Search Games with Predictions

Spyros Angelopoulos

From matchings to markets. A tale of Mathematics, Economics and Computer Science

CIRM, Marseille, Dec 11-15 2023

Parc des Buttes-Chaumont
0 m 100
Place A.-Carrel
rue Manin
rue de Crimée
Porte de Crimée
St-Pierre
Pavillon du lac
Chemin des Aiguilles
Temple de Vesta
Fondation ophtalmologique A.-de-Rotschild
Porte Secrétan
avenue Secrétan
avenue M.-Moreau
Pont des suicides
Grotte
Carrefour de la Colonne
Botzaris
Pavillon Puebla
BUTTES-CHAUMONT
Porte Fessart
Salon Weber
Cascade
Porte de la Villette
BOTZARIS
rue Botzaris
rue du Gl-Brunet
avenue Simon-Bolivar
rue des Dunes
rue Fessart
rue du Plateau
passage du Plateau
rue des Alouettes
Villa des Buttes-Chaumont
rue du Tunnel
rue Hassard
rue Compans
1
2
3
4
5
6
7
8
9
10
11

Quick summary of the presentation

- The power and limitations of **imperfect advice** in search games
- Motivated by recent advances on *algorithms with ML predictions / untrusted advice*
- Focus mainly on pure (deterministic) strategies

… but I will also touch on “real” search games with mixed strategies

Based on the following works:

Online Search with a Hint (*Information and Computation* 2023)

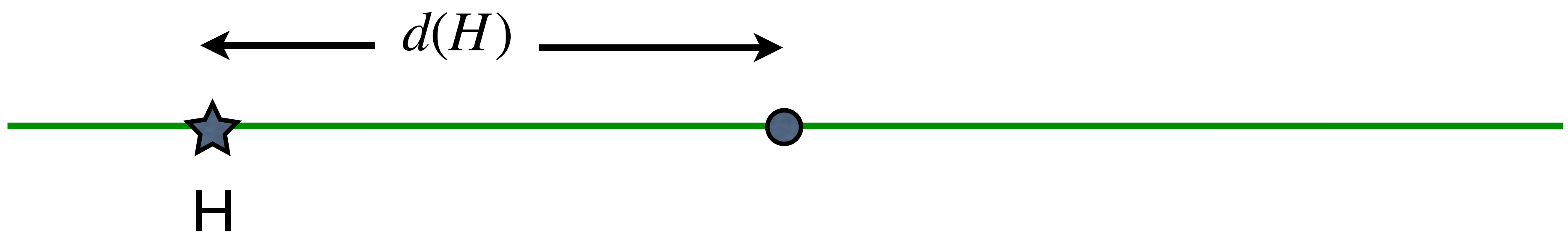
Competitive Search in the Line and the Star with Predictions (MFCS 2023)

Search Games with Predictions (ongoing work with T. Lidbetter and K. Panagiotou)

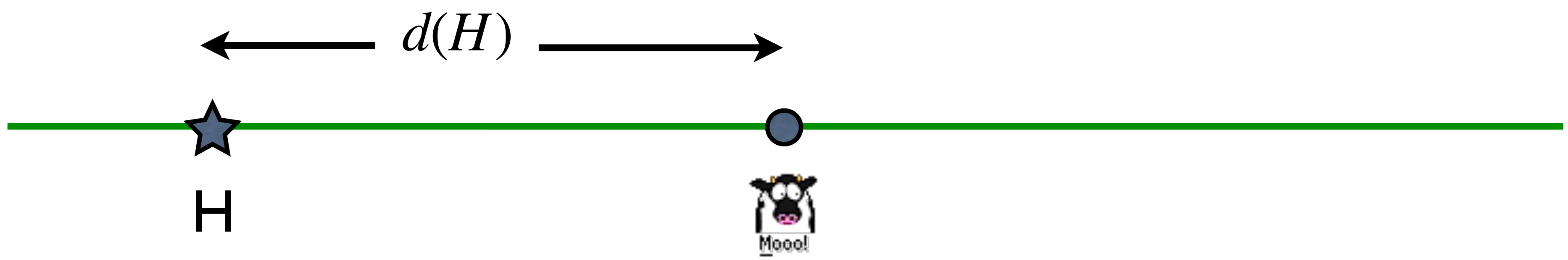
Case study: searching on the line (aka *cow-path* problem)

Case study: searching on the line (aka *cow-path* problem)

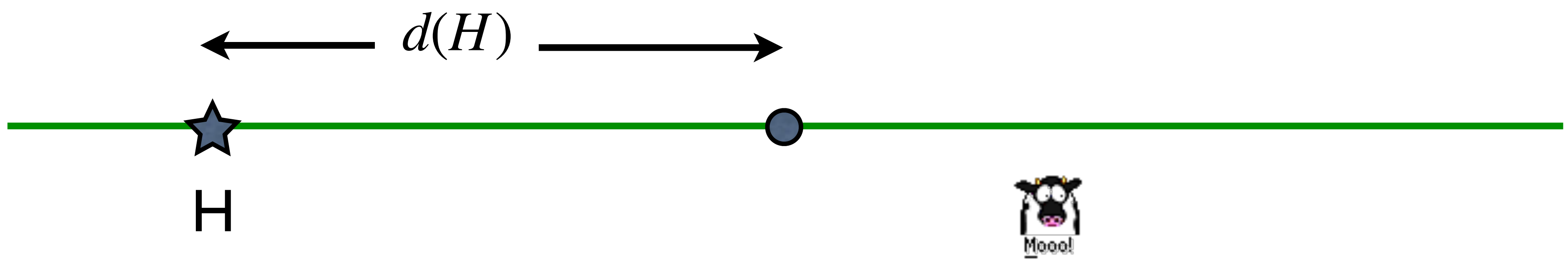
Case study: searching on the line (aka *cow-path* problem)



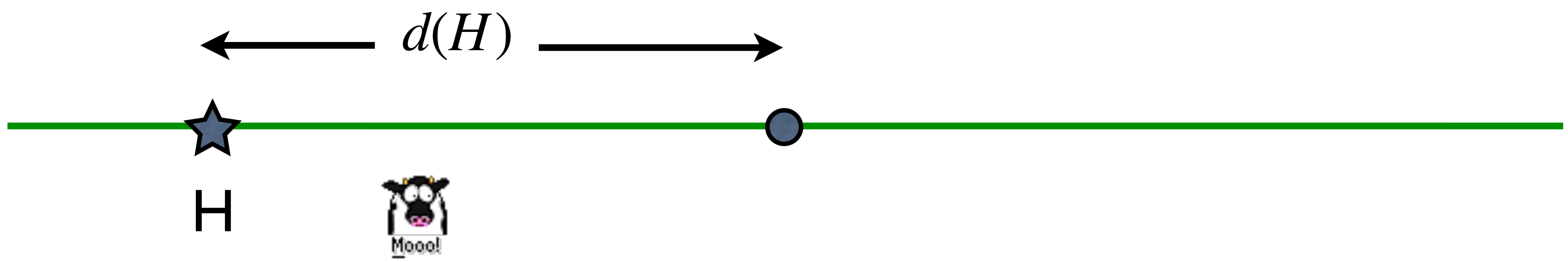
Case study: searching on the line (aka *cow-path* problem)



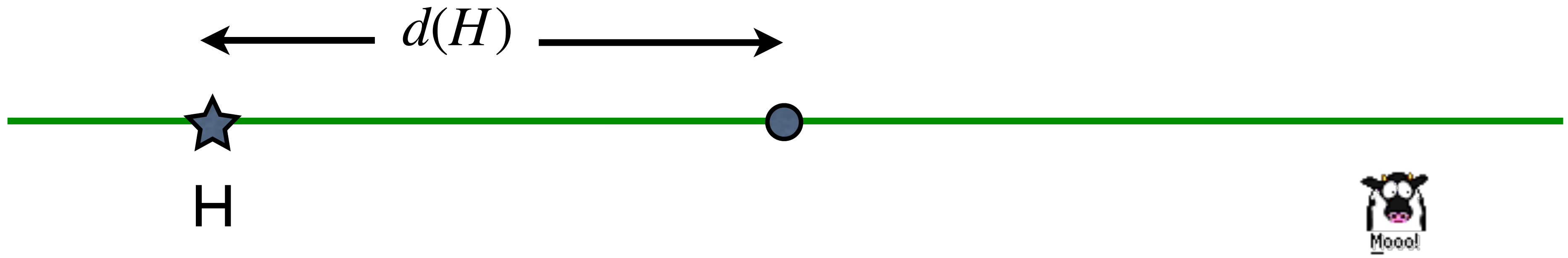
Case study: searching on the line (aka *cow-path* problem)



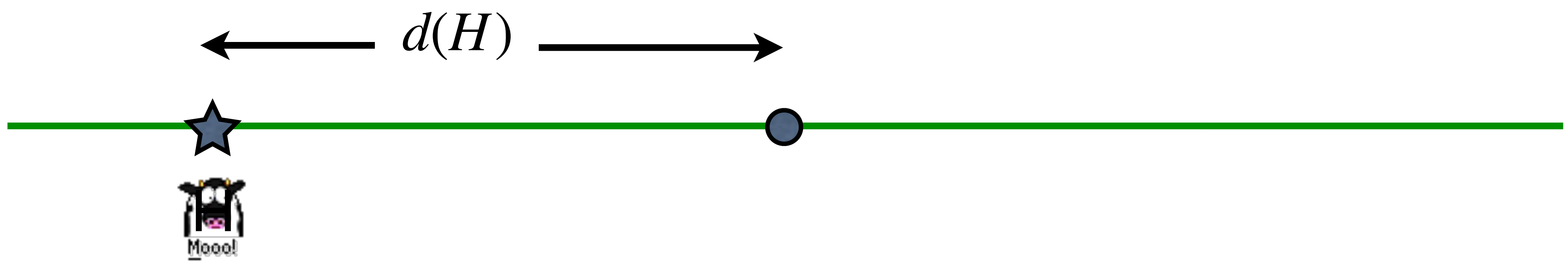
Case study: searching on the line (aka *cow-path* problem)



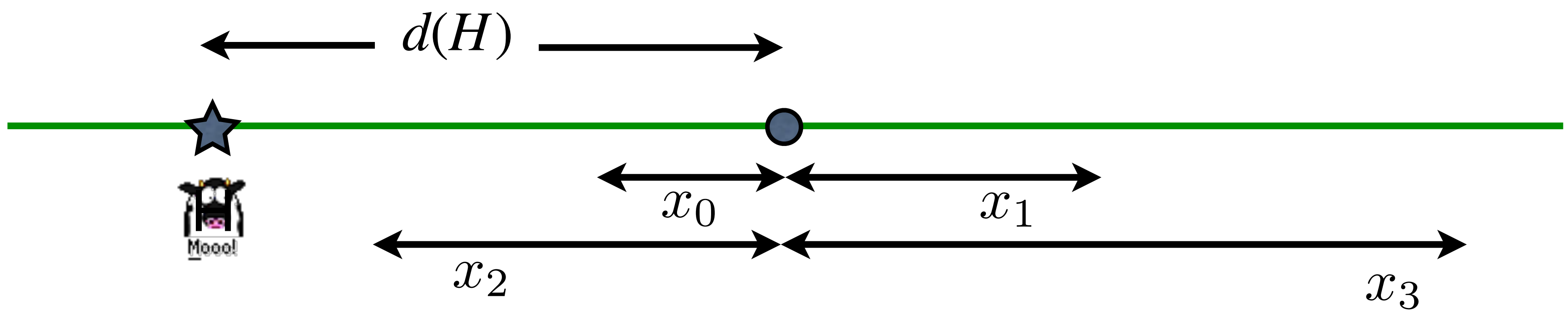
Case study: searching on the line (aka *cow-path* problem)



Case study: searching on the line (aka *cow-path* problem)

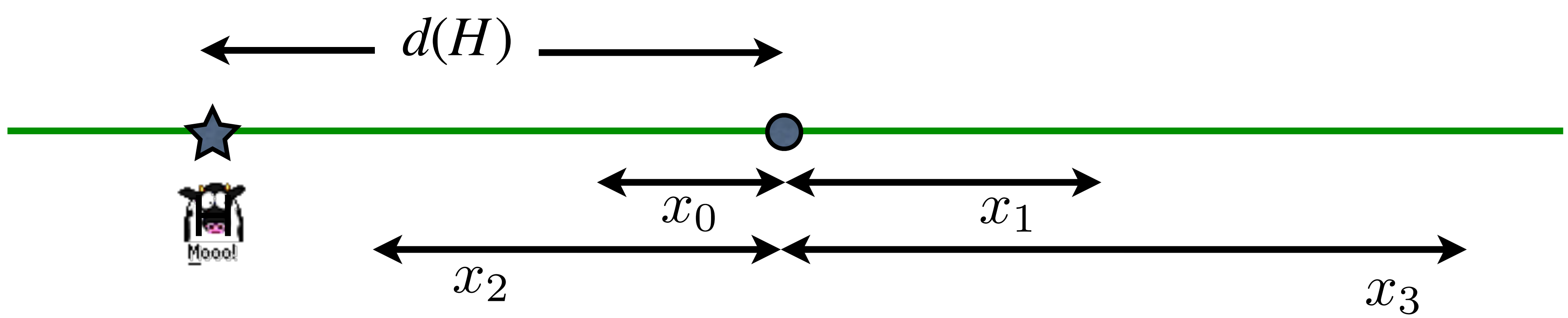


Case study: searching on the line (aka *cow-path* problem)



$$\mathsf{cr}(S) = \sup_H \frac{\text{distance traversed by the searcher using } S}{d(H)}$$

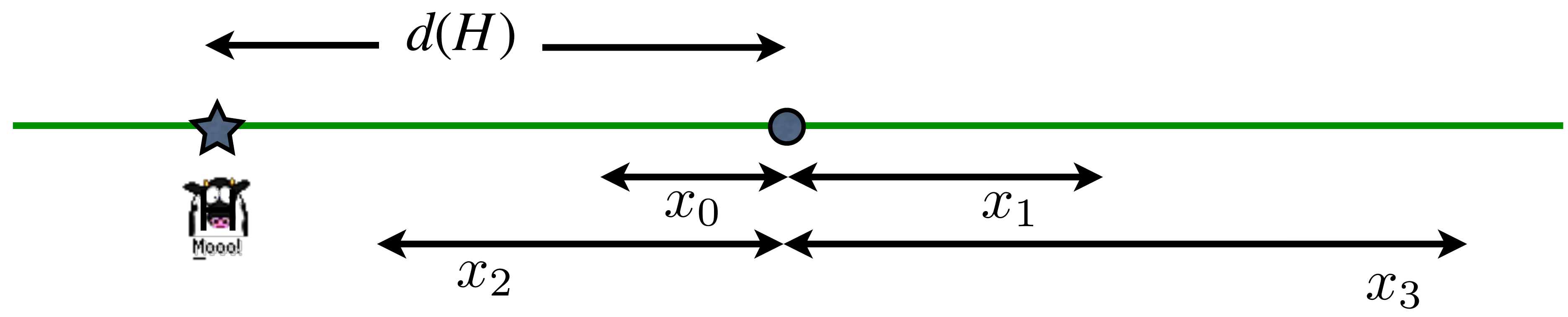
Case study: searching on the line (aka *cow-path* problem)



$$\mathrm{cr}(S) = \sup_H \frac{\text{distance traversed by the searcher using } S}{d(H)}$$

	Optimal competitive ratio	Techniques
Pure strategies	9 [Beck and Newman 1970]	Doubling + Gal's theorem
Mixed strategies	$1 + \min_{a>1} \frac{a}{\ln a} \approx 4.59$ [Gal 1980]	Randomized doubling + Minimax theorem

Case study: searching on the line (aka *cow-path* problem)



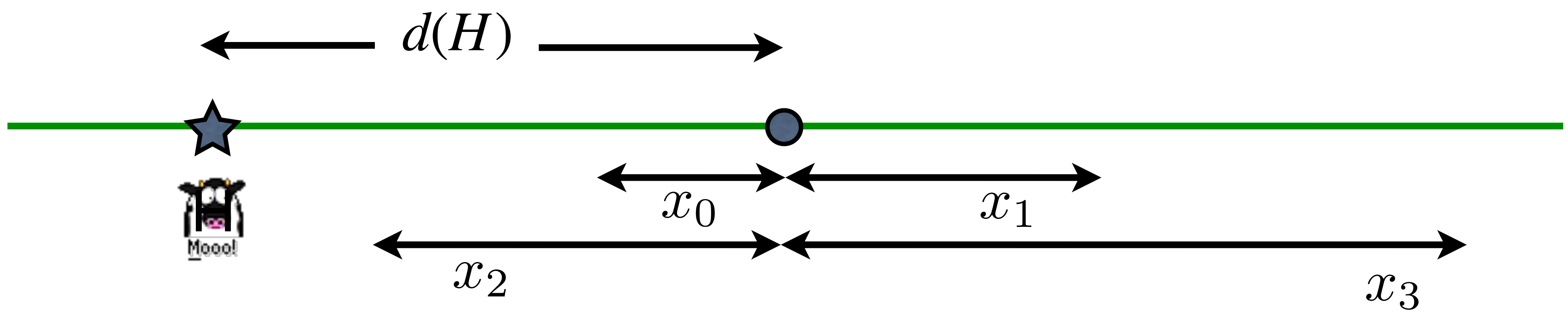
$$\text{cr}(S) = \sup_H \frac{\text{distance traversed by the searcher using } S}{d(H)}$$

	Optimal competitive ratio	
Pure strategies	9 [Beck and Newman 1970]	+ Gal’s theorem
Mixed strategies	$1 + \min_{a>1} \frac{a}{\ln a} \approx 4.59$ [Gal 1980]	Randomized doubling + Minimax theorem

$$\sup_i \frac{\sum_{j=0}^{i} x_j}{x_{i-1}} \geq \sup_i \frac{\sum_{j=0}^{i} \alpha_S^j}{\alpha_S^{i-1}}$$

$$\alpha_S = \limsup_{n \to \infty} x_n^{1/n}$$

Case study: searching on the line (aka *cow-path* problem)



$$\mathsf{cr}(S) = \sup_H \frac{\text{distance traversed by the searcher using } S}{d(H)}$$

	Optimal competitive ratio	Techniques
Pure strategies	9 [Beck and Newman 1970]	Doubling + Gal's theorem
Mixed strategies	$1 + \min_{a>1} \frac{a}{\ln a} \approx 4.59$ [Gal 1980]	Randomized doubling + Minimax theorem

Hints in search

Hints in search

Hint h: some information that is given to the searcher

The search strategy $S(h)$ is now a function of the hint

- If hint is trusted, then it is guaranteed to be correct
- If hint is untrusted, then it is generated adversarially

Hints in search

Hint h: some information that is given to the searcher

The search strategy $S(h)$ is now a function of the hint

- If hint is trusted, then it is guaranteed to be correct
- If hint is untrusted, then it is generated adversarially

$$\text{Competitiveness of S(h)} = (c_{S,h}, r_{S,h})$$

Hints in search

Hint h: some information that is given to the searcher

The search strategy $S(h)$ is now a function of the hint

- If hint is trusted, then it is guaranteed to be correct
- If hint is untrusted, then it is generated adversarially

Consistency: c.r. if hint is correct

$$\text{Competitiveness of S(h)} = (c_{S,h}, r_{S,h})$$

Hints in search

Hint h: some information that is given to the searcher

The search strategy $S(h)$ is now a function of the hint

- If hint is trusted, then it is guaranteed to be correct
- If hint is untrusted, then it is generated adversarially

Consistency: c.r. if hint is correct

$$\text{Competitiveness of S(h)} = (c_{S,h}, r_{S,h})$$

Robustness : c.r. if hint is adversarial

Hints in search

Hint h: some information that is given to the searcher

The search strategy $S(h)$ is now a function of the hint

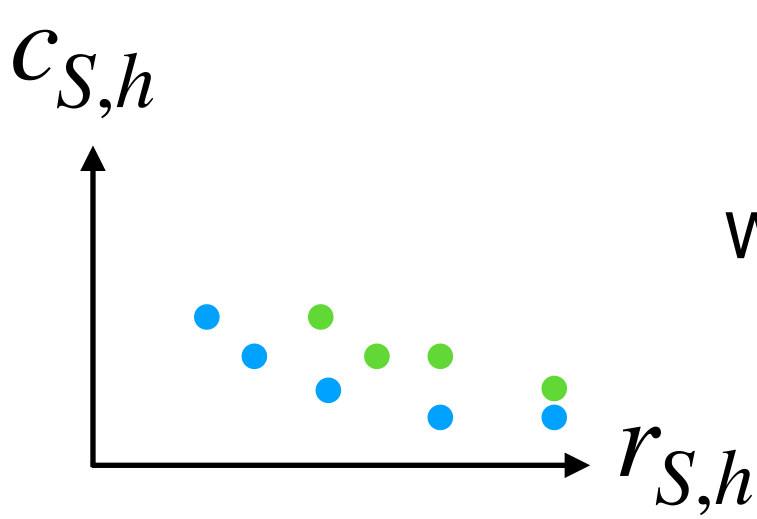
- If hint is trusted, then it is guaranteed to be correct
- If hint is untrusted, then it is generated adversarially

Consistency: c.r. if hint is correct

$$\text{Competitiveness of S(h)} = (c_{S,h}, r_{S,h})$$

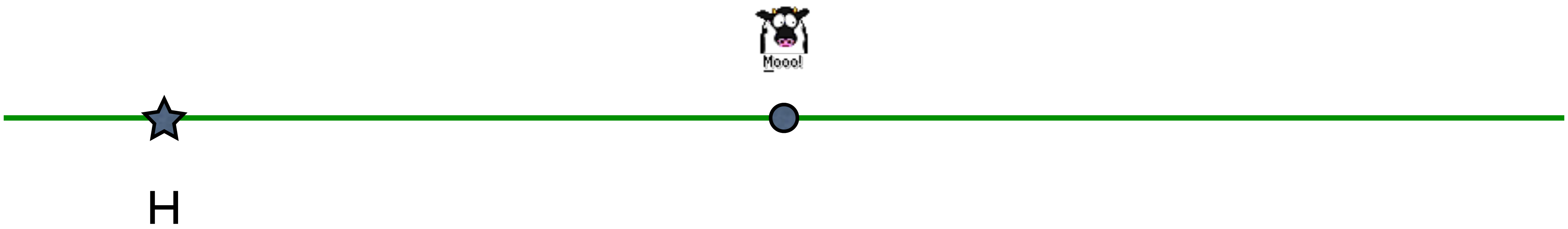
Robustness : c.r. if hint is adversarial

Pareto efficiency



What is the best consistency if we want the strategy to be $r-$robust?

An example



Suppose **hint= LEFT / RIGHT,** and that the searcher **blindly follows the hint**

This strategy is $(1,\infty)$ competitive

What are the other points on the Pareto frontier?

Some additional motivation: computing with ML predictions

[Lykouris and Vassilvitskii 2018]

Some additional motivation: computing with ML predictions

[Lykouris and Vassilvitskii 2018]

Access to a **prediction** associated with the input which is inherently **erroneous**

The prediction has error η (unknown to the algorithm)

Some additional motivation: computing with ML predictions

[Lykouris and Vassilvitskii 2018]

Access to a **prediction** associated with the input which is inherently **erroneous**

The prediction has error η (unknown to the algorithm)

Some additional motivation: computing with ML predictions

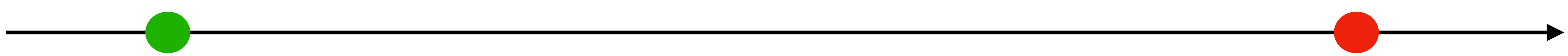
[Lykouris and Vassilvitskii 2018]

Access to a **prediction** associated with the input which is inherently **erroneous**

The prediction has error η (unknown to the algorithm)

Robustness : competitive ratio

with *adversarial* error



Consistency : competitive ratio

with *no* error

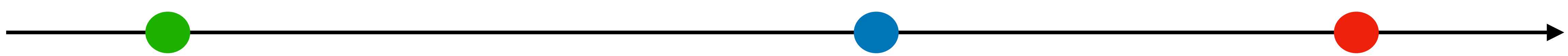
Some additional motivation: computing with ML predictions

[Lykouris and Vassilvitskii 2018]

Access to a **prediction** associated with the input which is inherently **erroneous**

The prediction has error η (unknown to the algorithm)

Robustness : competitive ratio with *adversarial* error



Consistency : competitive ratio with *no* error

competitive ratio with error η

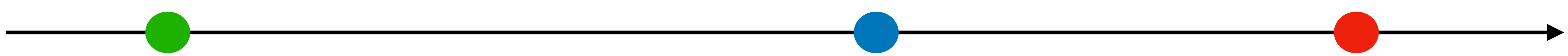
Some additional motivation: computing with ML predictions

[Lykouris and Vassilvitskii 2018]

Access to a **prediction** associated with the input which is inherently **erroneous**

The prediction has error η (unknown to the algorithm)

Robustness : competitive ratio with *adversarial* error



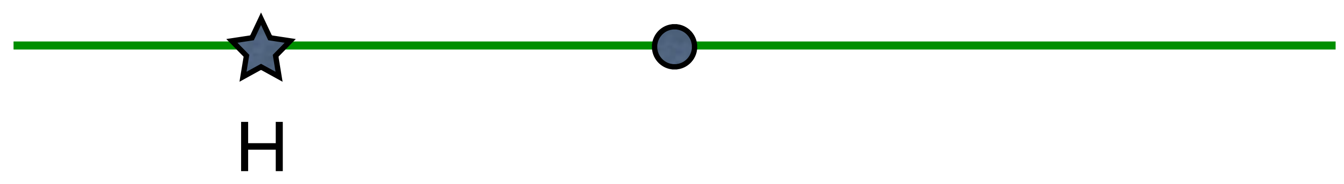
Consistency : competitive ratio with *no* error

competitive ratio with error η

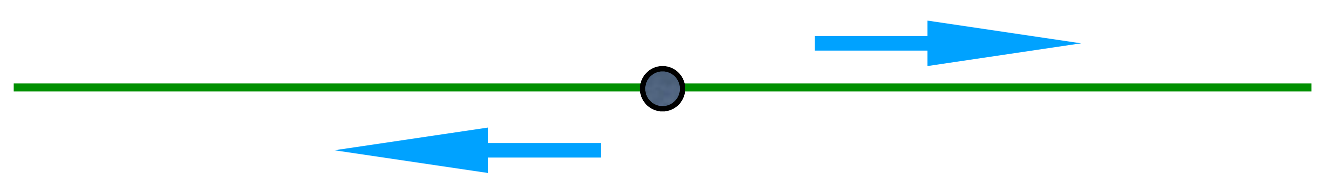
Many studies of online problems under this learning-enhanced framework

Types of hints

The hint is the exact **position** of the target



The hint is the **direction** of the search (left or right)



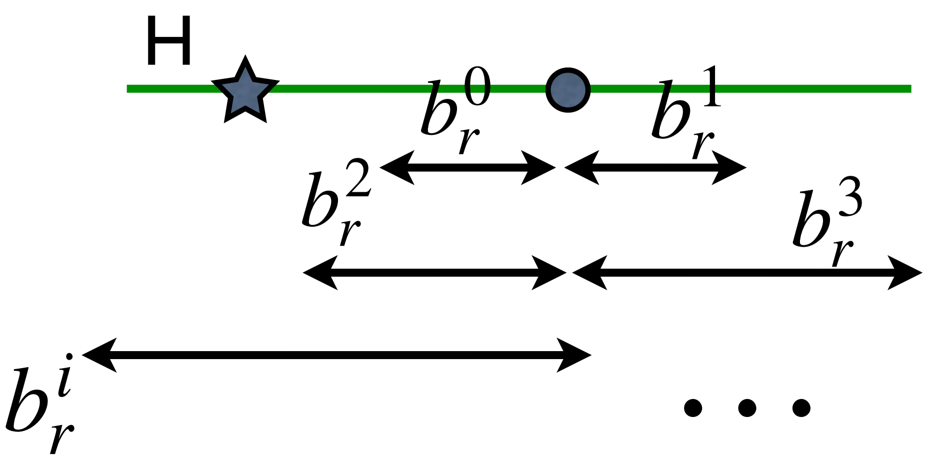
The hint is a **k-bit string**

01101...1

Positional and directional hints: summary of results

Positional hint

Upper bound



Lower bound

Properties of

$$\sum_{i=0}^{n} x_{i-1} \leq r x_i$$

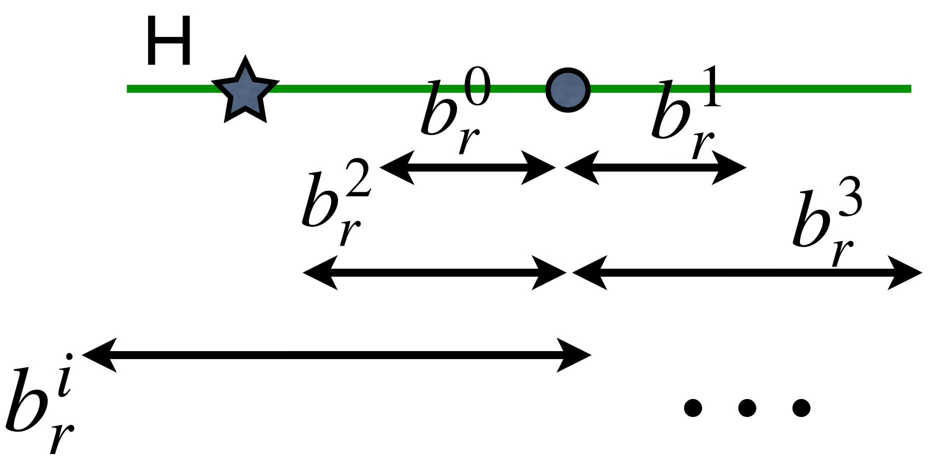
Result

$$\left(\frac{b_r+1}{b_r-1}, r\right)$$

Positional and directional hints: summary of results

Positional hint $b_r =$ largest base that guarantees $r-$robustness

Upper bound



Lower bound

Properties of

$$\sum_{i=0}^{n} x_{i-1} \leq r x_i$$

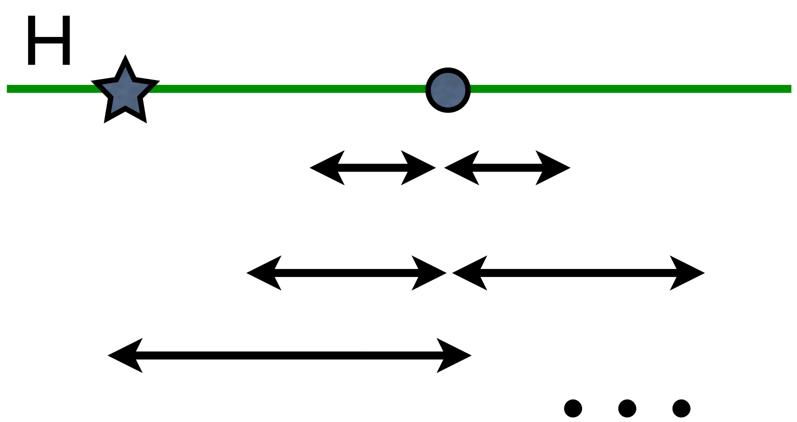
Result

$$\left(\frac{b_r + 1}{b_r - 1}, r\right)$$

Positional and directional hints: summary of results

Positional hint $b_r =$ largest base that guarantees $r-$robustness

Upper bound



Lower bound

Properties of

$$\sum_{i=0}^{n} x_{i-1} \leq r x_i$$

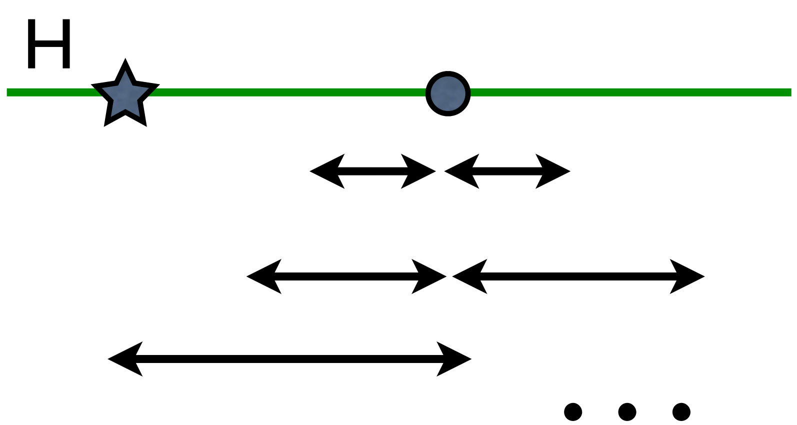
Result

$$\left(\frac{b_r+1}{b_r-1}, r\right)$$

Positional and directional hints: summary of results

Positional hint

Upper bound



Lower bound

Properties of

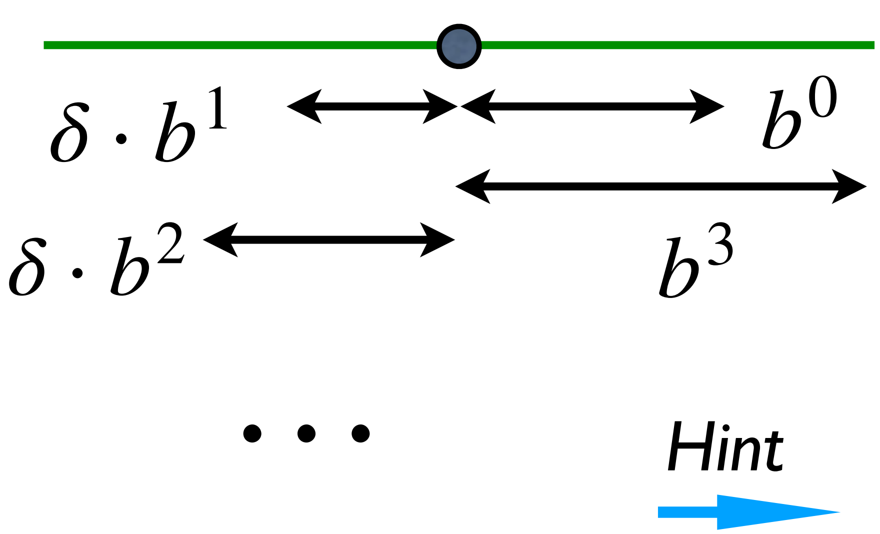
$$\sum_{i=0}^{n} x_{i-1} \leq r x_i$$

Result

$$\left(\frac{b_r + 1}{b_r - 1}, r\right)$$

Directional hint

Upper bound



Lower bound

Generalization

of Gal's Theorem

Result

$$c = 1 + 2\left(\frac{b^2}{b^2-1} + \delta \frac{b^3}{b^2-1}\right)$$

$$r = 1 + 2\left(\frac{b^2}{b^2-1} + \frac{1}{\delta}\frac{b^3}{b^2-1}\right)$$

k bits of (untrusted) advice

k bits of (untrusted) advice

Upper bound

1. Define 2^k appropriate "pseudo-geometric" strategies.

Half of them start on the right, the other half start on the left

2. Require that each of these strategies is **individually r-robust**

(this gives a *range* for b as function of r)

3. Consistency = competitive ratio of the **best** strategy,

optimize b within the range of step (2).

k bits of (untrusted) advice

Upper bound

1. Define 2^k appropriate "pseudo-geometric" strategies.

   Half of them start on the right, the other half start on the left

2. Require that each of these strategies is **individually r-robust**

   (this gives a *range* for b as function of r)

3. Consistency = competitive ratio of the **best** strategy,

   optimize b within the range of step (2).

$$\text{consistency} = 1 + 2\frac{b_r^{1/2^{k-1}}}{b_r - 1}$$

k bits of (untrusted) advice

Upper bound

1. Define 2^k appropriate "pseudo-geometric" strategies.
   Half of them start on the right, the other half start on the left

2. Require that each of these strategies is **individually r-robust**
   (this gives a *range* for b as function of r)

3. Consistency = competitive ratio of the **best** strategy,
   optimize b within the range of step (2).

$$\text{consistency} = 1 + 2\frac{b_r^{1/2^{k-1}}}{b_r - 1}$$

max b that guarantees
$r-$robustness

k bits of (untrusted) advice

Upper bound

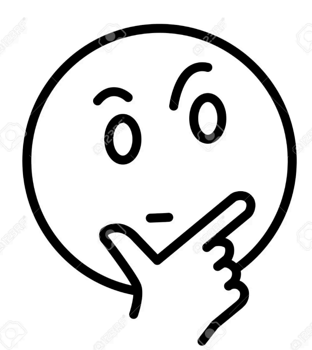
1. Define 2^k appropriate "pseudo-geometric" strategies.
   Half of them start on the right, the other half start on the left

2. Require that each of these strategies is **individually r-robust**
   (this gives a *range* for b as function of r)

3. Consistency = competitive ratio of the **best** strategy,
   optimize b within the range of step (2).

$$\text{consistency} = 1 + 2\frac{b_r^{1/2^{k-1}}}{b_r - 1}$$

max b that guarantees
$r-$robustness



Tricky part: showing
that this is optimal

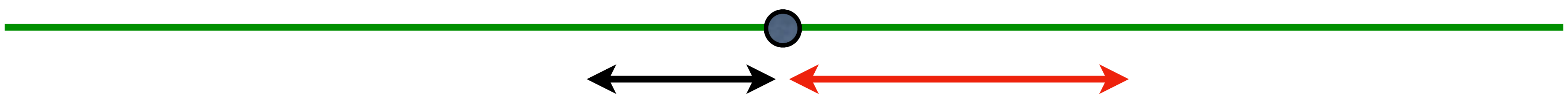
Lower bound: proof idea

Lower bound: proof idea

Suppose k=1 $\rightarrow$ two searchers

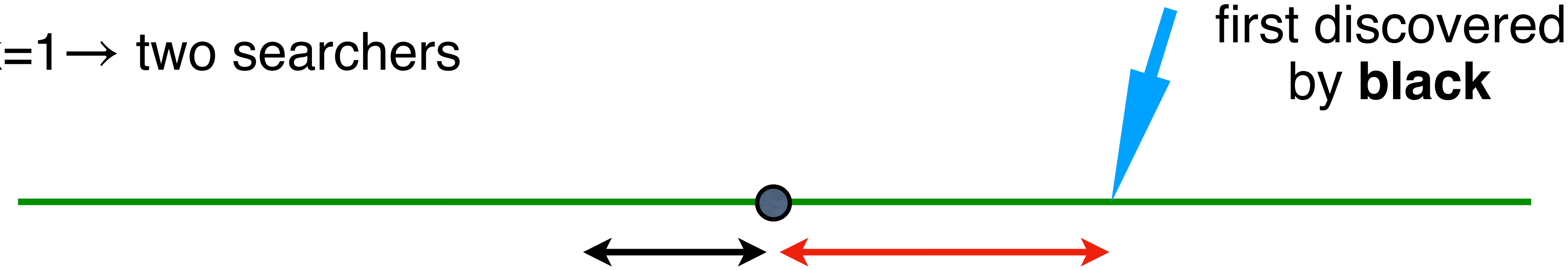
Lower bound: proof idea

Suppose k=1$\rightarrow$ two searchers



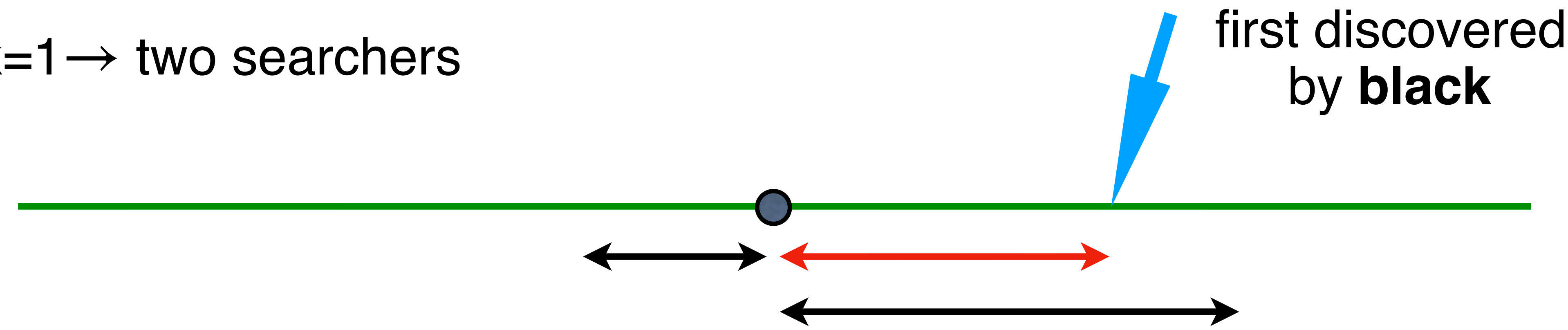
Lower bound: proof idea

Suppose k=1→ two searchers



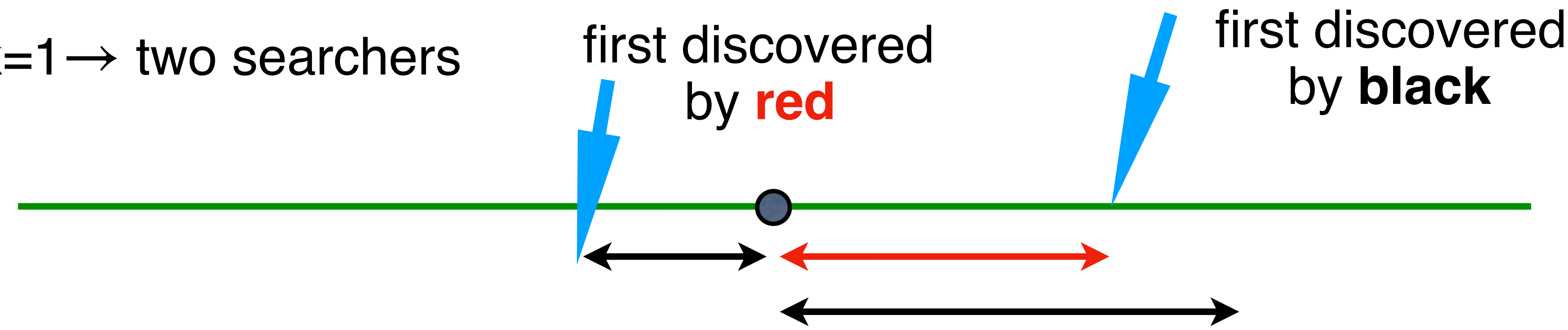
Lower bound: proof idea

Suppose k=1 → two searchers



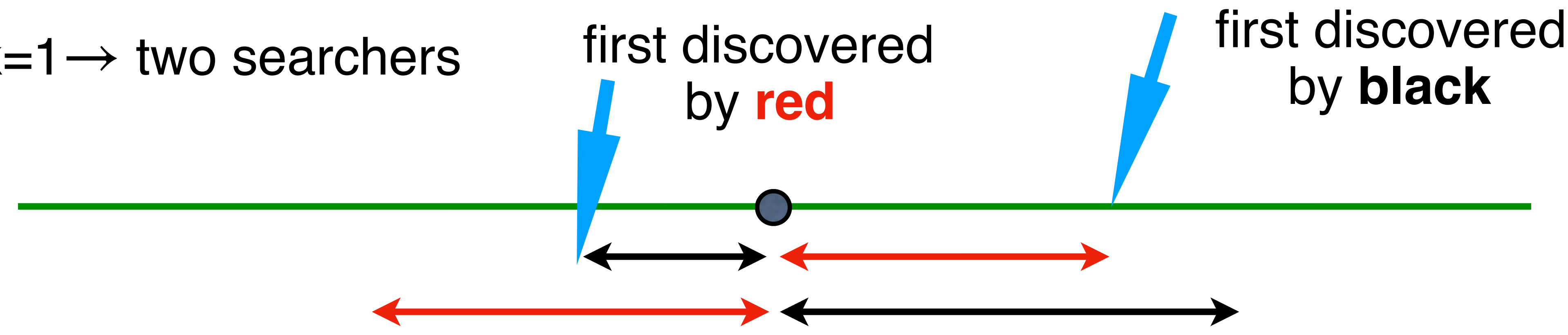
Lower bound: proof idea

Suppose k=1 → two searchers



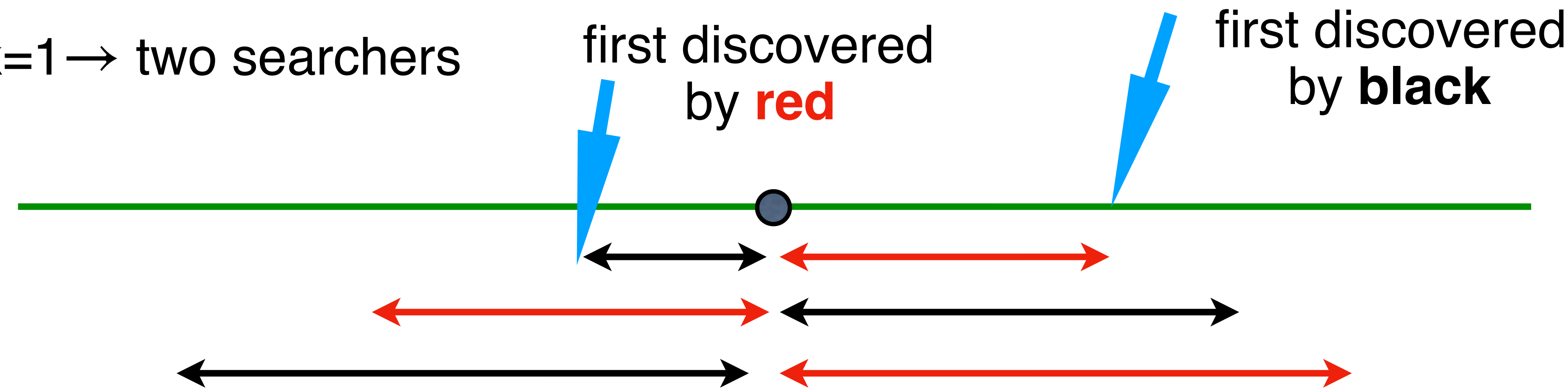
Lower bound: proof idea

Suppose k=1 → two searchers



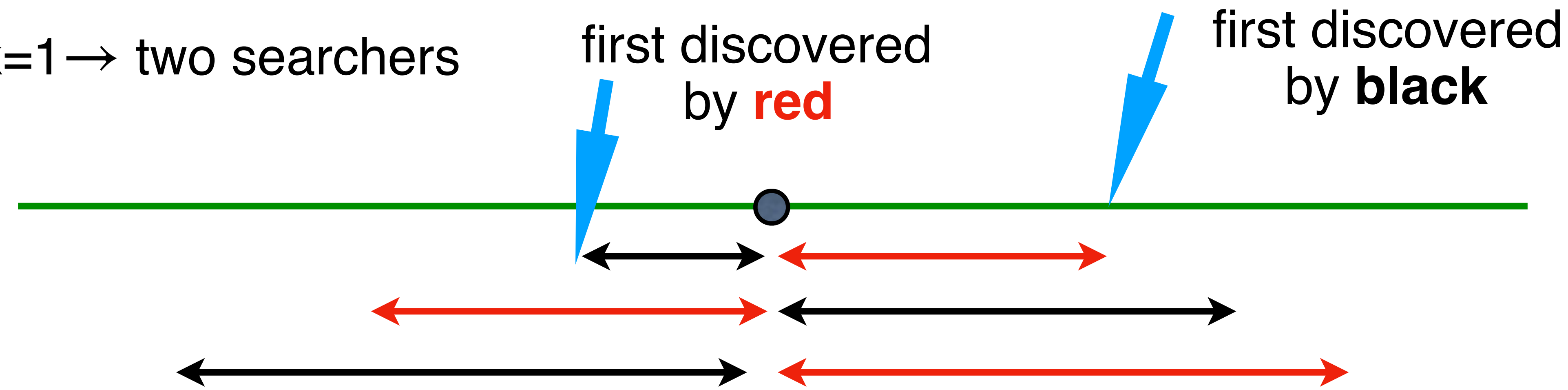
Lower bound: proof idea

Suppose k=1 → two searchers



Lower bound: proof idea

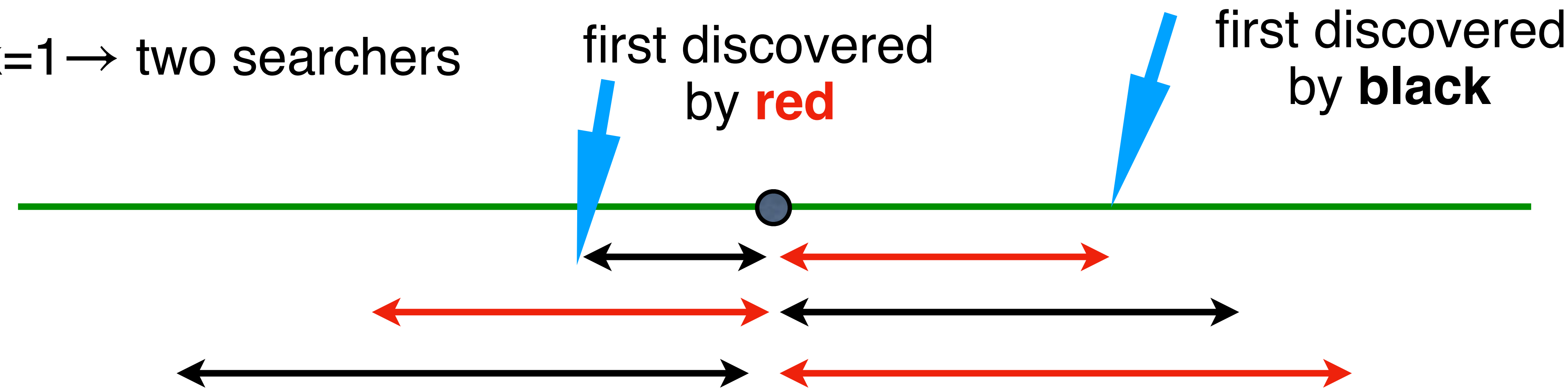
Suppose k=1 → two searchers



$$\text{Consistency} \geq \sup_i \frac{\text{cost}_i}{d_i}$$

Lower bound: proof idea

Suppose k=1 $\rightarrow$ two searchers

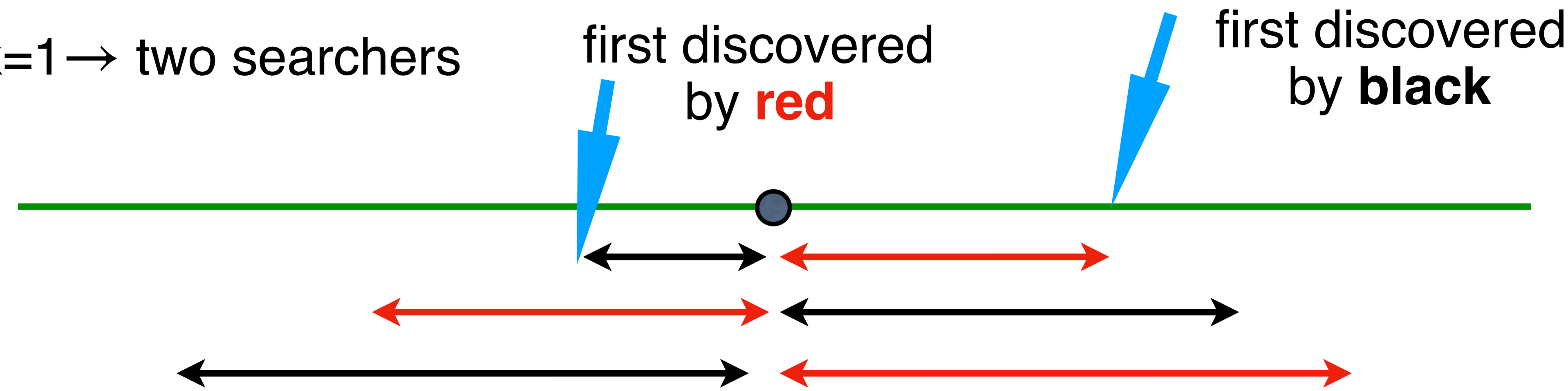


$$\text{Consistency} \geq \sup_i \frac{\text{cost}_i}{d_i}$$

Apply Gal's theorem on this expression

Lower bound: proof idea

Suppose k=1 → two searchers



$$\text{Consistency} \geq \sup_i \frac{\text{cost}_i}{d_i}$$

Apply Gal's theorem on this expression

Complications:

1. Need to incorporate individual robustness of searchers

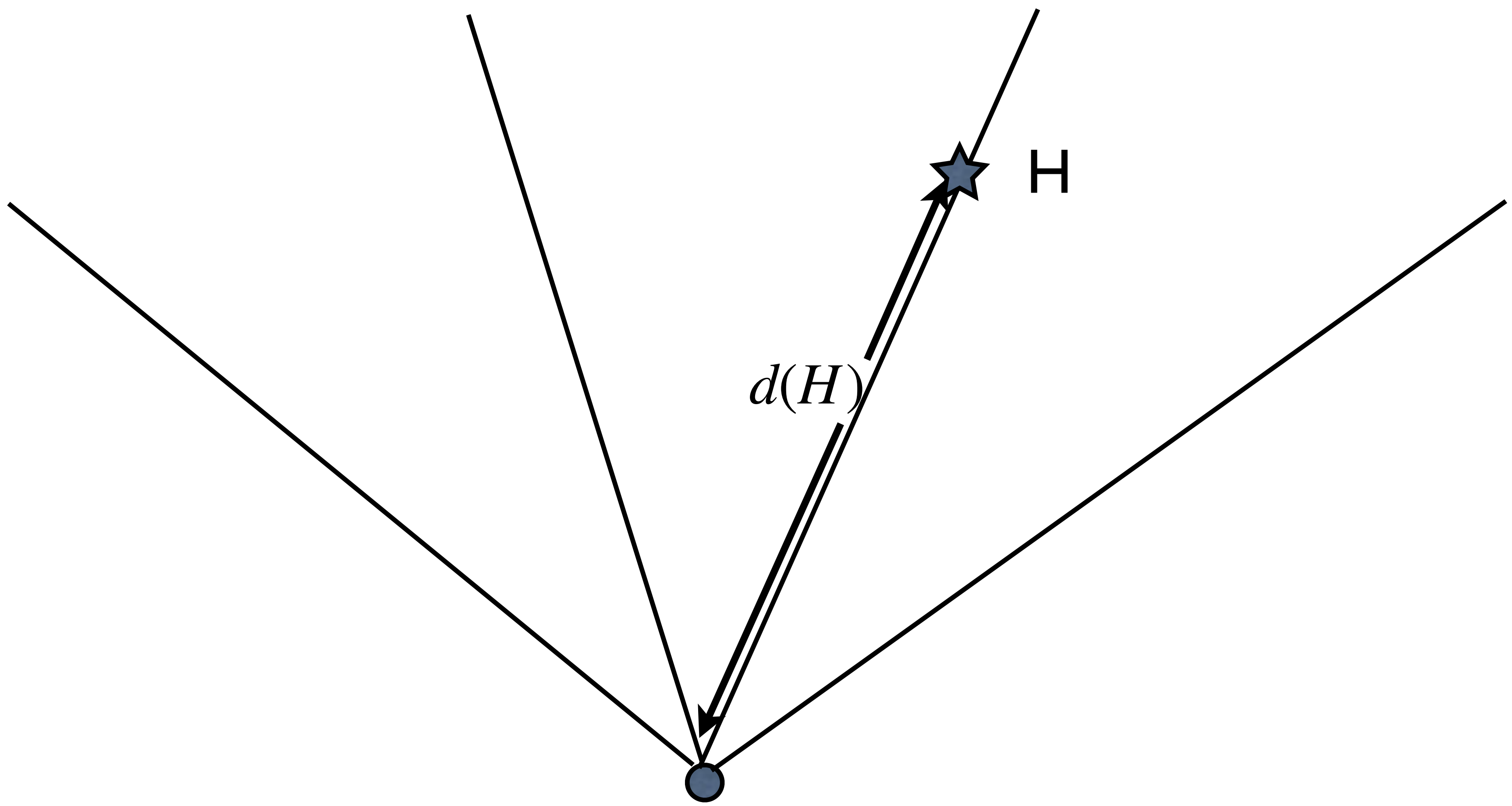
   Workaround: get into the details of the limsup

2. Searchers may be asymmetric (more on one side than the other)

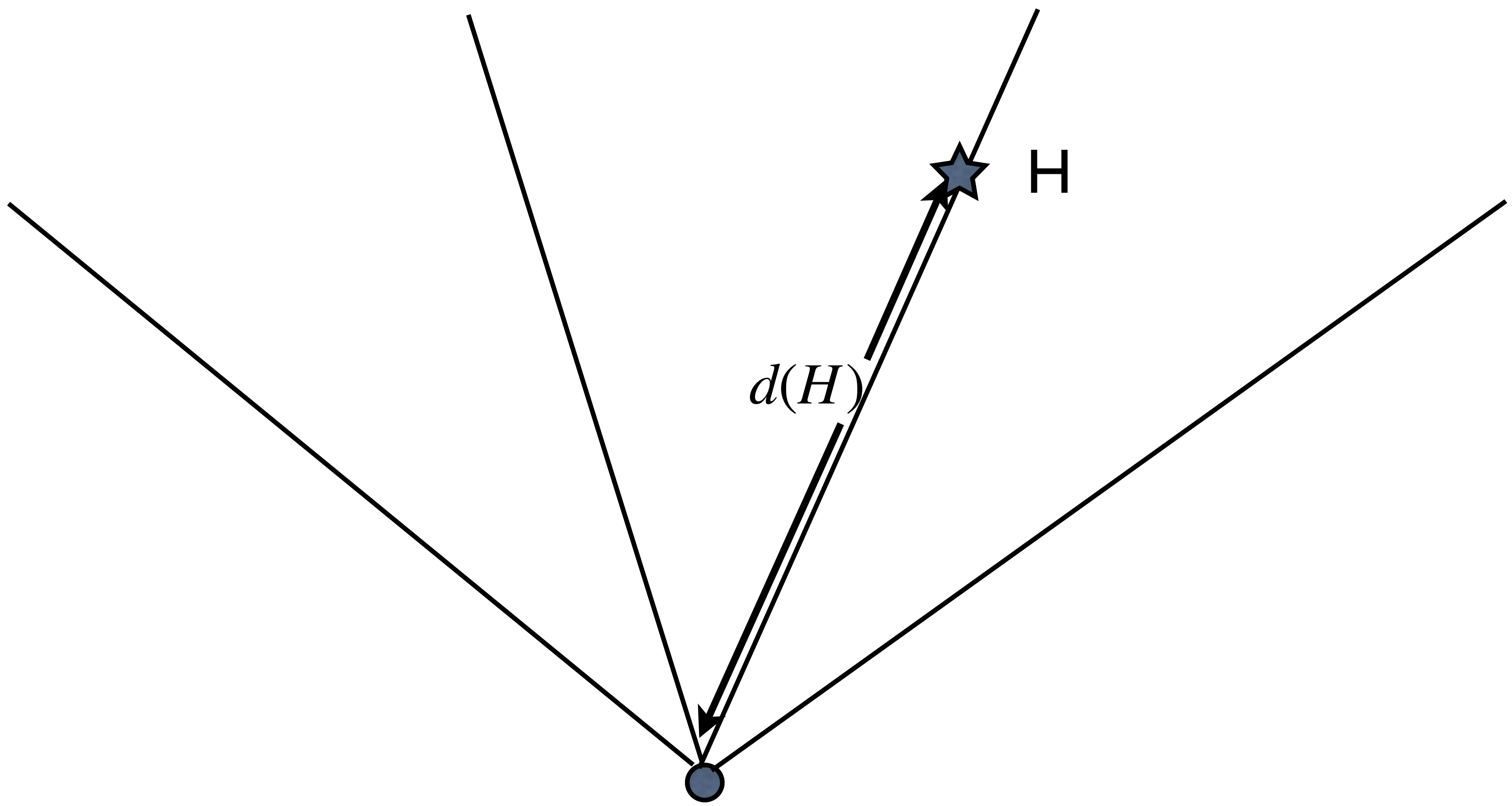
   Workaround: Bijective mapping over the search lengths that “balances" things up

Extensions to star search

Extensions to star search



Extensions to star search



Results

- Tight bounds for positional hints
- Asymptotically tight results for directional hints
- k-bit advice is open (in particular: lower bounds)

Dealing with errors

- **Positional hint:** *Error :* distance of the hider from the hint

  Strategy: Pretend that the searcher is "close and beyond" the predicted position

- **Directional hint:** No concept of error for the line ☺

  For the star: rank difference between the predicted/actual ray

- **k-bit advice hint:** *Error :* # of erroneous advice bits (or wrong query responses)

  Strategy : Search the space of 2^k pseudo-geometric strategies by using **fault-tolerant binary search**

  Takeaway: Upper and lower bounds via *Rényi-Ulam games*

Search games (work in progress)

Consistency / robustness tradeoffs for mixed strategy games

- **Box search**: n boxes, each with a search cost, payoff= expected search time

  hint = hider's box

  Extends results of [Lidbetter 2013]

- **Tree search**: Expanding search in a tree-like network Q

  hint = a connected branch of Q

  Extends results of [Alpern and Lidbetter 2023]

- **Linear search**: Randomized search on the infinite line

  hint = direction of search

  Extends a result of [Gal 1980]

Linear search game : techniques and results

Linear search game : techniques and results

- Biased randomized doubling with base $\alpha > 1$: The non-predicted brach is searched **less**, say by a factor $\mu \in (0,1)$

$$\text{consistency} = 1 + \frac{1 + \mu\alpha}{\ln \alpha} \qquad \text{robustness} = 1 + \frac{1 + \frac{\alpha}{\mu}}{\ln \alpha}$$

Linear search game : techniques and results

- Biased randomized doubling with base $\alpha > 1$: The non-predicted brach is searched **less**, say by a factor $\mu \in (0,1)$

$$\text{consistency} = 1 + \frac{1 + \mu\alpha}{\ln \alpha} \qquad \text{robustness} = 1 + \frac{1 + \frac{\alpha}{\mu}}{\ln \alpha}$$

- But how can we show that this is Pareto-optimal?

Answer: Suffices to study a new game in which the Searcher must minimize

a given linear combination of consistency and robustness, i.e,

constistency $+ \lambda \cdot$ robustness, with $\lambda \in (0,1]$

Linear search game : techniques and results

- Biased randomized doubling with base $\alpha > 1$: The non-predicted brach is searched **less**, say by a factor $\mu \in (0,1)$

$$\text{consistency} = 1 + \frac{1 + \mu\alpha}{\ln \alpha} \qquad\qquad \text{robustness} = 1 + \frac{1 + \frac{\alpha}{\mu}}{\ln \alpha}$$

- But how can we show that this is Pareto-optimal?

Answer: Suffices to study a new game in which the Searcher must minimize

a given linear combination of consistency and robustness, i.e,

constistency $+ \lambda \cdot$ robustness, with $\lambda \in (0,1]$

- We show that the value of this new game is $1 + \lambda + \min_{\alpha>1} \frac{1 + \lambda + 2\sqrt{\lambda\alpha}}{\ln a}$,

which is matched by the randomized doubling strategy with $\mu = \sqrt{\lambda}$

Conclusion

- Searching with untrusted information under several prediction models
- Even simple search problems become challenging under predictions

Future work

- Searching in graphs
- Patrolling and rendezvous games
- Combining advice complexity and learnability

Looking for a postdoc (or a phd opening) ?

3 postdocs over the next four years

http://lip6.fr/Spyros.Angelopoulos

Looking for a postdoc (or a phd opening) ?

3 postdocs over the next four years

http://lip6.fr/Spyros.Angelopoulos

Thank you!