Searching with turn cost and related problems

Spyros Angelopoulos

CNRS-UPMC

Joint work with

Diogo Arsenio | Christoph Dürr | Alex Lopez-Ortiz

CNRS | CNRS | Univ. of Waterloo

Outline of the presentation

Outline of the presentation

- Setting : A searcher that must locate a fixed target
  Starting point: environment = set of rays
- Objective : As quickly as possible -> performance guarantees
- Variant : Turning direction incurs a fixed cost

Outline of the presentation

- **Setting** : A searcher that must locate a fixed target
  Starting point: environment = set of rays
- **Objective** : As quickly as possible -> performance guarantees
- **Variant** : Turning direction incurs a fixed cost
- **Main result** :

Outline of the presentation

- Setting : A searcher that must locate a fixed target
  Starting point: environment = set of rays

- Objective : As quickly as possible -> performance guarantees

- Variant : Turning direction incurs a fixed cost

- Main result : ★ Tight bounds on the performance measures

Outline of the presentation

- Setting : A searcher that must locate a fixed target
  Starting point: environment = set of rays
- Objective : As quickly as possible -> performance guarantees
- Variant : Turning direction incurs a fixed cost
- Main result : ★ Tight bounds on the performance measures

  ★ Explore the role of infinite LPs + duality

Outline of the presentation

- **Setting** : A searcher that must locate a fixed target
  Starting point: environment = set of rays

- **Objective** : As quickly as possible -> performance guarantees

- **Variant** : Turning direction incurs a fixed cost

- **Main result** :
  ★ Tight bounds on the performance measures

  ★ Explore the role of infinite LPs + duality

  ★ Connections with problems in AI

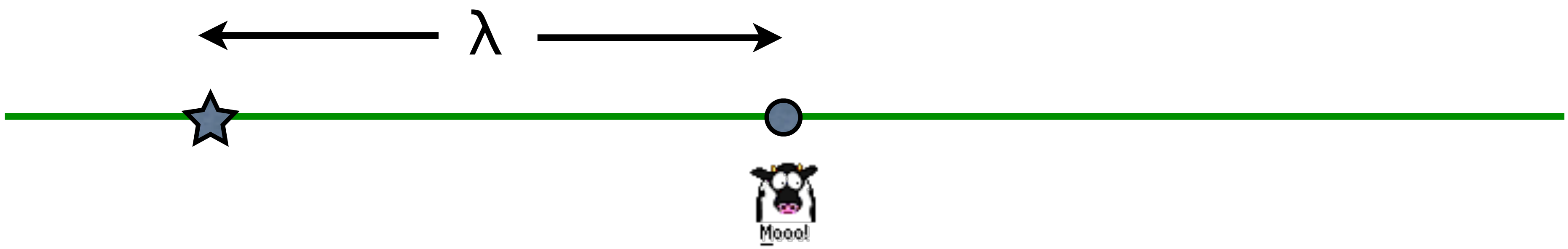
Warm-up : The cow-path problem

Warm-up : The cow-path problem

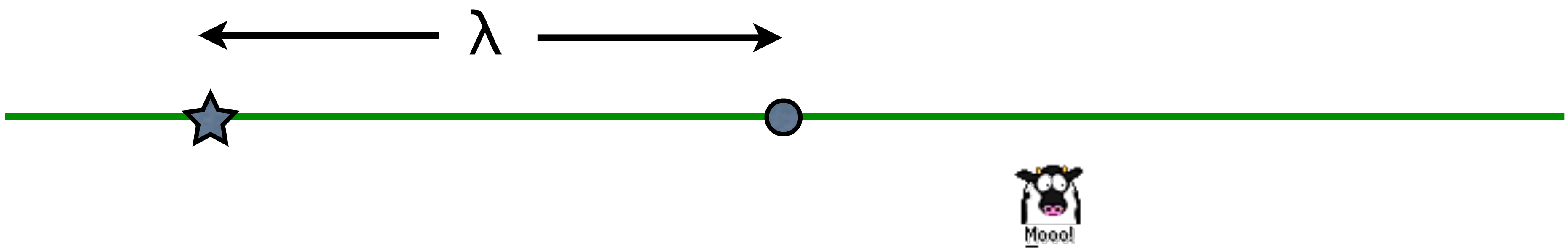
Warm-up : The cow-path problem

Warm-up : The cow-path problem

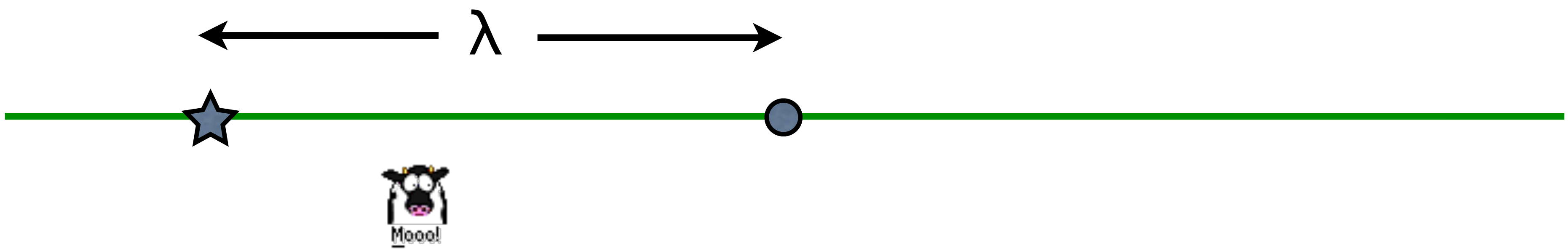
Warm-up : The cow-path problem



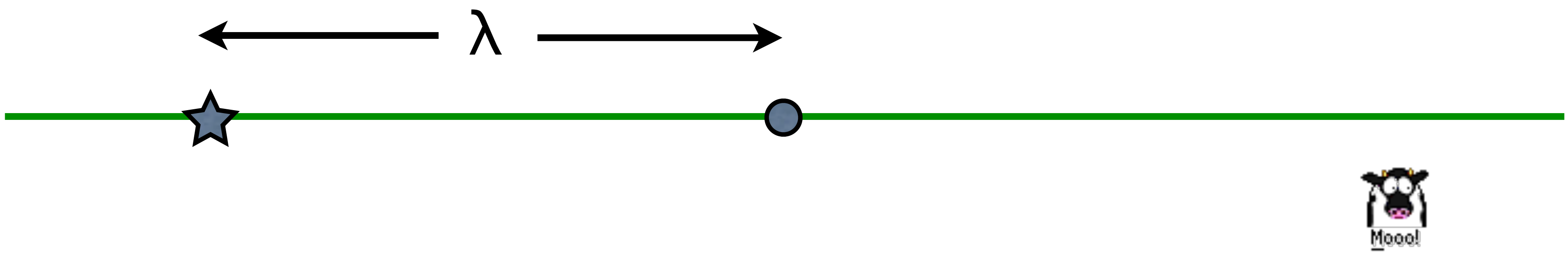
Warm-up : The cow-path problem



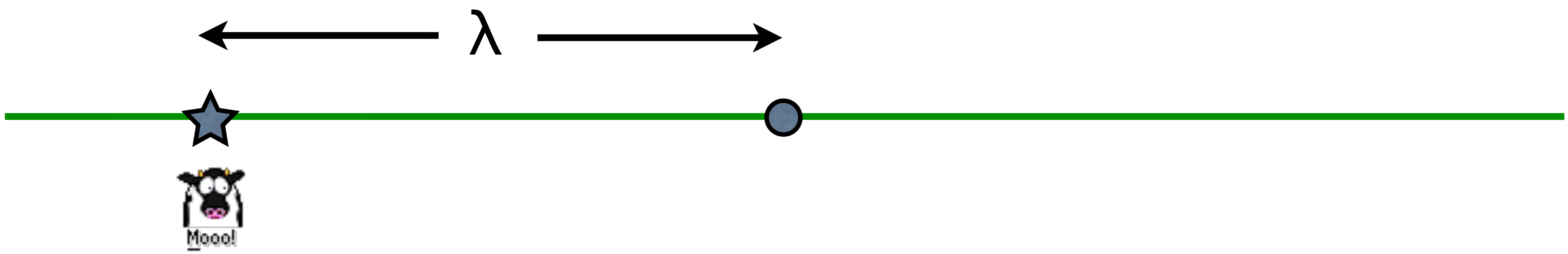
Warm-up : The cow-path problem



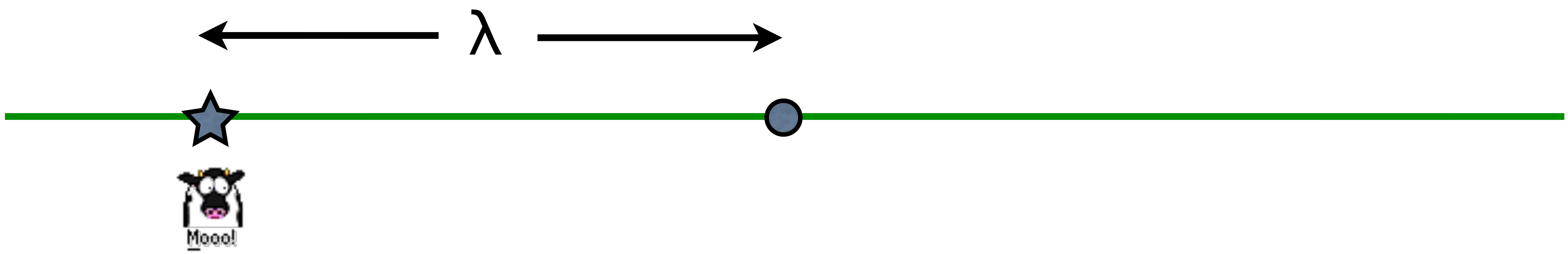
Warm-up : The cow-path problem



Warm-up : The cow-path problem

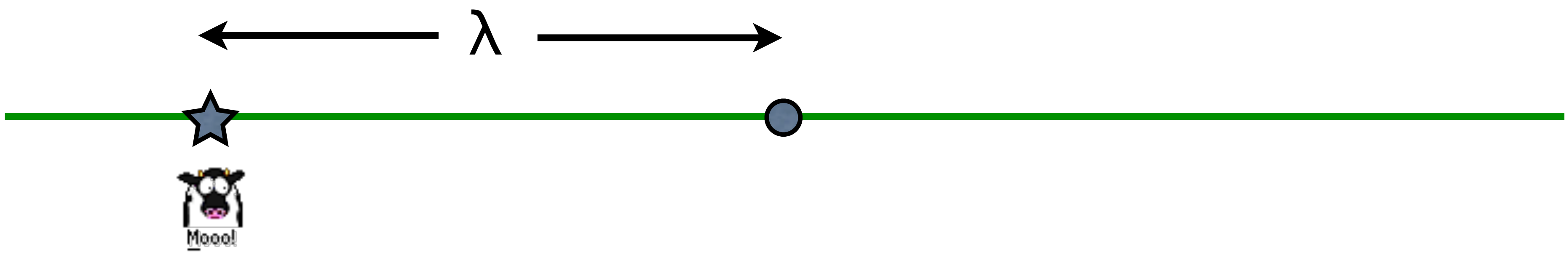


Warm-up : The cow-path problem



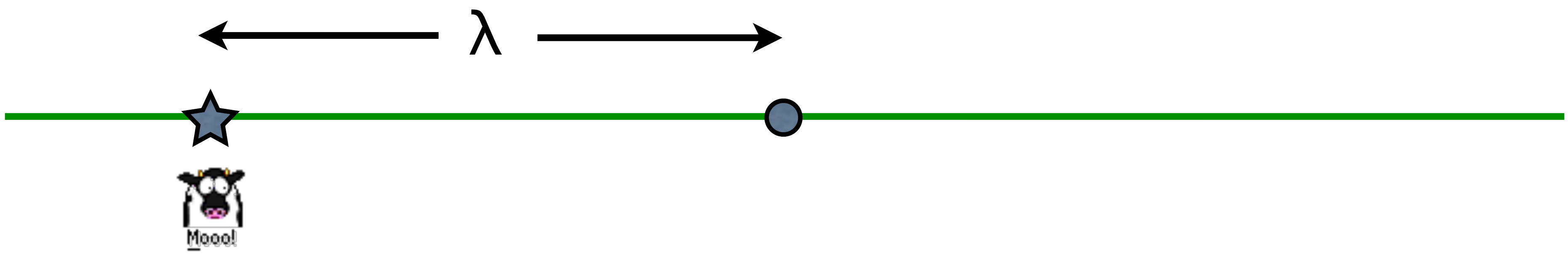
- Search strategy : an (infinite) sequence of turn points

Warm-up : The cow-path problem



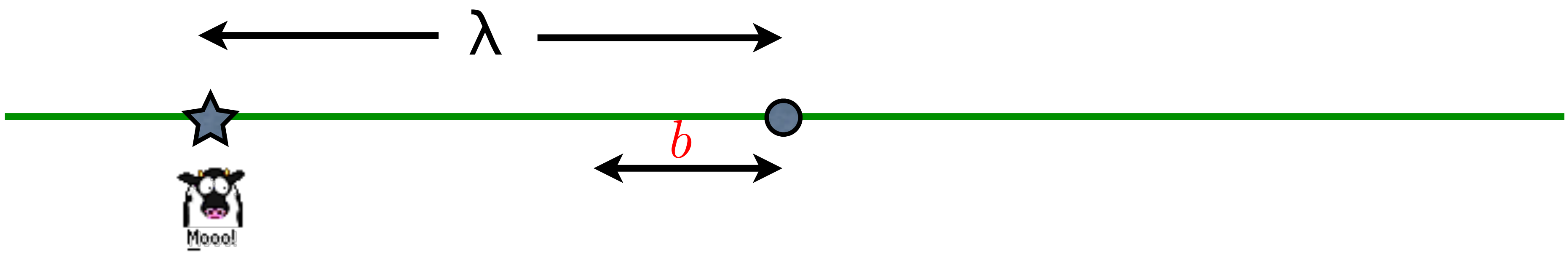
- Search strategy : an (infinite) sequence of turn points
- Competitive ratio = $\sup_{\star} \dfrac{\text{total distance of searcher}}{\lambda}$

Warm-up : The cow-path problem



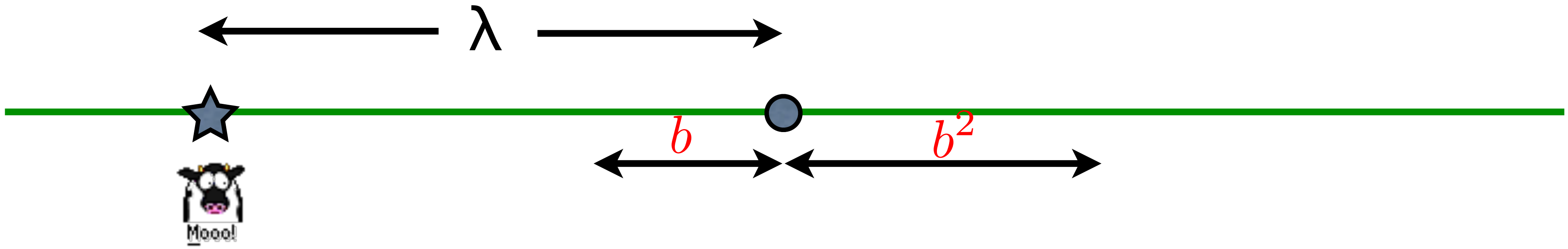
- **Search strategy** : an (infinite) sequence of turn points
- **Competitive ratio** = $\sup_{\star} \dfrac{\text{total distance of searcher}}{\lambda}$
- **Optimal strategy** : Geometric search

Warm-up : The cow-path problem



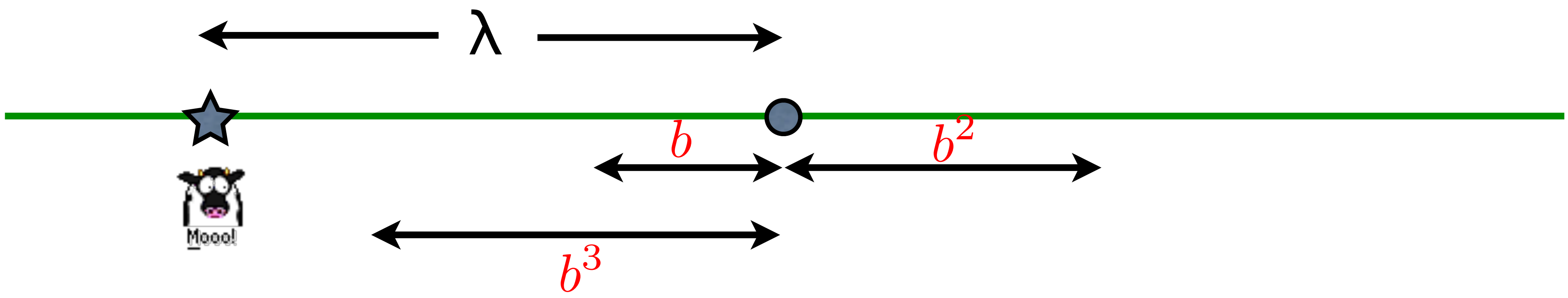
- Search strategy : an (infinite) sequence of turn points
- Competitive ratio = $\sup_{\star} \dfrac{\text{total distance of searcher}}{\lambda}$
- Optimal strategy : Geometric search

Warm-up : The cow-path problem



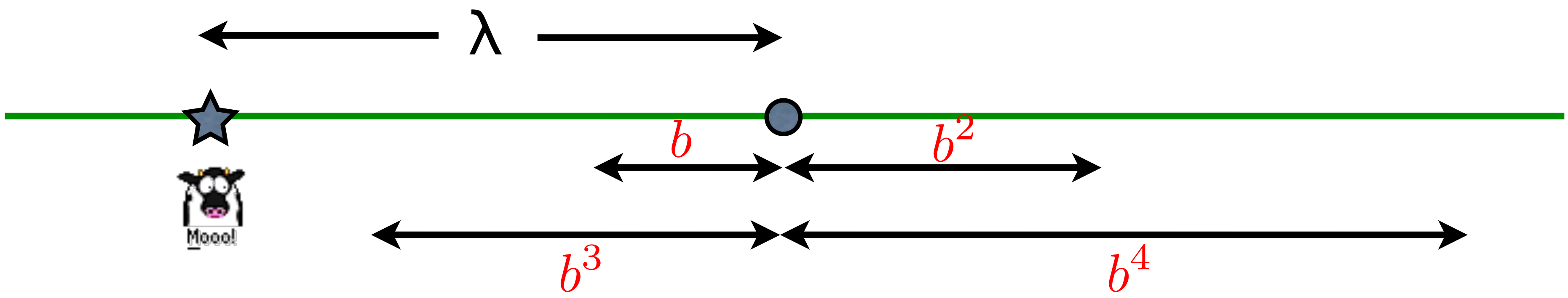
- Search strategy : an (infinite) sequence of turn points
- Competitive ratio = $\sup_{\star} \dfrac{\text{total distance of searcher}}{\lambda}$
- Optimal strategy : Geometric search

Warm-up : The cow-path problem



- Search strategy : an (infinite) sequence of turn points
- Competitive ratio = $\sup_{\star} \dfrac{\text{total distance of searcher}}{\lambda}$
- Optimal strategy : Geometric search

Warm-up : The cow-path problem



- **Search strategy** : an (infinite) sequence of turn points
- **Competitive ratio** = $\sup_{\star} \dfrac{\text{total distance of searcher}}{\lambda}$
- **Optimal strategy** : Geometric search

A long history of previous research

A long history of previous research

- Initially proposed by Bellman and Beck (1963) in a Bayesian context
- First solved by Beck and Newman : optimal competitive ratio of 9

A long history of previous research

- Initially proposed by Bellman and Beck (1963) in a Bayesian context
- First solved by Beck and Newman : optimal competitive ratio of 9

[Beck 64] On the linear search problem

A long history of previous research

- Initially proposed by Bellman and Beck (1963) in a Bayesian context
- First solved by Beck and Newman : optimal competitive ratio of 9

[Beck 64] On the linear search problem

[Beck 65] More on the linear search problem

A long history of previous research

- Initially proposed by Bellman and Beck (1963) in a Bayesian context
- First solved by Beck and Newman : optimal competitive ratio of 9

[Beck 64] On the linear search problem

[Beck 65] More on the linear search problem

[Beck& Newman 70] Yet more on the linear search problem

A long history of previous research

- Initially proposed by Bellman and Beck (1963) in a Bayesian context
- First solved by Beck and Newman : optimal competitive ratio of 9

[Beck 64] On the linear search problem

[Beck 65] More on the linear search problem

[Beck& Newman 70] Yet more on the linear search problem

[Beck& Warren 73] The return of the linear search problem

A long history of previous research

- Initially proposed by Bellman and Beck (1963) in a Bayesian context
- First solved by Beck and Newman : optimal competitive ratio of 9

[Beck 64] On the linear search problem

[Beck 65] More on the linear search problem

[Beck& Newman 70] Yet more on the linear search problem

[Beck& Warren 73] The return of the linear search problem

[Beck& Beck 84] Son of the linear search problem

A long history of previous research

- Initially proposed by Bellman and Beck (1963) in a Bayesian context
- First solved by Beck and Newman : optimal competitive ratio of 9

[Beck 64] On the linear search problem

[Beck 65] More on the linear search problem

[Beck& Newman 70] Yet more on the linear search problem

[Beck& Warren 73] The return of the linear search problem

[Beck& Beck 84] Son of the linear search problem

[Beck& Beck 86] The linear search problem rides again

A long history of previous research

- Initially proposed by Bellman and Beck (1963) in a Bayesian context
- First solved by Beck and Newman : optimal competitive ratio of 9

[Beck 64] On the linear search problem

[Beck 65] More on the linear search problem

[Beck& Newman 70] Yet more on the linear search problem

[Beck& Warren 73] The return of the linear search problem

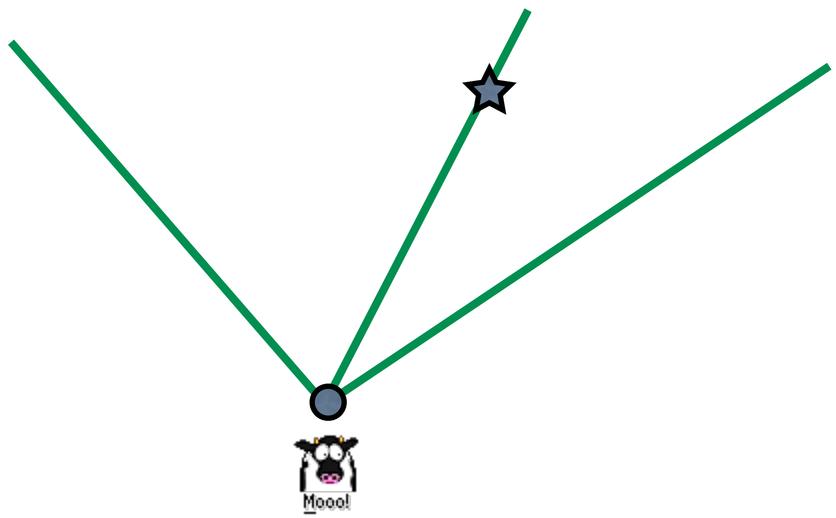
[Beck& Beck 84] Son of the linear search problem

[Beck& Beck 86] The linear search problem rides again

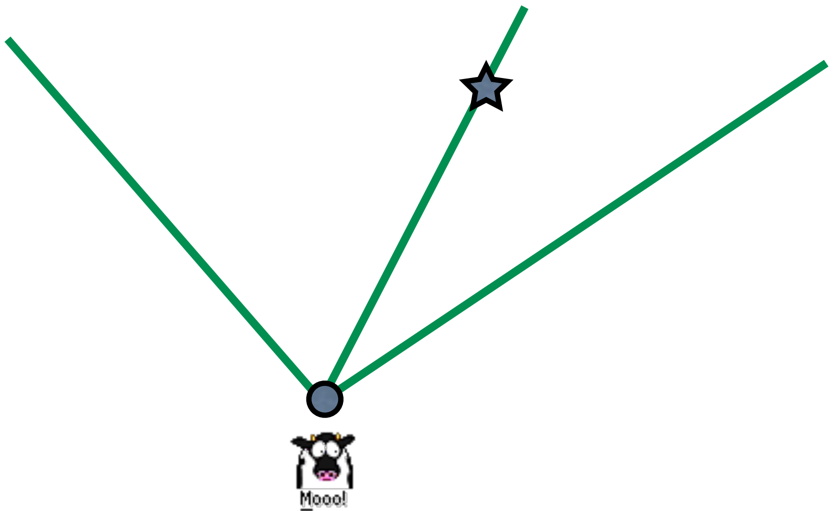
[Beck& Beck 92] Revenge of the linear search problem

A generalization : Star search or ray search

A generalization : Star search or ray search



A generalization : Star search or ray search

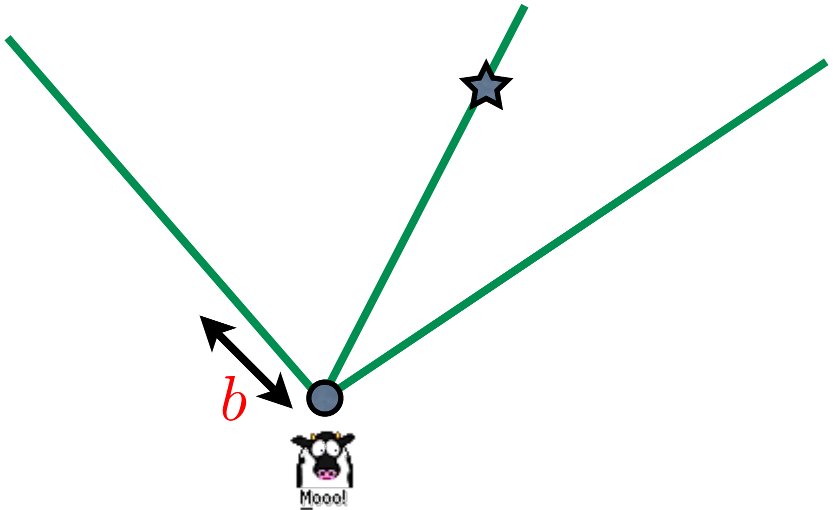


m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) $/\lambda$

Optimal strategy : geometric search [Gal 72]

A generalization : Star search or ray search

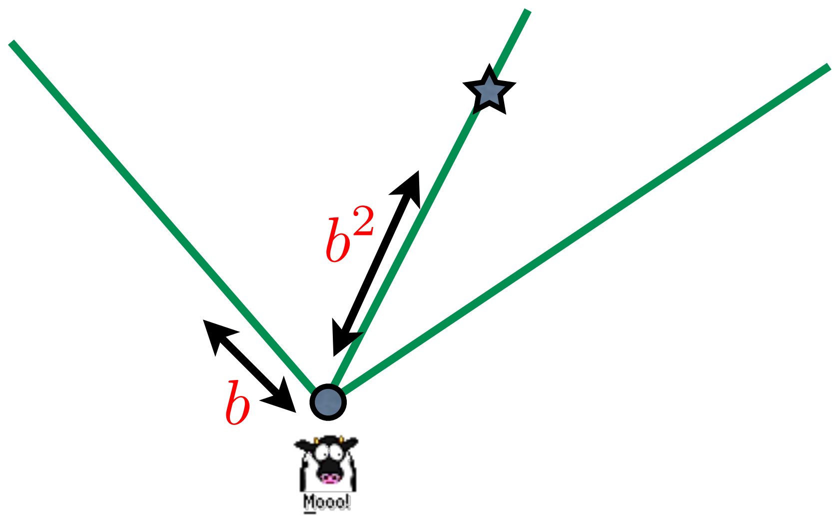


m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

A generalization : Star search or ray search

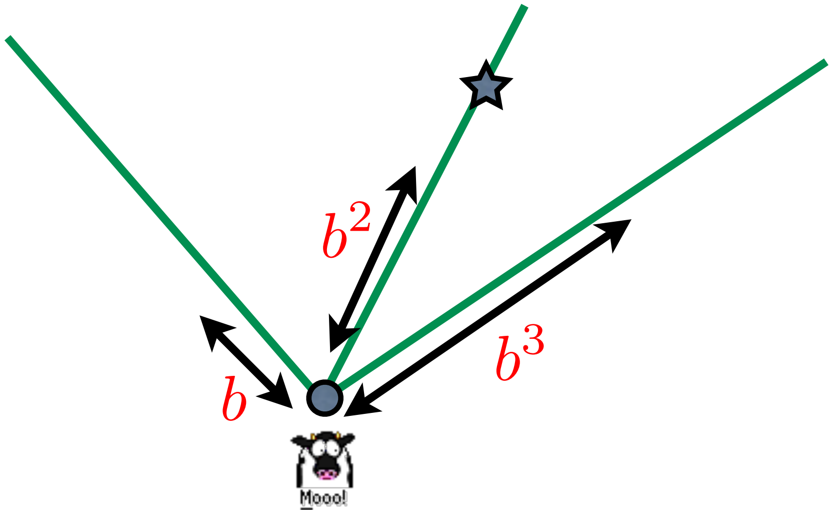


m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

A generalization : Star search or ray search

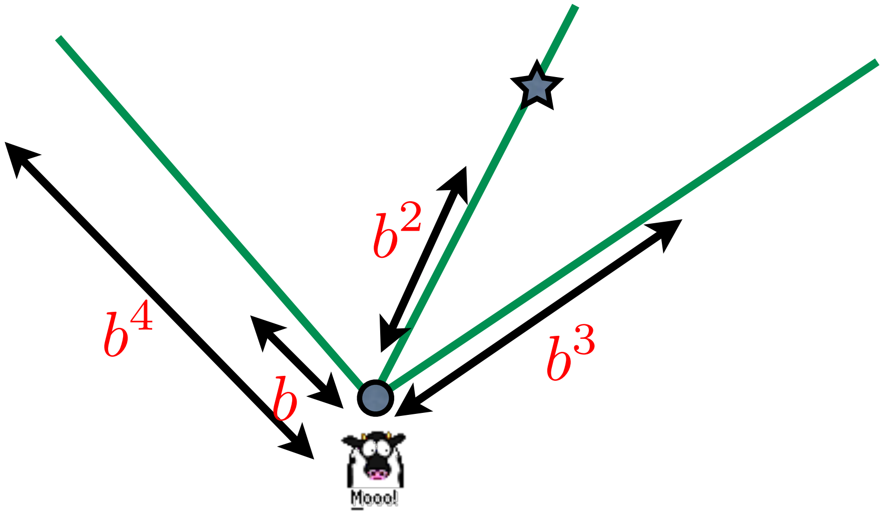


m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

A generalization : Star search or ray search

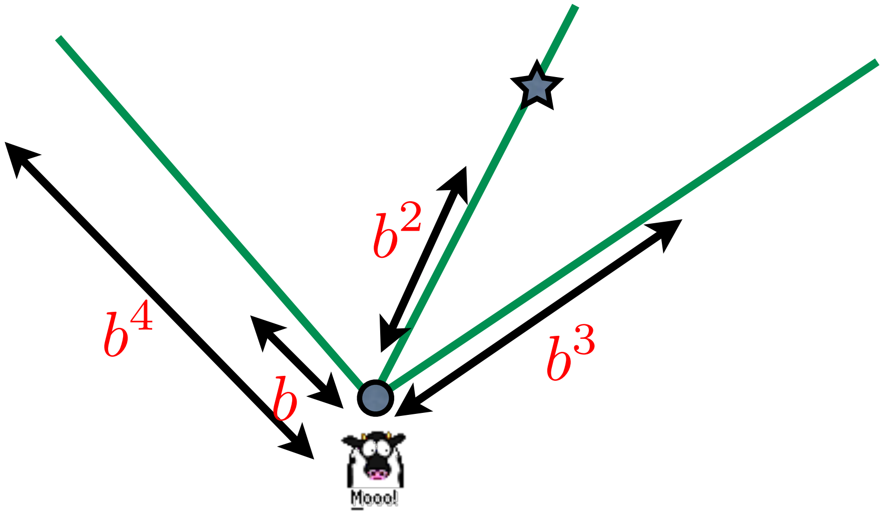


m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

A generalization : Star search or ray search



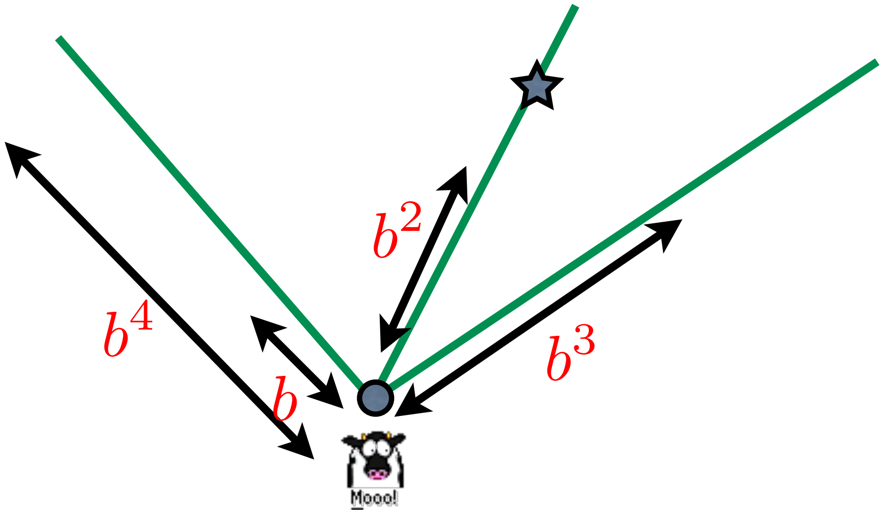
m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

Many variants:

A generalization : Star search or ray search



m infinite rays, one target

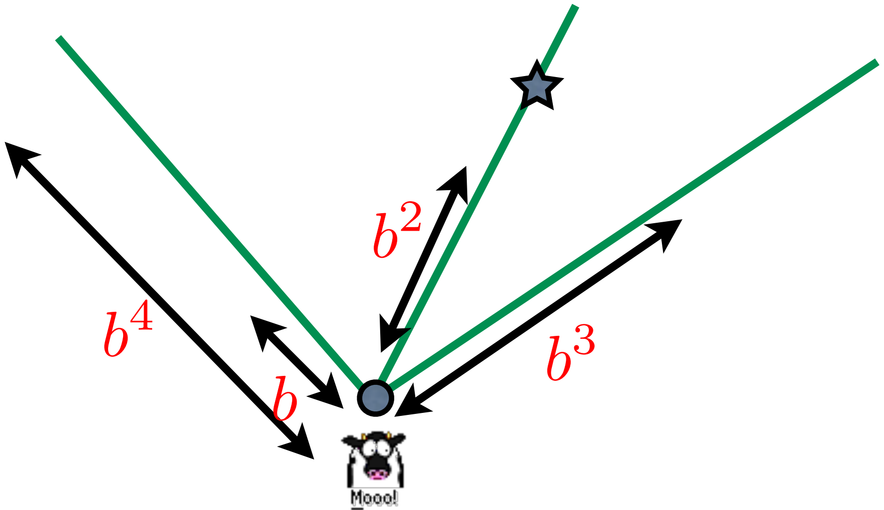
Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

Multiple searchers [Lopez-Ortiz and Schuierer 2002]

Many variants:

A generalization : Star search or ray search



m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

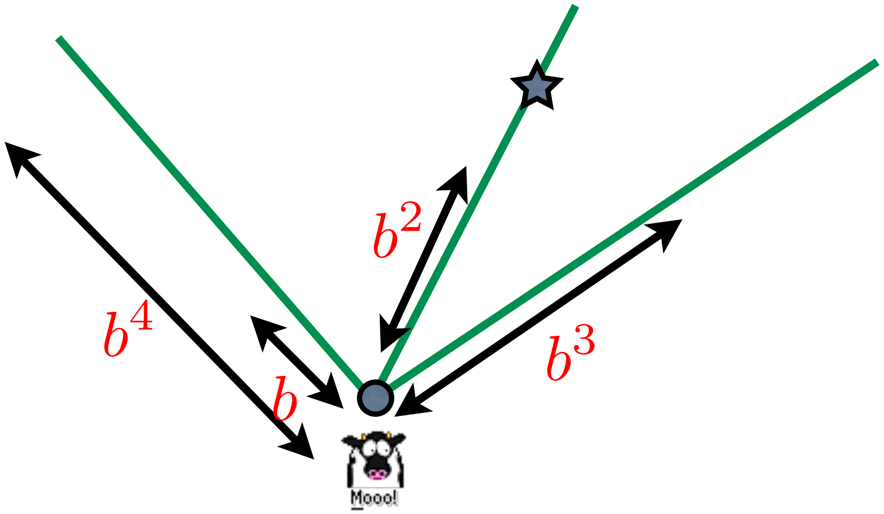
Optimal strategy : geometric search [Gal 72]

Many variants:

Multiple searchers [Lopez-Ortiz and Schuierer 2002]

Turn cost [Demaine et al. 2004]

A generalization : Star search or ray search



m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

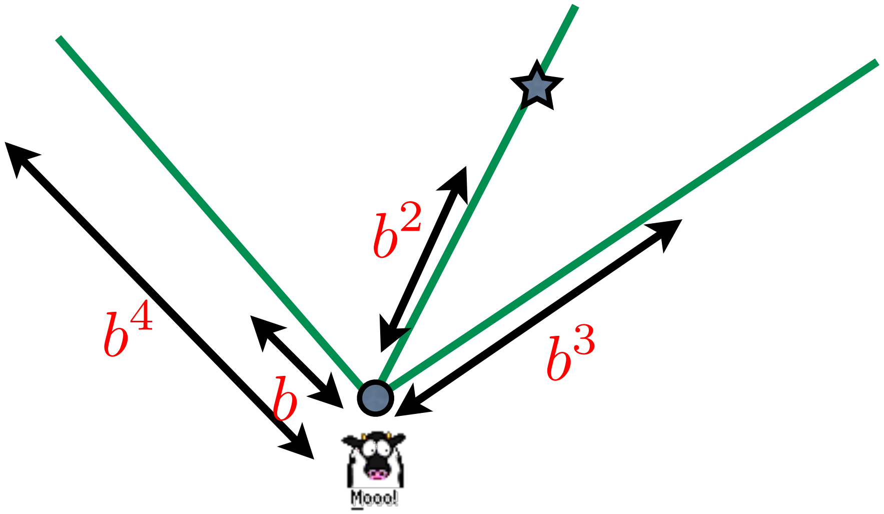
Many variants:

Multiple searchers [Lopez-Ortiz and Schuierer 2002]

Turn cost [Demaine et al. 2004]

Randomized strategies [Kao et al. 1996]

A generalization : Star search or ray search



m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

Many variants:

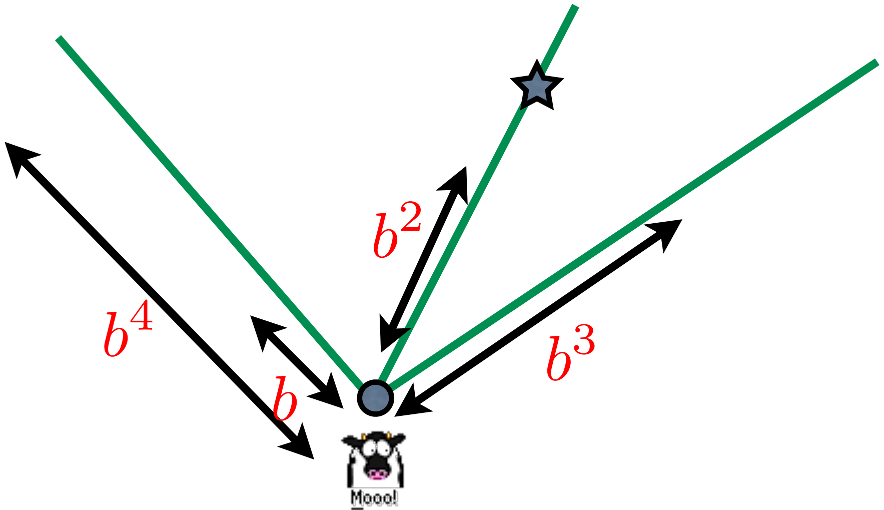
Multiple searchers [Lopez-Ortiz and Schuierer 2002]

Turn cost [Demaine et al. 2004]

Randomized strategies [Kao et al. 1996]

Searching on the plane [Gal 1980 and Langetepe 2010]

A generalization : Star search or ray search



m infinite rays, one target

Competitive ratio = $\sup_{\star}$ (search cost) /λ

Optimal strategy : geometric search [Gal 72]

Many variants:

Multiple searchers [Lopez-Ortiz and Schuierer 2002]

Turn cost [Demaine et al. 2004]

Randomized strategies [Kao et al. 1996]

Searching on the plane [Gal 1980 and Langetepe 2010]

Textbook by Alpern and Gal: The Theory of Search Games and Rendezvous

More recent related work

More recent related work

- [Kirkpatrick 2009] and [Mc Gregor et al. 2009] : New measures for analysis: OPT does not have the complete picture of the instance

More recent related work

- [Kirkpatrick 2009] and [Mc Gregor et al. 2009] : New measures for analysis: OPT does not have the complete picture of the instance
- [A. et al. 2011] Multi-target search
  [Tseng and Kirkpatrick 2011] Input-thrifty algorithms

More recent related work

- [Kirkpatrick 2009] and [Mc Gregor et al. 2009] : New measures for analysis: OPT does not have the complete picture of the instance

- [A. et al. 2011] Multi-target search
  [Tseng and Kirkpatrick 2011] Input-thrifty algorithms

- [Bose et al., 2013] Linear search with distance bounds

More recent related work

- [Kirkpatrick 2009] and [Mc Gregor et al. 2009] : New measures for analysis: OPT does not have the complete picture of the instance

- [A. et al. 2011] Multi-target search
  [Tseng and Kirkpatrick 2011] Input-thrifty algorithms

- [Bose et al., 2013] Linear search with distance bounds

- [Bernrstein et al. 2003] Ray search Interruptible algorithms

  [A. et al., 2006 & 2009]

More recent related work

- [Kirkpatrick 2009] and [Mc Gregor et al. 2009] : New measures for analysis: OPT does not have the complete picture of the instance

- [A. et al. 2011] Multi-target search
  [Tseng and Kirkpatrick 2011] Input-thrifty algorithms

- [Bose et al., 2013] Linear search with distance bounds

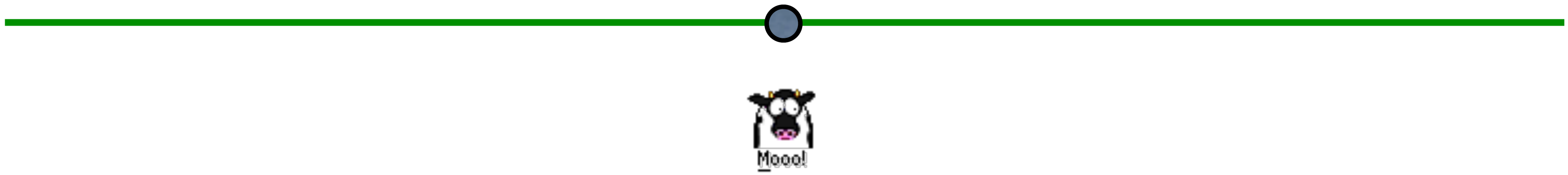
- [Bernrstein et al. 2003] Ray search Interruptible algorithms
  [A. et al., 2006 & 2009]

- In preparation : More connections between searching and interruptible algorithms

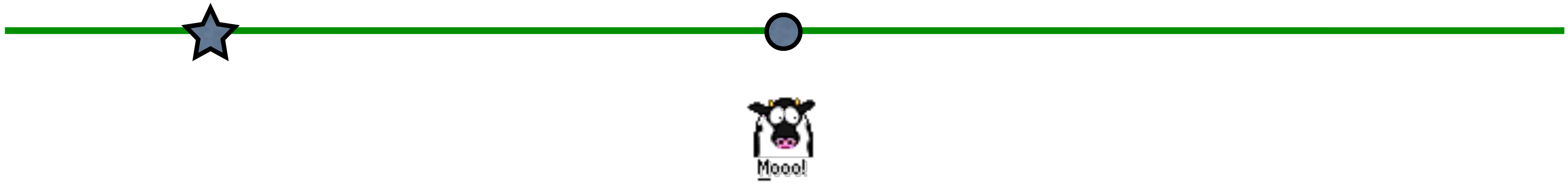
Online searching with turn cost

Online searching with turn cost

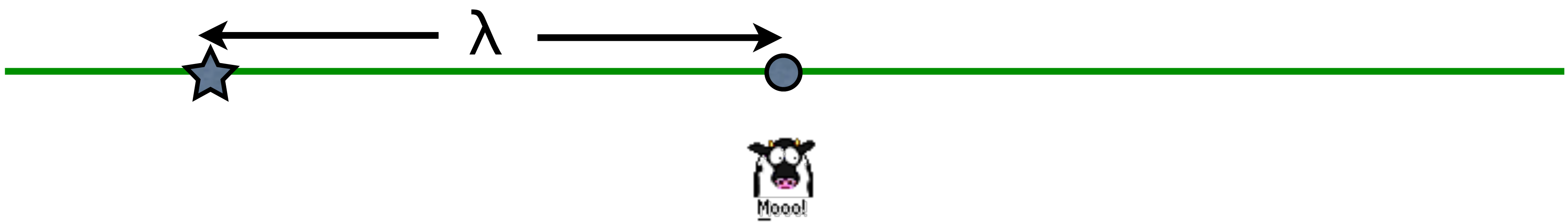
Online searching with turn cost



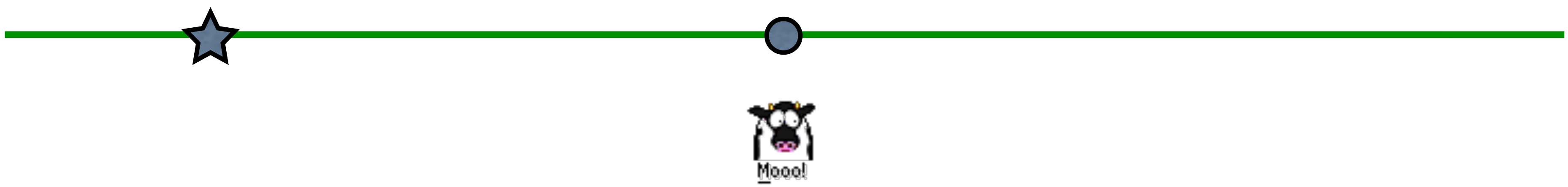
Online searching with turn cost



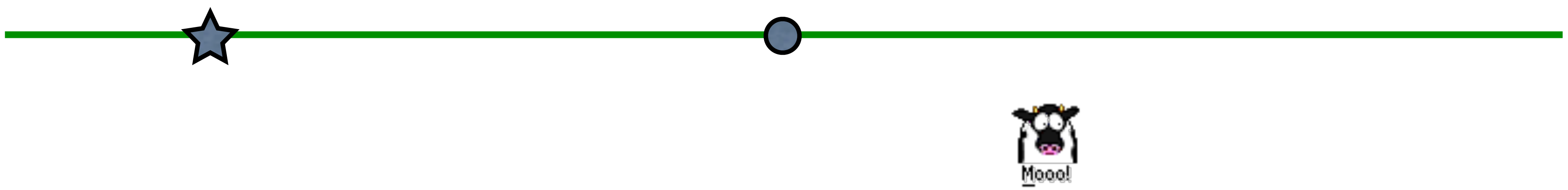
Online searching with turn cost



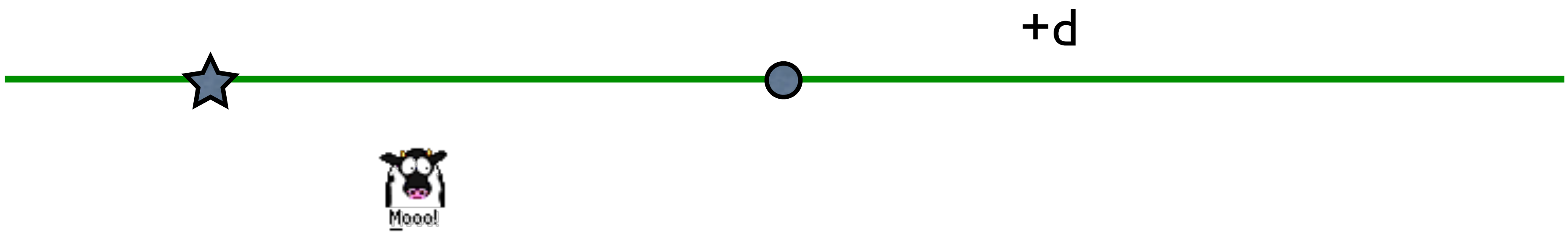
Online searching with turn cost



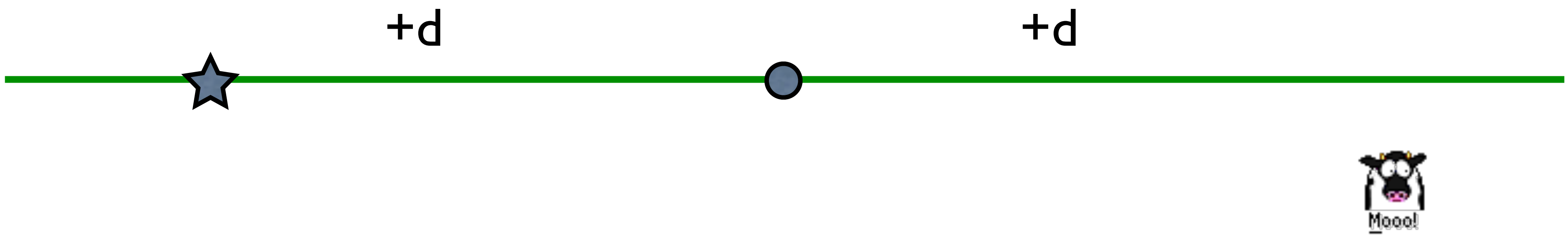
Online searching with turn cost



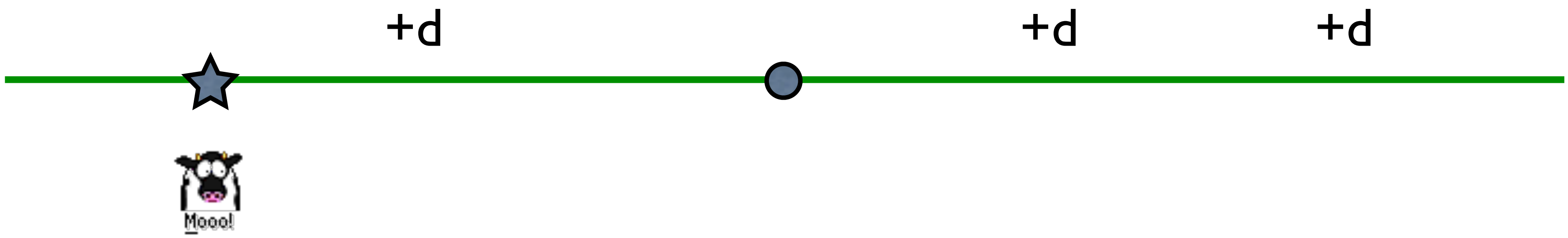
Online searching with turn cost



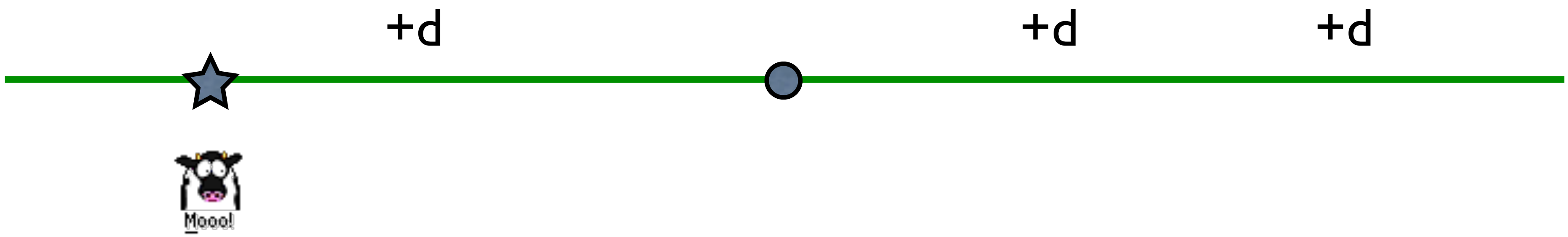
Online searching with turn cost



Online searching with turn cost

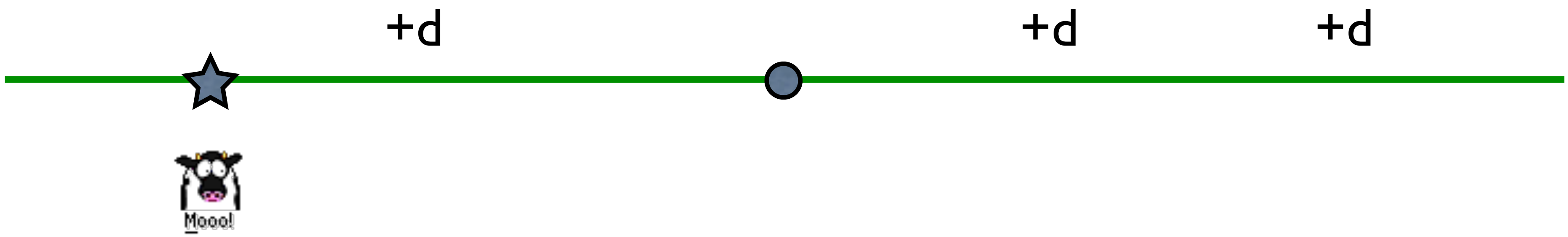


Online searching with turn cost



- Total cost = Distance traversed + overall turn cost

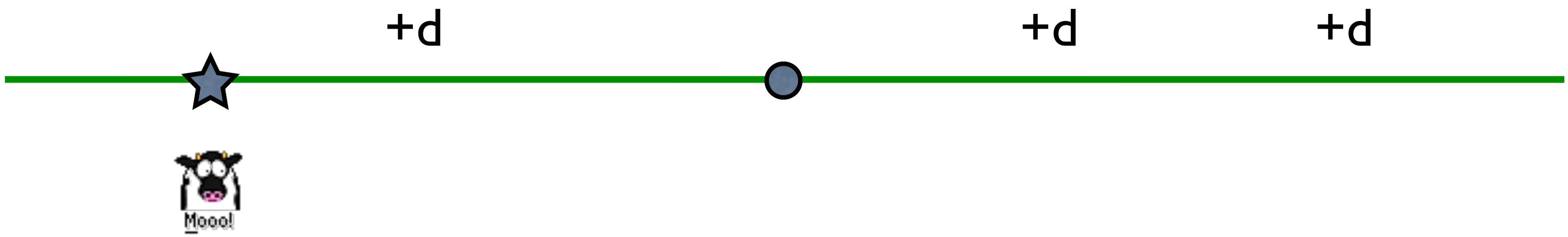
Online searching with turn cost



- Total cost = Distance traversed + overall turn cost

- Competitive ratio = $\sup_{\bigstar} \dfrac{\text{total search cost}}{\lambda}$

Online searching with turn cost



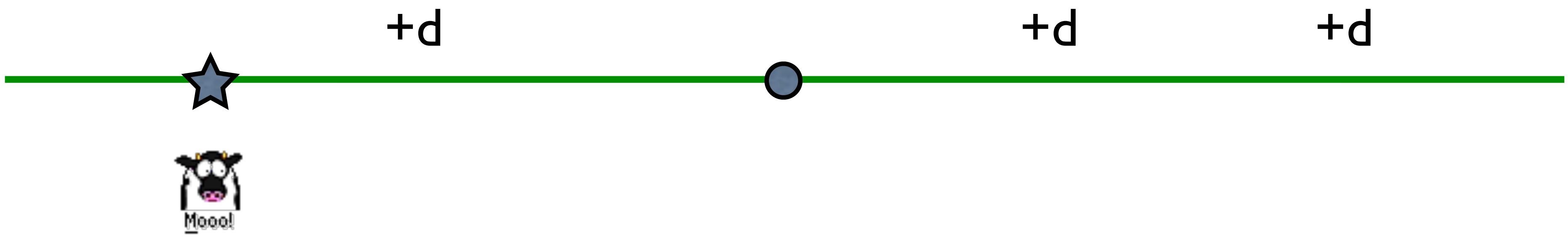
- Total cost = Distance traversed + overall turn cost

- Competitive ratio = $\sup_{\star} \dfrac{\text{total search cost}}{\lambda}$

- Objective : Find the smallest B such that

$$\text{Total cost of searching} \leq (\text{ Comp. ratio }) \cdot \lambda + B$$

Online searching with turn cost



- **Total cost** = Distance traversed + overall turn cost
- **Competitive ratio** = $\sup_{\bigstar} \dfrac{\text{total search cost}}{\lambda}$
- **Objective** : Find the smallest B such that

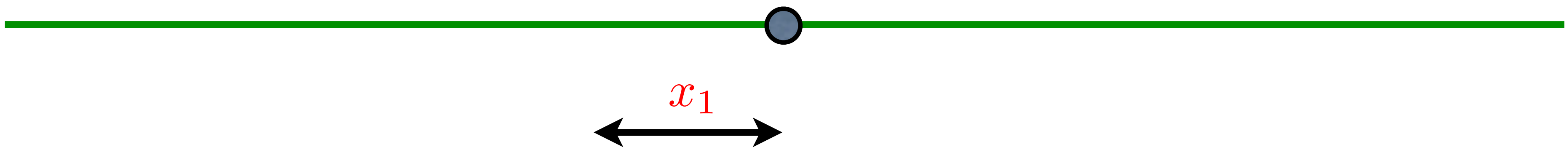
$$\text{Total cost of searching} \leq (\text{Comp. ratio}) \cdot \lambda + B$$

competitive ratio = 9

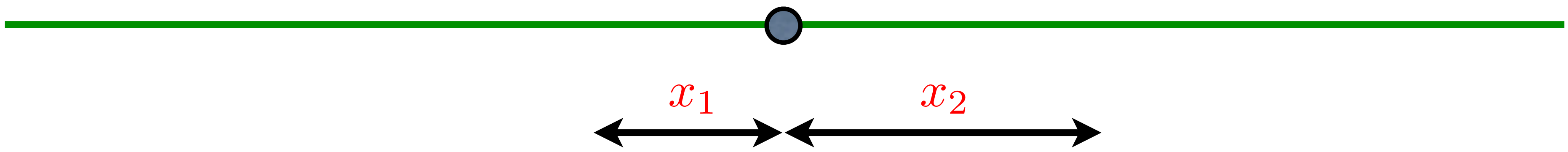
LP formulations of the problem (Demaine et al.)

LP formulations of the problem (Demaine et al.)

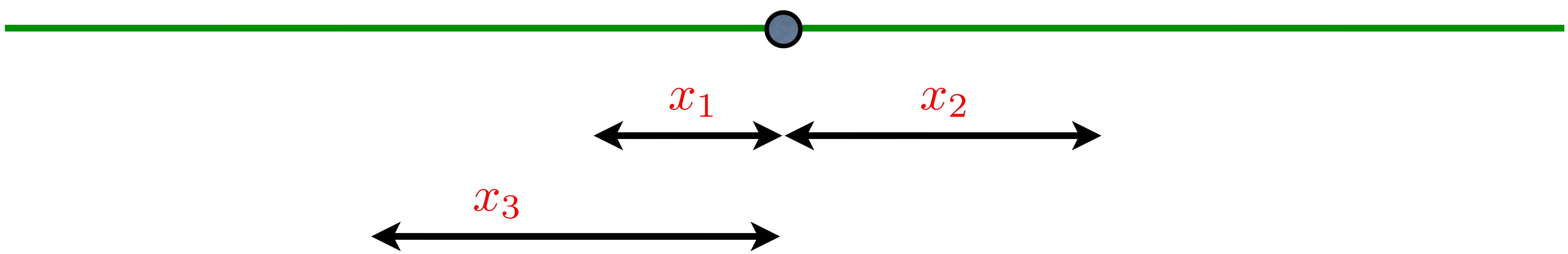
LP formulations of the problem (Demaine et al.)



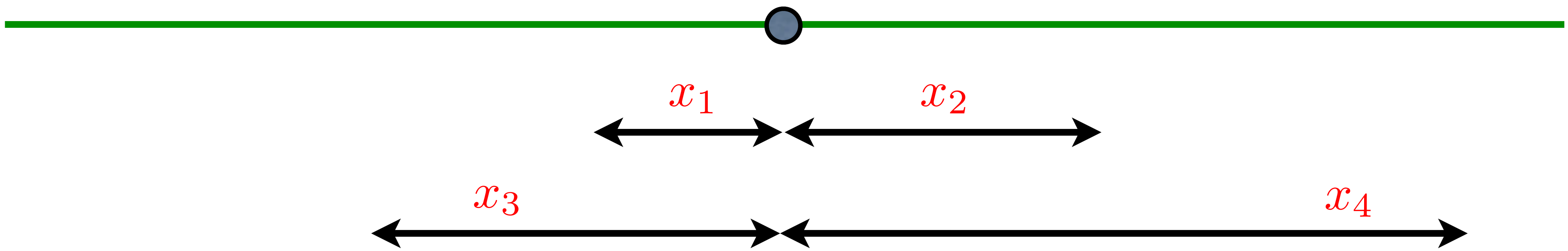
LP formulations of the problem (Demaine et al.)



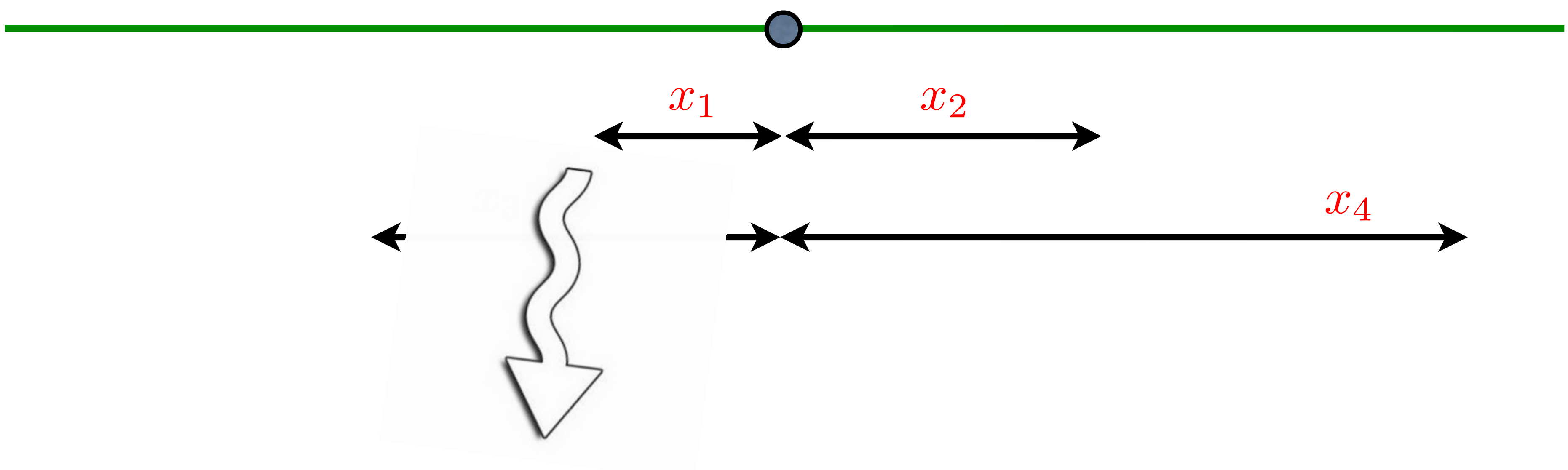
LP formulations of the problem (Demaine et al.)



LP formulations of the problem (Demaine et al.)

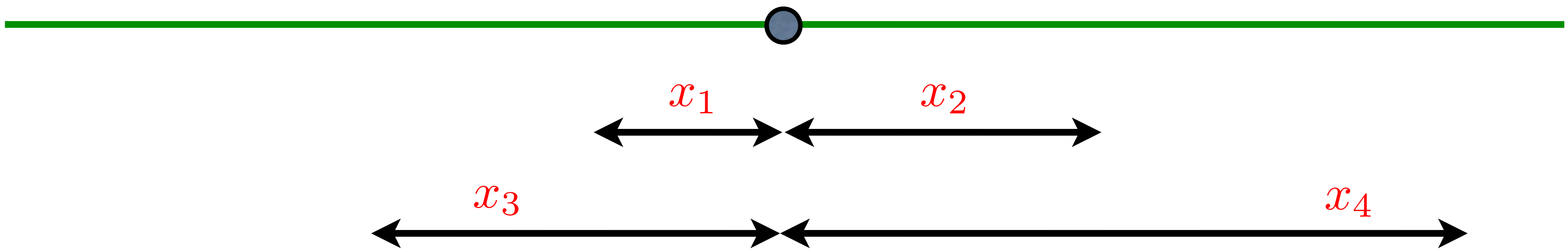


LP formulations of the problem (Demaine et al.)

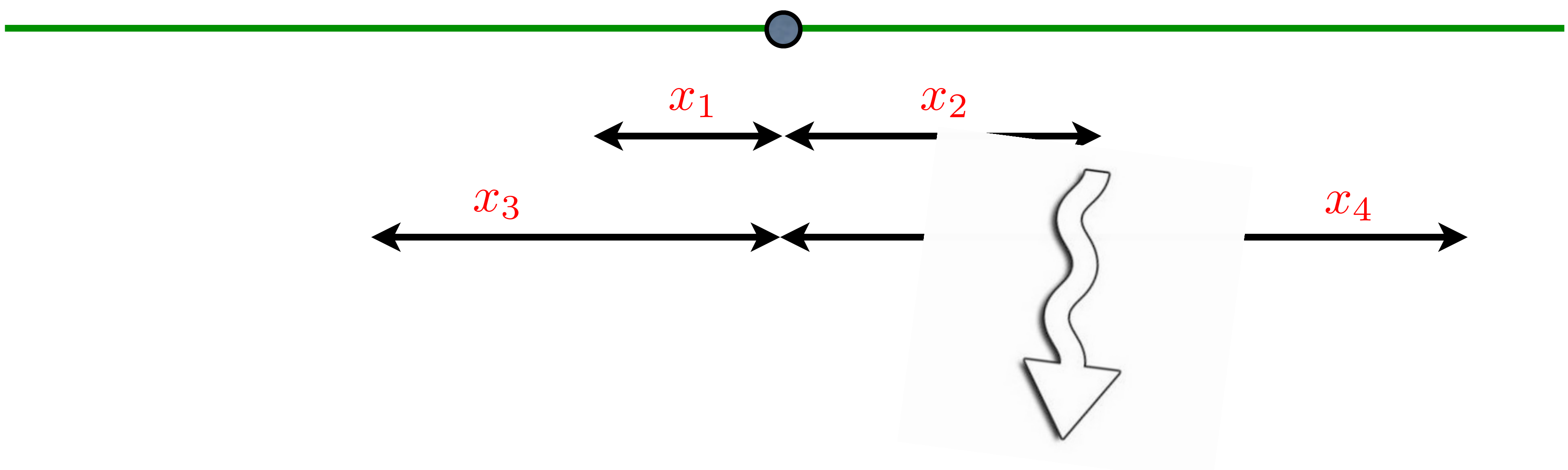


$$2x_1 + 2x_2 + x_1 + 2d \leq 9x_1 + B$$

LP formulations of the problem (Demaine et al.)

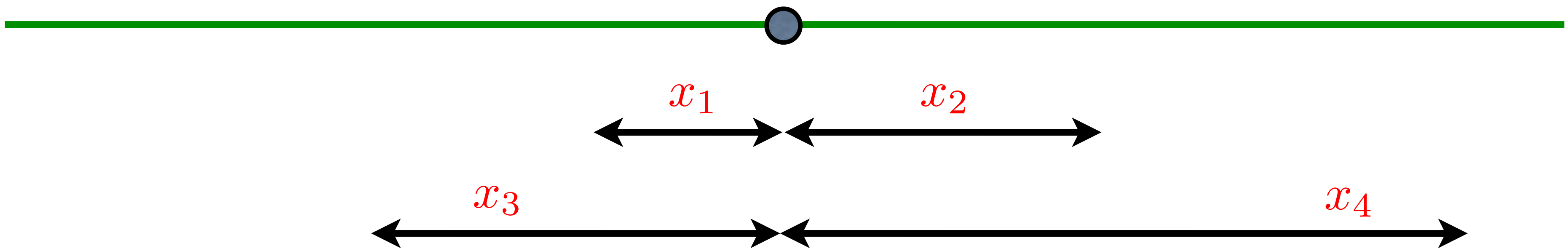


LP formulations of the problem (Demaine et al.)

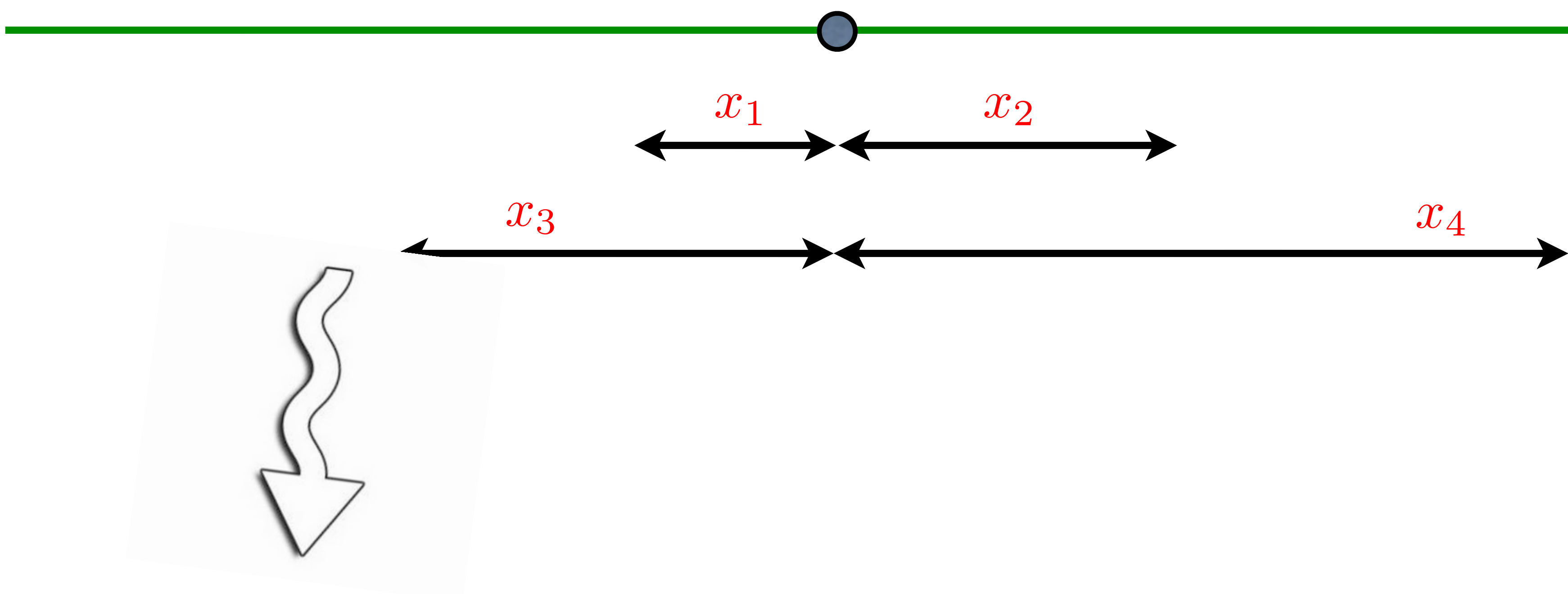


$$2x_1 + 2x_2 + 2x_3 + x_2 + 3d \leq 9x_2 + B$$

LP formulations of the problem (Demaine et al.)

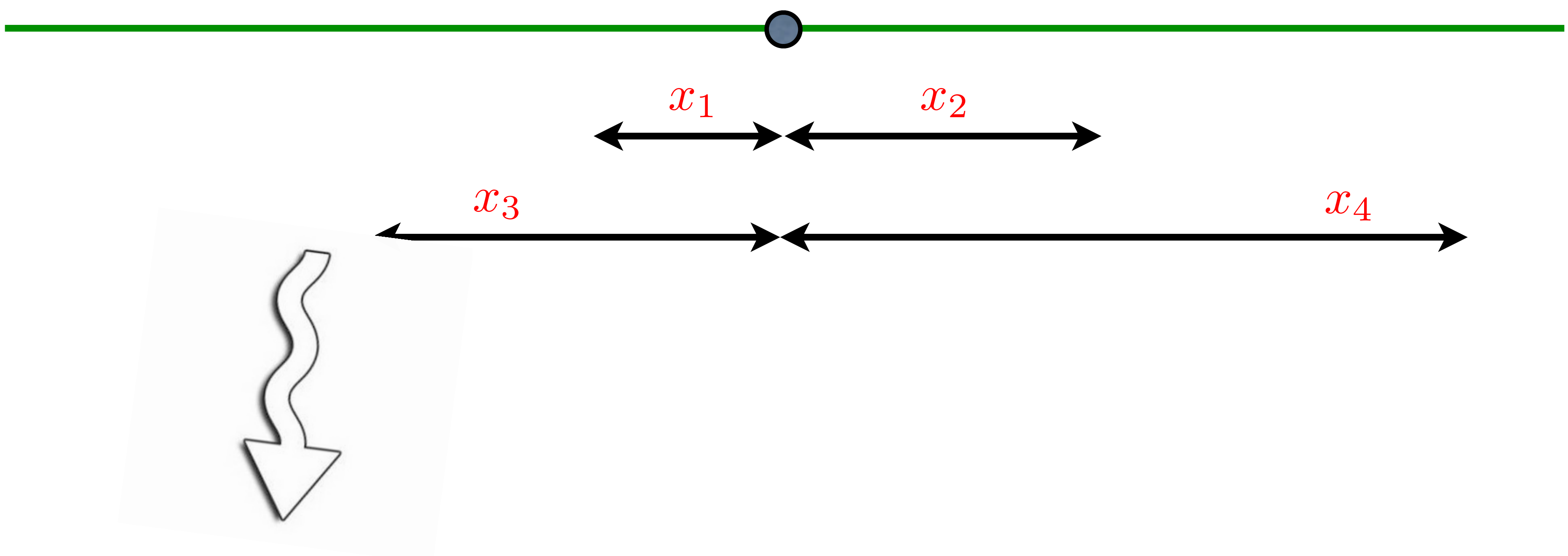


LP formulations of the problem (Demaine et al.)



$$2x_1 + 2x_2 + 2x_3 + 2x_4 + x_3 + 4d \leq 9x_3 + B$$

LP formulations of the problem (Demaine et al.)

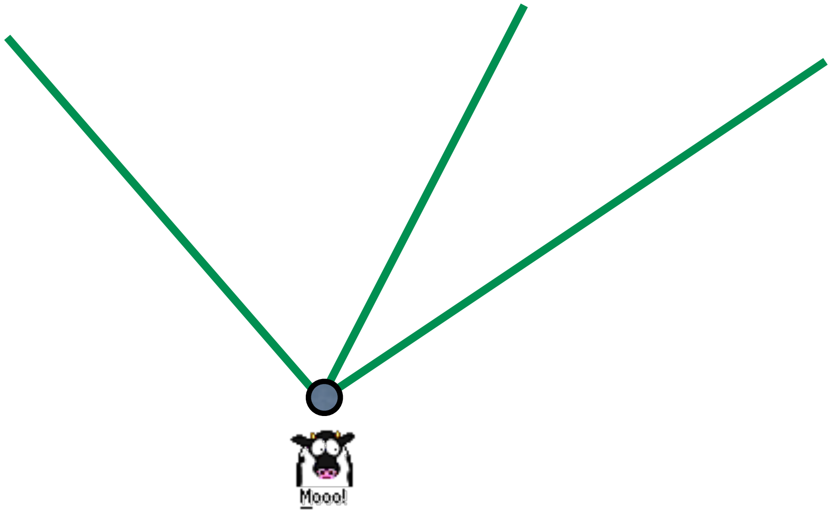


$$2x_1 + 2x_2 + 2x_3 + 2x_4 + x_3 + 4d \leq 9x_3 + B$$

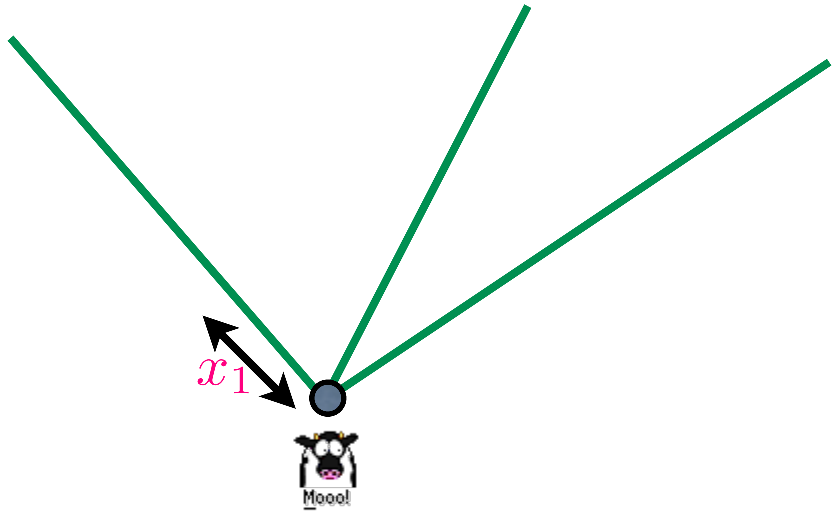
Infinite LP : Min B subject to an infinite number of constraints

Ray searching with turn cost

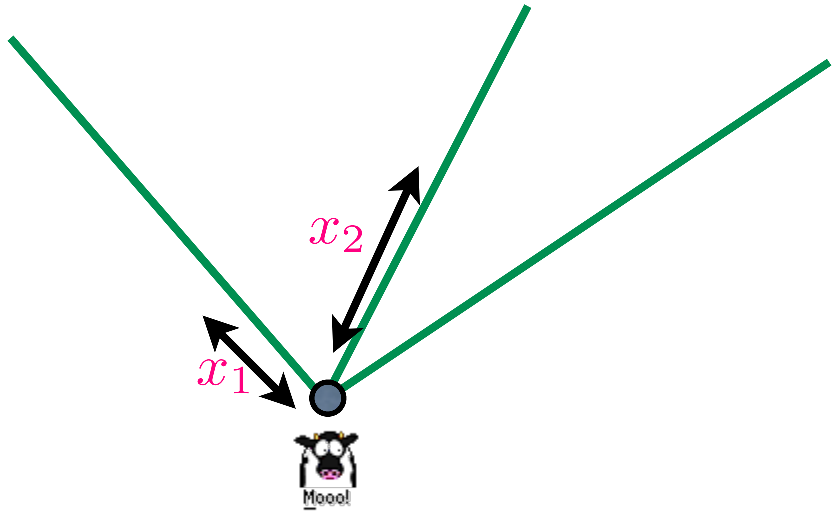
Ray searching with turn cost



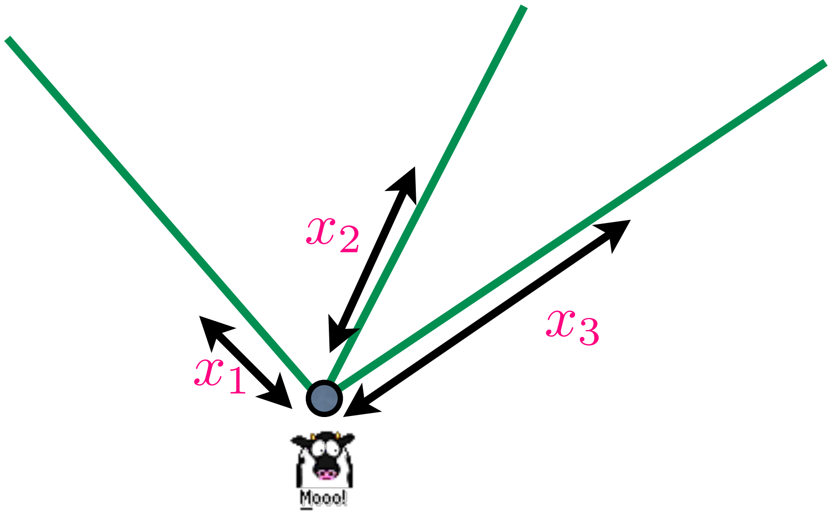
Ray searching with turn cost



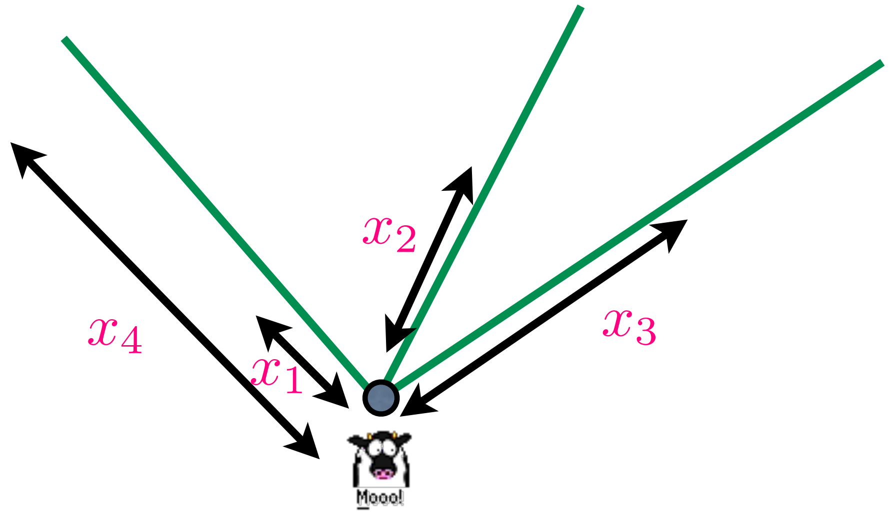
Ray searching with turn cost



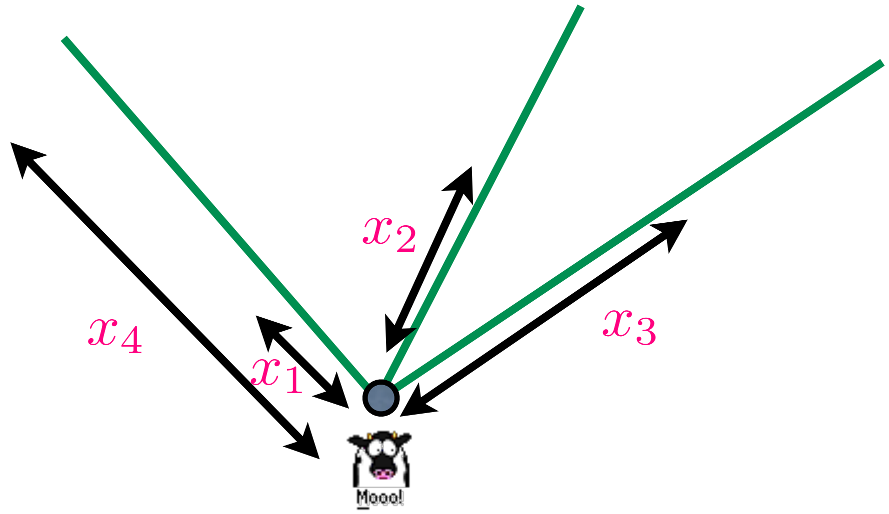
Ray searching with turn cost



Ray searching with turn cost



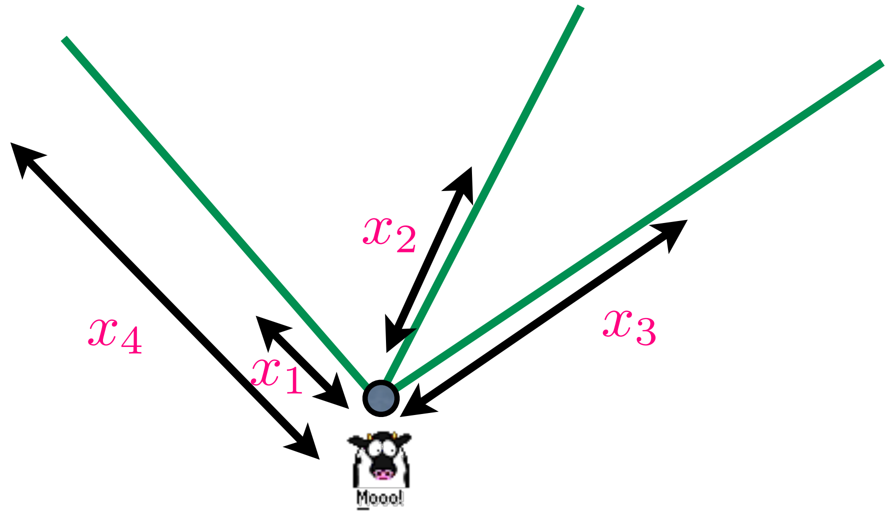
Ray searching with turn cost



$$\text{comp. ratio} = 1 + 2M$$

$$\text{where } M = \frac{b^m}{b-1} \text{ and } b = \frac{m}{m-1}$$

Ray searching with turn cost



$\text{comp. ratio} = 1 + 2M$

where $M = \frac{b^m}{b-1}$ and $b = \frac{m}{m-1}$

Using the strategy $x_i = \frac{d}{2}(b^i - 1)$ yields $B = (M - m)d$

Extension to ray searching (Demaine et al.)

Extension to ray searching (Demaine et al.)

We can obtain an infinite family of LP formulations

$$
\begin{aligned}
\min \quad & B && (P_1)\\
\text{s.t.} \quad & 2\sum_{j=1}^{m-1} x_j - B \leqslant -d(m-1) && \\
& 2\sum_{j=1}^{m+i} x_j - 2Mx_{i+1} - B \leqslant -d(m+i) && \forall i = 0 \ldots k\\
& B, x_1, \ldots, x_{m+k} \geqslant 0, &&
\end{aligned}
$$

Extension to ray searching (Demaine et al.)

We can obtain an infinite family of LP formulations

$$
\begin{aligned}
\min \quad & B & (P_1) \\
\text{s.t.} \quad & 2\sum_{j=1}^{m-1} x_j - B \leqslant -d(m-1) & \\
& 2\sum_{j=1}^{m+i} x_j - 2Mx_{i+1} - B \leqslant -d(m+i) & \forall i = 0 \ldots k \\
& B, x_1, \ldots, x_{m+k} \geqslant 0, &
\end{aligned}
$$

with corresponding dual LPs

$$
\begin{aligned}
\max \quad & \left((m-1)z + \sum_{i=0}^{k} y_i(m+i)\right) d & (D_1) \\
\text{s.t.} \quad & z + \sum_{i=0}^{k} y_i \leqslant 1 & \\
& \begin{Bmatrix} z \ , & j \leq m-1 \\ 0 \ , & \text{otherwise} \end{Bmatrix} + \sum_{i=\max(0,j-m)}^{k} y_i - My_{j-1} \geqslant 0 & \forall j = 1 \ldots m+k \\
& z, y_0, \ldots, y_k \geqslant 0 &
\end{aligned}
$$


THE REST OF YOUR LIFE...
AVERAGE
MEMORABLE

Dealing with the infinite LP

Dealing with the infinite LP

Demaine et al. focus on the infinite dual LP

$$
\begin{aligned}
\max \quad & \Big((m-1)z + \sum\nolimits_{i=0}^{\infty} y_i(m+i)\Big)\, d && (D_1^{\infty}) \\
\text{s.t.} \quad & z + \sum\nolimits_{i=0}^{\infty} y_i \leqslant 1 \\
& \left\{ \begin{array}{ll} z\ , & j \leq m-1 \\ 0\ , & \text{otherwise} \end{array} \right\} + \sum\nolimits_{i=\max(0,j-m)}^{\infty} y_i - M y_i \geqslant 0 && \forall j = 1,2,\ldots \\
& z, y_0, y_1, \ldots \geqslant 0
\end{aligned}
$$

Dealing with the infinite LP

Demaine et al. focus on the infinite dual LP

$$
\begin{aligned}
\max \quad & \left((m-1)z + \sum_{i=0}^{\infty} y_i(m+i)\right) d && (D_1^{\infty}) \\
\text{s.t.} \quad & z + \sum_{i=0}^{\infty} y_i \leqslant 1 \\
& \left\{ \begin{array}{ll} z \ , & j \leq m-1 \\ 0 \ , & \text{otherwise} \end{array} \right\} + \sum_{i=\max(0,j-m)}^{\infty} y_i - M y_i \geqslant 0 && \forall j = 1, 2, \ldots \\
& z, y_0, y_1, \ldots \geqslant 0
\end{aligned}
$$

and argue that the following is a feasible solution to the infinite dual LP

$$z = \frac{m}{M} \qquad y_0 = y_1 = y_{m-2} = \ldots = \frac{1}{M}, y_{m-1} = \frac{1}{M}(1-z)$$

$$\text{and } y_i = y_{i-1} - \frac{1}{M} y_{i-m}, \text{ for all } i \geq m,$$

Dealing with the infinite LP

Demaine et al. focus on the infinite dual LP

$$
\begin{aligned}
\max \quad & \left((m-1)z + \sum_{i=0}^{\infty} y_i(m+i)\right) d & (D_1^{\infty}) \\
\text{s.t.} \quad & z + \sum_{i=0}^{\infty} y_i \leqslant 1 \\
& \left\{ \begin{array}{ll} z \ , & j \leq m-1 \\ 0 \ , & \text{otherwise} \end{array} \right\} + \sum_{i=\max(0,j-m)}^{\infty} y_i - M y_i \geqslant 0 \quad \forall j = 1, 2, \ldots \\
& z, y_0, y_1, \ldots \geqslant 0
\end{aligned}
$$

and argue that the following is a feasible solution to the infinite dual LP

$$z = \frac{m}{M} \qquad y_0 = y_1 = y_{m-2} = \ldots = \frac{1}{M}, y_{m-1} = \frac{1}{M}(1-z)$$

$$\text{and } y_i = y_{i-1} - \frac{1}{M} y_{i-m}, \text{ for all } i \geq m,$$

which yields an objective value of (M-m)d (which is optimal)

but there is a problem.....

but there is a problem.....

- One can show that a different feasible solution to the infinite dual LP yields an objective of $Md > (M-m)d = \text{upper bound}$
- This means we cannot trust the infinite dual LP
- Instead we should work on finite LPs (and obtain the best bound at the limit)

but there is a problem.....

- One can show that a different feasible solution to the infinite dual LP yields an objective of $Md > (M - m)d = \text{upper bound}$

- This means we cannot trust the infinite dual LP

- Instead we should work on finite LPs (and obtain the best bound at the limit)

- Finding the best dual solution : establishing some properties of the linear recurrence **y**

- More precisely: for which initial data does **y** become eventually negative?

A related problem in AI : Anytime algorithms

A related problem in AI : Anytime algorithms

- Quality of output improves as a function of time
  [Dean and Boddy 1987], [Russell and Zilberstein 1991]

- Interruptible algorithms: Can be interrupted at any time, must be able to produce a solution

- Contract algorithms: Prespecified amount of execution time given as part of the algorithm's input

A related problem in AI : Anytime algorithms

- Quality of output improves as a function of time
  [Dean and Boddy 1987], [Russell and Zilberstein 1991]

- Interruptible algorithms: Can be interrupted at any time, must be able to produce a solution

  (+) flexible!
  (-) complicated, no performance guarantees

- Contract algorithms: Prespecified amount of execution time given as part of the algorithm's input

A related problem in AI : Anytime algorithms

- Quality of output improves as a function of time
  [Dean and Boddy 1987], [Russell and Zilberstein 1991]

- Interruptible algorithms: Can be interrupted at any time, must be able to produce a solution

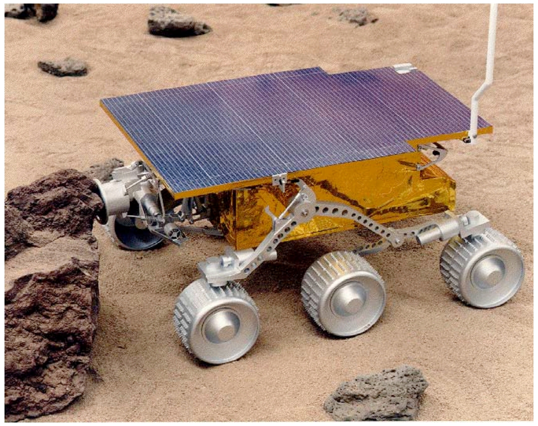
  (+) flexible!
  (-) complicated, no performance guarantees

- Contract algorithms: Prespecified amount of execution time given as part of the algorithm's input

  (-) less flexible
  (+) easier to program, analyze

From contract to interruptible algorithms

- Main goal: "Black-box" techniques for turning every contract algorithm to its interruptible version
- Establish measures of how good this simulation is
- Find efficient simulations in this measure
- Survey: "Using anytime algorithms in intelligent systems" (Zilberstein)

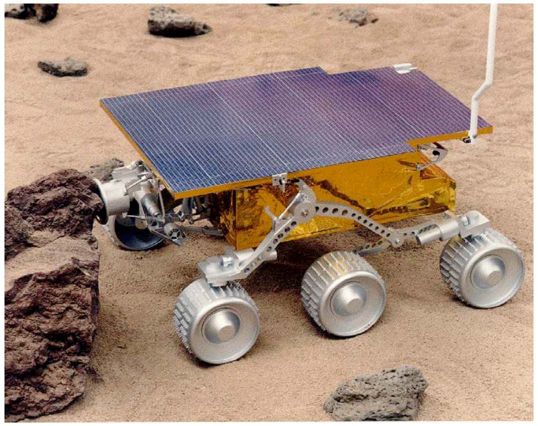


"Plan synthesis must have anytime, incremental characteristics. It should be possible to stop a plan synthesis algorithm at any time during its execution and expect useful results. One should expect the "quality" of the results to improve continuously as a function of time."

John Bressina and Mark Drummond
NASA Ames Research Center

From contract to interruptible algorithms

- Main goal: "Black-box" techniques for turning every contract algorithm to its interruptible version
- Establish measures of how good this simulation is
- Find efficient simulations in this measure
- Survey: "Using anytime algorithms in intelligent systems" (Zilberstein)



"Plan synthesis must have anytime, incremental characteristics. It should be possible to stop a plan synthesis algorithm at any time during its execution and expect useful results. One should expect the "quality" of the results to improve continuously as a function of time."

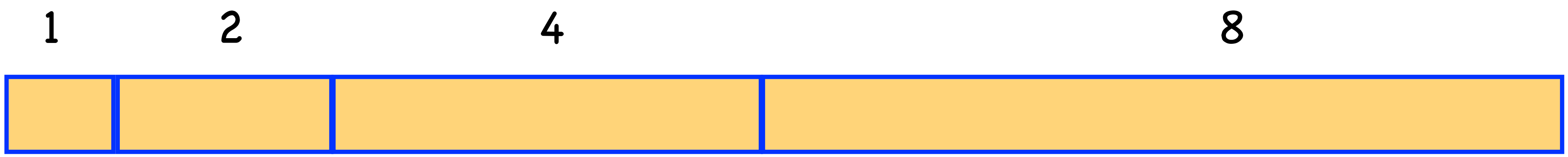
John Bressina and Mark Drummond
NASA Ames Research Center

A quick example

- Suppose we are given a contract algorithm
- Run the algorithm for 1 step, then for 2 steps, then for 4 steps and so forth

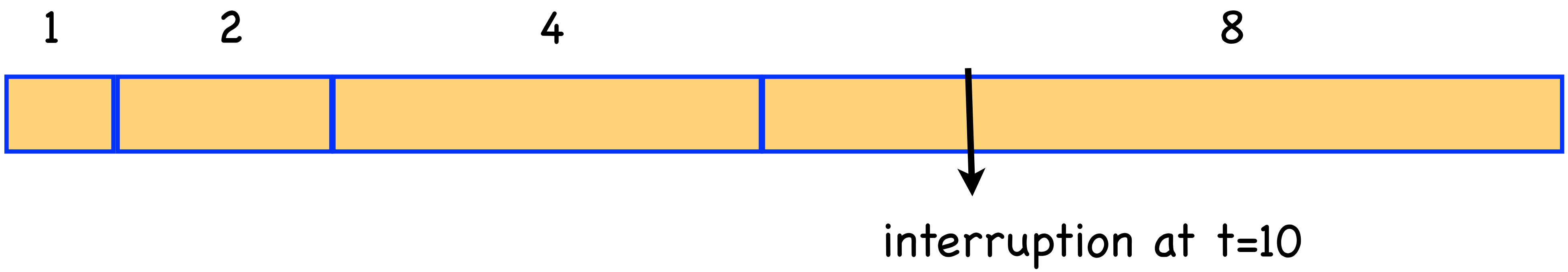
A quick example

- Suppose we are given a contract algorithm
- Run the algorithm for 1 step, then for 2 steps, then for 4 steps and so forth



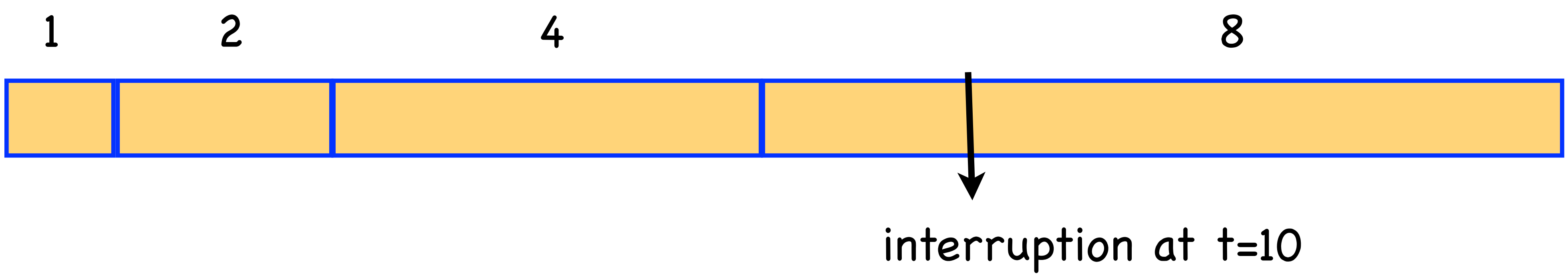
A quick example

- Suppose we are given a contract algorithm
- Run the algorithm for 1 step, then for 2 steps, then for 4 steps and so forth



A quick example

- Suppose we are given a contract algorithm
- Run the algorithm for 1 step, then for 2 steps, then for 4 steps and so forth



- In hindsight we could have run the algorithm for 10 units, but the best we achieved is a running time of 4
- Inefficiency = $\max_t \frac{t}{\text{longest contract finished by } t}$

Conclusion and outlook

Conclusion and outlook

- We study online search problems with turn cost using infinite LP formulations
- Caveats of duality in infinite LPs
- Further applications : Search problems in unbounded domains

  Resource allocation problems with infinite horizon
- Many other variants of ray searching remain open

Conclusion and outlook

- We study online search problems with turn cost using infinite LP formulations
- Caveats of duality in infinite LPs
- Further applications : Search problems in unbounded domains

  Resource allocation problems with infinite horizon
- Many other variants of ray searching remain open

Thank you!