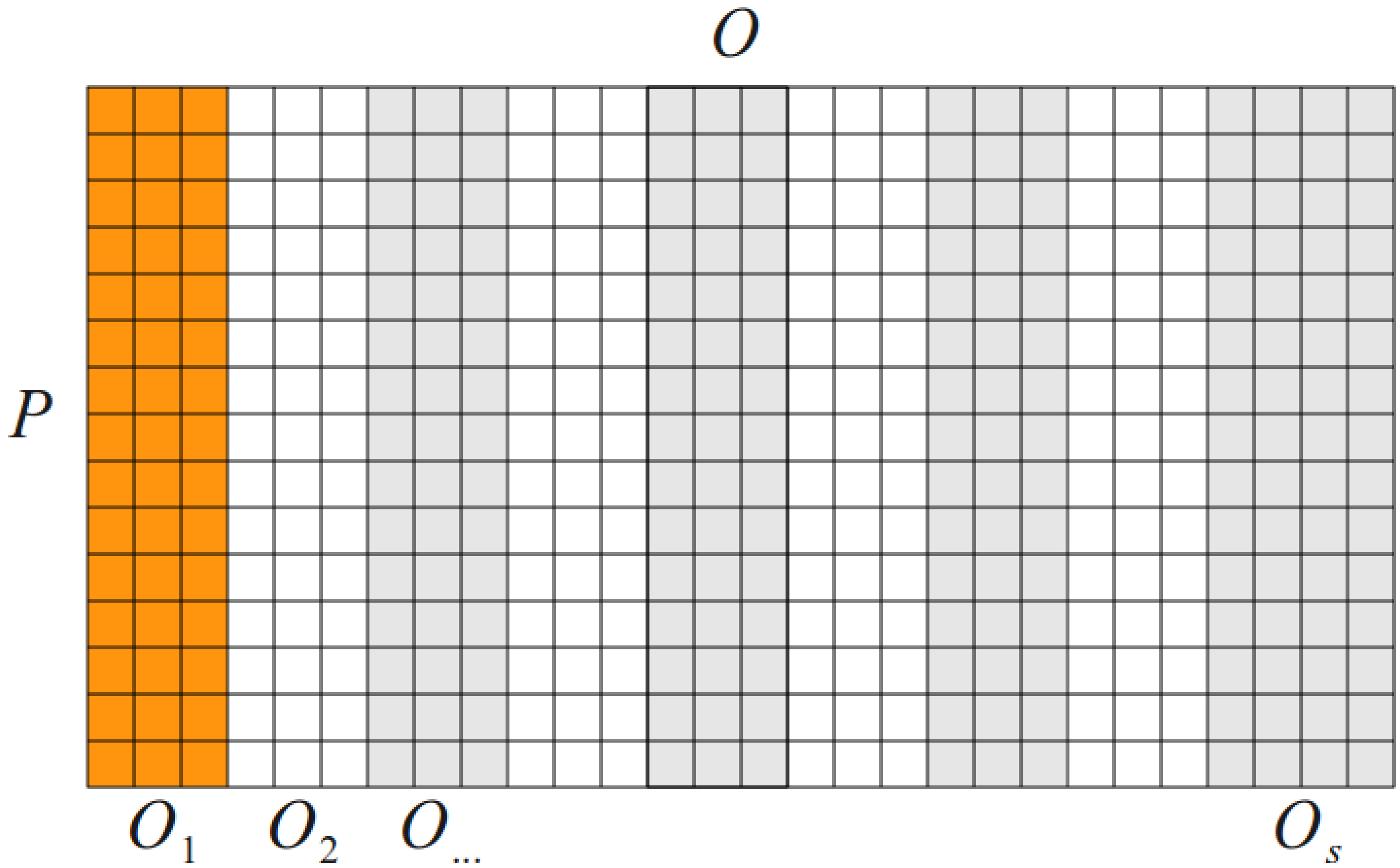
1) Partition randomly $O = O_1 \cup \ldots \cup O_s$ with $s = D^{3/2}$



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

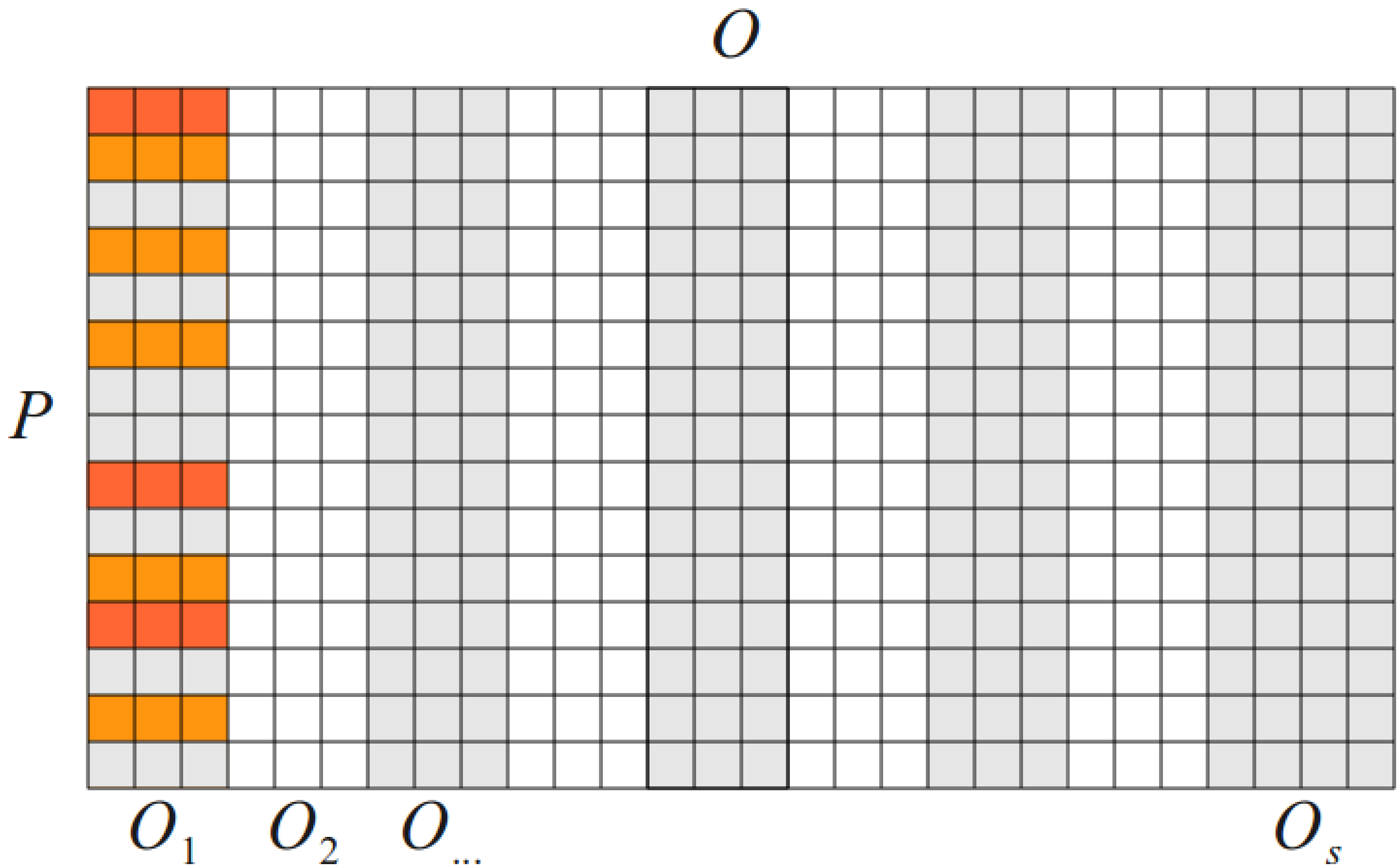
The SMALL_RADIUS Algorithm

2) For every O_i execute ZERO_RADIUS with all players and parameter $\alpha/5$



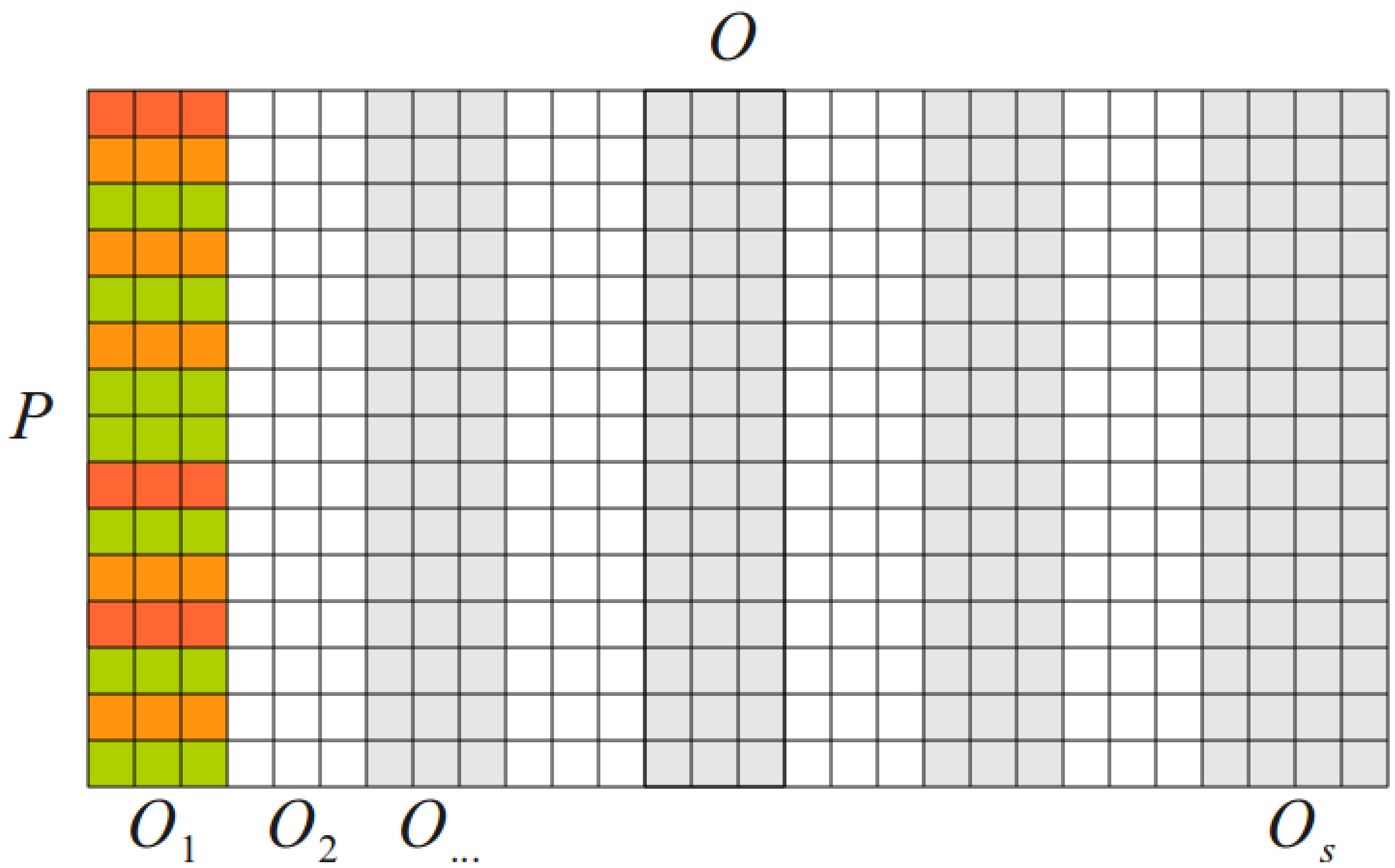
The SMALL_RADIUS Algorithm

3) Within the set O_i , only use vectors output by at least $\alpha n/5$ players



The SMALL_RADIUS Algorithm

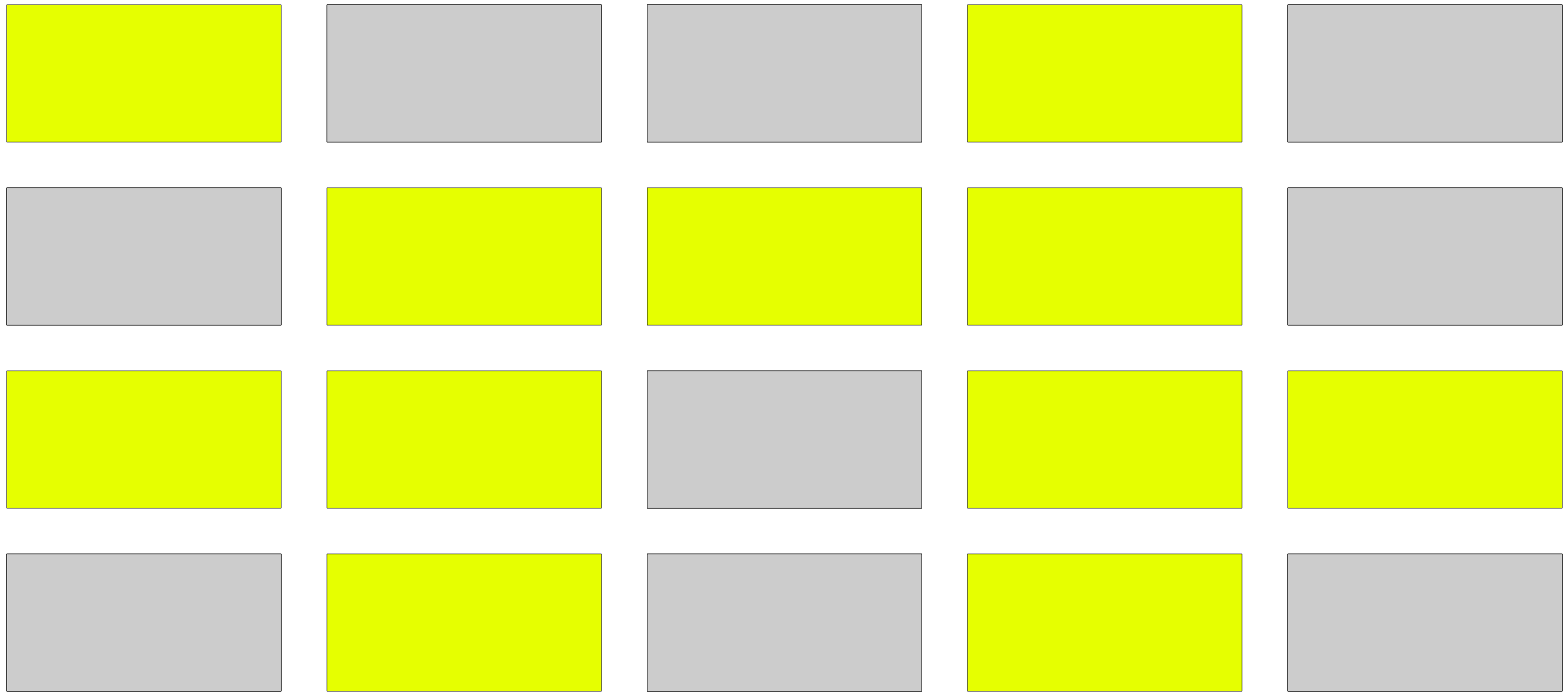
4) Within the set O_i , player P applies procedure SELECT to the remaining vectors with distance bound D



The SMALL_RADIUS Algorithm

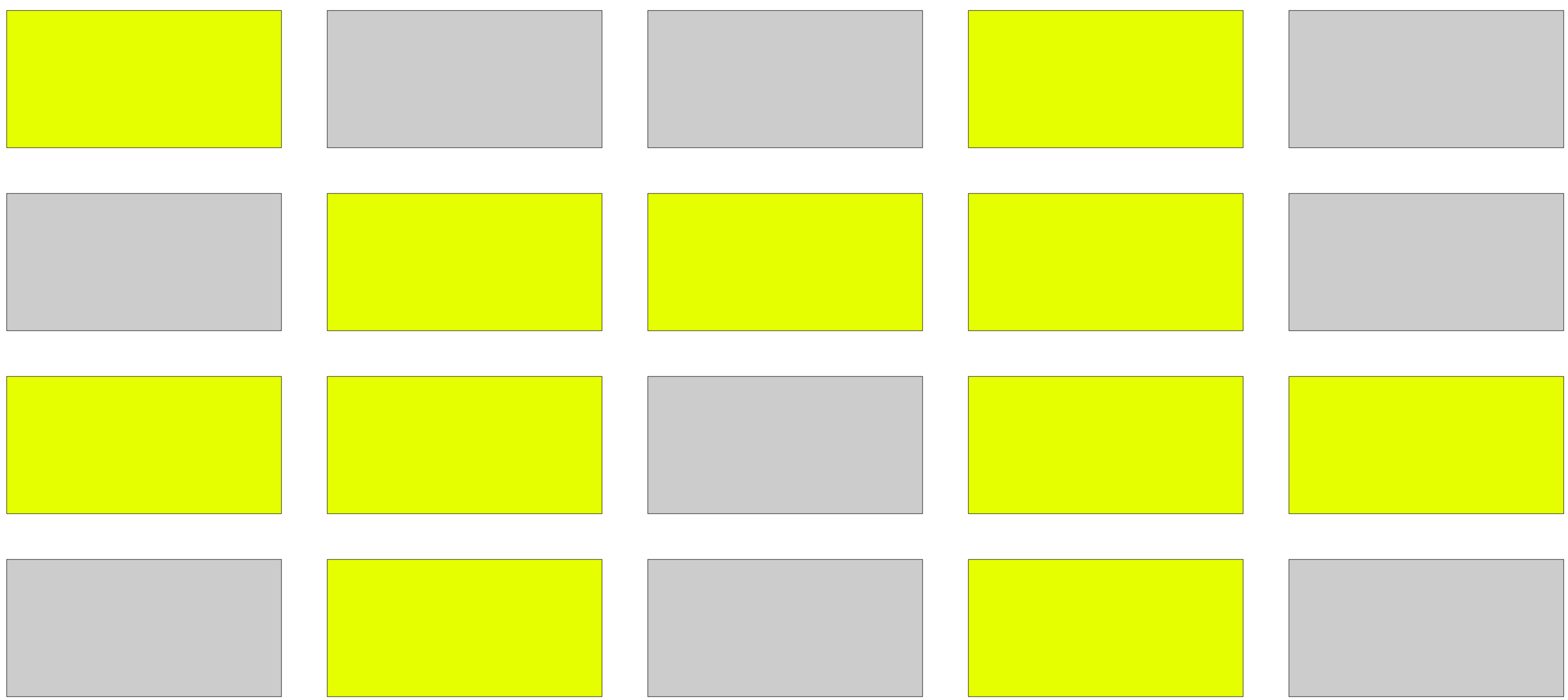
5) Do this K times.

Probability that one of the K independent executions succeed is $1-2^{-\Omega(K)}$



The SMALL_RADIUS Algorithm

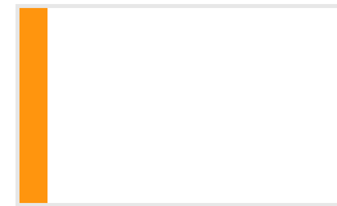
6) On the successful executions, all players execute SELECT with distance bound 5D and output the result.



SMALL_RADIUS: Cost

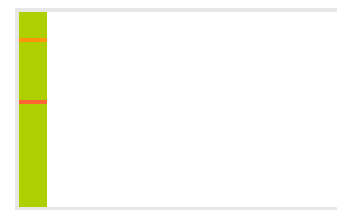
- Step 2): ZERO_RADIUS invoked
  - $s = O(D^{3/2})$ times with n users and m/s objects

    $O\left(\left(\frac{m}{n} + D^{3/2}\right) \cdot \frac{\log n}{\alpha}\right)$



- Step 4): SELECT invoked
  - $s = O(D^{3/2})$ times with bound D and at most $O(1/\alpha)$ candidates

    $O(D^{5/2}/\alpha)$



- Step 6): SELECT invoked $O(KD)$
- Overall complexity $O\left(K \frac{m}{\alpha n} D^{3/2} (\log n + D)\right)$

Summary

- SELECT
  - Find closest of k vectors within distance D
  - $k(D+1)$
- ZERO_RADIUS
  - Find correct preference vector for players in $(\alpha, 0)$-typical sets
  - $\mathcal{O}(\lceil m/n \rceil \log n/\alpha)$
- SMALL_RADIUS
  - Find preference vectors of (α, D)-typical sets with $\rho \leqslant 5$
  - $\mathcal{O}\left(K \frac{m}{\alpha n} D^{3/2} (\log n + D)\right)$

The LARGE_RADIUS Algorithm

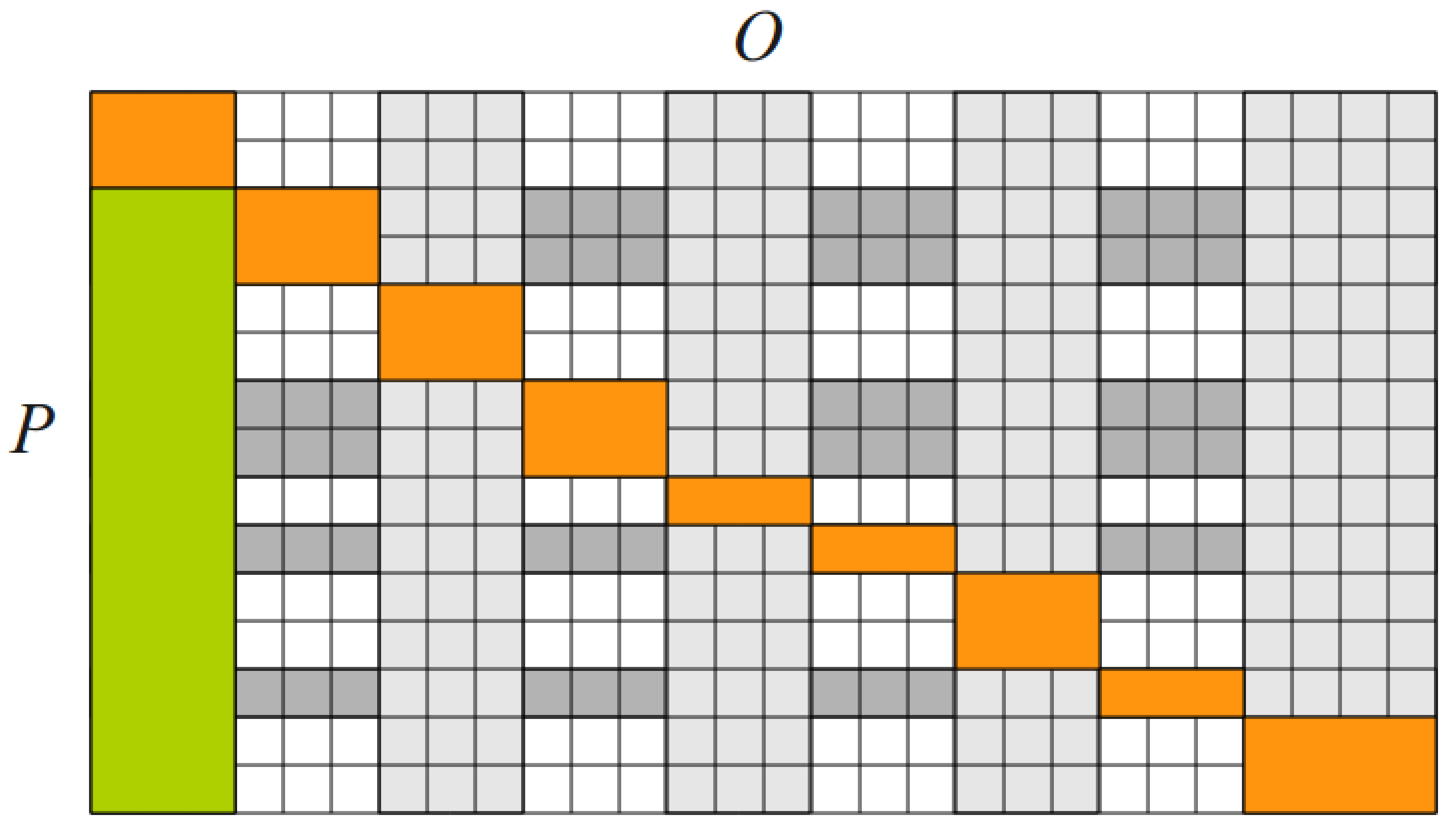
- Input
  - Parameter α
  - Parameter $D \geqslant \Omega(\log n)$
- Output
  - An estimate vector $w(p)$ for every player p which is a member of a (α, D)-typical set A with

$$dist(w(p), v(p)) = \mathcal{O}(D/\alpha), \quad p \in A$$

$$\Rightarrow \Delta(A) = \mathcal{O}(D/\alpha)$$

$$\Rightarrow \rho(A) = \mathcal{O}(1/\alpha)$$

LARGE_RADIUS: Idea



Main Algorithm

- Given α and D
  - If $D=0$ use ZERO_RADIUS
  - If $D=O(\log n)$ use SMALL_RADIUS
  - If $D \geqslant \Omega(\log n)$ use LARGE_RADIUS
- For every (α, D)-typical set A
  - w.h.p. $\Delta(A)=O(D/\alpha)$
  - the number of probes performed by each player is

$$O\left(\left\lceil \frac{m}{n} \right\rceil \cdot \frac{\log^{7/2} n}{\alpha^2}\right)$$

Unknown Input Characteristics

- Known α, unknown D
  - Run $O(\log n)$ independent versions of the main algorithm with $D = \{0, 2^1, 2^2, \ldots, 2^{\log n}\}$
  - Choose closest of all $O(\log n)$ output vectors
  - Increase running time by a factor of $O(\log n)$
  - Decrease quality of output by a constant factor

$$O\left(\left\lceil \frac{m}{n} \right\rceil \cdot \frac{\log^{9/2} n}{\alpha^2}\right)$$

Unknown Input Characteristics

- Unknown α, unknown D
  - Given α => number of probing rounds $\tau = O\left(\left\lceil \frac{m}{n} \right\rceil \cdot \frac{\log^{9/2} n}{\alpha^2}\right)$
  - Given τ => minimum $\alpha(\tau)$
  - Start parallel versions with $\alpha(\tau = 2^j)$ and unknown D
  - After every round, choose closest output vector

Conclusion

- Distributed algorithm for an interactive recommendation system
  - No restrictions on the input set
  - Has polylogarithmic running time
- First algorithm published that combines these two properties