Further Connections between Contract-Scheduling and Ray-Searching Problems

Spyros Angelopoulos

CNRS and University Pierre and Marie Curie

Outline and motivation

Outline and motivation

- Two well-studied problems:
  one from OR/TCS, the other from AI

Outline and motivation

- Two well-studied problems:
  one from OR/TCS, the other from AI

- The first problem: searching on a star
  (unbounded domain)

Outline and motivation

- Two well-studied problems:
  one from OR/TCS, the other from AI

- The first problem: searching on a star
  (unbounded domain)

- The second problem: how to obtain
  efficient anytime algorithms

Outline and motivation

- Two well-studied problems:
  one from OR/TCS, the other from AI

- The first problem: searching on a star
  (unbounded domain)

- The second problem: how to obtain
  efficient anytime algorithms

- Objective: address variants of these
  problems with a common approach

Outline and motivation

- Two well-studied problems:
  one from OR/TCS, the other from AI
- The first problem: searching on a star
  (unbounded domain)
- The second problem: how to obtain
  efficient anytime algorithms
- Objective: address variants of these
  problems with a common approach


USNews NEWS

News | Opinion | National Issues | Special Reports | Cartoons | Photos | The Report

Ken Walsh's Washington | Newsgram | Washington Whispers | At the Edge | Data Mine | The Run 20

Musk, Hawking Call to Ban Robots That Could Kill You All by Themselves

A massive coalition of leading thinkers and innovators has a chilling warning about the future of warfare.

An unmanned Predator drone flies over Afghanistan in 2009. A group of thought leaders wants to prevent the development of artificially intelligent machines that could attack with much more autonomy.

By Tom Risen | July 27, 2015 | 5:52 p.m. EDT

The U.S. military is investing heavily in drones as the future of war, but a group of 1,000 scientific leaders wants to draw the line at developing weapons that essentially can think for themselves.

Futurists like Elon Musk, scientists like Stephen Hawking and innovators like Apple co-founder Steve Wozniak have signed an open letter calling for a ban on the use of autonomous weapons, with the petition scheduled to be unveiled at the opening of this week's International Joint Conference on Artificial Intelligence in Argentina. Hawking and Musk have previously urged scientists to be wary while developing artificial intelligence, cautioning that autonomous machines may not be able to understand the possible good and evil consequences of their actions.

Outline and motivation

- Two well-studied problems: one from OR/TCS, the other from AI
- The first problem: searching on a star (unbounded domain)
- The second problem: how to obtain efficient anytime algorithms
- Objective: address variants of these problems with a common approach


USNews NEWS

News, Op

News | Opinion | National Issues | Special Reports | Cartoons | Photos | The Report

Ken Walsh's Washington | Newsgram | Washington Whispers | At the Edge | Data Mine | The Run 20

Musk, Hawking Call to Ban Robots That Could Kill You All by Themselves

A massive coalition of leading thinkers and innovators has a chilling warning about the future of warfare.

An unmanned Predator drone flies over Afghanistan in 2009. A group of thought leaders wants to prevent the development of artificially intelligent machines that could attack with much more autonomy.

By Tom Risen | July 27, 2015 | 5:52 p.m. EDT

+ More

The U.S. military is investing heavily in drones as the future of war, but a group of 1,000 scientific leaders wants to draw the line at developing weapons that essentially can think for themselves.

Futurists like Elon Musk, scientists like Stephen Hawking and innovators like Apple co-founder Steve Wozniak have signed an open letter calling for a ban on the use of autonomous weapons, with the petition scheduled to be unveiled at the opening of this week's International Joint Conference on Artificial Intelligence in Argentina. Hawking and Musk have previously urged scientists to be wary while developing artificial intelligence, cautioning that autonomous machines may not be able to understand the possible good and evil consequences of their actions.

Outline and motivation

- Two well-studied problems:
  one from OR/TCS, the other from AI

- The first problem: searching on a star (unbounded domain)

- The second problem: how to obtain efficient anytime algorithms

- Objective: address variants of these problems with a common approach


U.S.News & World Report NEWS News, Op

News Opinion National Issues Special Reports Cartoons Photos The Report

Ken Walsh's Washington Newsgram Washington Whispers At the Edge Data Mine The Run 20

Musk, Hawking Call to Ban Robots That Could Kill You All by Themselves

A massive coalition of leading thinkers and innovators has a chilling warning about the future of warfare.

An unmanned Predator drone flies over Afghanistan in 2009. A group of thought leaders wants to prevent the development of artificially intelligent machines that could attack with much more autonomy.

By Tom Risen | July 27, 2015 | 5:52 p.m. EDT

The U.S. military is investing heavily in drones as the future of war, but a group of 1,000 scientific leaders wants to draw the line at developing weapons that essentially can think for themselves.

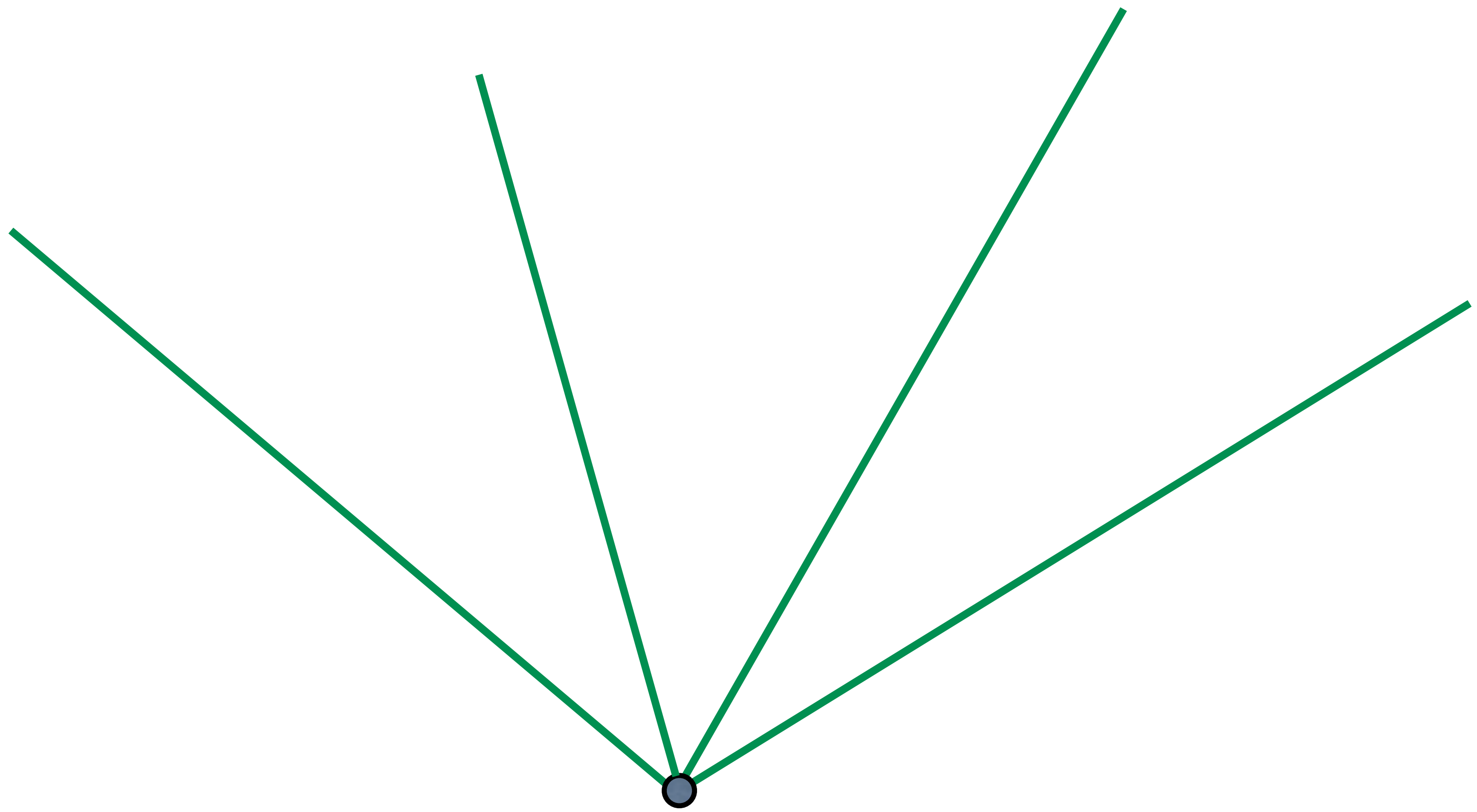
Futurists like Elon Musk, scientists like Stephen Hawking and innovators like Apple co-founder Steve Wozniak have signed an open letter calling for a ban on the use of autonomous weapons, with the petition scheduled to be unveiled at the opening of this week's International Joint Conference on Artificial Intelligence in Argentina. Hawking and Musk have previously urged scientists to be wary while developing artificial intelligence, cautioning that autonomous machines may not be able to understand the possible good and evil consequences of their actions.

Robots in this presentation are benign!*

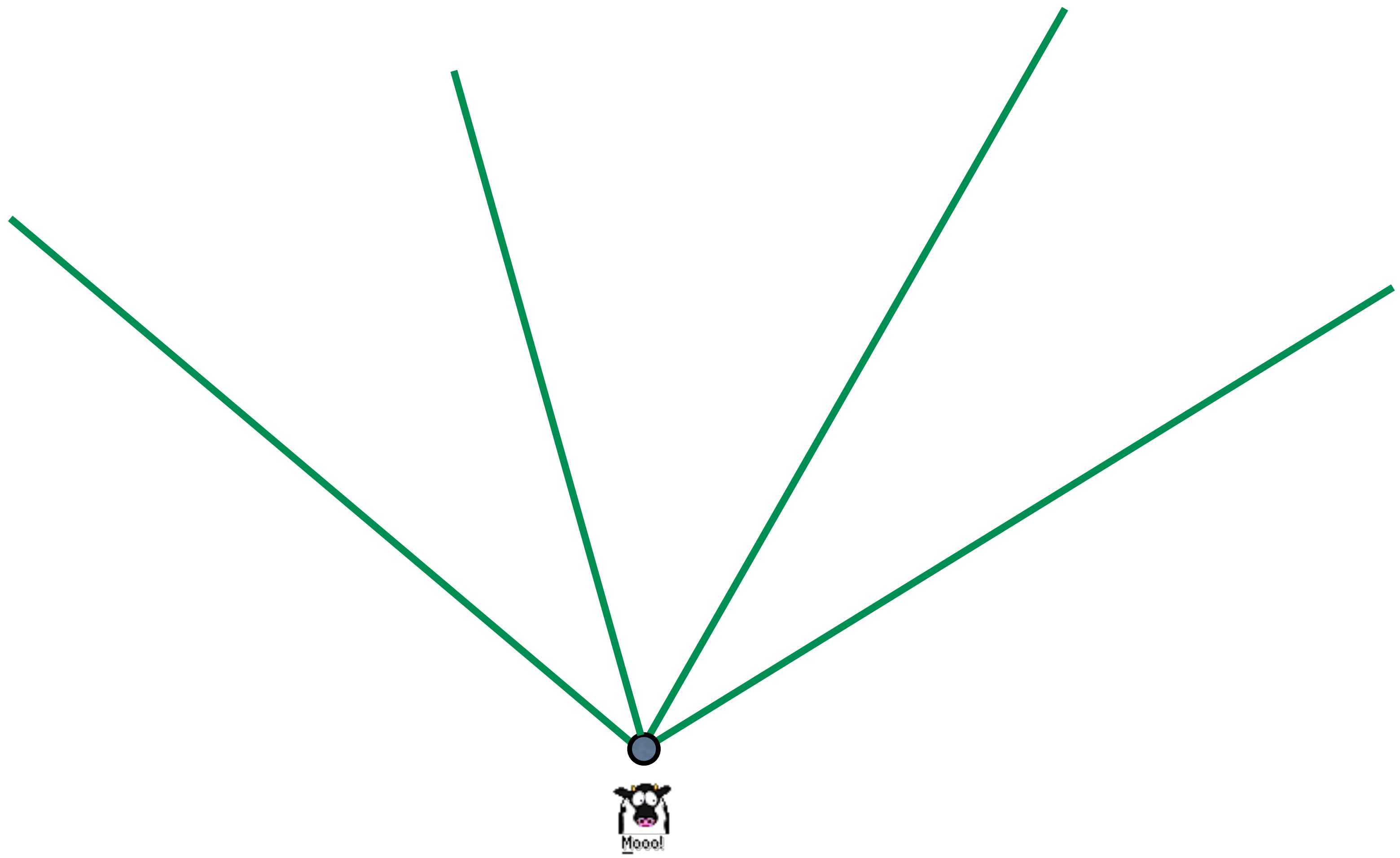
*certain conditions may apply

The first problem: Ray searching

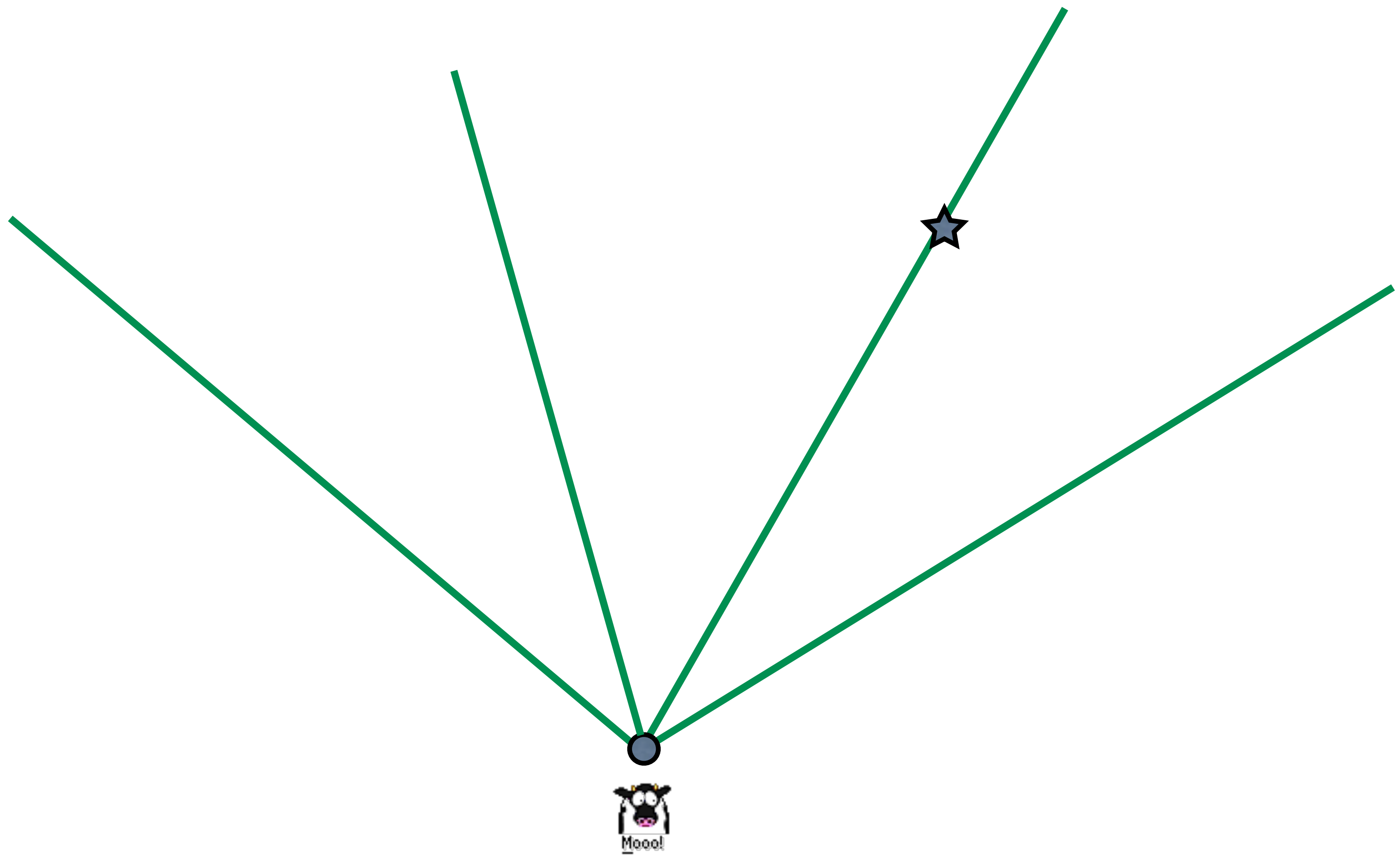
The first problem: Ray searching



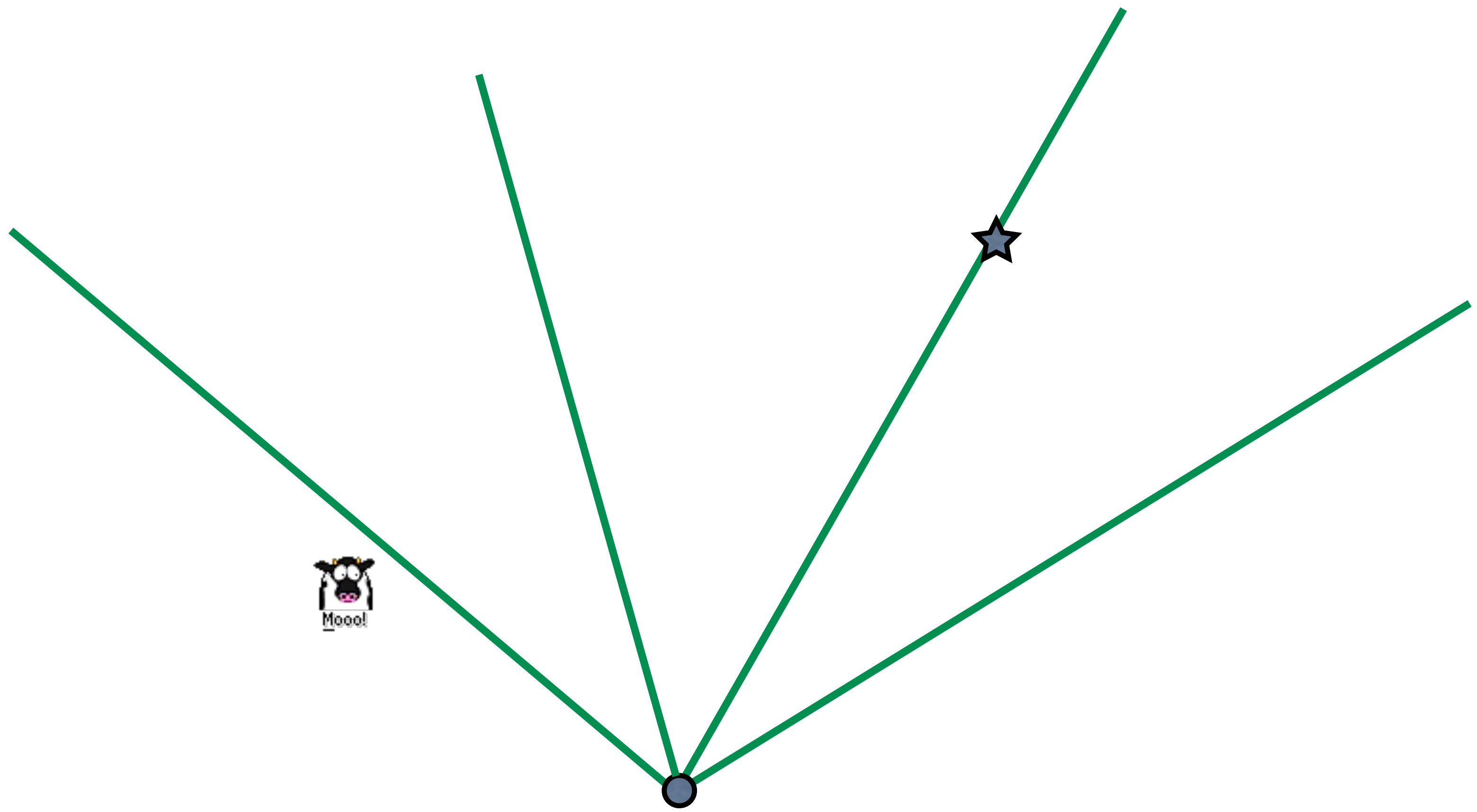
The first problem: Ray searching



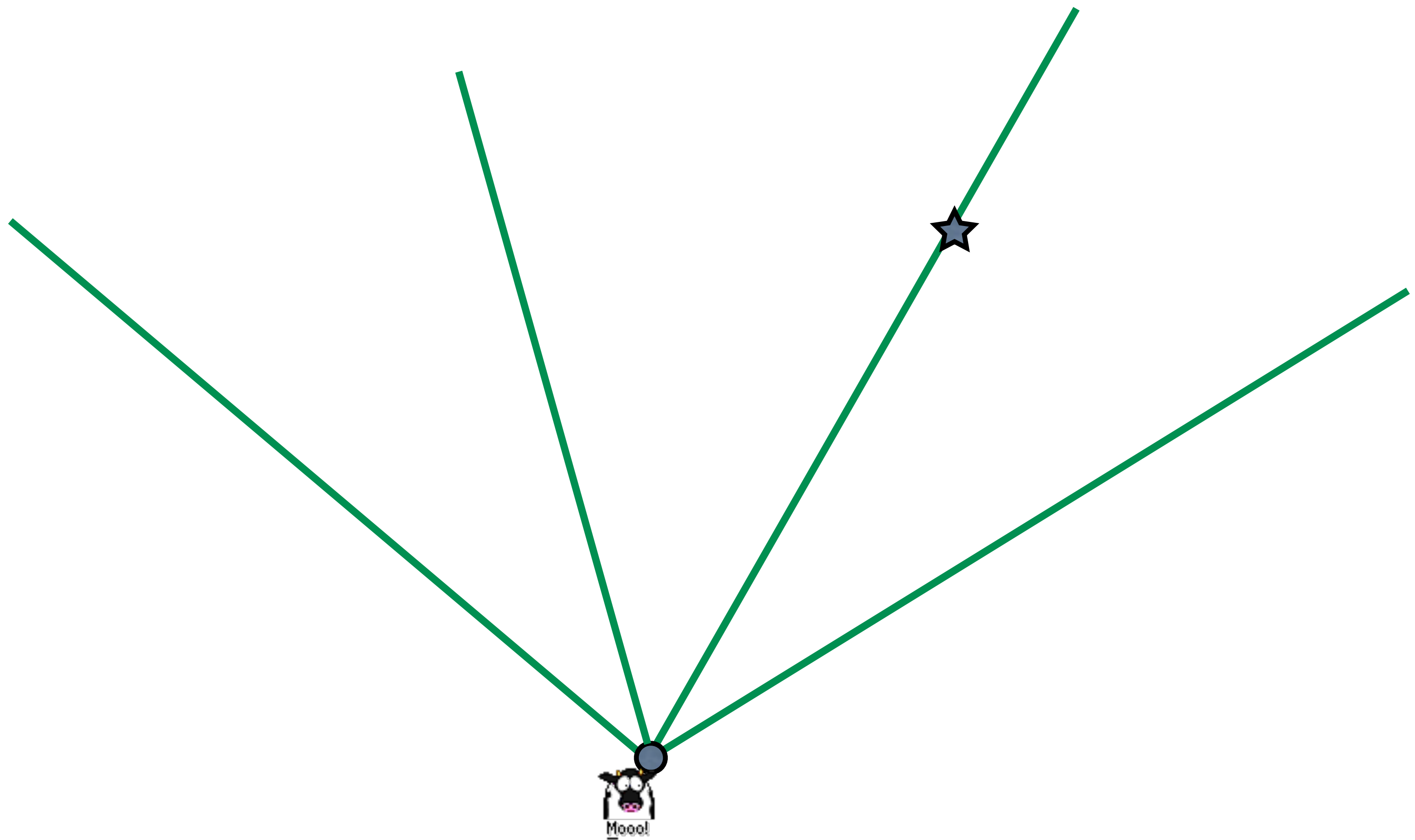
The first problem: Ray searching



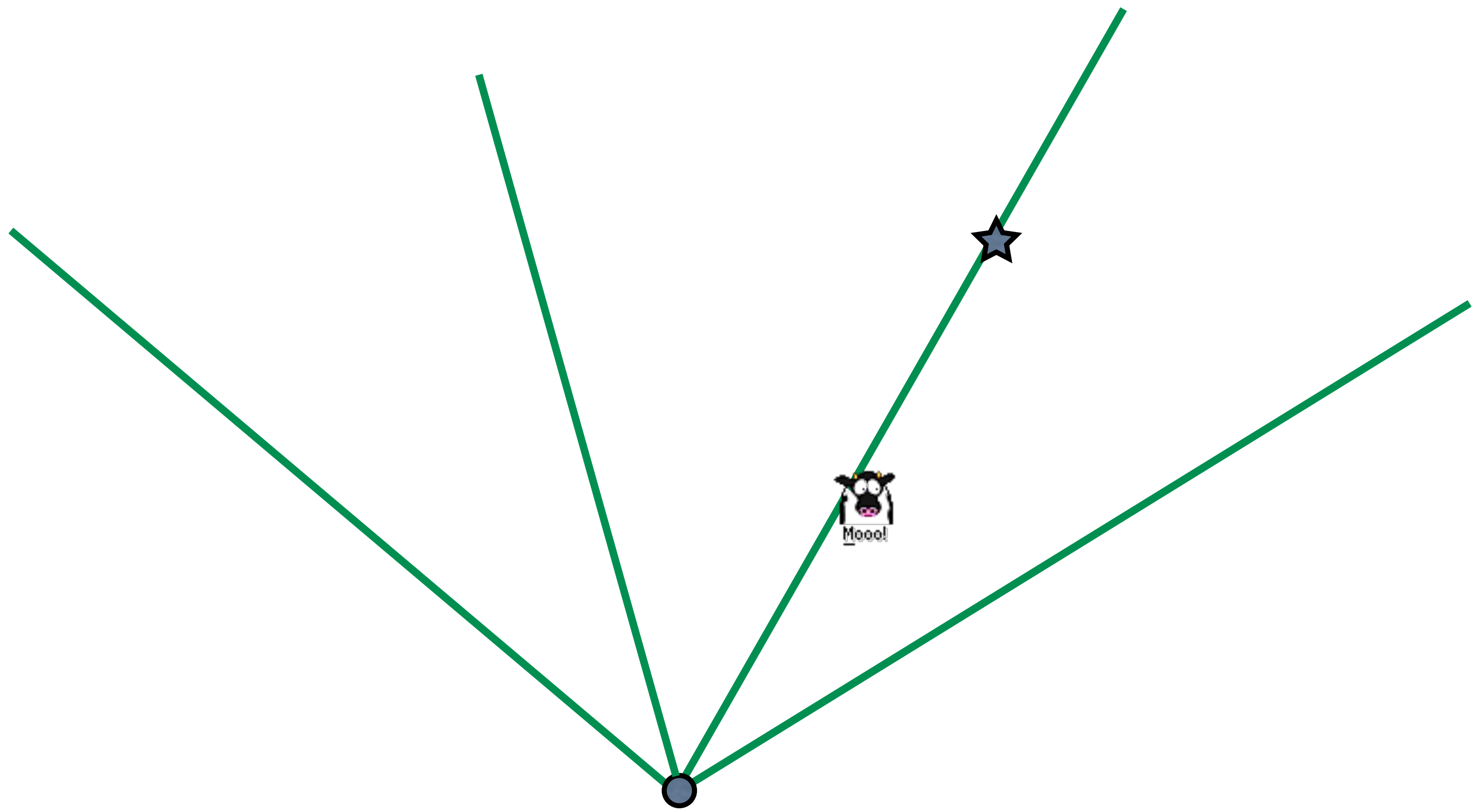
The first problem: Ray searching



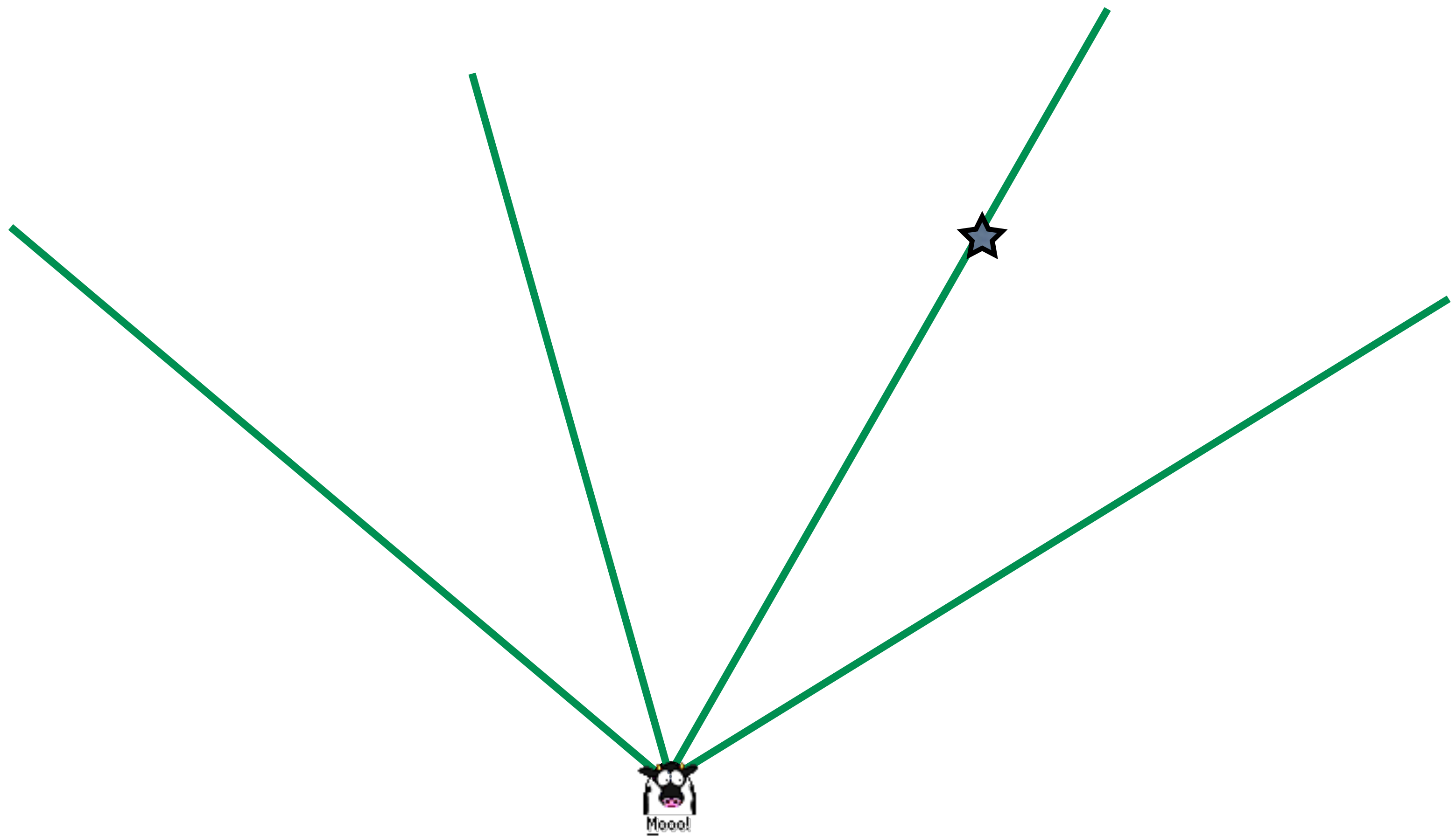
The first problem: Ray searching



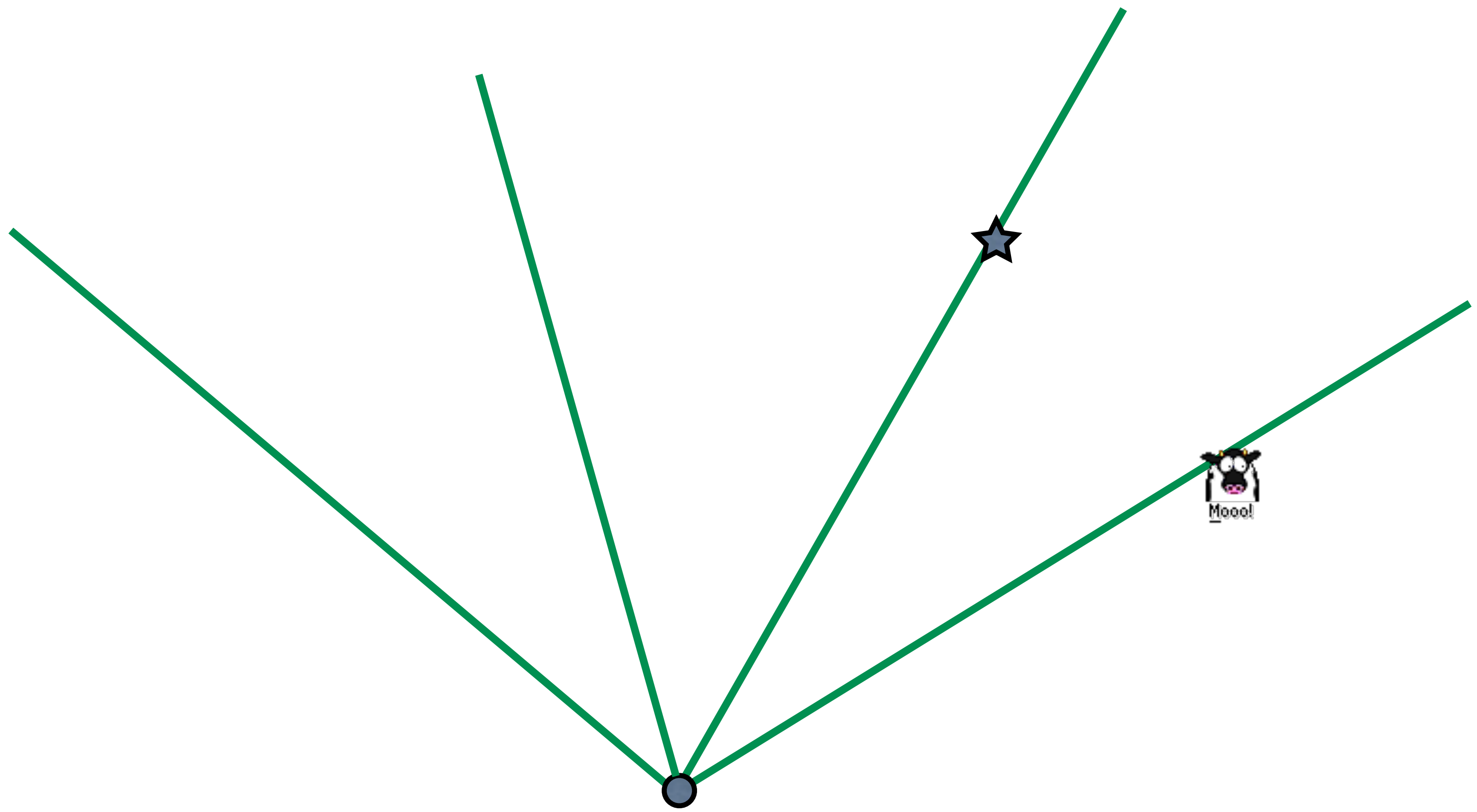
The first problem: Ray searching



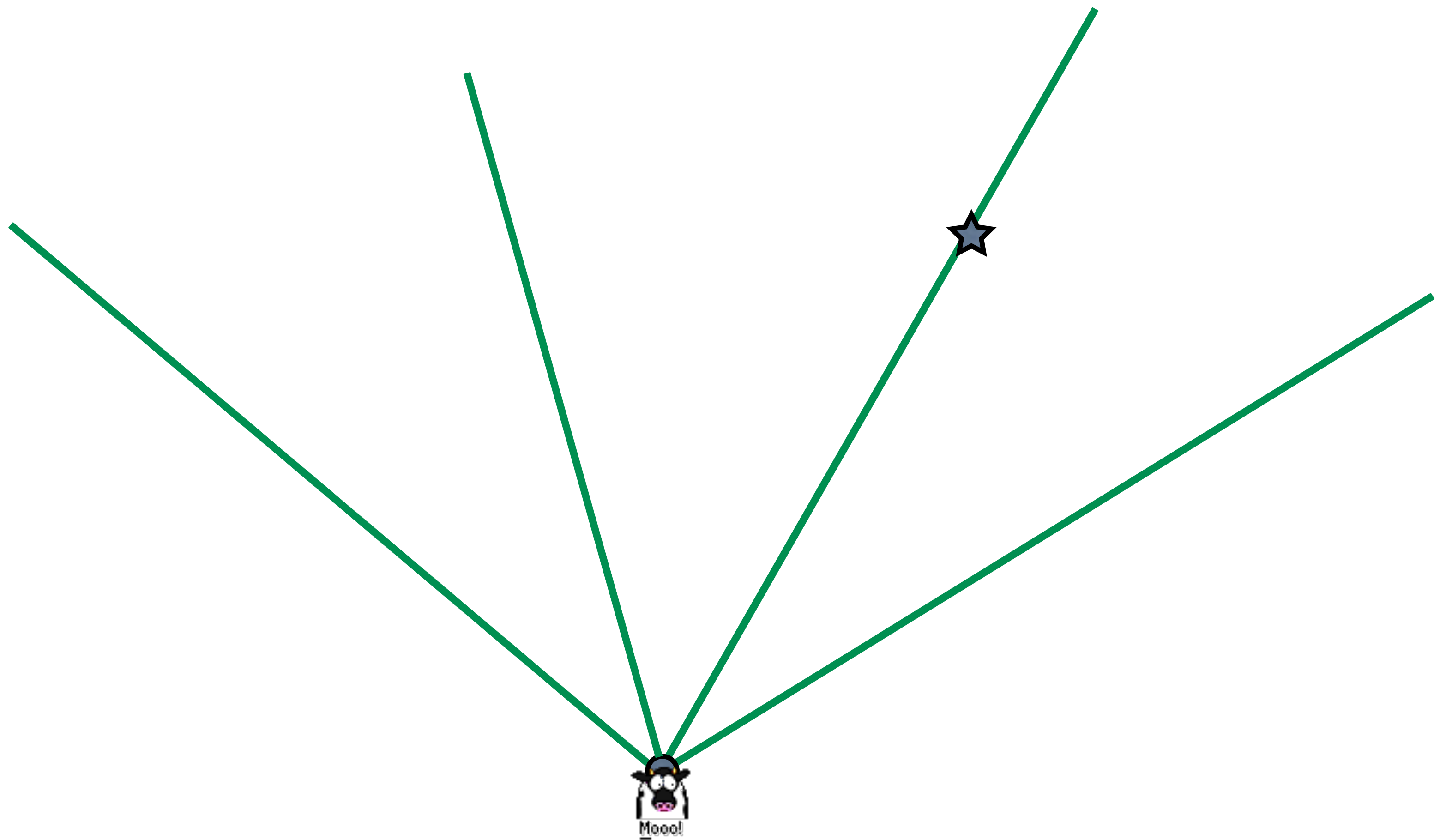
The first problem: Ray searching



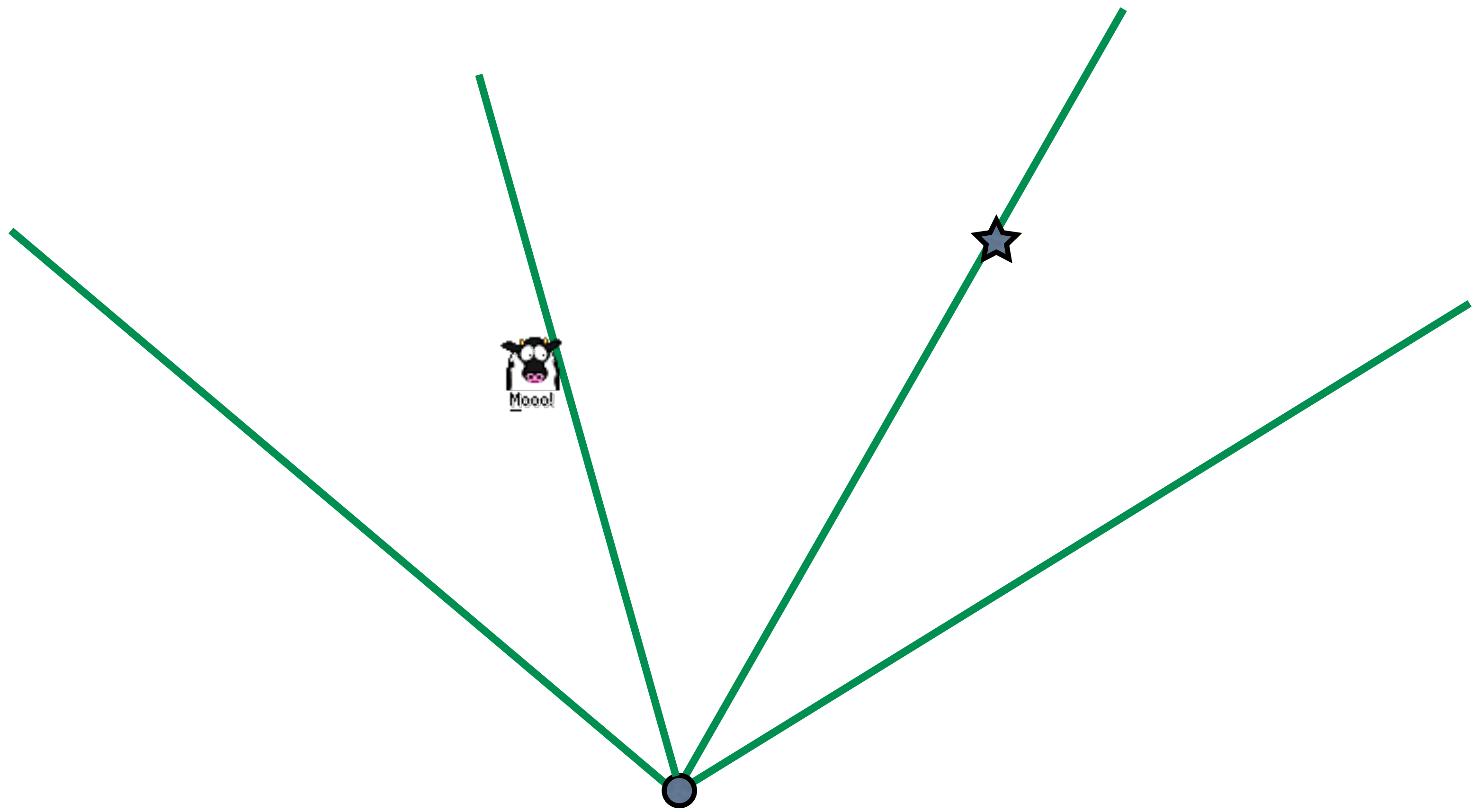
The first problem: Ray searching



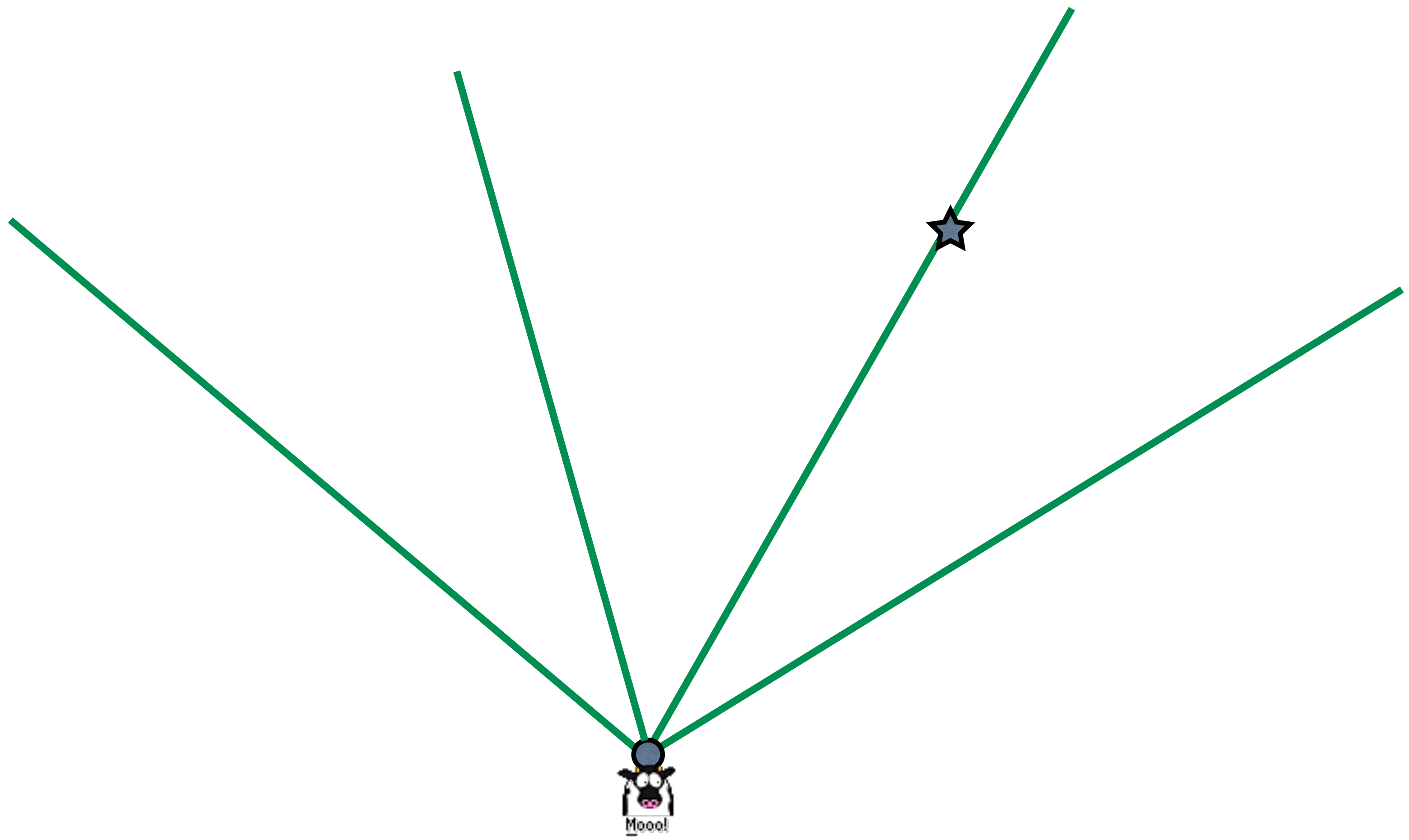
The first problem: Ray searching



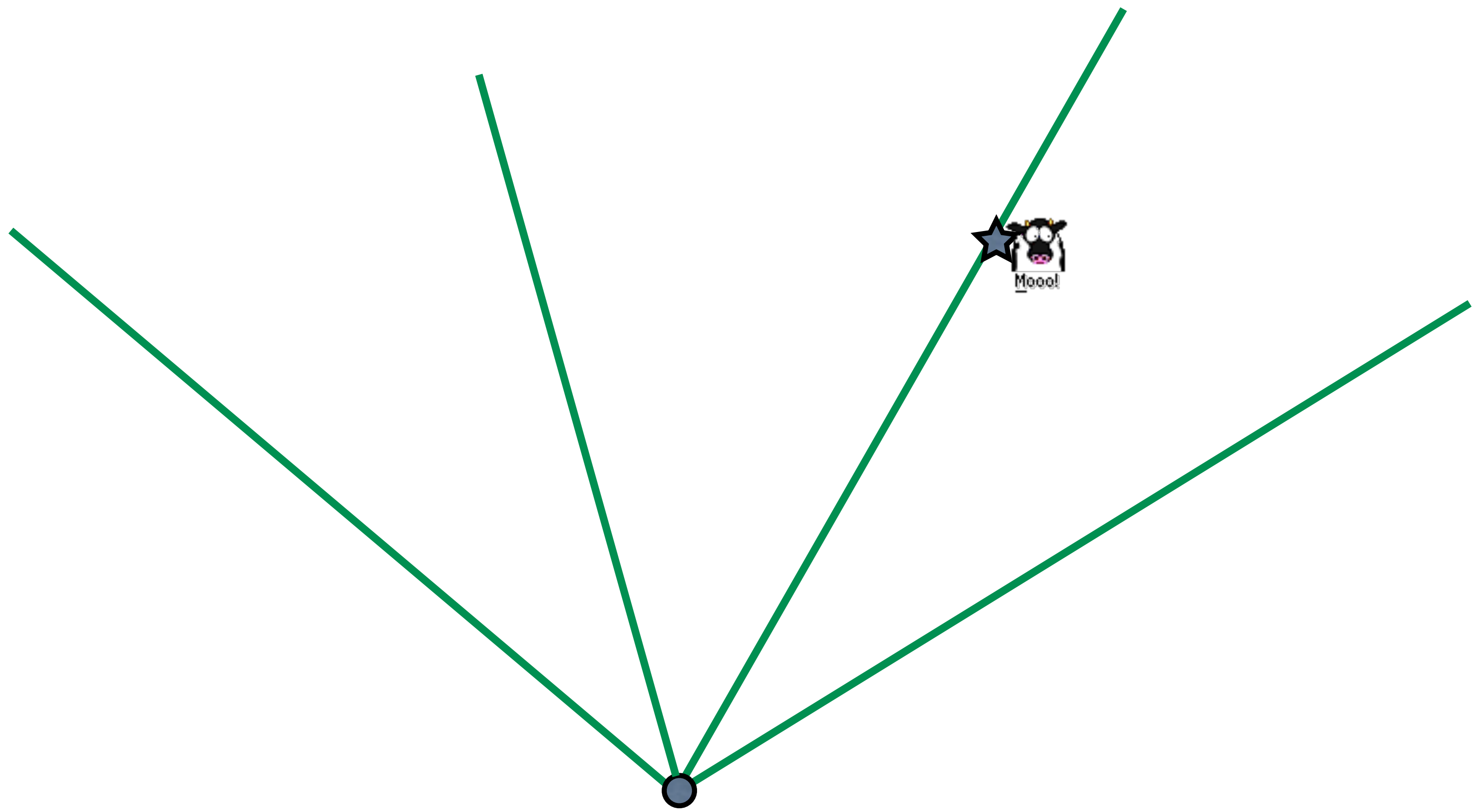
The first problem: Ray searching



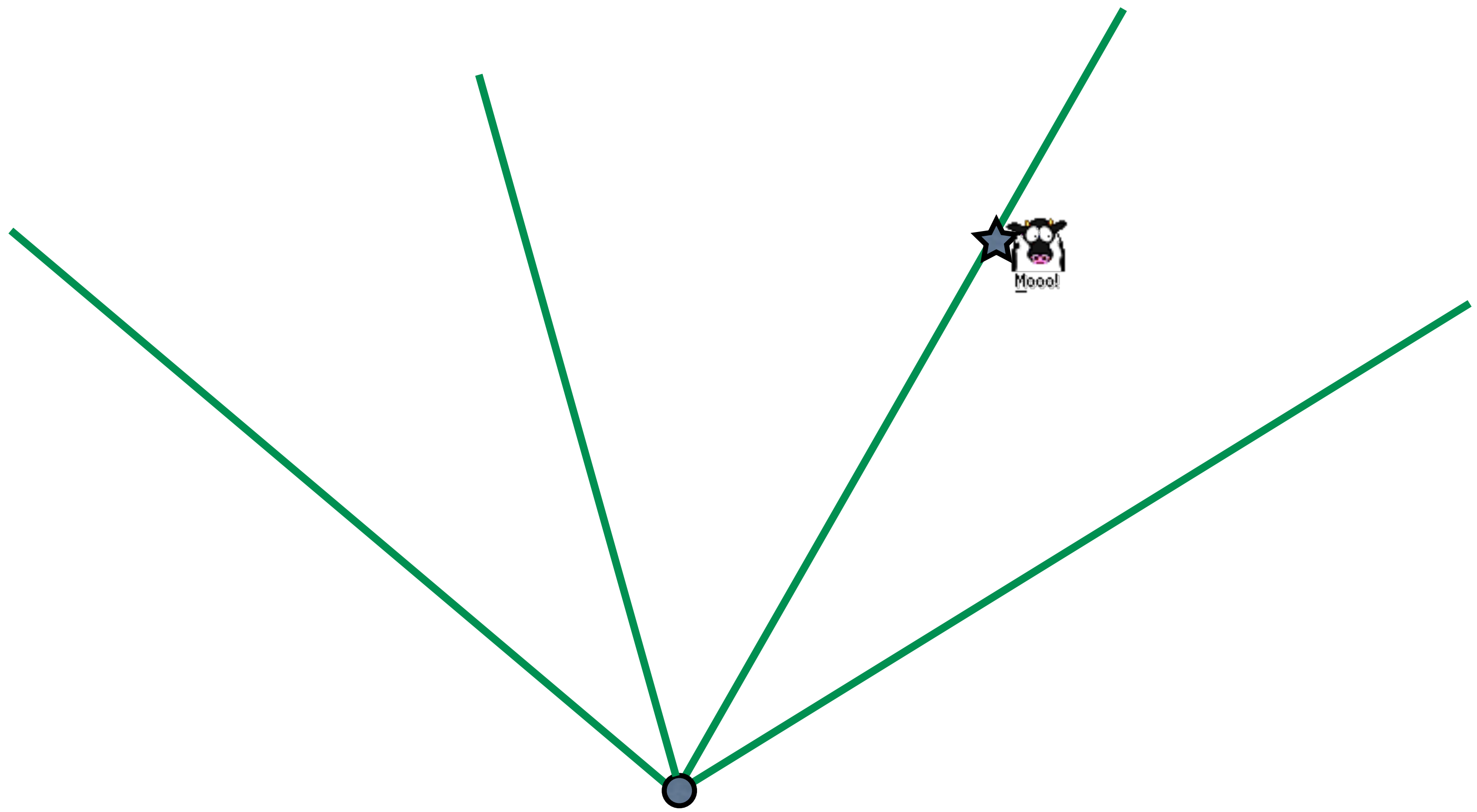
The first problem: Ray searching



The first problem: Ray searching

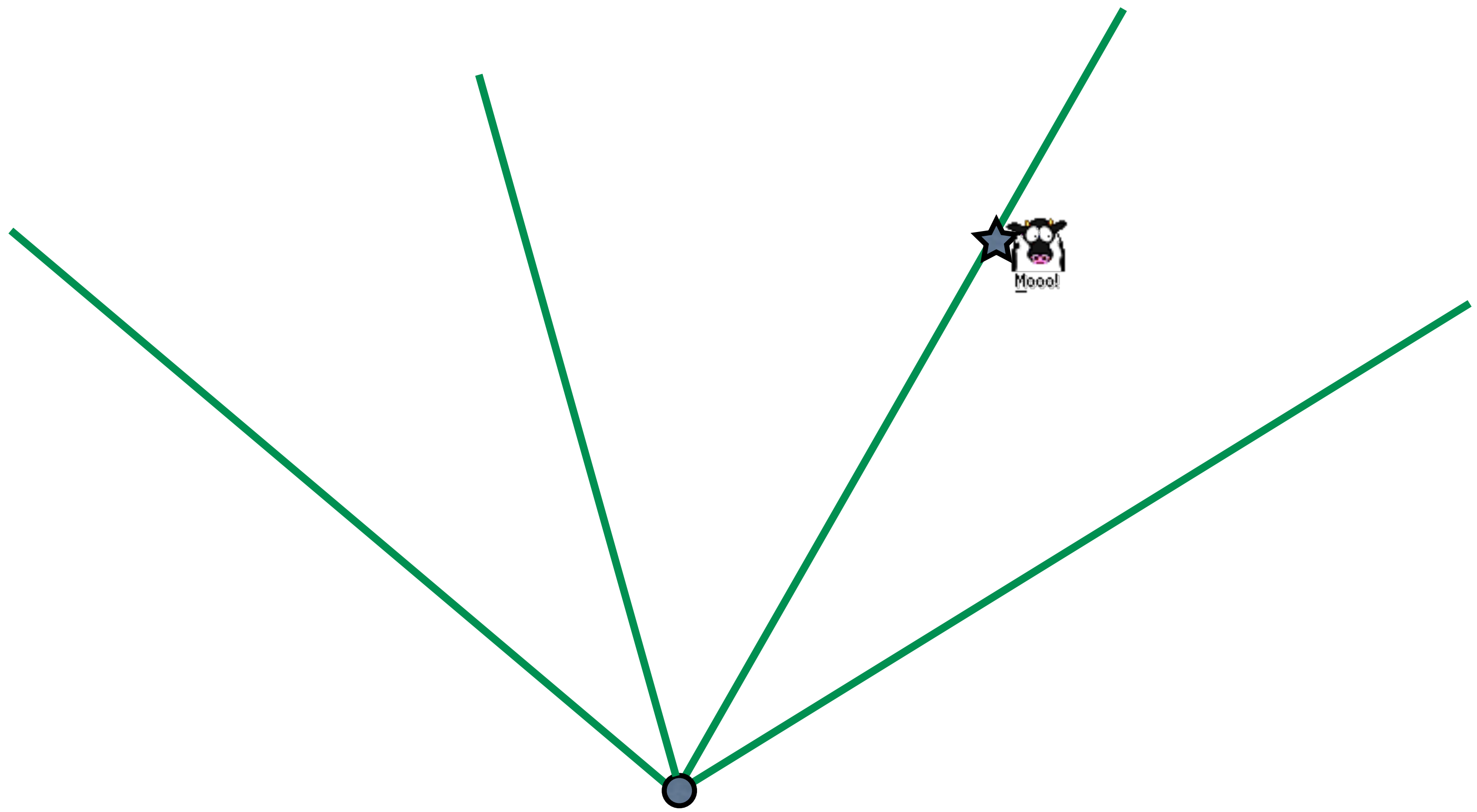


The first problem: Ray searching



Performance evaluation: Competitive ratio

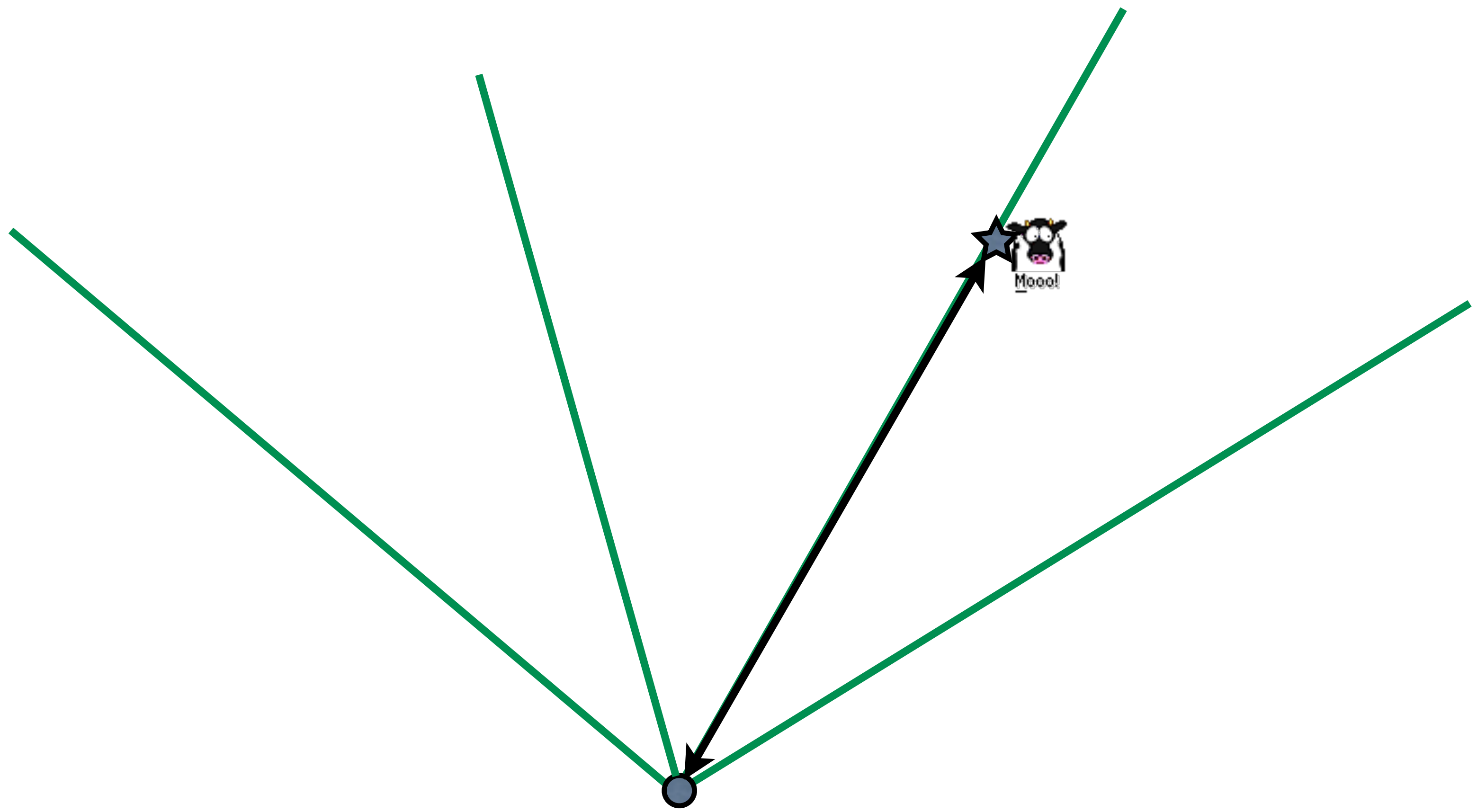
The first problem: Ray searching



Performance evaluation: Competitive ratio

$$\alpha = \sup_{\star} \frac{\text{total distance to find } \star}{\text{distance of } \star \text{ from origin}}$$

The first problem: Ray searching



Performance evaluation: Competitive ratio

$$\alpha = \sup_{\star} \frac{\text{total distance to find } \star}{\text{distance of } \star \text{ from origin}}$$

The second problem: Scheduling contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via
schedules of executions of contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

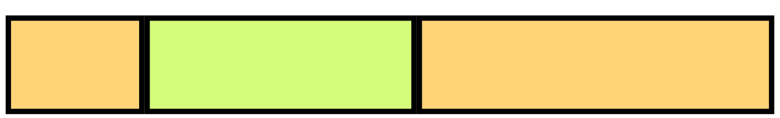
- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms



The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via
schedules of executions of contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

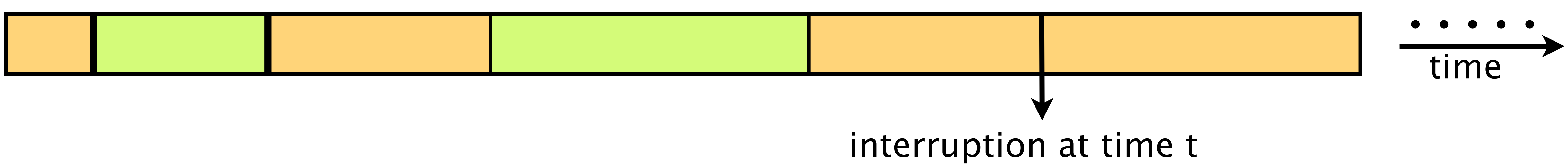
- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms



The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

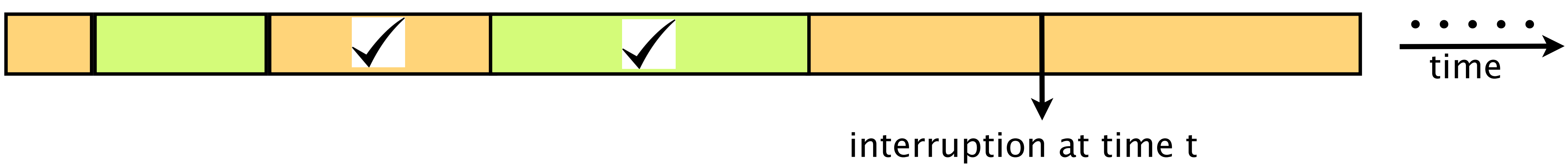
?

→

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via
schedules of executions of contract algorithms



The second problem: Scheduling contract algorithms

Contract algorithms

- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms

The second problem: Scheduling contract algorithms

Contract algorithms

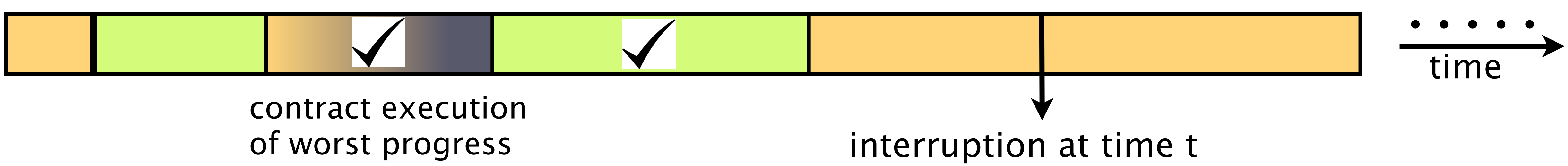
- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms



The second problem: Scheduling contract algorithms

Contract algorithms

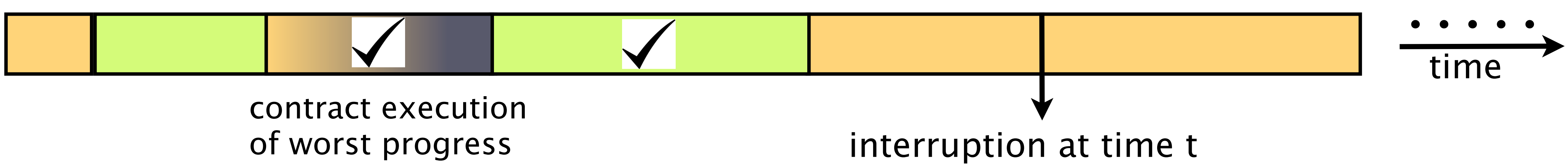
- execution time given as input
- may return non-meaningful solutions if interrupted

?

Interruptible algorithms

- may be interrupted at will
- always return meaningful solutions (improving with time)

[Russell and Zilberstein 1991]: Interruptible algorithms via schedules of executions of contract algorithms

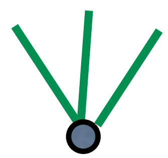


Performance evaluation: Acceleration ratio $\beta = \sup_t \frac{\text{interruption time } t}{\text{length of } \boxed{}}$

Some related previous work

Ray searching

m rays



- Early work by Bellman, Beck and Newman for $m = 2$
- Optimal strategies by [Gal 74]
- Re-discovered in CS context [Baeza-Yates et al. 93]
- Several other settings:

  Randomization [Kao et al. 96]

  Multiple searchers [Lopez-Ortiz and Schuierer 04]

  Turn cost [Demaine et al. 06], [A. et al. 14+]

  New measures [Kirkpatrick 09]

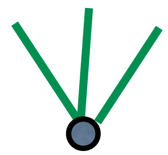
Contract scheduling

n problems

- [Russell and Zilberstein 91]: $n = 1$
- [Bernstein et al. 02]: general n
- [Zilberstein et al. 03]: $n = 1$, multiple processors
- [Bernstein et al. 03], [Lopez-Ortiz et al. 06] general n, multiple processors
- [A. and Lopez-Ortiz 08]: soft interruptions
- [A. and Lopez-Ortiz 09]: new measures

Some related previous work

Ray searching
m rays



Contract scheduling
n problems

- Early work by Bellman, Beck and Newman for $m = 2$
- Optimal strategies by [Gal 74]
- Re-discovered in CS context [Baeza-Yates et al. 93]
- Several other settings:

  Randomization [Kao et al. 96]

  Multiple searchers [Lopez-Ortiz and Schuierer 04]

  Turn cost [Demaine et al. 06], [A. et al. 14+]

  New measures [Kirkpatrick 09]

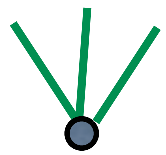
- [Russell and Zilberstein 91]: $n = 1$
- [Bernstein et al. 02]: general n
- [Zilberstein et al. 03]: $n = 1$, multiple processors
- [Bernstein et al. 03], [Lopez-Ortiz et al. 06] general n, multiple processors
- [A. and Lopez-Ortiz 08]: soft interruptions
- [A. and Lopez-Ortiz 09]: new measures

[Bernstein et al. 03]: Connections between cyclic strategies for the two problems

Results

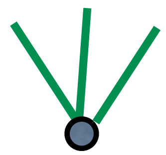
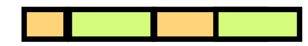
Results

Ray searching
m rays

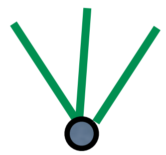


Contract scheduling
n problems

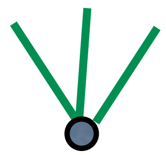
Results

	Ray searching m rays 	**Contract scheduling** n problems 
Stochastic setting	**Setting**: Target detection with probability p **Results**: Strategy with competitive ratio $\Theta(m/p^2)$ + No strategy is better than $m/(2p)$ –competitive	**Setting**: Contracts are Monte Carlo algorithms with success prob. p **Results**: Schedule with accel. ratio $(e(n+1))/p$ + No schedule better than n/p – acceleration

Results

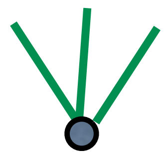
	Ray searching m rays 	Contract scheduling n problems 
Stochastic setting	**Setting**: Target detection with probability p **Results**: Strategy with competitive ratio $\Theta(m/p^2)$ + No strategy is better than $m/(2p)$ –competitive	**Setting**: Contracts are Monte Carlo algorithms with success prob. p **Results**: Schedule with accel. ratio $(\mathrm{e}(n+1))/p$ + No schedule better than n/p – acceleration
Fault tolerance / redundancy	**Setting**: Target detection on the r –th visit **Results**: Strategy with competitive ratio $r(m-1)\left(\frac{m}{m-1}\right)^m + 2 - r$ + no strategy better than $rm/2$ –competitive	**Setting**: Output the r –th smallest contract **Results**: Strategy with acceleration ratio $r(n+1)\left(1+\frac{1}{rn}\right)^{rn}$ + no strategy better than rn –competitive

Results

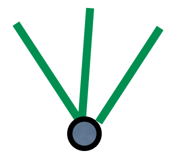
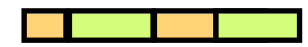
	Ray searching  m rays	Contract scheduling n problems 	Methodology
Stochastic setting	**Setting**: Target detection with probability p **Results**: Strategy with competitive ratio $\Theta(m/p^2)$ + No strategy is better than $m/(2p)$ –competitive	**Setting**: Contracts are Monte Carlo algorithms with success prob. p **Results**: Schedule with accel. ratio $(\mathrm{e}(n+1))/p$ + No schedule better than n/p – acceleration	Non–trivial analysis of cyclic strategies
Fault tolerance / redundancy	**Setting**: Target detection on the r –th visit **Results**: Strategy with competitive ratio $r(m-1)\left(\frac{m}{m-1}\right)^m + 2 - r$ + no strategy better than $rm/2$ –competitive	**Setting**: Output the r –th smallest contract **Results**: Strategy with acceleration ratio $r(n+1)\left(1+\frac{1}{rn}\right)^{rn}$ + no strategy better than rn –competitive	Non–cyclic strategies that improve upon the best cyclic ones

Results (continued)

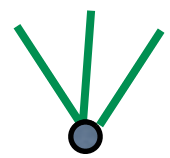
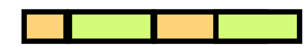
Results (continued)

	Ray searching m rays 	Contract scheduling n problems 
Randomized scheduling	**Known**: Randomization helps improve the competitive ratio [Kao et al. 96]	**Result**: Randomized schedule of acceleration ratio about 0.6 times the deterministic acceleration ratio

Results (continued)

	Ray searching m rays 	Contract scheduling n problems 
Randomized scheduling	**Known**: Randomization helps improve the competitive ratio [Kao et al. 96]	**Result**: Randomized schedule of acceleration ratio about 0.6 times the deterministic acceleration ratio
Trade offs between performance and turns / executions	**Setting**: we are interested in the # of turns **Results**: Optimal trade-offs between competitive ratio and # of turns	**Setting**: we are interested in the # of executions **Results**: Optimal trade-offs between acceleration ratio and # of executions

Results (continued)

	Ray searching m rays 	Contract scheduling n problems 	Methodology
Randomized scheduling	**Known**: Randomization helps improve the competitive ratio [Kao et al. 96]	**Result**: Randomized schedule of acceleration ratio about 0.6 times the deterministic acceleration ratio	Similar strategies but different analysis (no closed form in the case of contract scheduling)
Trade offs between performance and turns / executions	**Setting**: we are interested in the # of turns **Results**: Optimal trade-offs between competitive ratio and # of turns	**Setting**: we are interested in the # of executions **Results**: Optimal trade-offs between acceleration ratio and # of executions	Combination of uniform and exponentially increasing strategies

The stochastic setting

The stochastic setting

- Easy bound of $\Omega(m/p)$

The stochastic setting

- Easy bound of $\Omega(m/p)$
- We analyze a round-robing strategy: search with lengths $x_1, x_2, \ldots$

The stochastic setting

- Easy bound of $\Omega(m/p)$
- We analyze a round-robing strategy: search with lengths $x_1, x_2, \ldots$

1. Upper bound the expected cost of the strategy

   (we get an expression that contains the x_i's and p

The stochastic setting

- Easy bound of $\Omega(m/p)$
- We analyze a round-robing strategy: search with lengths $x_1, x_2, \ldots$

1. Upper bound the expected cost of the strategy
   (we get an expression that contains the x_i's and p

2. Assume exponential strategies: $x_i = b^i$ for fixed $b > 1$
   In the resulting expression, there is a term of the form $\sum_{j=2}^{\infty}((b^m(1-p))^{j-1}$

The stochastic setting

- Easy bound of $\Omega(m/p)$
- We analyze a round-robing strategy: search with lengths $x_1, x_2, \ldots$

1. Upper bound the expected cost of the strategy
   (we get an expression that contains the x_i's and p

2. Assume exponential strategies: $x_i = b^i$ for fixed $b > 1$
   In the resulting expression, there is a term of the form $\sum_{j=2}^{\infty}((b^m(1-p))^{j-1}$

3. Setting $\lambda := b^m(1-p)$ and requiring that $\lambda < 1$ we obtain

$$\alpha \leq 1 + 2\frac{b^m}{b^m - 1} \cdot \frac{1}{1-\lambda}$$

The stochastic setting

- Easy bound of $\Omega(m/p)$
- We analyze a round-robing strategy: search with lengths $x_1, x_2, \ldots$

1. Upper bound the expected cost of the strategy
   (we get an expression that contains the x_i's and p

2. Assume exponential strategies: $x_i = b^i$ for fixed $b > 1$
   In the resulting expression, there is a term of the form $\sum_{j=2}^{\infty}((b^m(1-p))^{j-1}$

3. Setting $\lambda := b^m(1-p)$ and requiring that $\lambda < 1$ we obtain

$$\alpha \leq 1 + 2\frac{b^m}{b^m - 1} \cdot \frac{1}{1-\lambda}$$

4. Applying some calculus we show that $\alpha \leq 1 + 8m/p^2$

Randomized scheduling of contract algorithms

Randomized scheduling of contract algorithms

Algorithm

1. Choose a random permutation of the n problems and random $\epsilon \in (0, 1)$

2. In the i-th step schedule a contract for problem $i \bmod n$ and of length $b^{i+\epsilon}$

Randomized scheduling of contract algorithms

Algorithm

1. Choose a random permutation of the n problems and random $\epsilon \in (0,1)$

2. In the i-th step schedule a contract for problem $i \bmod n$ and of length $b^{i+\epsilon}$

Inspired by the randomized ray-searching algorithm of [Kao et al. 95]

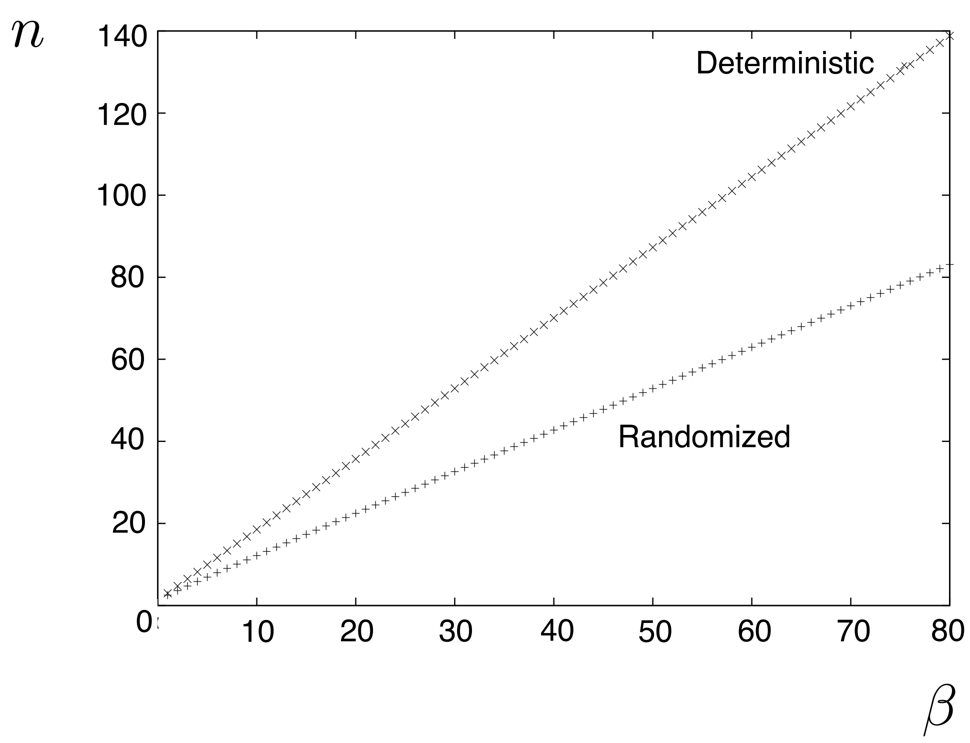
Randomized scheduling of contract algorithms

Algorithm

1. Choose a random permutation of the n problems and random $\epsilon \in (0,1)$

2. In the i-th step schedule a contract for problem $i \bmod n$ and of length $b^{i+\epsilon}$

Inspired by the randomized ray-searching algorithm of [Kao et al. 95]



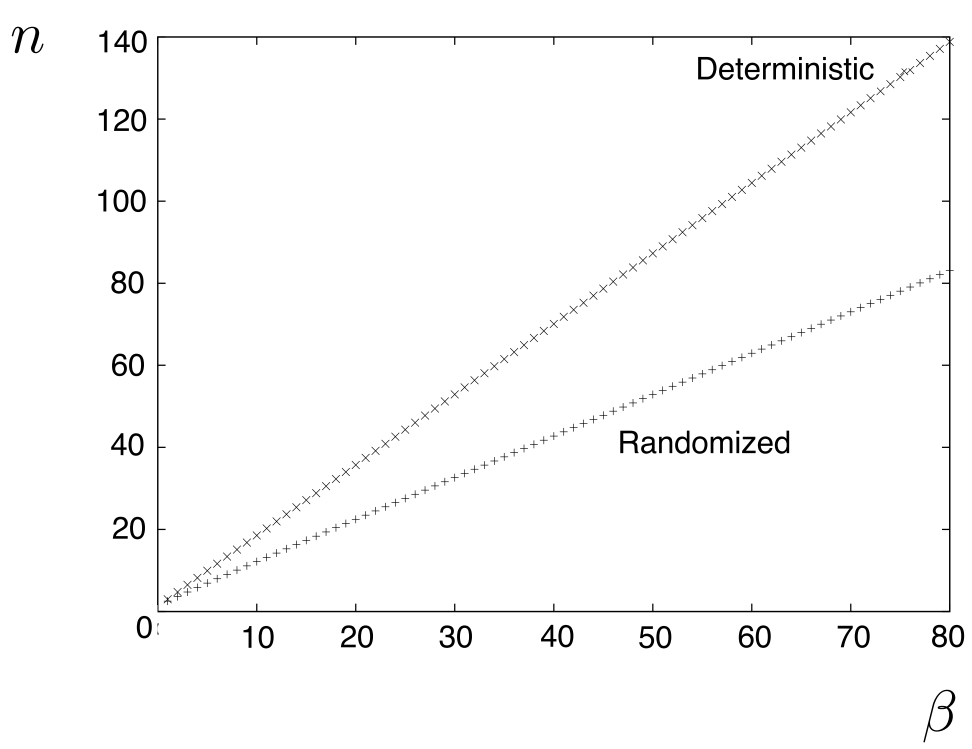
Randomized scheduling of contract algorithms

Algorithm

1. Choose a random permutation of the n problems and random $\epsilon \in (0,1)$

2. In the i-th step schedule a contract for problem $i \bmod n$ and of length $b^{i+\epsilon}$

Inspired by the randomized ray-searching algorithm of [Kao et al. 95]

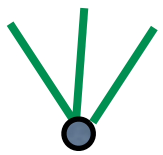


- A closed formula does not appear to exist
- We can give analytical bounds for $n \to \infty$

Tradeoffs between performance and turns/executions

Tradeoffs between performance and turns/executions

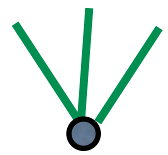
Ray searching
m rays



Contract scheduling
n problems

Tradeoffs between performance and turns/executions

Ray searching
m rays



Contract scheduling
n problems

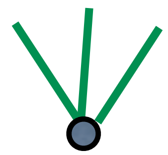
"Given a target at distance d what is the minimum number of turns required to guarantee a certain competitive ratio?"

"Given an interruption at time t what is the minimum number of contracts required to guarantee a certain acceleration ratio?"

Tradeoffs between performance and turns/executions

Ray searching

m rays



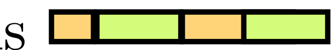
"Given a target at distance d what is the minimum *number of turns* *required to guarantee a certain competitive ratio?"*

pathwise search

expanding search

Contract scheduling

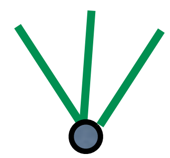
n problems



"Given an interruption at time t what is the minimum *number of contracts* *required to guarantee a certain acceleration ratio?"*

Tradeoffs between performance and turns/executions

Ray searching

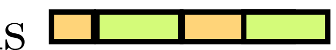
m rays 

"Given a target at distance d what is the minimum number of turns required to guarantee a certain competitive ratio?"

pathwise search

expanding search

Contract scheduling

n problems 

"Given an interruption at time t what is the minimum number of contracts required to guarantee a certain acceleration ratio?"

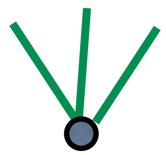
contract scheduling

scheduling with preemptions

Tradeoffs between performance and turns/executions

Ray searching

m rays



"Given a target at distance d what is the minimum number of turns required to guarantee a certain competitive ratio?"

pathwise search

expanding search

Contract scheduling

n problems

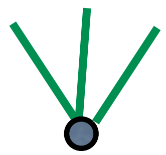
"Given an interruption at time t what is the minimum number of contracts required to guarantee a certain acceleration ratio?"

contract scheduling

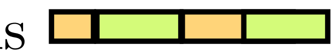
scheduling with preemptions

Tradeoffs between performance and turns/executions

Ray searching
m rays



Contract scheduling
n problems



"Given a target at distance d what is the minimum number of turns required to guarantee a certain competitive ratio?"

"Given an interruption at time t what is the minimum number of contracts required to guarantee a certain acceleration ratio?"

pathwise search

expanding search

contract scheduling

scheduling with preemptions

Theorem: For any strategy with acceleration ratio $n(b+1)-\epsilon$, for $b>1$ and $\epsilon>0$ there exists t such that the # of preemptions up to t is at least $n\log_b\left(\frac{t(b-1)}{n}+1\right)-n$.

Furthermore, there is a strategy with acceleration ratio $n(b+1)$ and at most $n\log_b\left(\frac{t(b-1)}{n}+1\right)+n$ preemptions.

Conclusions and outlook

Conclusions and outlook

- Connections between two well-studied problems
- Similarities in settings, common algorithmic approach
- The two problems are similar, but also have certain differences

Conclusions and outlook

- Connections between two well-studied problems
- Similarities in settings, common algorithmic approach
- The two problems are similar, but also have certain differences
- More work needed on the stochastic setting (tight bounds for probabilistic searching, more elaborate study of stochastic contract scheduling)
- Heterogeneous environments? (e.g., every ray has its own probability of target location)
- Connections with other problems? (older work: speedup of Las Vegas algorithms [Luby et al. 93]; newer work: progressive algorithms [Alewijnse 15])

Conclusions and outlook

- Connections between two well-studied problems
- Similarities in settings, common algorithmic approach
- The two problems are similar, but also have certain differences
- More work needed on the stochastic setting (tight bounds for probabilistic searching, more elaborate study of stochastic contract scheduling)
- Heterogeneous environments? (e.g., every ray has its own probability of target location)
- Connections with other problems? (older work: speedup of Las Vegas algorithms [Luby et al. 93]; newer work: progressive algorithms [Alewijnse 15])

Full version of the paper available at www.arxiv.org or at www.lip6.fr/Spyros.Angelopoulos

Conclusions and outlook

Thank you!

- Connections between two well-studied problems
- Similarities in settings, common algorithmic approach
- The two problems are similar, but also have certain differences
- More work needed on the stochastic setting (tight bounds for probabilistic searching, more elaborate study of stochastic contract scheduling)
- Heterogeneous environments? (e.g., every ray has its own probability of target location)
- Connections with other problems? (older work: speedup of Las Vegas algorithms [Luby et al. 93]; newer work: progressive algorithms [Alewijnse 15])

Full version of the paper available at www.arxiv.org or at www.lip6.fr/Spyros.Angelopoulos