Beliefs We Can Believe In:
Replacing Assumptions with Data in Real-time Search

Maximilian Fickert[1] and Tianyi Gu[2] and Leonhard Staut[1] and
Wheeler Ruml[2] and Jörg Hoffmann[1] and Marek Petrik[2]

1 UNIVERSITÄT DES SAARLANDES

2 University of New Hampshire

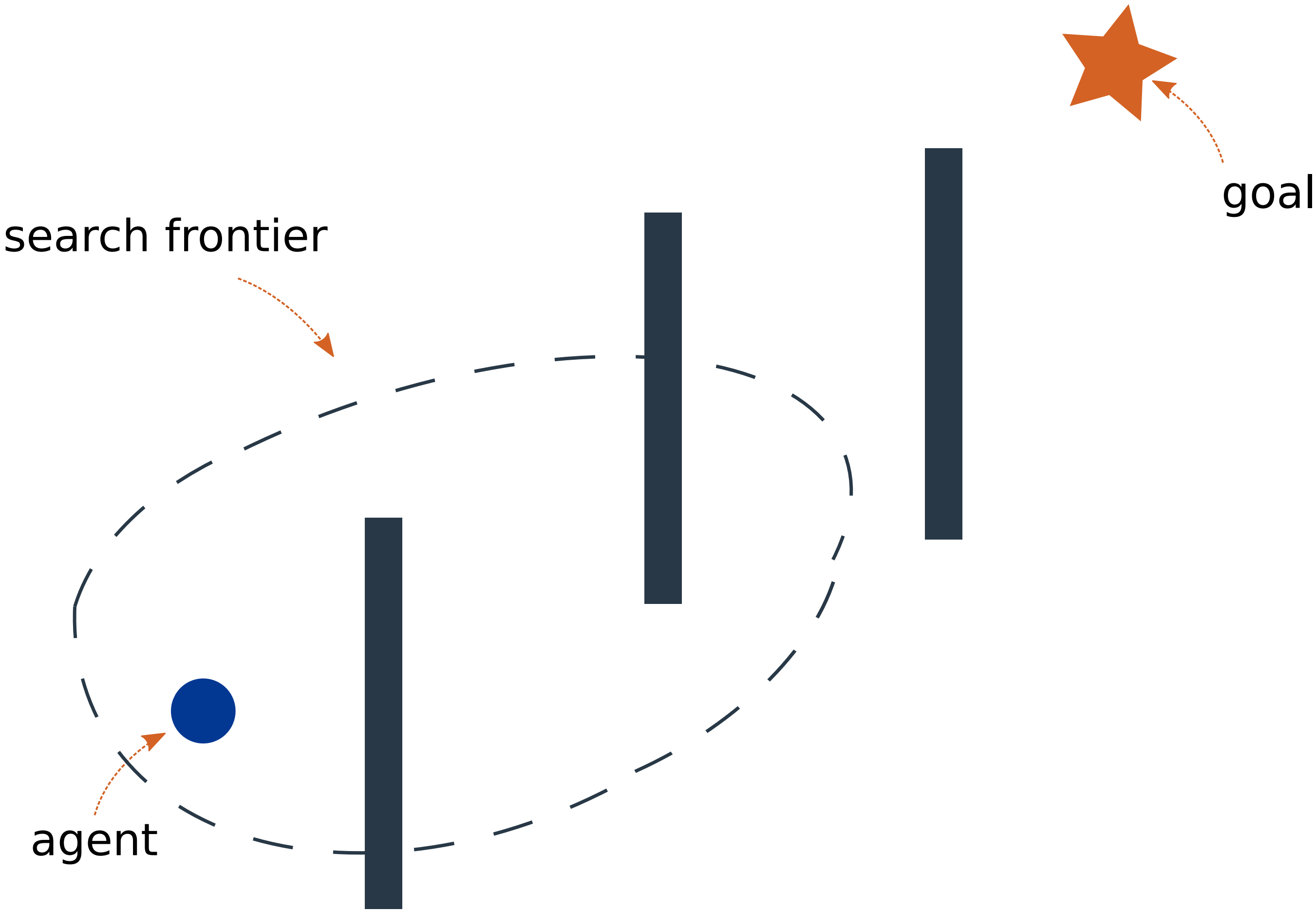
What is Real-time Heuristic Search?

An example: path finding

agent performs search for a bounded time

What is Real-time Heuristic Search?

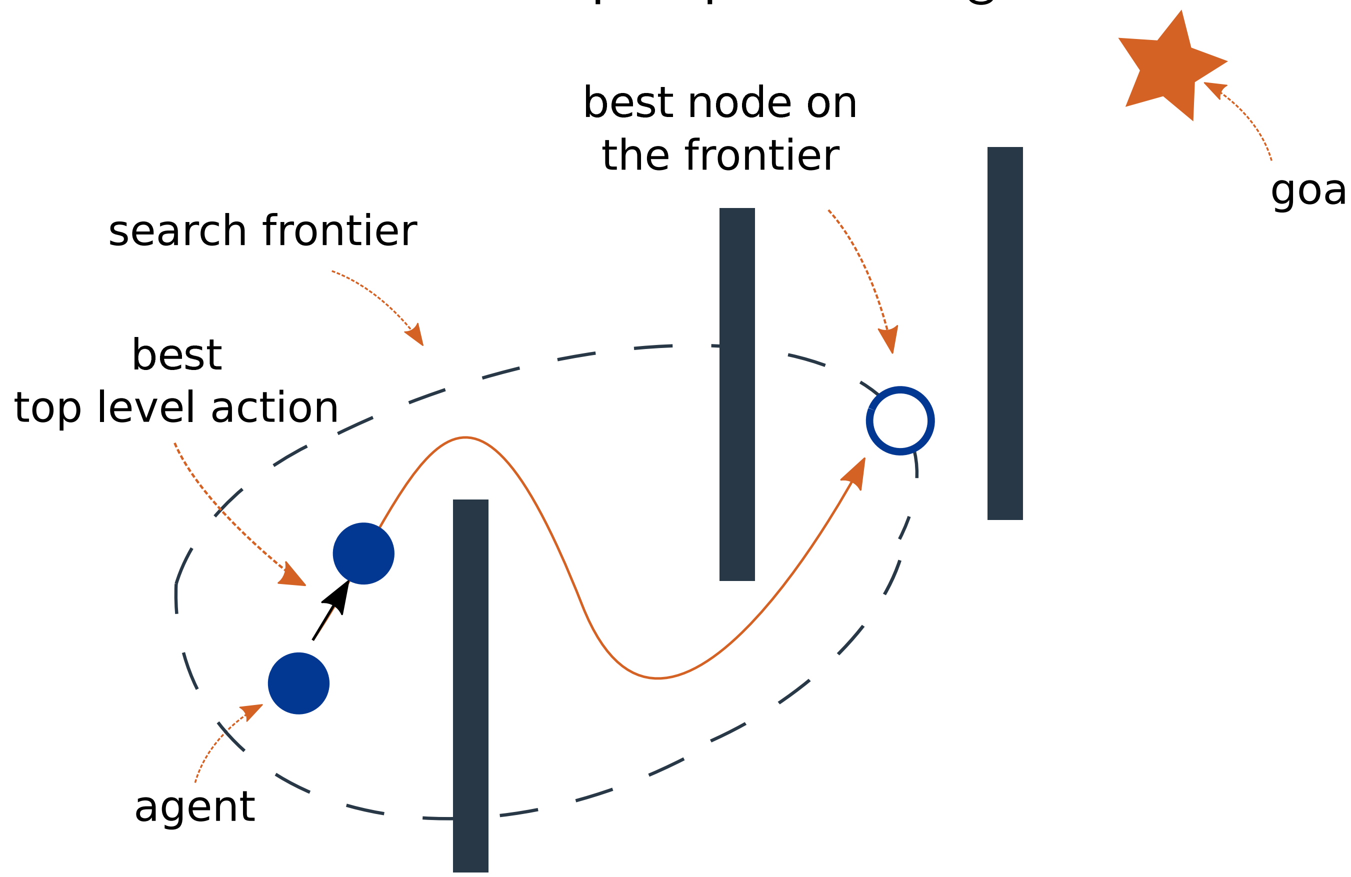
An example: path finding



agent performs search for a bounded time

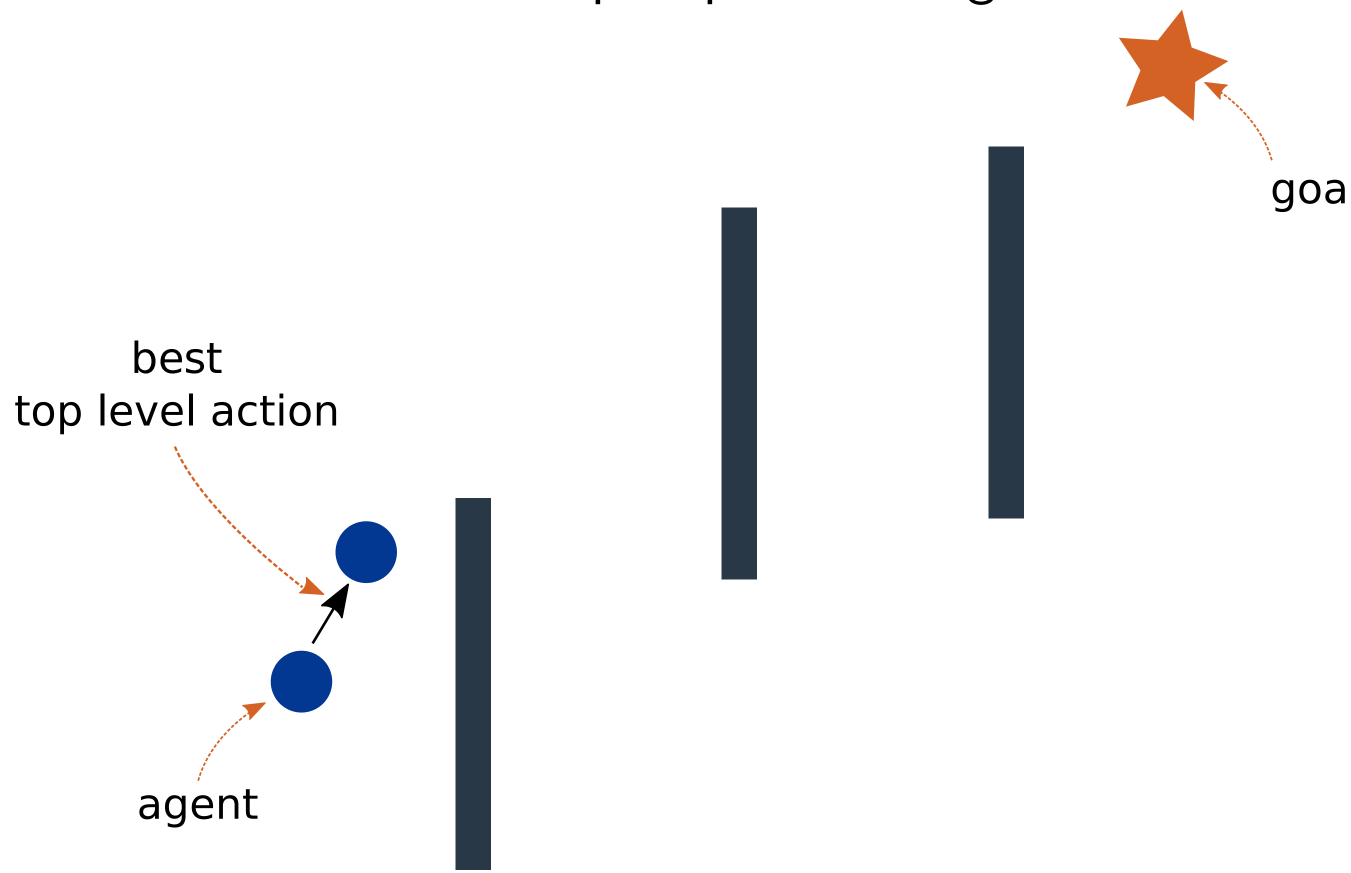
What is Real-time Heuristic Search?

An example: path finding



agent commits to best action and executes

An example: path finding



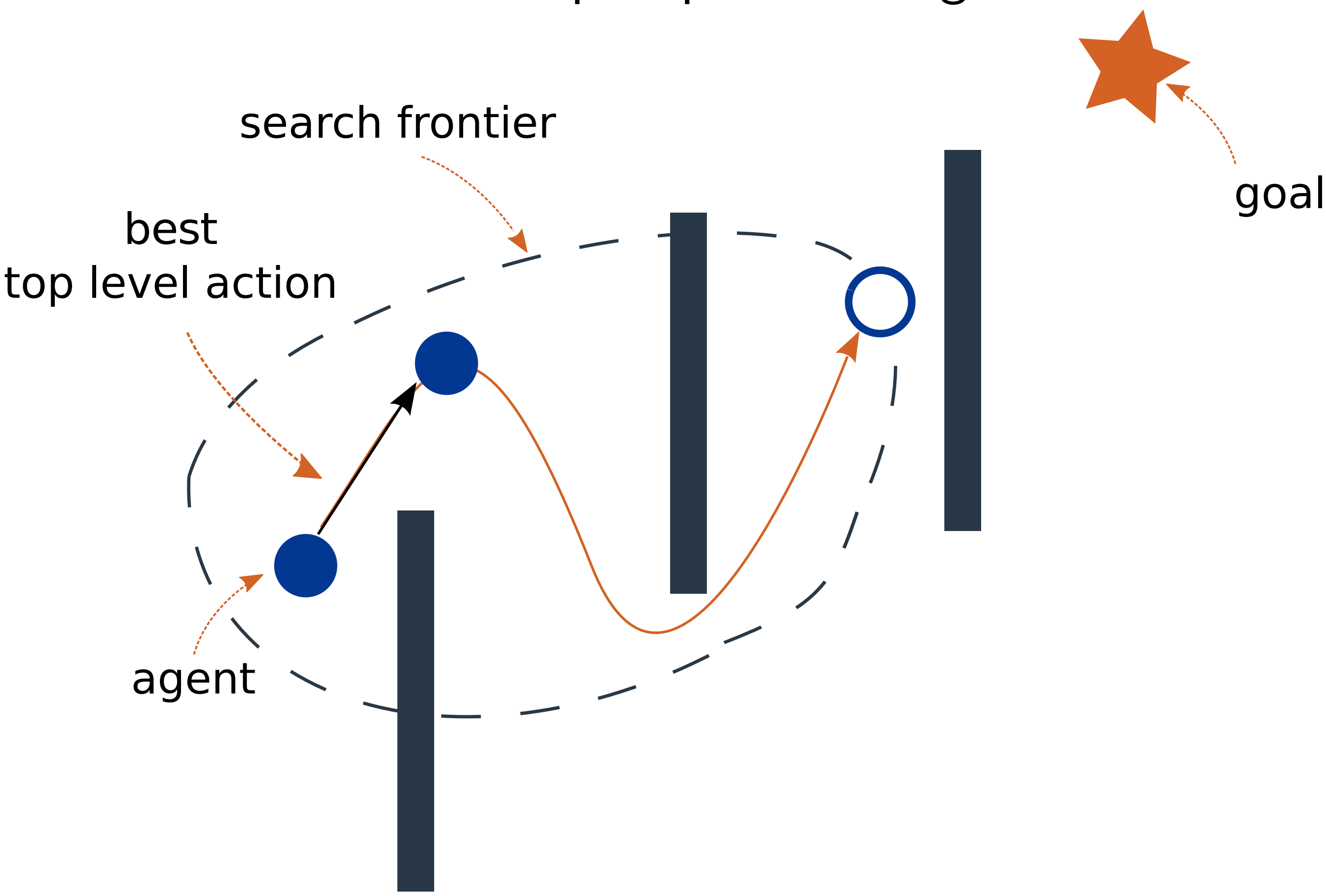
agent commits to best action and executes

What is Real-time Heuristic Search?

An example: path finding

agent commits to best action and executes

An example: path finding



online planning: interleaving search and action execution

"receding horizon control"

Real-time heuristic search:

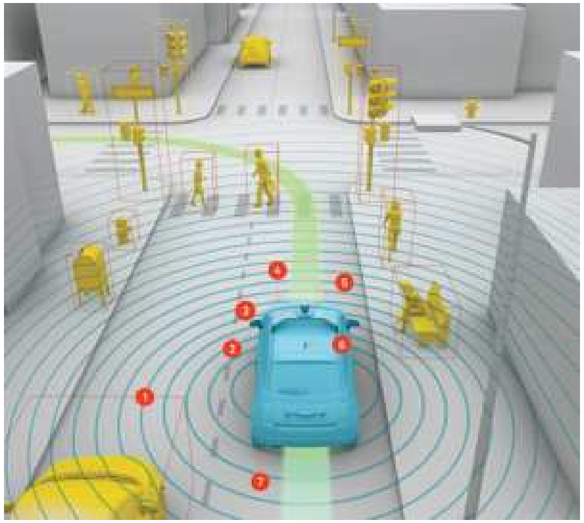
return the next action within a time bound

Applications:

interacting with humans

dynamic environment

eg, robotics



A Classic Approach: LSS-LRTA* (Koenig&Sun 2008)

three phases:

1. Lookahead Phase:
   expands nodes with minimum f
   to explore the search space

A Classic Approach: LSS-LRTA* (Koenig&Sun 2008)

three phases:

1. Lookahead Phase:
   expands nodes with minimum f
   to explore the search space
2. Decision-making Phase:
   backup the minimum f from search frontier ('minimin')
   select top level action with minimum f to execute

A Classic Approach: LSS-LRTA* (Koenig&Sun 2008)

three phases:

1. Lookahead Phase:
   expands nodes with minimum f
   to explore the search space
2. Decision-making Phase:
   backup the minimum f from search frontier ('minimin')
   select top level action with minimum f to execute
3. Learning Phase:
   update heuristic values
   (to escape local minima and avoid infinite loops)

A Classic Approach: LSS-LRTA* (Koenig&Sun 2008)

three phases:

1. Lookahead Phase:
   expands nodes with minimum f
   to explore the search space
2. Decision-making Phase:
   backup the minimum f from search frontier ('minimin')
   select top level action with minimum f to execute
3. Learning Phase:
   update heuristic values
   (to escape local minima and avoid infinite loops)

repeat until at a goal

A Classic Approach: LSS-LRTA* (Koenig&Sun 2008)

three phases:

1. Lookahead Phase:
   expands nodes with minimum f
   to explore the search space
2. Decision-making Phase:
   backup the minimum f from search frontier ('minimin')
   select top level action with minimum f to execute
3. Learning Phase:
   update heuristic values
   (to escape local minima and avoid infinite loops)

repeat until at a goal

derived from offline search, but optimal for online?

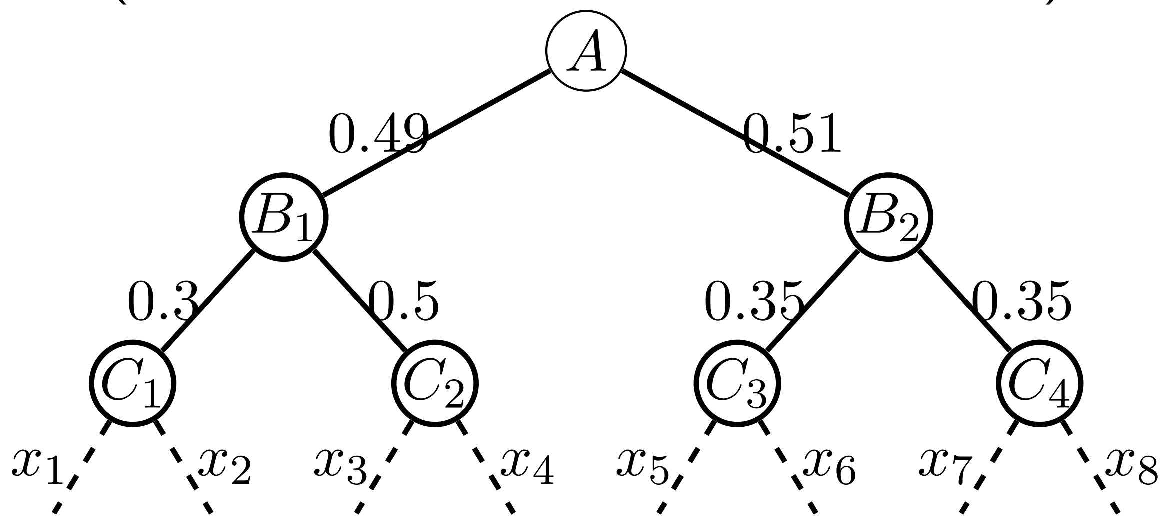
Overview

- Introduction
- AAAI-19 Recap: the Nancy framework
- This work: Data-Driven Nancy
  - h error distribution
  - completeness proof
- Results
- Conclusions

AAAI-19 Recap: The Nancy Framework

Decision-making Phase: A Troublesome Example

Should an agent at A move to B_1 or B_2?
(x_i are unknown but i.i.d. uniform 0-1)

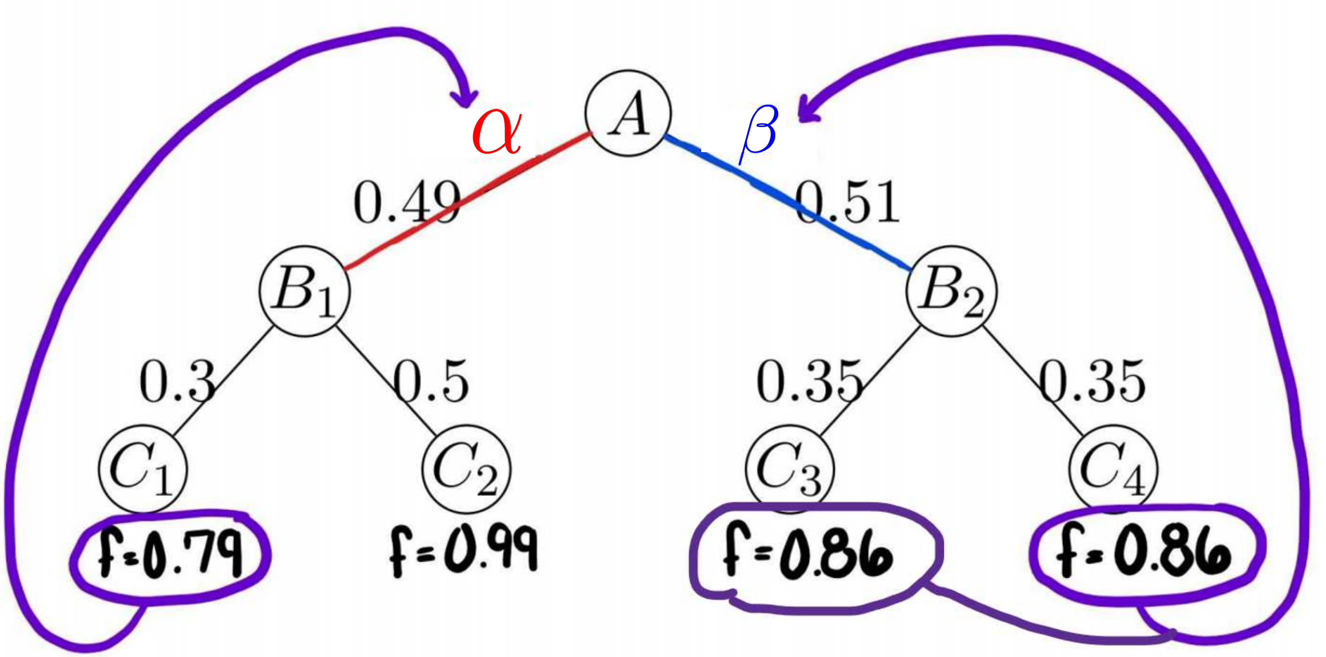


random tree domain (Pemberton & Korf 1995)

$f = g + h = g + 0$ is lower bound on optimal plan cost

Decision-making Phase: A Troublesome Example

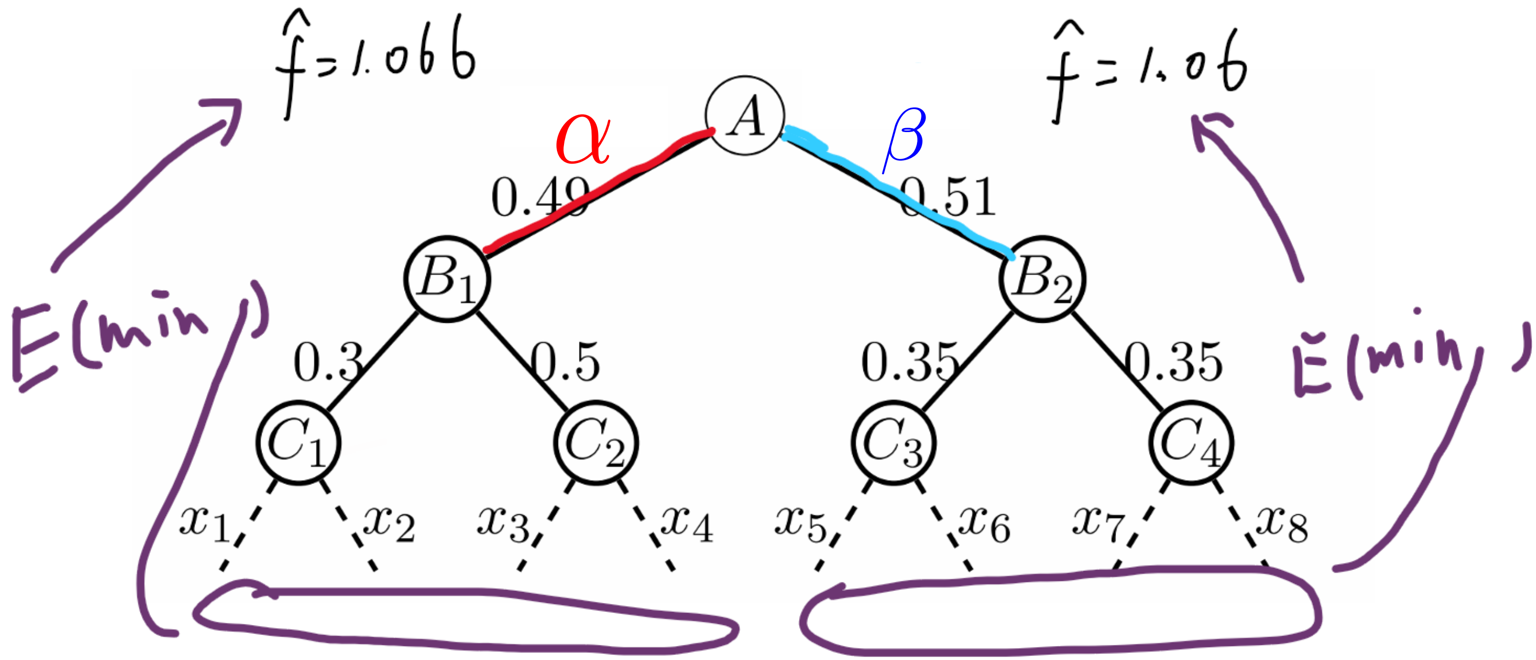
Should an agent at A move to B_1 or B_2?
(x_i are unknown but i.i.d. uniform 0-1)



decision theory says minimize expected value
lower bound: not suitable for rational action selection

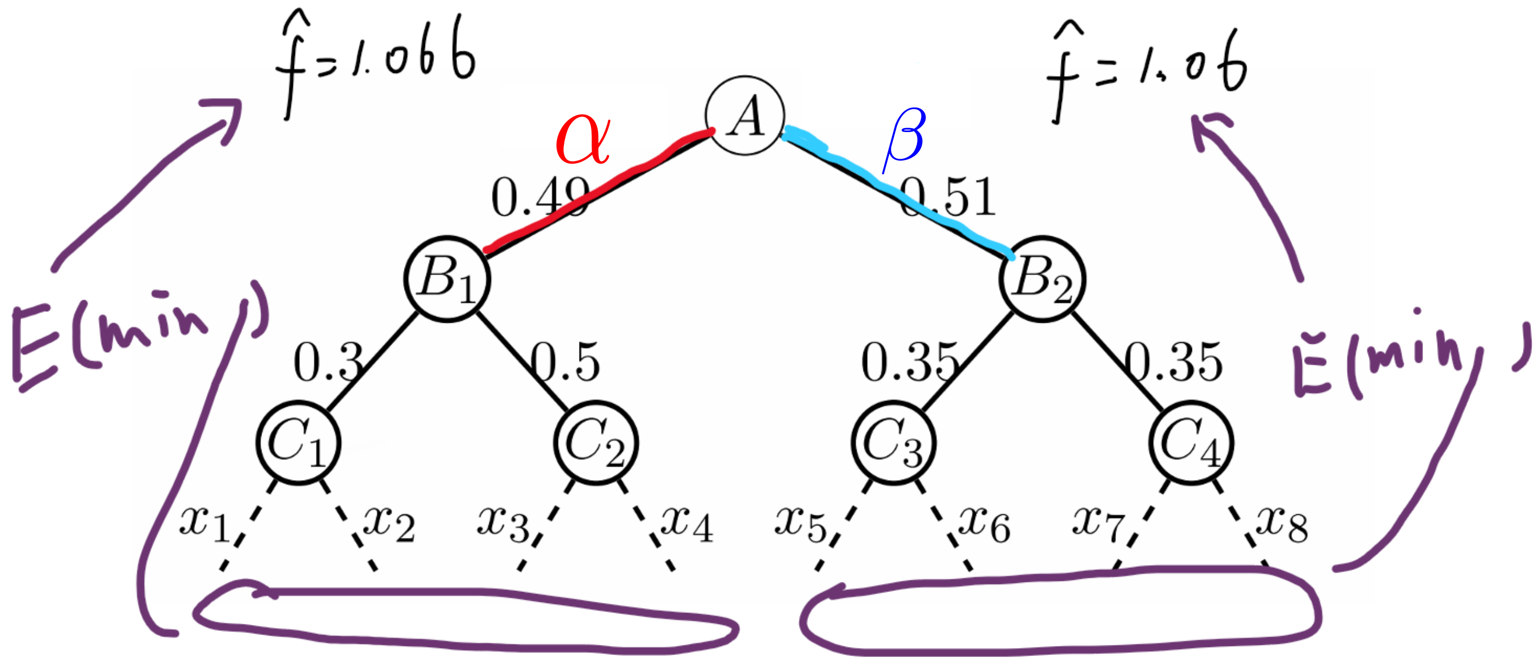
Decision-making Phase: A Troublesome Example

Should an agent at A move to B_1 or B_2?
(x_i are unknown but i.i.d. uniform 0-1)



$\hat{f}$ is expected plan cost

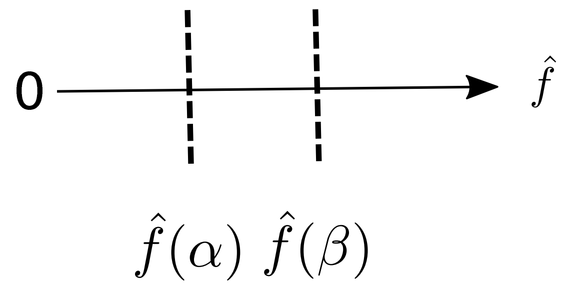
Should an agent at A move to B_1 or B_2?
(x_i are unknown but i.i.d. uniform 0-1)



$\hat{f}$ is expected plan cost

f is not the answer: should minimize expected value!

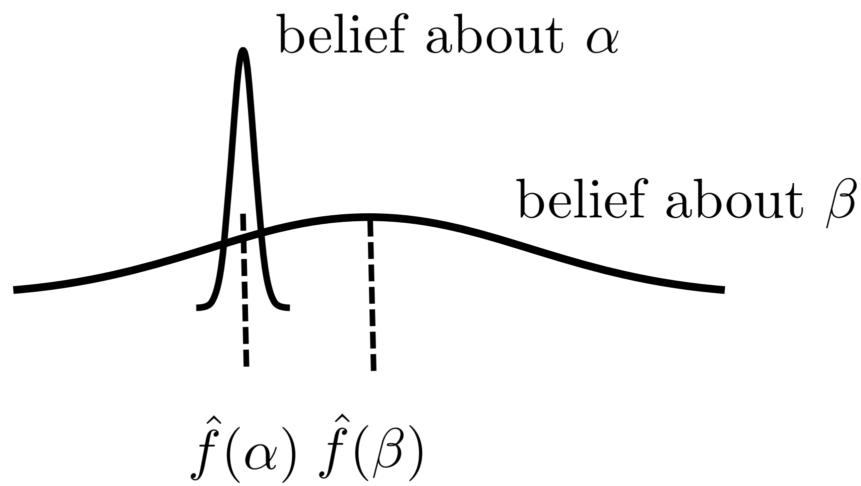
Lookahead Phase: A Troublesome Example



$\hat{f}$ is expected value

Should an agent expand nodes under α or β?

Lookahead Phase: A Troublesome Example



$\hat{f}$ is expected value

Should an agent expand nodes under α or β?

$\hat{f}$ is not the answer: what to do?
want to maximize value of information
need to consider uncertainty of estimates

Risk-based Lookahead

Risk-based lookahead (AAAI-19):

want to maximize value of information

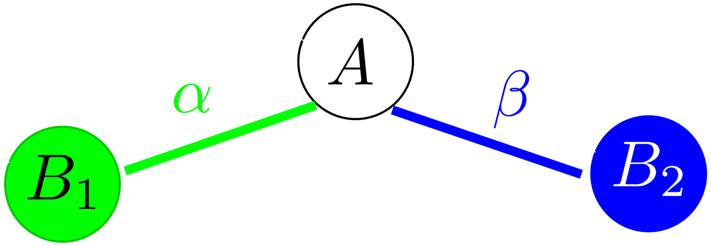
expand nodes which minimize expected regret

relies on belief of values

choose expansions that decrease uncertainty in beliefs

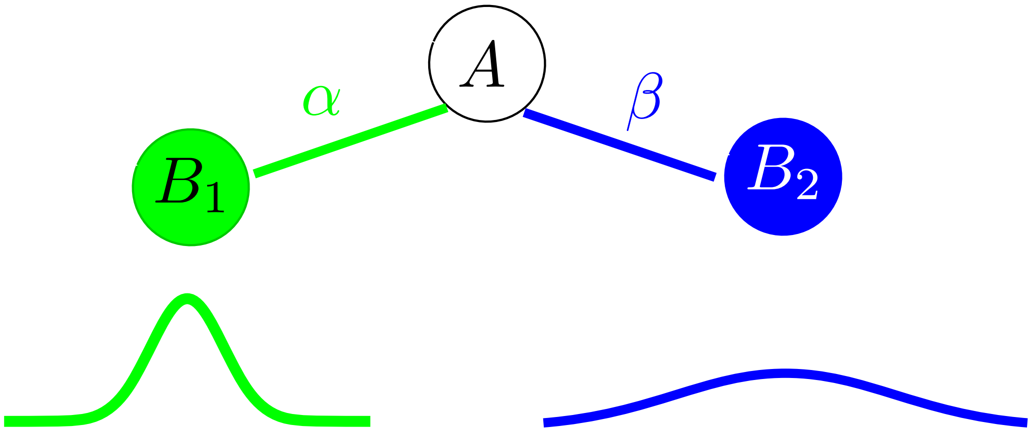
Risk-based Lookahead Example

expand under α or β?



Risk-based Lookahead Example

expand under α or β?

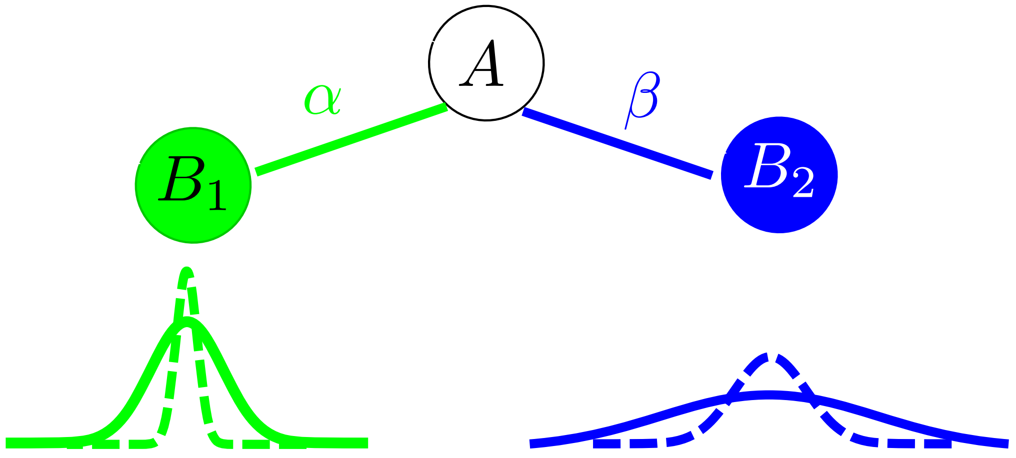


need 2 things:

1) current beliefs
2) estimate of how beliefs might change with search

Risk-based Lookahead Example

expand under α or β?

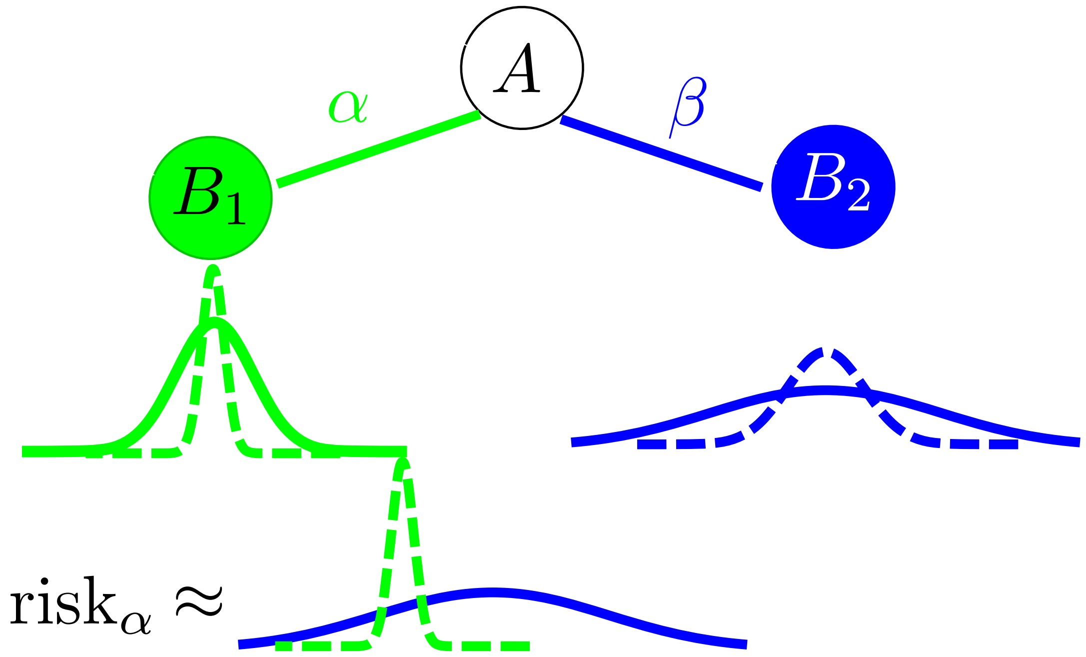


need 2 things:

1) current beliefs
2) estimate of how beliefs might change with search

Risk-based Lookahead Example

expand under α or β?

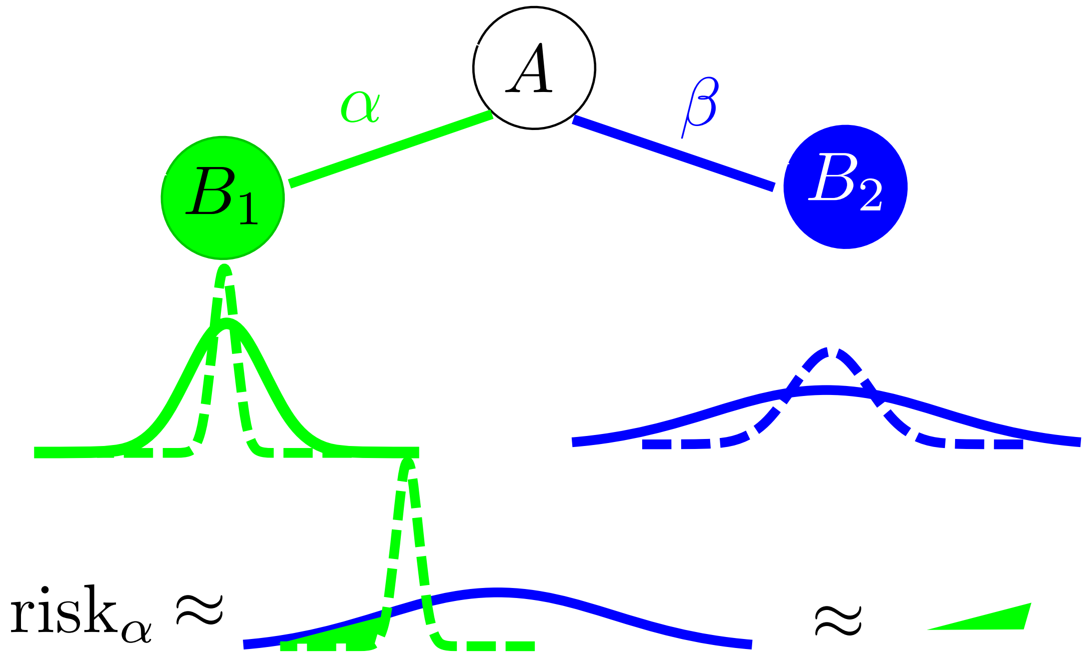


Risk: expected regret if a suboptimal action is selected
α is TLA with lowest expected value, other is β

$$\mathbb{E}\left[\underbrace{f^*(\alpha) - f^*(\beta)}_{\text{what is our regret}} \;\middle|\; \underbrace{f^*(\beta) < f^*(\alpha)}_{\text{in cases when } \alpha \text{ not best}}\right]$$

Risk-based Lookahead Example

expand under α or β?



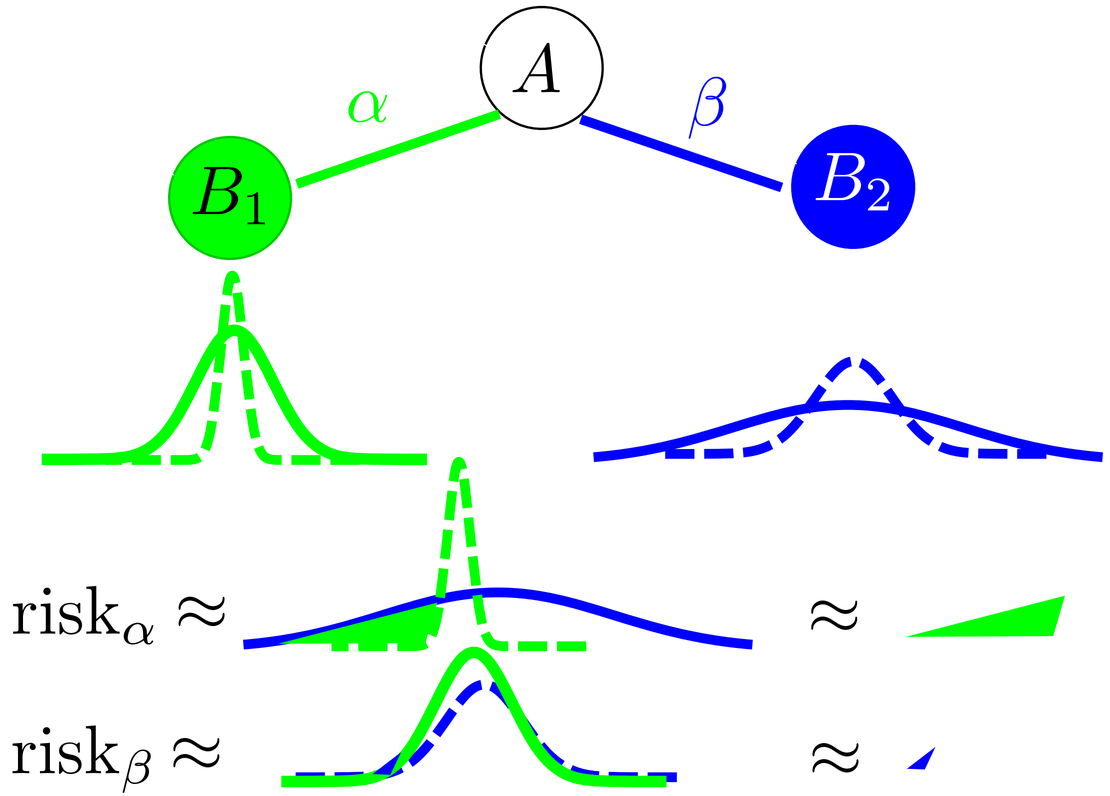
Risk: expected regret if a suboptimal action is selected

α is TLA with lowest expected value, other is β

$$\mathbb{E}\left[\underbrace{f^*(\alpha) - f^*(\beta)}_{\text{what is our regret}} \;\middle|\; \underbrace{f^*(\beta) < f^*(\alpha)}_{\text{in cases when } \alpha \text{ not best}}\right]$$

Risk-based Lookahead Example

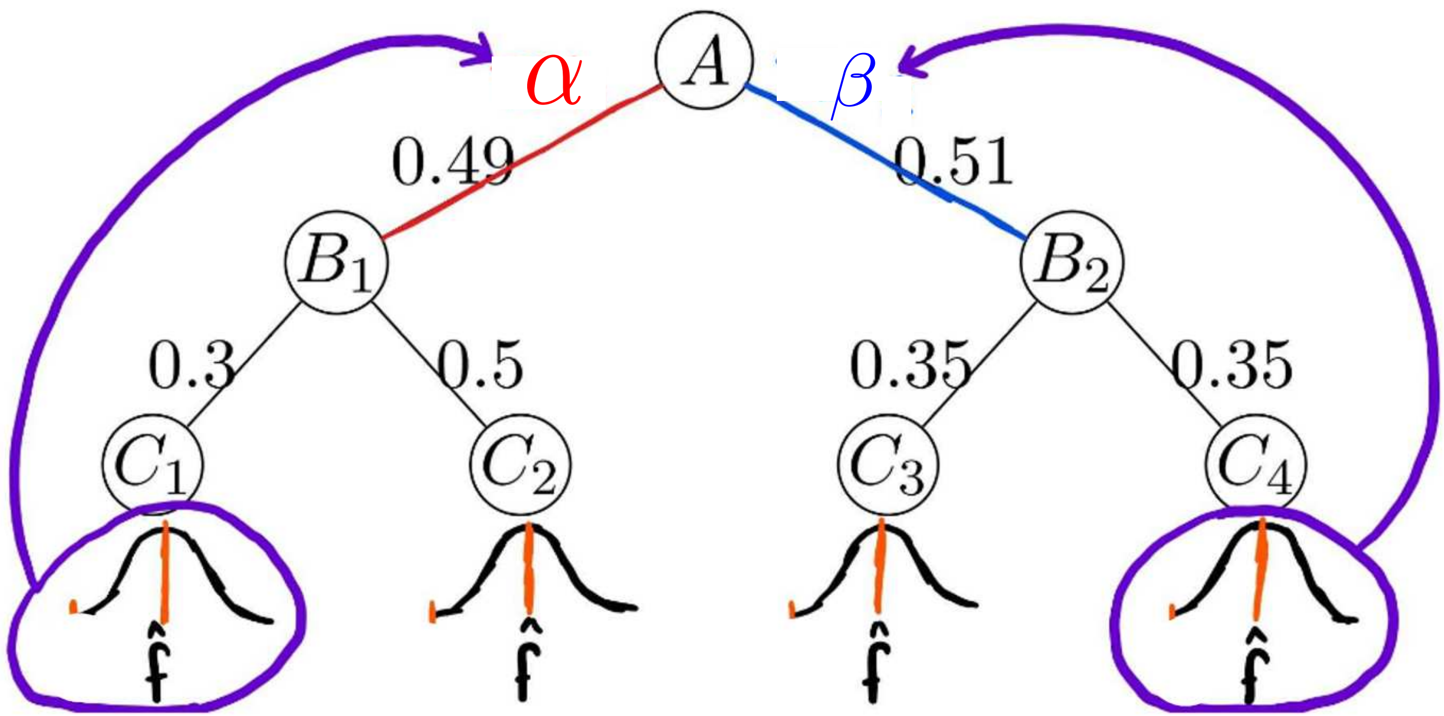
expand under α or β?



expand under the TLA that minimizes risk!

expand under β!

Backup Rules: Nancy



Nancy (AAAI-19):

parent $\leftarrow$ belief with minimum $\hat{f}$ among successors

conveys an entire belief distribution

How to Form The Belief Distribution?

Nancy: Heuristic values: scalar $\rightarrow$ probability distribution (belief)

But where do beliefs come from?

How to Form The Belief Distribution?

Nancy: Heuristic values: scalar $\rightarrow$ probability distribution (belief)

But where do beliefs come from?

Nancy: beliefs based on assumptions

truncated Gassian based on h cost-to-go and d hops-to-go
online learning with few parameters

Data-Driven Nancy:

Replace the assumptions with actual data.
offline learning with many parameters (histogram)

Data-Driven Nancy

Learning a Model of Heuristic Error

belief: distribution of h^* given features of state (h)

Gathering data:

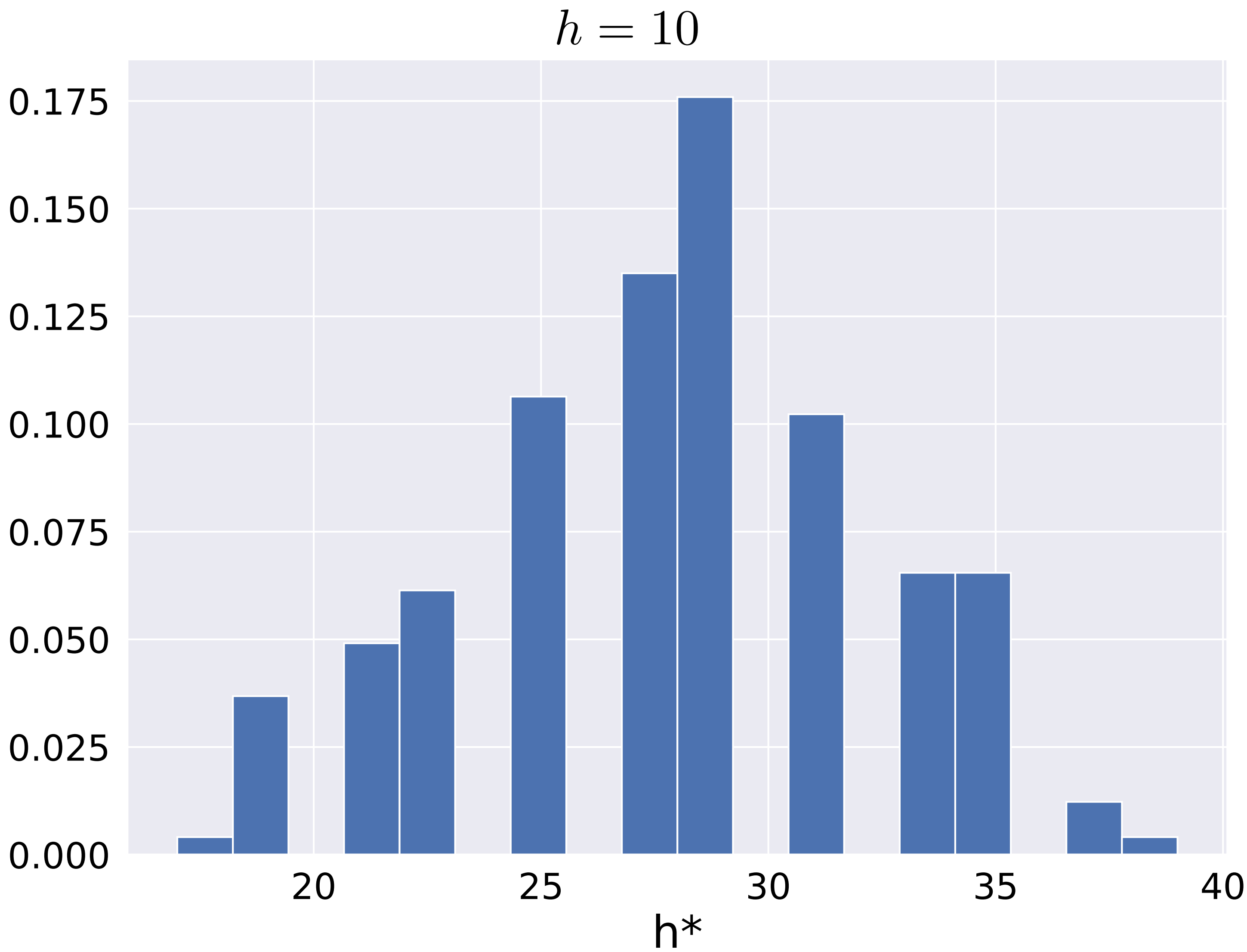
run weighted-A* on random problems and collect all states

for each observed h value:

pick most common 200 states from the collection,

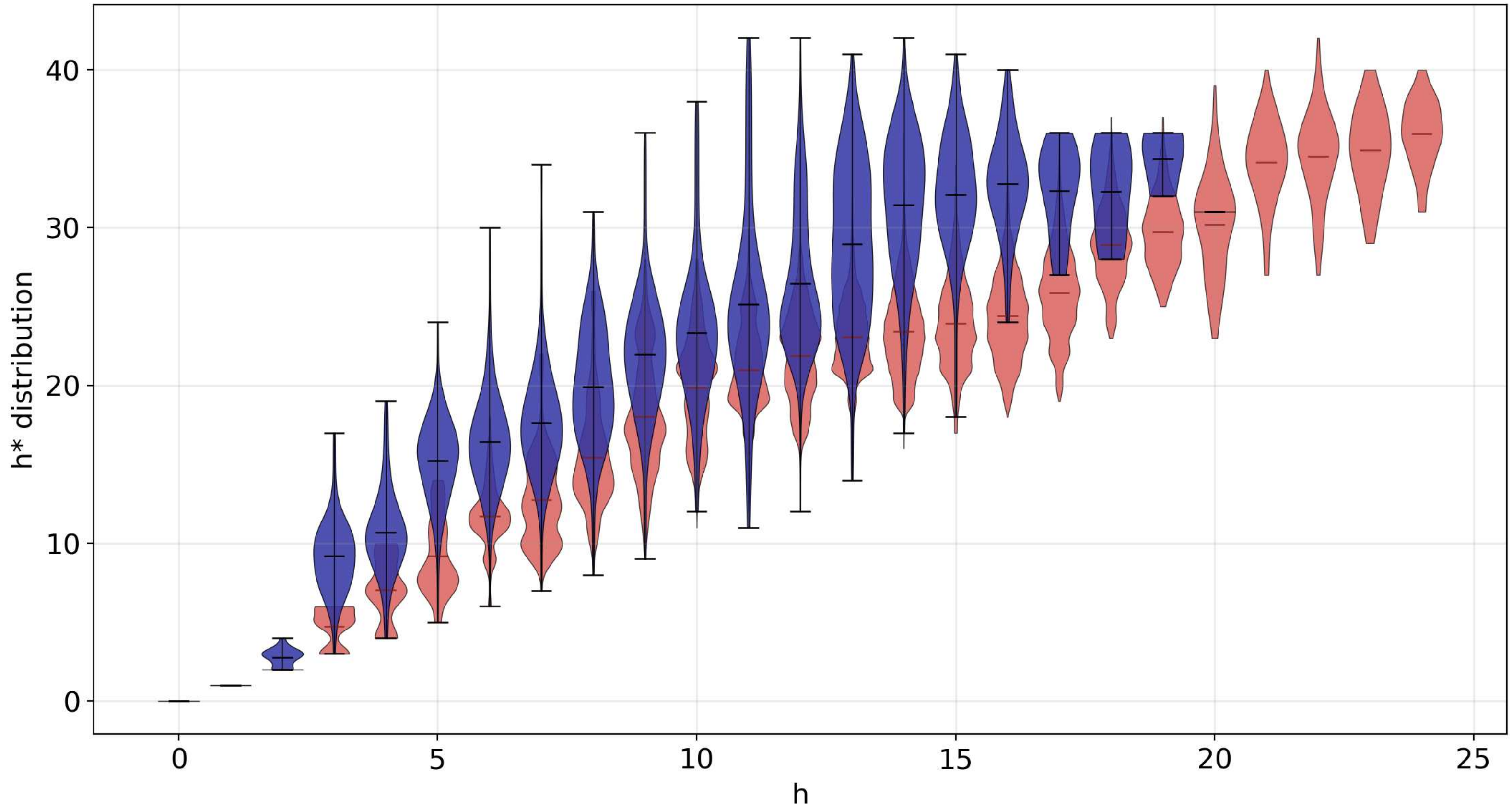
compute h^*

Example h* distribution: Sliding Puzzle



Example h^* distribution: Transport vs Blocks World

What is the data-driven distribution looks like?



Beliefs are different from domain to domain!
In many domains, data are not Gaussian!

Completeness proof

Problem:

Original Nancy is incomplete due to subtle issue:
not guaranteed to see best node from previous iteration in next one

Completeness proof

Problem:

Original Nancy is incomplete due to subtle issue:
not guaranteed to see best node from previous iteration in next one

Our solution:

Persist on the previous target state if current lookahead does not yield a better one (with lower $\hat{f}$)

This proof applies to any LSS-LRTA*-style algorithm

Results

Mean Solution Cost on Planning Domains

Domain	L	LSS-LRTA*	Nancy ('19)	Nancy (P)	Nancy (P+DD)
Blocksw.	100	46	67	**33**	38
	300	36	46	**30**	34
	1000	30	44	32	**27**
Transport	100	631	1116	615	**496**
	300	519	705	559	**485**
	1000	499	607	567	**422**
Transport (unit-cost)	100	48	79	40	**31**
	300	47	43	**30**	34
	1000	35	36	29	**27**
Elevators (unit-cost)	100	50	55	**35**	39
	300	32	40	**29**	30
	1000	34	31	27	**26**

Data lets Nancy work when assumptions fail!

Conclusions

Summary

- Nancy start to explore an optimal way of doing online heuristic search
- Nancy outperforms conventional LSS-LRTA* in cost and run time
- Replacing assumptions with data increase robustness
- General completeness proof

More broadly:

- Setting isolates the issue: unlike in MDPs or RL, all uncertainty is due to bounded rationality
- Metareasoning about uncertainty pays off, even for deterministic domains!

Questions?