MAX PLANCK INSTITUTE
FOR INFORMATICS

SIC Saarland Informatics Campus

Strategic Facility Location Problem

Mechanism Design Without Money

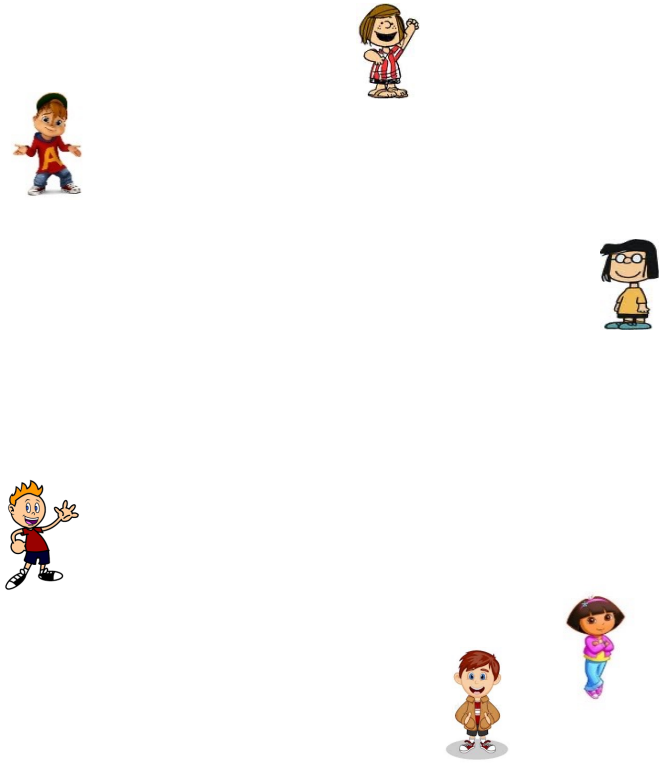
Kurt Mehlhorn, Javier Cembrano, **Golnoosh Shahkarami**

May 6, 2025

Mechanism Design

Strategyproof Mechanism:

- Agents have **private** information
- No agent has an incentive to misreport her data



Goal:

- Approximation guarantees of strategyproof mechanisms

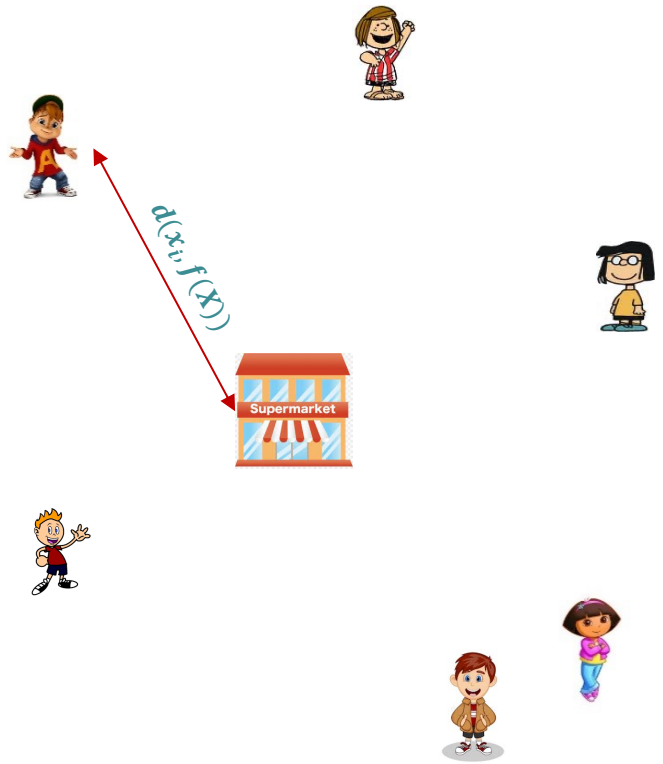
Single Facility Location

- N: set of n strategic agents
- $X = \{x_1, \dots, x_n\}$: each agent i has a preferred location $x_i \in \mathbb{R}^2$
- $f(X) \in \mathbb{R}^2$: location of the facility
- $d(x_i, f(X)) \geq 0$: Euclidean distance of agent i from the facility
- **Goal**: minimize some social cost objective
  - Egalitarian Social Cost:

$$\max_{i \in N} d(x_i, f(X))$$

  - Utilitarian Social Cost:

$$\sum_{i \in N} d(x_i, f(X))$$



Utilitarian Social Cost

- Goal: minimize $\sum_{i \in N} d(x_i, f(X))$

Median — 1-D

Input:

$$X = \{x_1, \ldots, x_n\}$$

Output:

$$f(X) = \underset{i \in N}{median}\, x_i$$

Strategyproof

Optimal

Egalitarian Social Cost

- Goal: minimize $\max_{i \in N} d(x_i, f(X))$

OPT — D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

$$f(X) = (\max_{i \in N} x_i - \min_{i \in N} x_i)/2$$

Egalitarian Social Cost

- Goal: minimize $\max_{i \in N} d(x_i, f(X))$

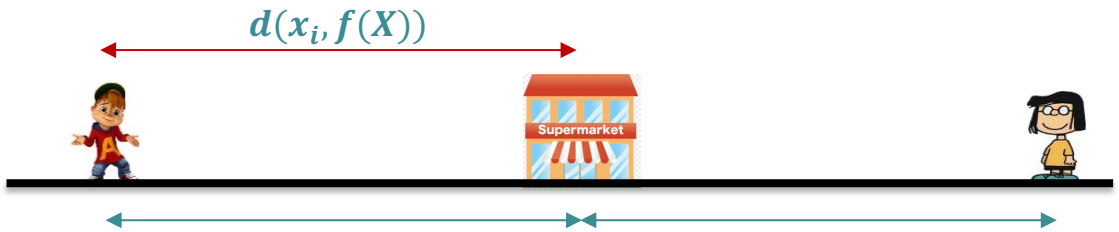
OPT — D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

$$f(X) = (\max_{i \in N} x_i - \min_{i \in N} x_i)/2$$



Egalitarian Social Cost

- Goal: minimize $\max_{i \in N} d(x_i, f(X))$

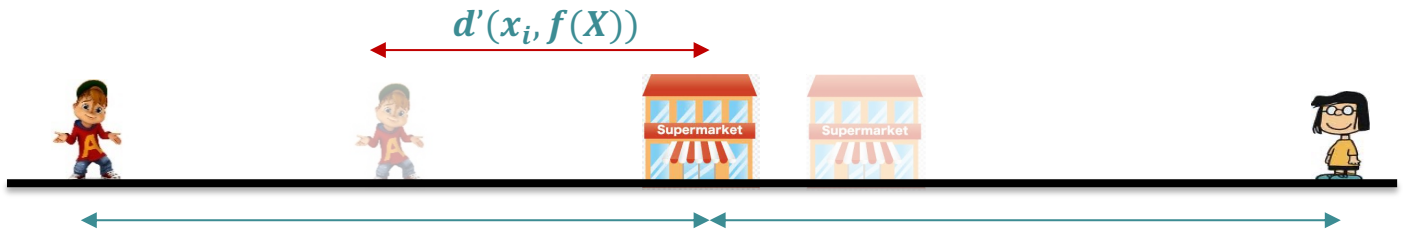
OPT — D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

$$f(X) = (\max_{i \in N} x_i - \min_{i \in N} x_i)/2$$



Strategyproof Mechanisms - Deterministic

Median — D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

$$f(X) = \underset{i \in N}{median}\, x_i$$

- **Strategyproof**:

  Returning any fixed x_i is strategyproof

- **2-approximation**:

  Any point in the interval $[x_1, x_n]$ is 2-approximation

- **Tight**:

  Any deterministic strategyproof mechanism has an approximation ratio of at least 2

Strategyproof Mechanisms - Randomized

LRM

R

Input:

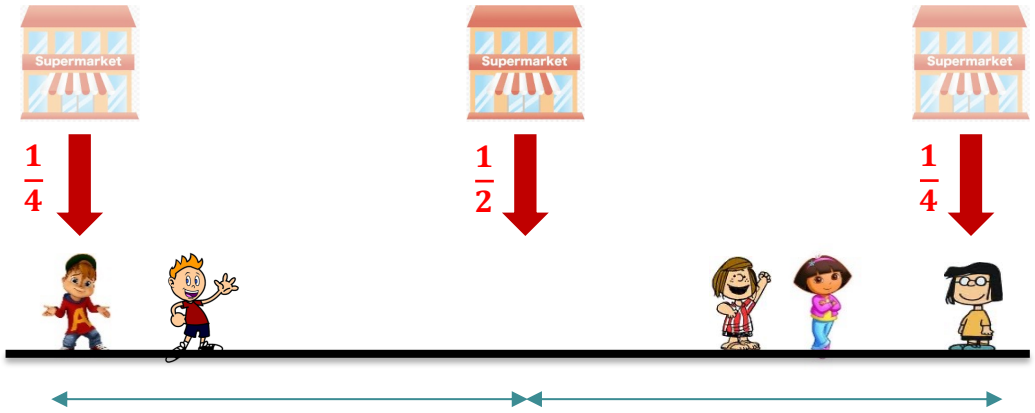
$$X = \{x_1, \dots, x_n\}$$

Output:

w.p. $\frac{1}{4}$: $f(X) = \min_{i \in N} x_i$

w.p. $\frac{1}{2}$: $f(X) = \text{mid}_{i \in N} x_i$

w.p. $\frac{1}{4}$: $f(X) = \max_{i \in N} x_i$



- **Strategyproof**
- **1.5-approximation**:

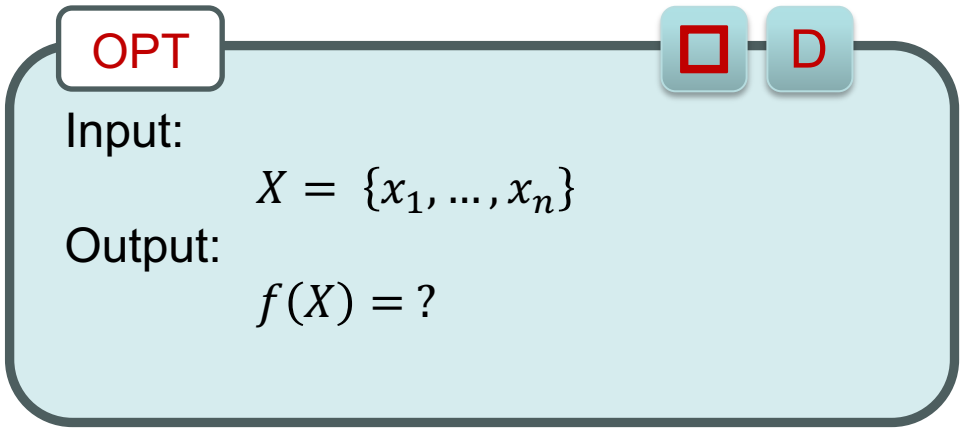
$$\frac{1}{4} \times 2 + \frac{1}{2} \times 1 + \frac{1}{4} \times 2 = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = \frac{3}{2}$$

- **Tight**:

  Any randomized strategyproof mechanism has an approximation ratio of at least 1.5 on the line metric

Strategyproof Mechanisms – Euclidean Plane

- Goal: minimize $\sum_{i \in N} d(x_i, f(X))$



- Coordinatewise median: $\sqrt{2}$ – apx

Strategyproof Mechanisms – Euclidean Plane

- Goal: minimize $\max_{i \in N} d(x_i, f(X))$

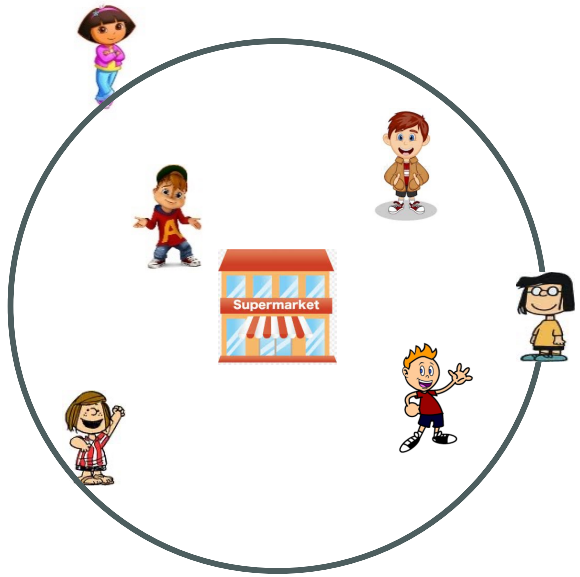
OPT

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

$$f(X) = center\ of\ the\ minimum\ bounding\ circle$$



- Random Dictatorship: 2 – apx
- Centroid Mechanism: $2 - \frac{1}{n}$ - apx

Strategyproof Mechanisms – Other Settings

- Two Facilities
  - PROCACCIA, A., TENNENHOLTZ, M., **Approximate Mechanism Design without Money (EC'2013)**
- Other Metrics (Circle, Tree, …)
  - Alon, N., Feldman, M., PROCACCIA, A., TENNENHOLTZ, M., **Strategyproof Approximation of the Minimax on Networks (MOR'2010)**
  - Meir, R., **Strategyproof Facility Location for Three Agents on a Circle (SAGT'19)**

MAX PLANCK INSTITUTE
FOR INFORMATICS

SIC Saarland Informatics Campus

Learning-Augmented Mechanism Design

Randomized Strategic Facility Location with Predictions

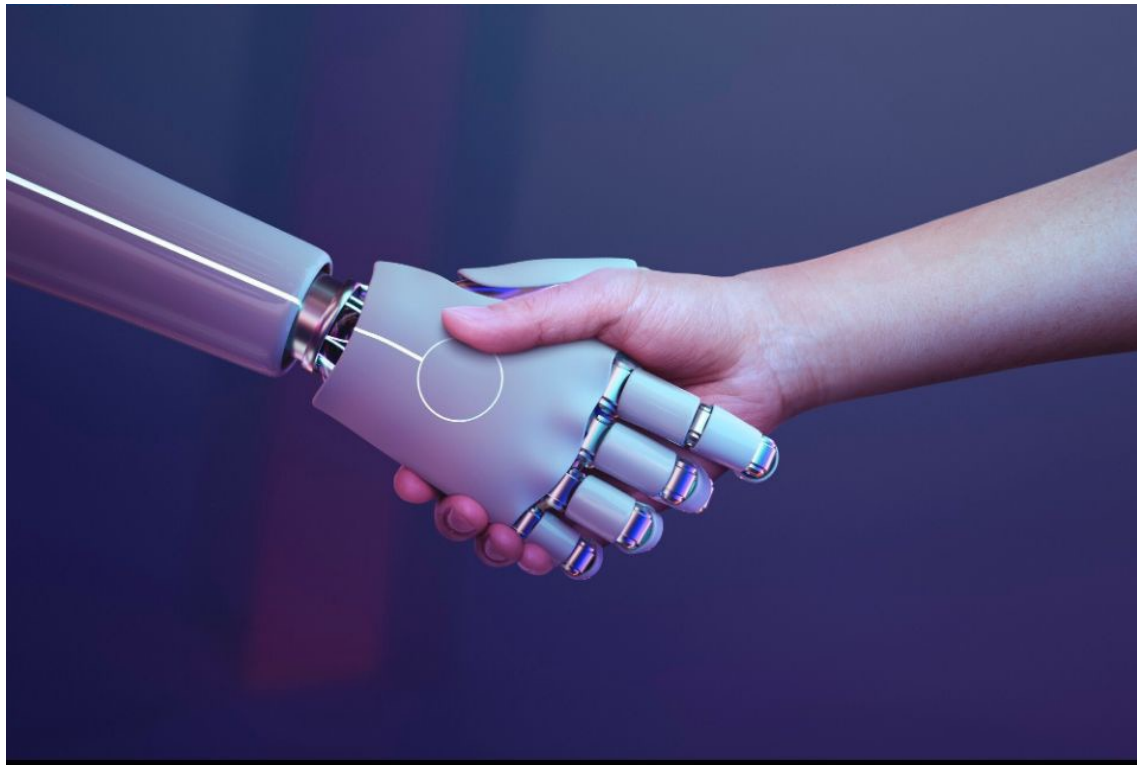
Eric Balkanski, Vasilis Gkatzelis , **Golnoosh Shahkarami**

Columbia University

Drexel University

MPI-INF

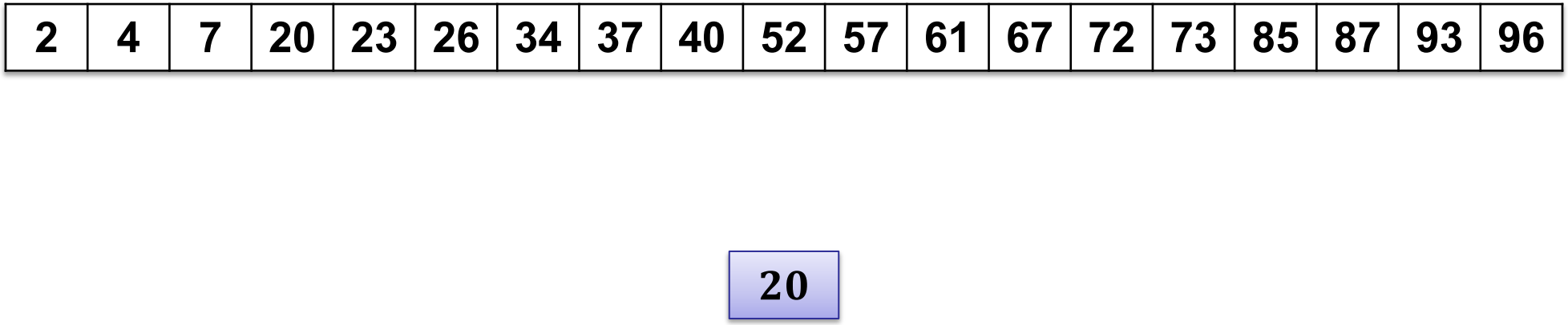
Learning-Augmented Algorithms



- Motivation
- Application
- Formal setting

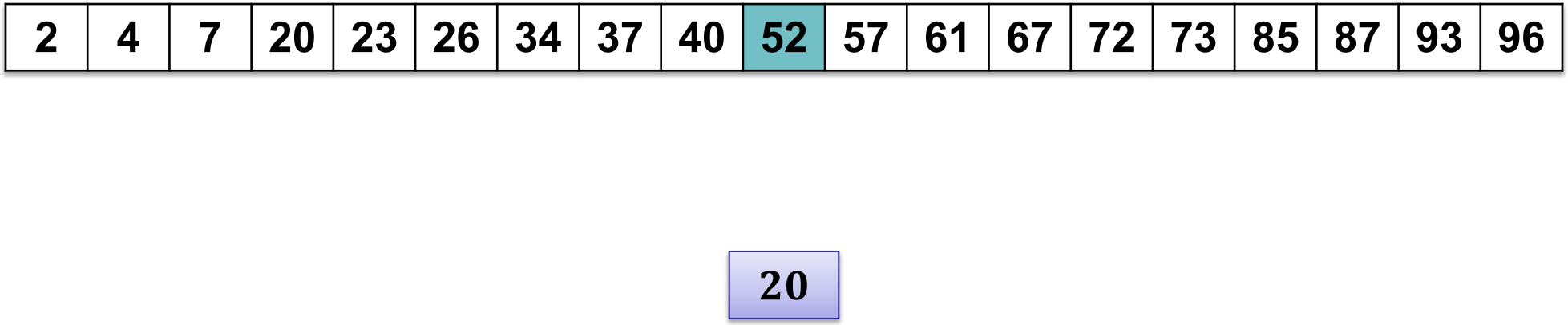
Learning-Augmented Algorithms

Example: Binary Search



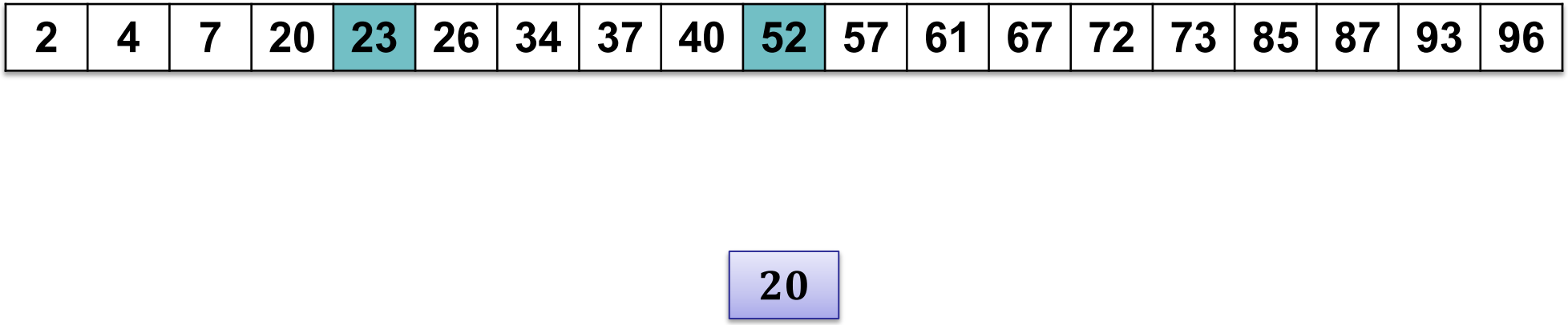
Learning-Augmented Algorithms

Example: Binary Search



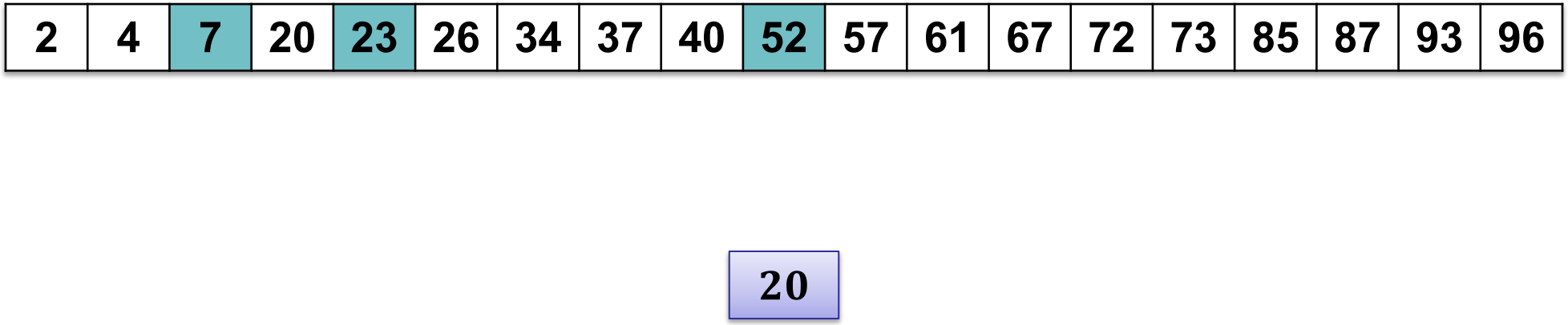
Learning-Augmented Algorithms

Example: Binary Search



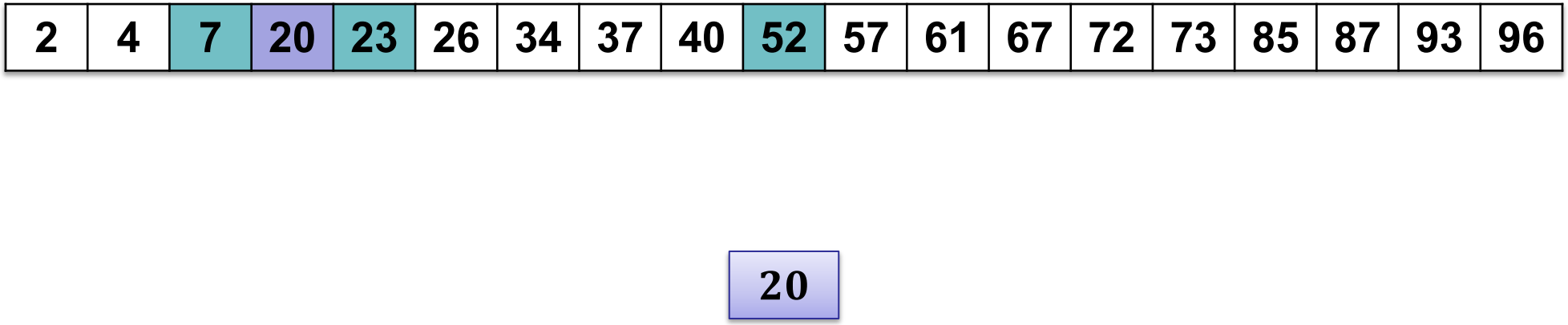
Learning-Augmented Algorithms

Example: Binary Search



Learning-Augmented Algorithms

Example: Binary Search



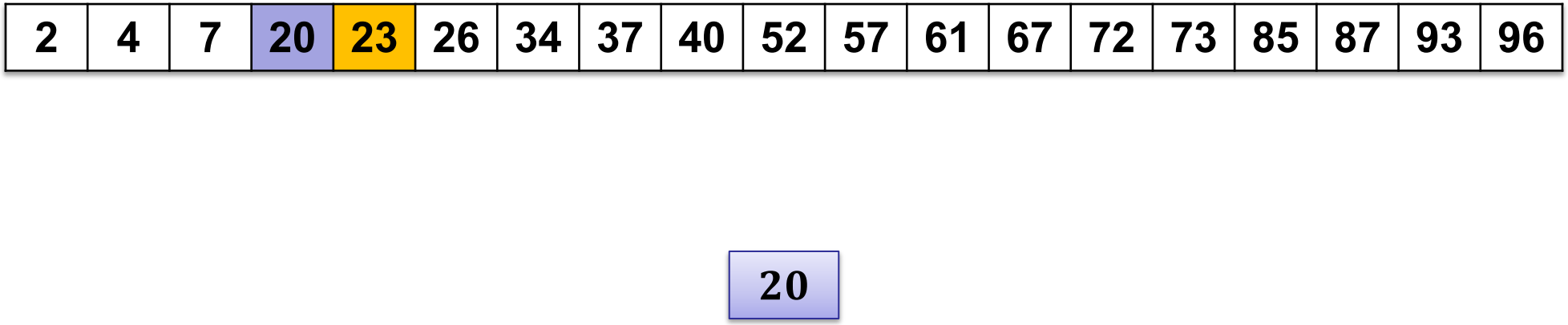
Learning-Augmented Algorithms

Example: Binary Search

20

Learning-Augmented Algorithms

Example: Binary Search

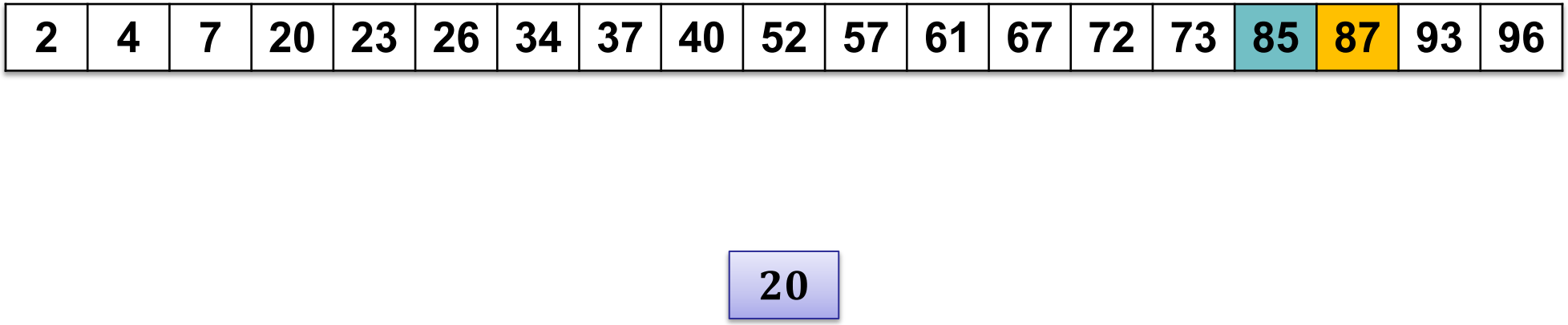


Learning-Augmented Algorithms

Example: Binary Search

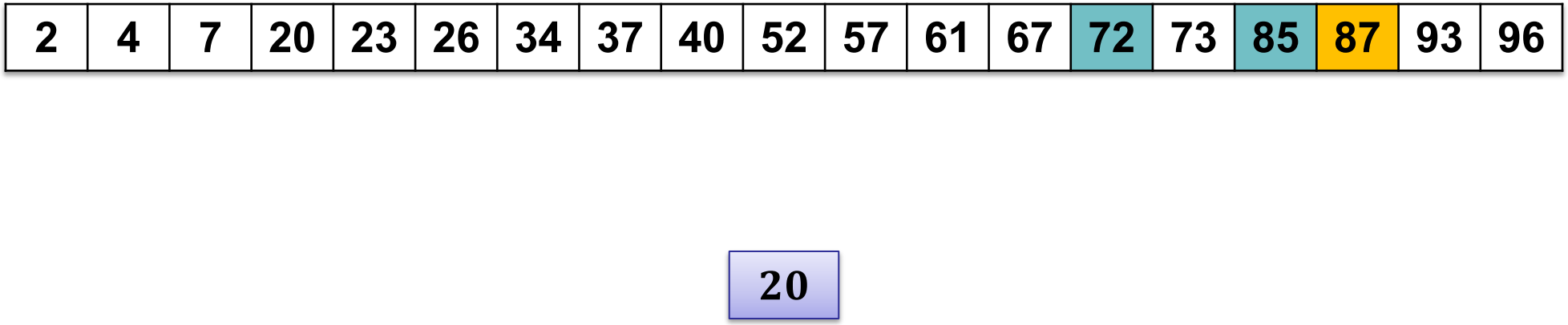
Learning-Augmented Algorithms

Example: Binary Search



Learning-Augmented Algorithms

Example: Binary Search



Learning-Augmented Algorithms

Example: Binary Search

20

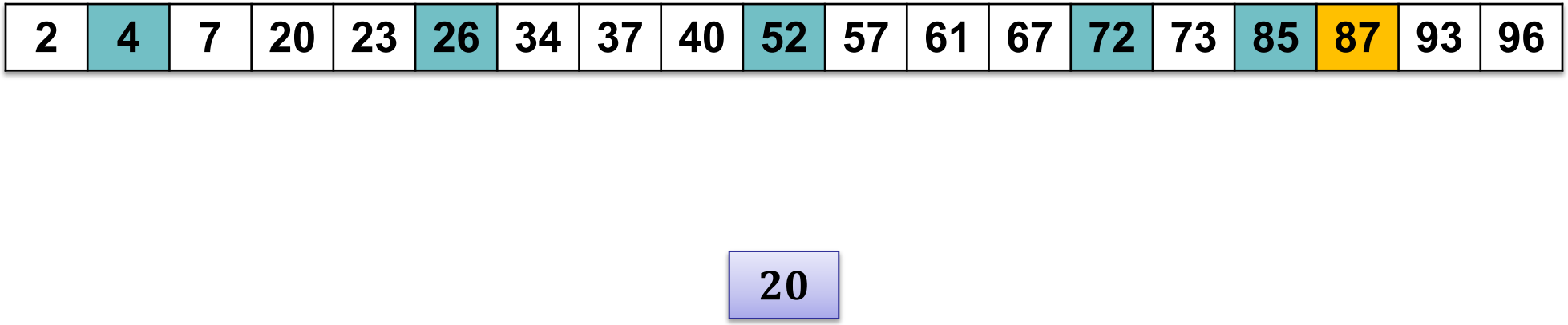
Learning-Augmented Algorithms

Example: Binary Search

20

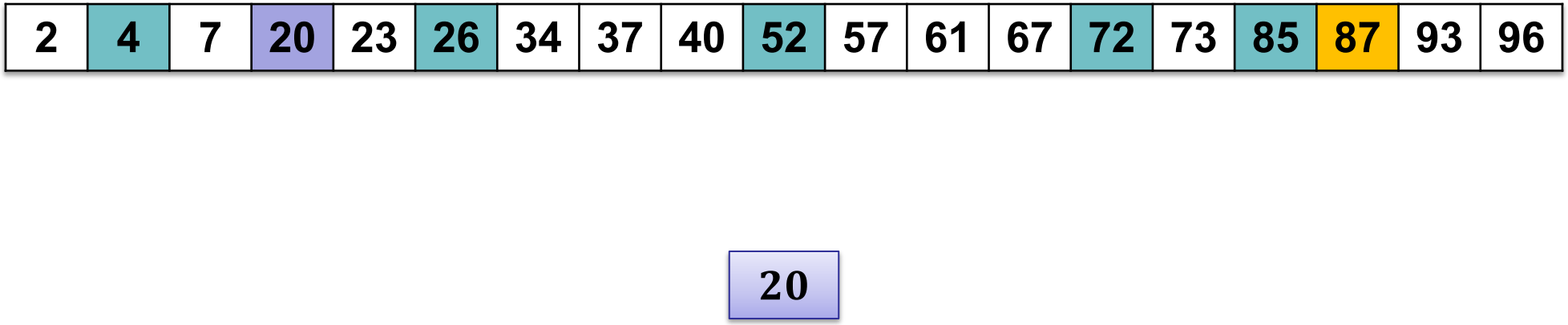
Learning-Augmented Algorithms

Example: Binary Search

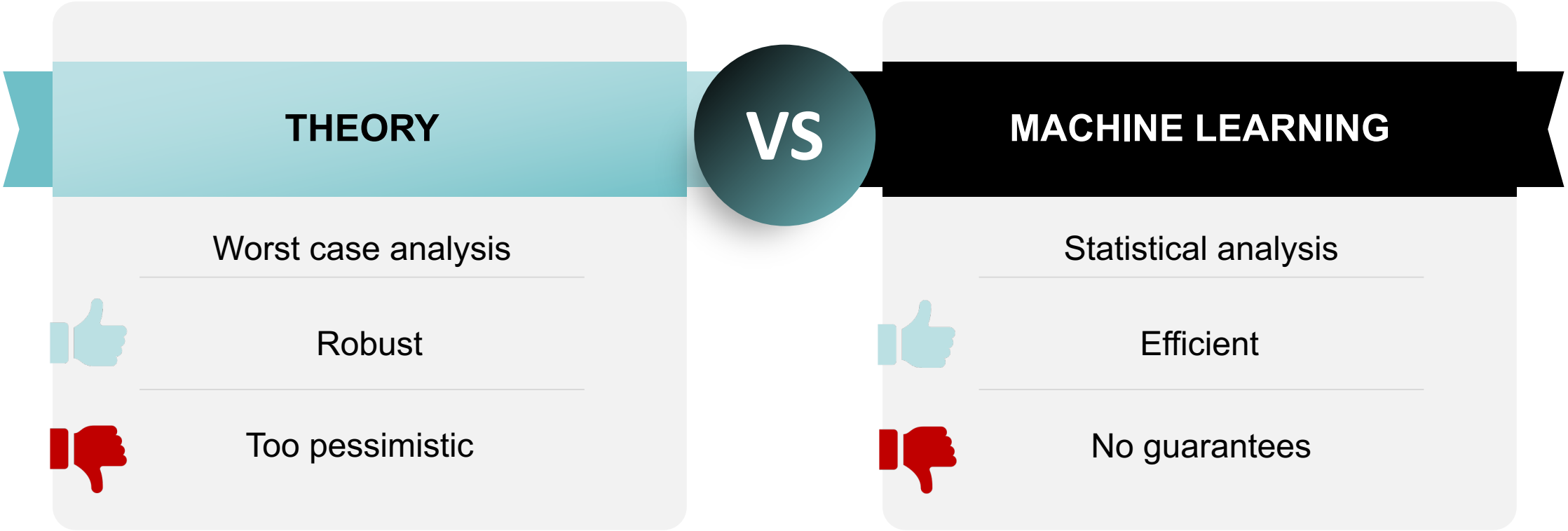


Learning-Augmented Algorithms

Example: Binary Search

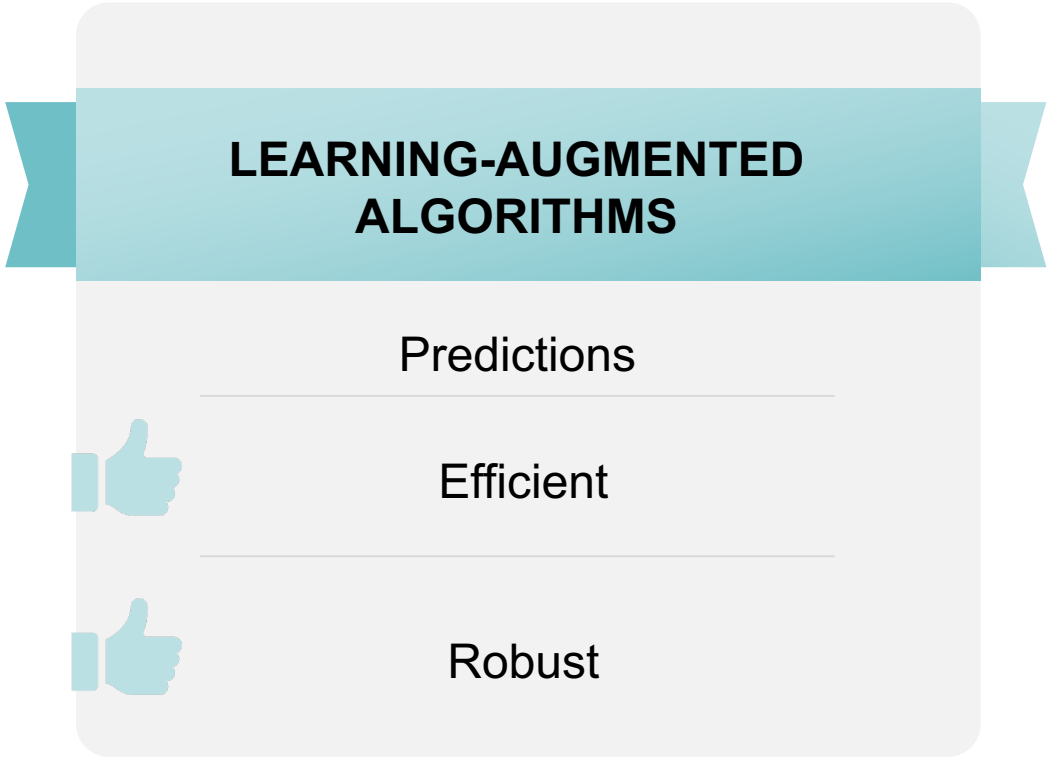


Learning-Augmented Algorithms



Can we get the best of both worlds?

Learning-Augmented Algorithms



Given some predictions

- Improve the performance guarantees when having good predictions
- Being robust against bad predictions

Learning-Augmented Algorithms

Online Algorithms

Input that has not yet arrived
Improve competitive ratio

Mechanism Design

Private data of agents
Improve approximation guarantees

Graph Algorithms

Optimum Solution
Improve running time

Distortion in Voting

Utilities of agents
Improve approximation guarantees

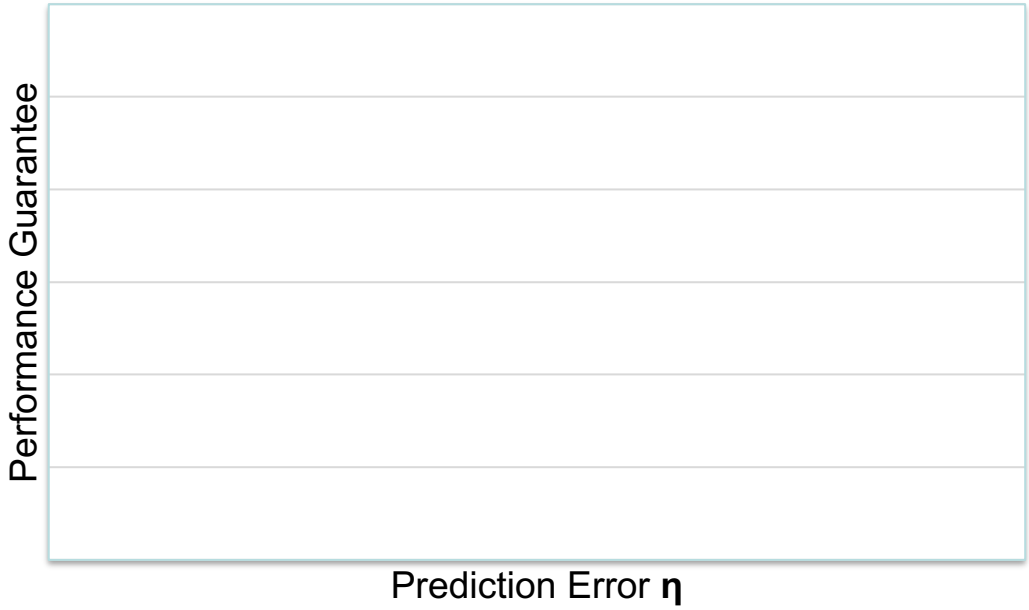
Data Structure, Sorting

(Offline) optimal solution
Improve running time

Learning-Augmented Algorithms

We do not know quality of the predictions

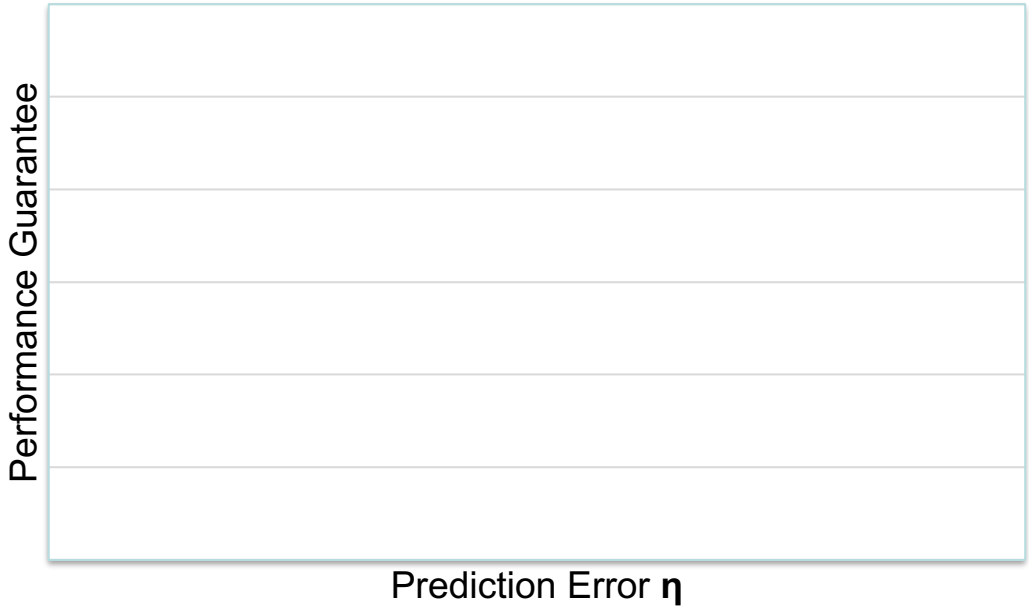
- Define prediction error **η**
- There are several ways to do so



Learning-Augmented Algorithms

Properties of learning-augmented algorithms

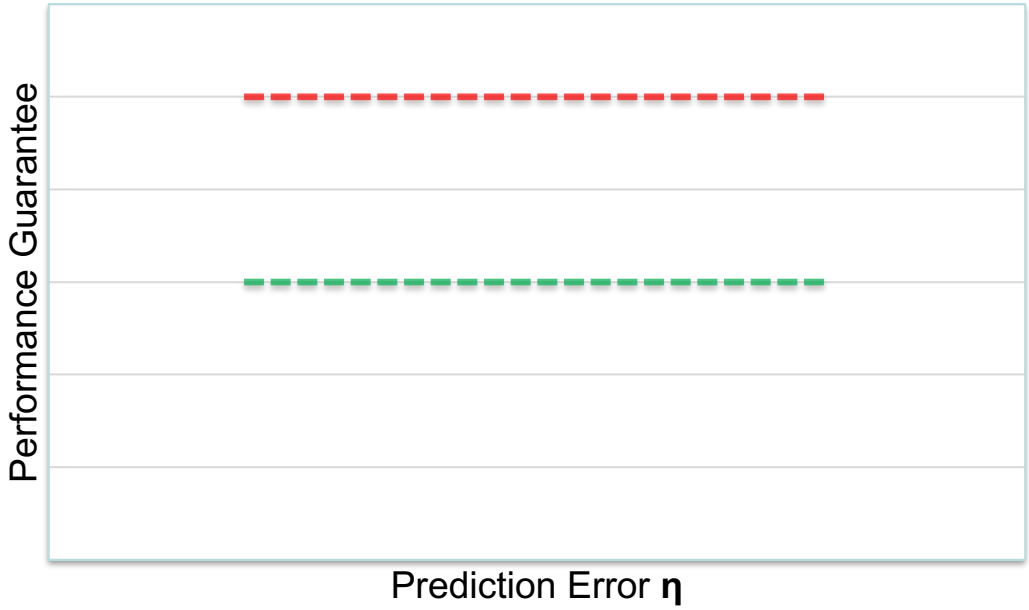
- Robustness: Losing only a constant factor of the algorithm's existing worst-case performance guarantee if **η is large**.



Learning-Augmented Algorithms

Properties of learning-augmented algorithms

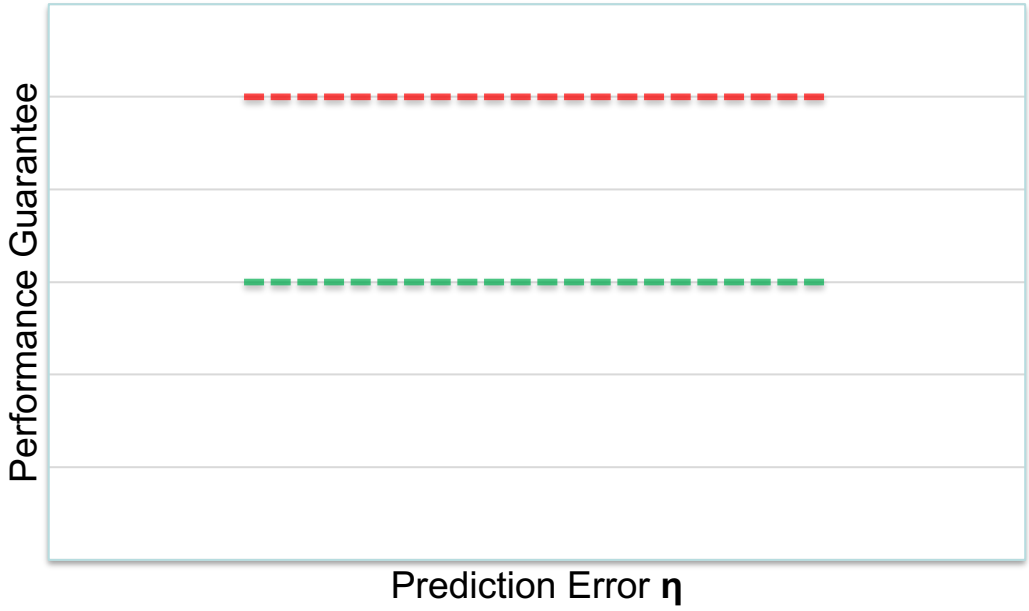
- Robustness: Losing only a constant factor of the algorithm's existing worst-case performance guarantee if **η is large**.



Learning-Augmented Algorithms

Properties of learning-augmented algorithms

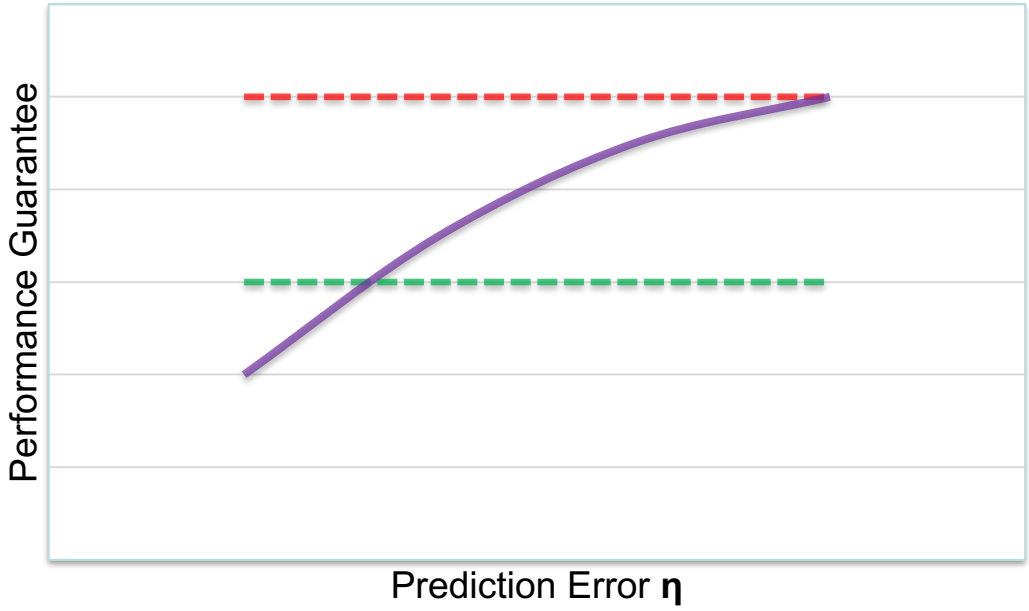
- Robustness: Losing only a constant factor of the algorithm's existing worst-case performance guarantee if **η is large**.
- Consistency: Improve the algorithm's performance guarantee if **η is zero**.
- Smoothness: The performance guarantee degrades gracefully with the quality of the predictions.



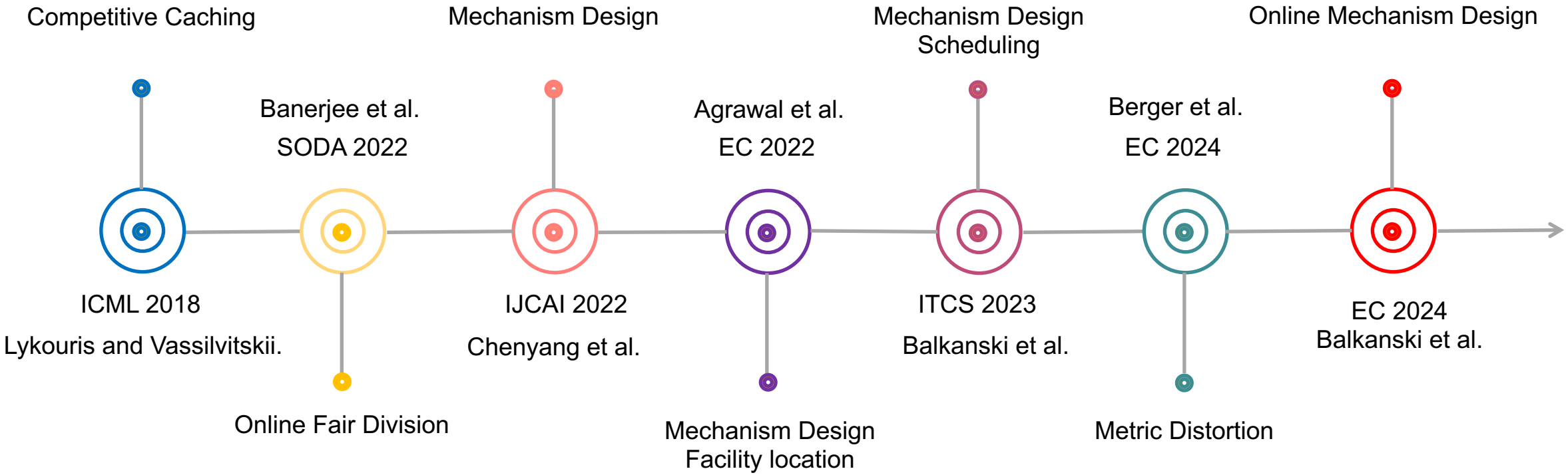
Learning-Augmented Algorithms

Properties of learning-augmented algorithms

- Robustness: Losing only a constant factor of the algorithm's existing worst-case performance guarantee if **η is large**.
- Consistency: Improve the algorithm's performance guarantee if **η is zero**.
- Smoothness: The performance guarantee degrades gracefully with the quality of the predictions.



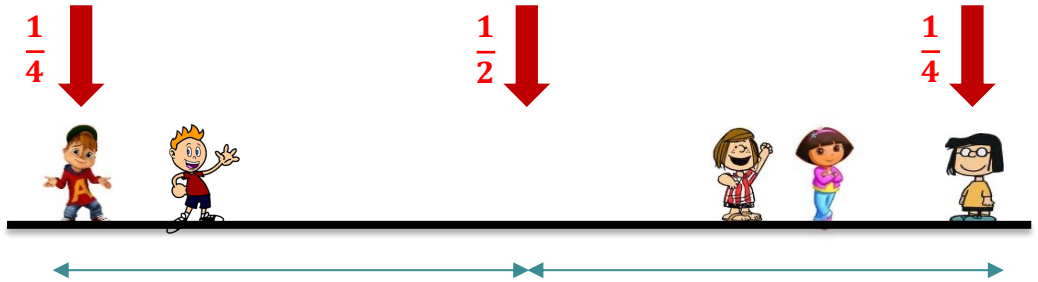
Related Work



Strategyproof Mechanisms

Can we improve the approximation factors using predictions?

What would be a reasonable notion of prediction?



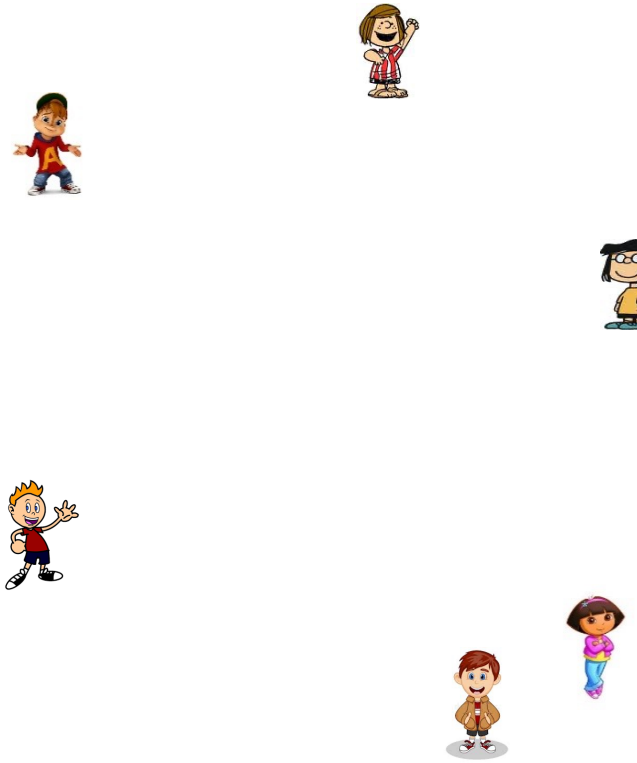
Learning-Augmented Mechanism Design

Strategyproof Mechanism:

- Agents have **private** information
- No agent has an incentive to misreport her data

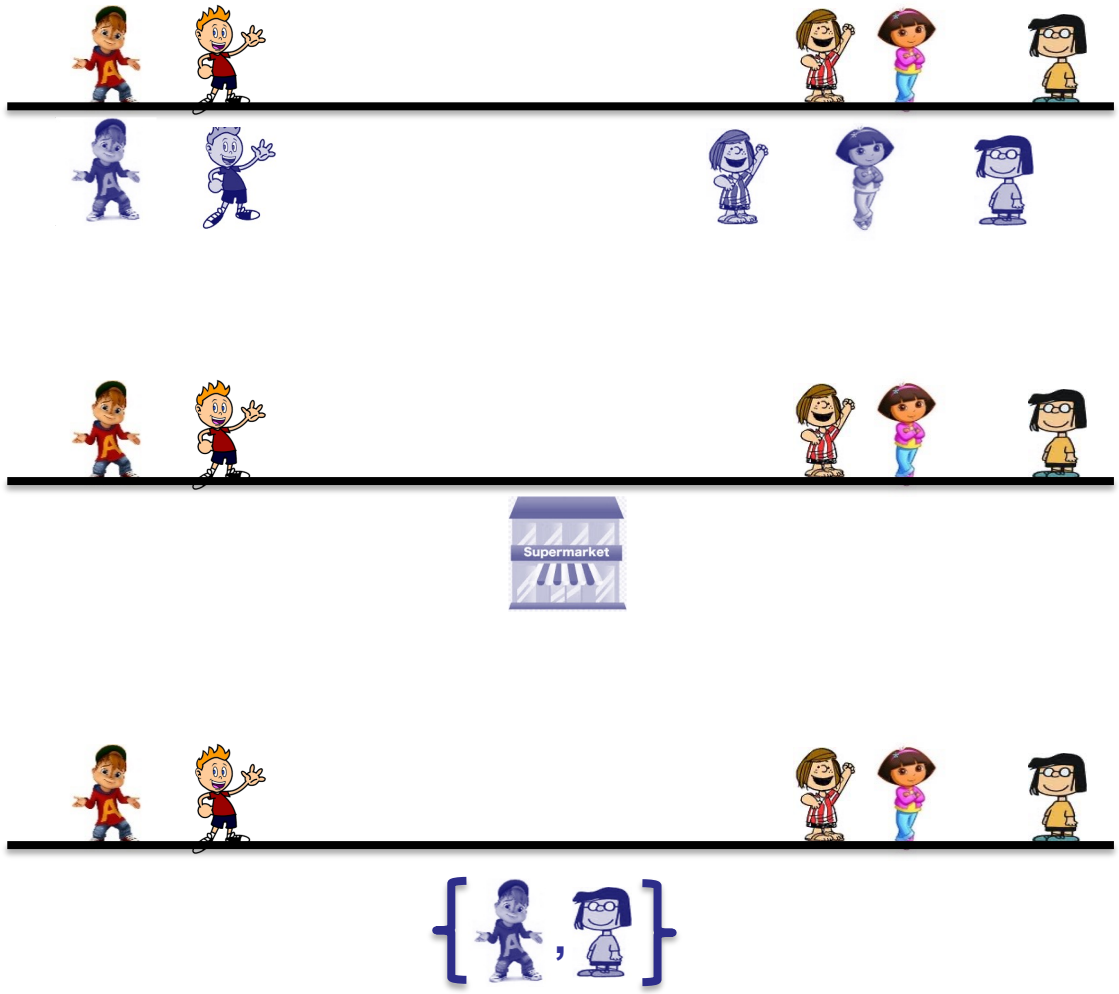
Goal:

- Improve approximation guarantees of strategyproof mechanisms using predictions



Prediction Settings

- Every agents' location and IDs
  - $X^* = \{x_1^*, \dots, x_n^*\}$
- Optimal facility location
  - F^*
- ID of extreme agents
  - $e^* = \{e_1^*, e_2^*\}$



Deterministic Mechanisms with F^* Prediction

MinMaxP

* — D

Input:

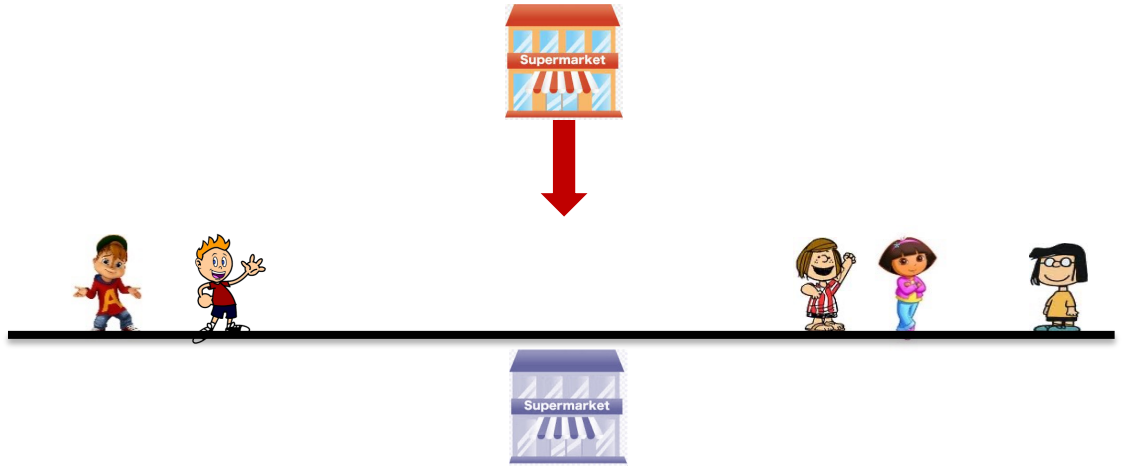
$$X = \{x_1, \dots, x_n\}$$

Output:

If $F^* \in \left[\min_{i \in N} x_i, \max_{i \in N} x_i\right]$: $f(X) = F^*$

If $F^* < \min_{i \in N} x_i$: $f(X) = \min_{i \in N} x_i$

If $F^* > \max_{i \in N} x_i$: $f(X) = \max_{i \in N} x_i$



Learning-Augmented Mechanism Design:
Leveraging Predictions for Facility Location

Agrawal et al. EC 2022

Deterministic Mechanisms with F^* Prediction

MinMaxP * – D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

If $F^* \in \left[\min_{i \in N} x_i, \max_{i \in N} x_i\right]$: $f(X) = F^*$

If $F^* < \min_{i \in N} x_i$: $f(X) = \min_{i \in N} x_i$

If $F^* > \max_{i \in N} x_i$: $f(X) = \max_{i \in N} x_i$

Learning-Augmented Mechanism Design:
Leveraging Predictions for Facility Location

[Agrawal et al., 2022]

Deterministic Mechanisms with F^* Prediction

MinMaxP * – D

Input:

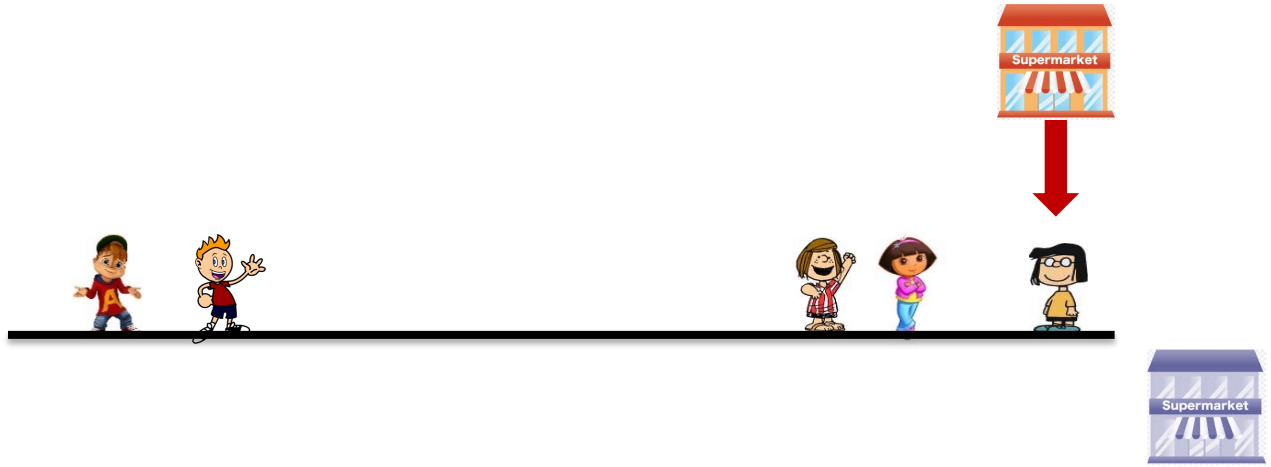
$$X = \{x_1, \dots, x_n\}$$

Output:

If $F^* \in \left[\min_{i \in N} x_i, \max_{i \in N} x_i\right]$: $f(X) = F^*$

If $F^* < \min_{i \in N} x_i$: $f(X) = \min_{i \in N} x_i$

If $F^* > \max_{i \in N} x_i$: $f(X) = \max_{i \in N} x_i$



Learning-Augmented Mechanism Design:
Leveraging Predictions for Facility Location

Agrawal et al. EC 2022

Deterministic Mechanisms with F^* Prediction

MinMaxP — * — 1 — D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

If $F^* \in \left[\min_{i \in N} x_i, \max_{i \in N} x_i\right]$: $f(X) = F^*$

If $F^* < \min_{i \in N} x_i$: $f(X) = \min_{i \in N} x_i$

If $F^* > \max_{i \in N} x_i$: $f(X) = \max_{i \in N} x_i$

1 consistent
2 robust

MinBoundingBox — * — □ — D

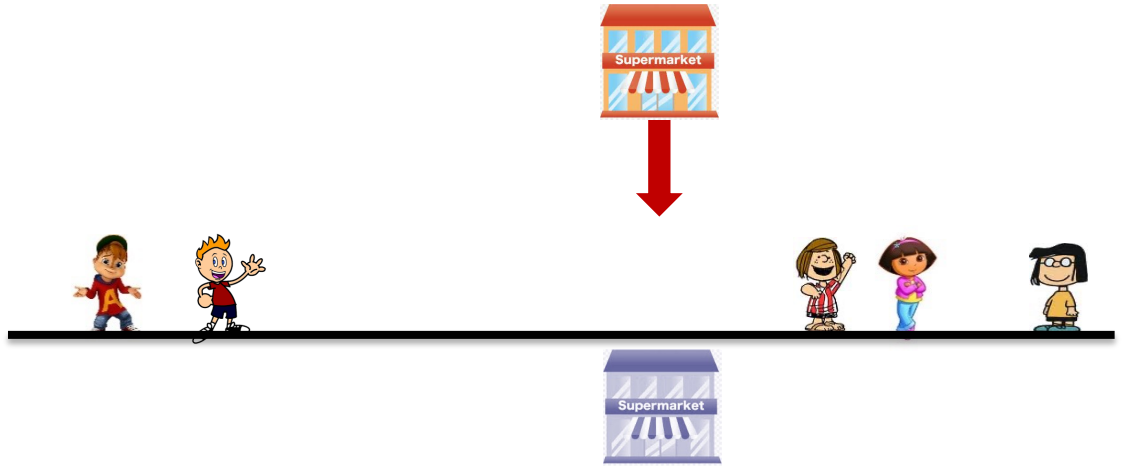
Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

MinMaxP on each axis

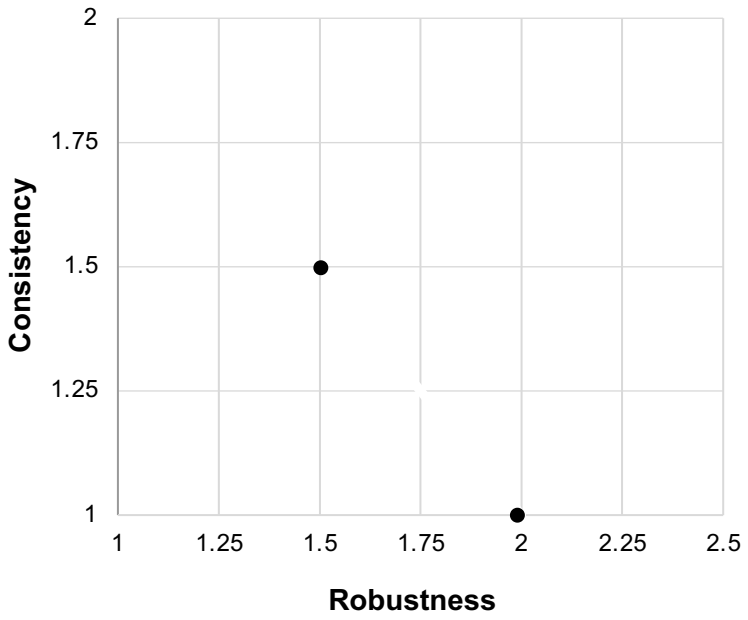
1 consistent
$1 + \sqrt{2}$ robust



Learning-Augmented Mechanism Design:
Leveraging Predictions for Facility Location

Agrawal et al. EC 2022

Randomized Strategic Facility Location with Predictions on the Line

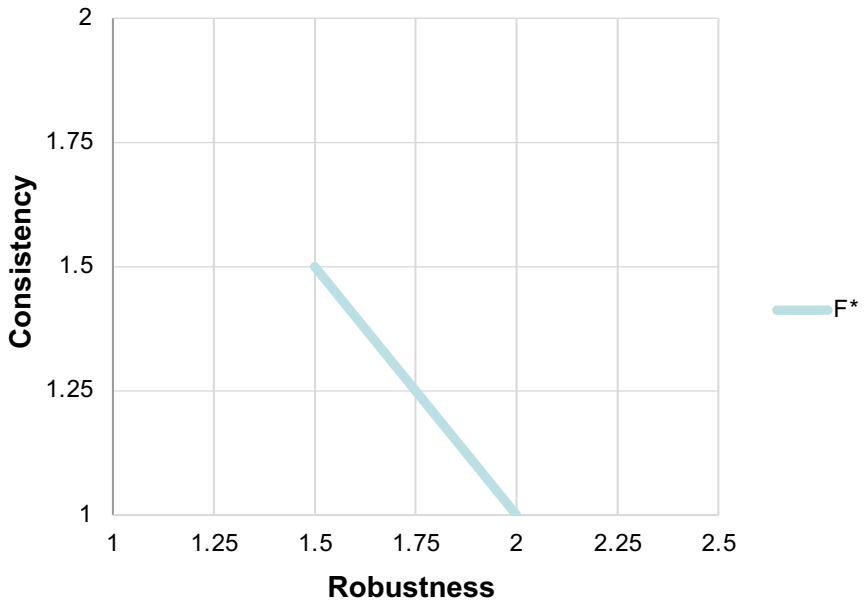


- Consistency-Robustness tradeoff

Given a prediction F^* (the optimal facility location):

- MinMaxP: 1 consistent, 2 robust [Agrawal et al., EC'22]
- LRM: 1.5 consistent, 1.5 robust

Randomized Mechanisms with Predictions - Line



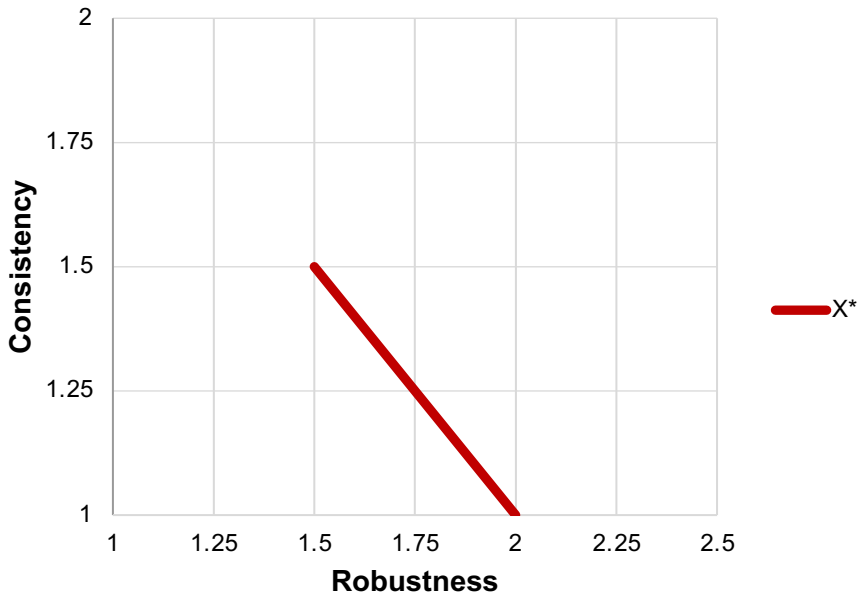
Given a prediction F^* (the optimal facility location):

Proposition – Upper bound

For any $\delta \in [0, 0.5]$, there exists a randomized strategyproof mechanism that is

- $1 + \delta$-consistent, and
- $2 - \delta$-robust.

Randomized Mechanisms with Predictions - Line

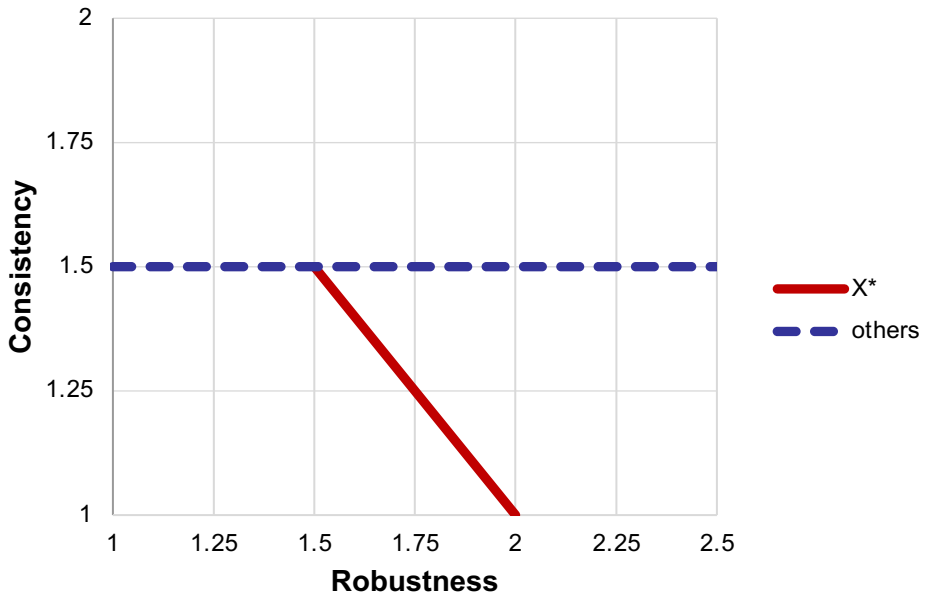


Given a prediction X^* (full predictions):

Theorem – Lower bound

No strategyproof mechanism that is $1 + \delta$-consistent for some $\delta \in [0, 0.5]$, can also guarantee robustness better than $2 - \delta$.

Randomized Mechanisms with Predictions - Line



Given other predictions X^*_{-1} (missing one location):

Theorem – Lower bound

There is no randomized strategyproof mechanism that is better than 1.5-consistent.

Randomized Strategic Facility Location with Predictions on the Euclidean Plane

OPT

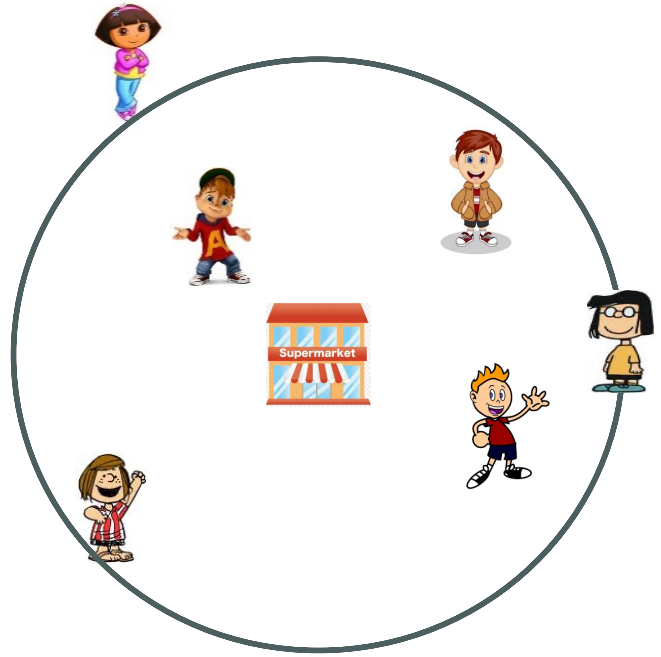
D

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

$$f(X) = center\ of\ the\ minimum\ bounding\ circle$$



- Random Dictatorship: 2 – apx
- Centroid Mechanism: $2 - \frac{1}{n}$ - apx

Randomized Mechanisms - Euclidean Plane

LRM

R

Input:

$$X = \{x_1, \dots, x_n\}$$

Output:

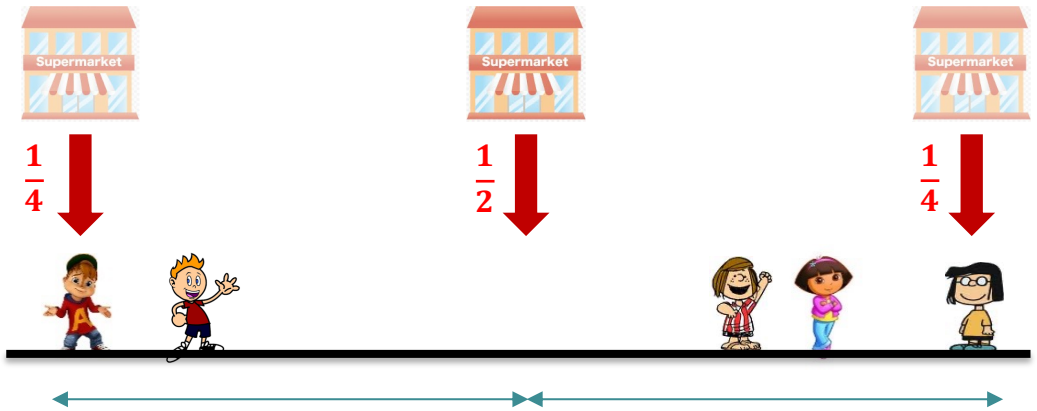
w.p. $\frac{1}{4}$: $f(X) = \min_{i \in N} x_i$

w.p. $\frac{1}{2}$: $f(X) = mid\ x_i$

w.p. $\frac{1}{4}$: $f(X) = \max_{i \in N} x_i$

✓ **1.5-apx**

✓ **tight**



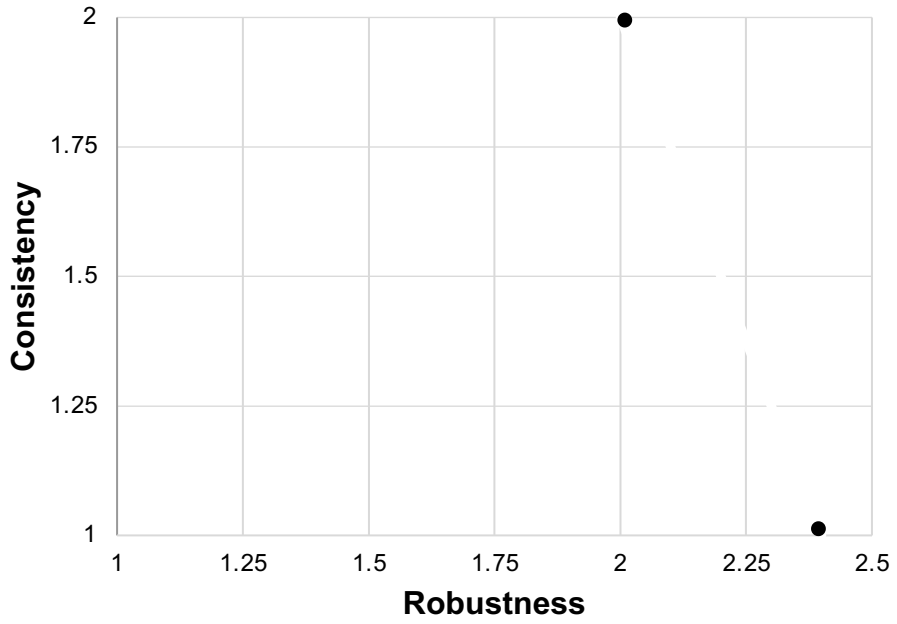
- Lower bound of 1.5 on the line does **not** hold!

Theorem – Lower bound

Any randomized strategyproof mechanism in the Euclidean metric space has an approximation ratio of at least $\mathbf{1.118}$.

- The gap of $(\mathbf{1.118}, 2 - \frac{1}{n})$ remains open!

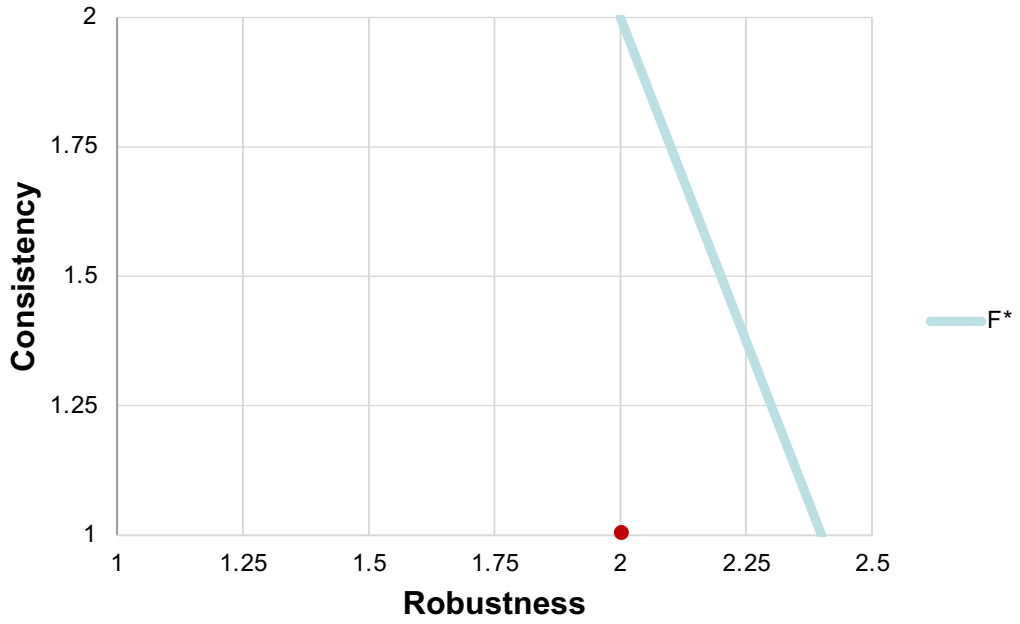
Randomized Mechanisms with Predictions - Euclidean Plane



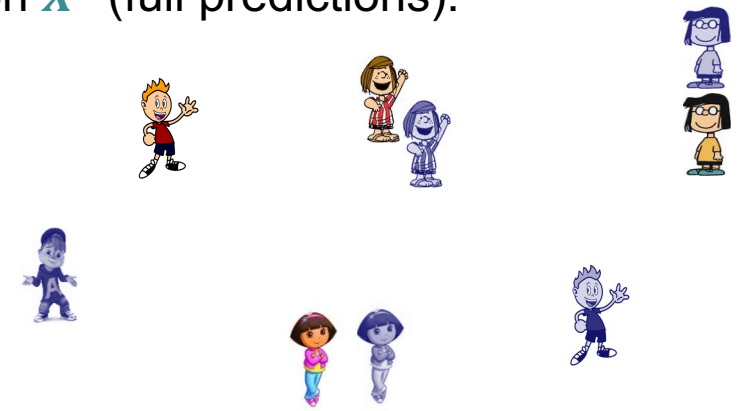
Given a prediction F^* (the optimal facility location):

- MinBoundingBox: 1 consistent, $1+\sqrt{2}$ robust
- Random Dictatorship: 2 consistent, 2 robust

Randomized Mechanisms with Predictions - Euclidean Plane



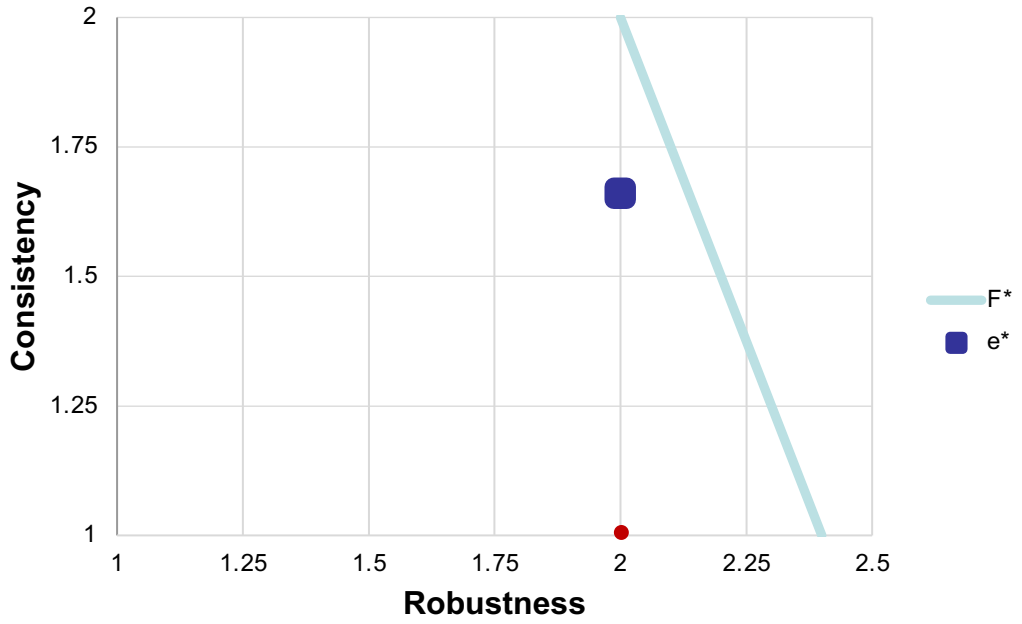
Given a prediction X^* (full predictions):



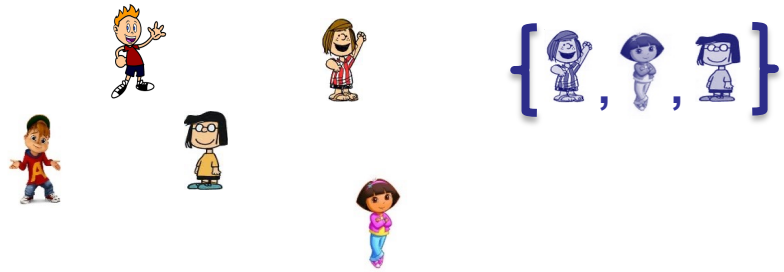
Theorem – Lower bound

No strategyproof mechanism that is 1-consistent can also guarantee robustness better than 2.

Randomized Mechanisms with Predictions - Euclidean Plane



Given a prediction e^* (IDs of extreme agents predictions):

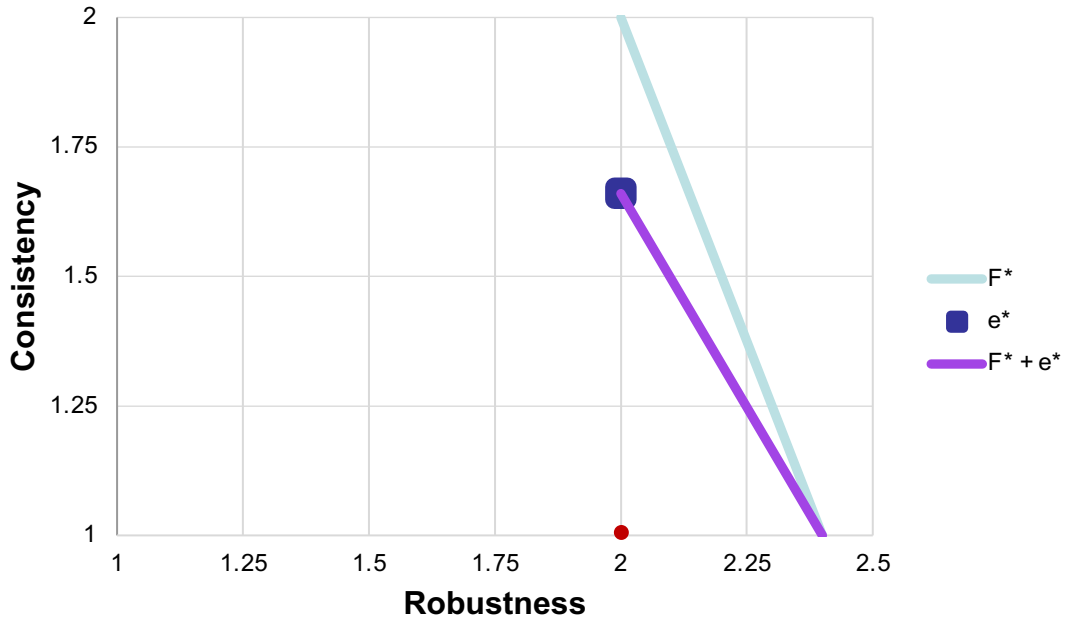


Mechanism – Centroid

There exists a randomized strategyproof mechanism that for any number of agents achieves

- 1.67 consistency, and
- 2 robustness.

Randomized Mechanisms with Predictions - Euclidean Plane



Given a prediction e^* (IDs of extreme agents predictions) and a prediction F^* (the optimal facility location):

Randomized Mechanisms with Predictions - Euclidean Plane

Centroid

Input:

$$X = \{x_1, \dots, x_n\}, e^* = \{e_1^*, e_2^*, e_3^*\}$$

Output:

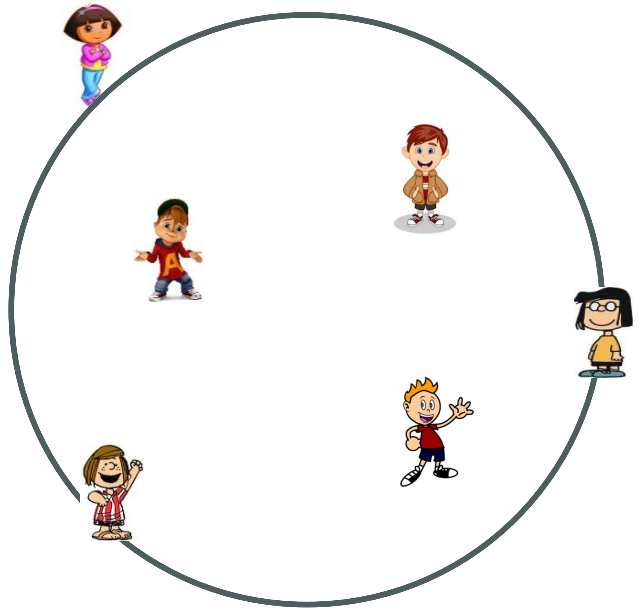
w.p. $\frac{1}{2}$: $f(X) = (x_{e_1^*} + x_{e_2^*} + x_{e_3^*})/3$

w.p. $\frac{1}{6}$: $f(X) = x_{e_1^*}$

w.p. $\frac{1}{6}$: $f(X) = x_{e_2^*}$

w.p. $\frac{1}{6}$: $f(X) = x_{e_3^*}$

1.67 consistent
2 robust



Randomized Mechanisms with Predictions - Euclidean Plane

Centroid

Input:

$$X = \{x_1, \dots, x_n\}, e^* = \{e_1^*, e_2^*, e_3^*\}$$

Output:

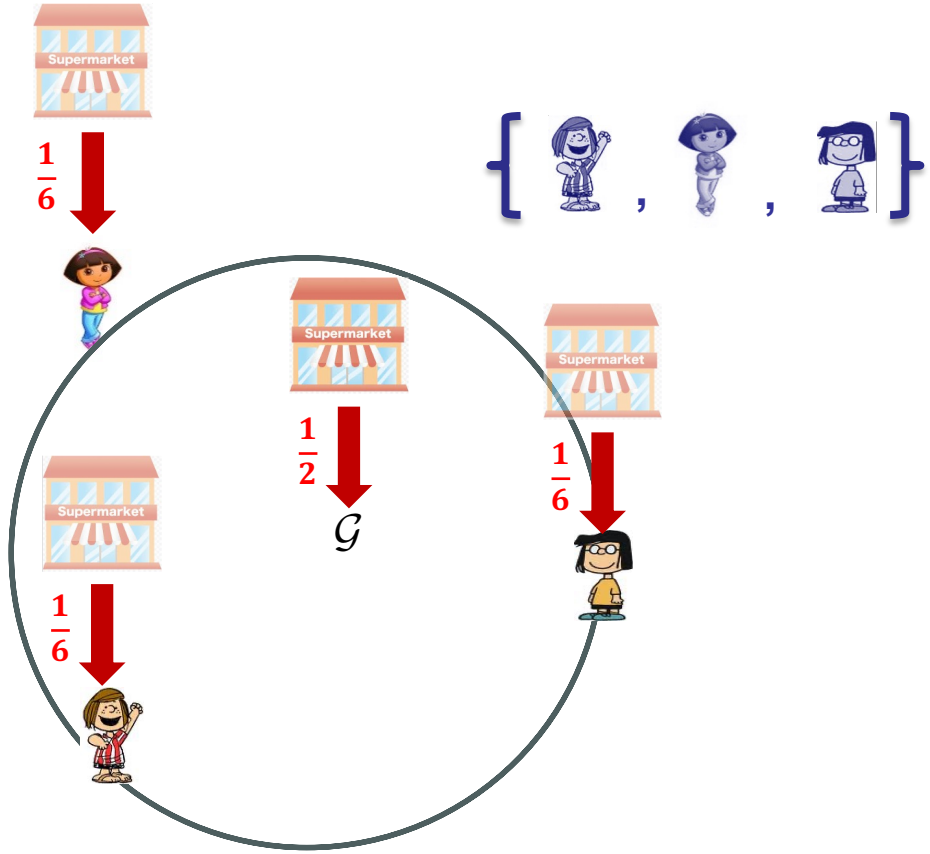
w.p. $\frac{1}{2}$: $f(X) = (x_{e_1^*} + x_{e_2^*} + x_{e_3^*})/3$

w.p. $\frac{1}{6}$: $f(X) = x_{e_1^*}$

w.p. $\frac{1}{6}$: $f(X) = x_{e_2^*}$

w.p. $\frac{1}{6}$: $f(X) = x_{e_3^*}$

1.67 consistent
2 robust



Randomized Mechanisms with Predictions

Centroid

Input:

$$X = \{x_1, \dots, x_n\}, e^* = \{e_1^*, e_2^*, e_3^*\}$$

Output:

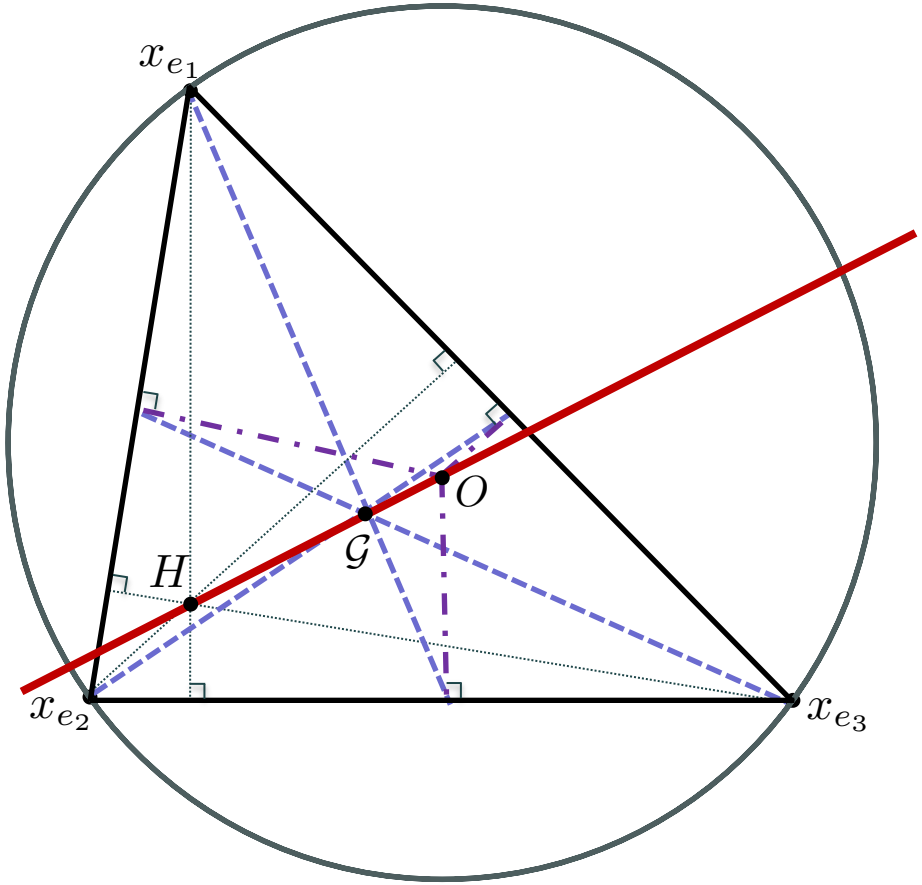
w.p. $\frac{1}{2}$: $f(X) = \mathcal{G}$

w.p. $\frac{1}{6}$: $f(X) = x_{e_1^*}$

w.p. $\frac{1}{6}$: $f(X) = x_{e_2^*}$

w.p. $\frac{1}{6}$: $f(X) = x_{e_3^*}$

1.67 consistent
2 robust



Thank you!

Randomized Strategic Facility Location with Predictions [Balkanski et al., 2024]