Distributional Methods for Heuristic Search

Tianyi Gu

Advisor: Wheeler Ruml

University of New Hampshire

About Me

6th year CS PhD at UNH

Research: heuristic search and planning

- real-time heuristic search
- suboptimal search
- metareasoning

heuristic search can benefit from representing uncertainty

scalar heuristic $\rightarrow$ belief distribution that represents uncertainty

About Me

6th year CS PhD at UNH

Research: heuristic search and planning

- **real-time heuristic search**
- suboptimal search
- metareasoning

heuristic search can benefit from representing uncertainty

scalar heuristic $\rightarrow$ belief distribution that represents uncertainty

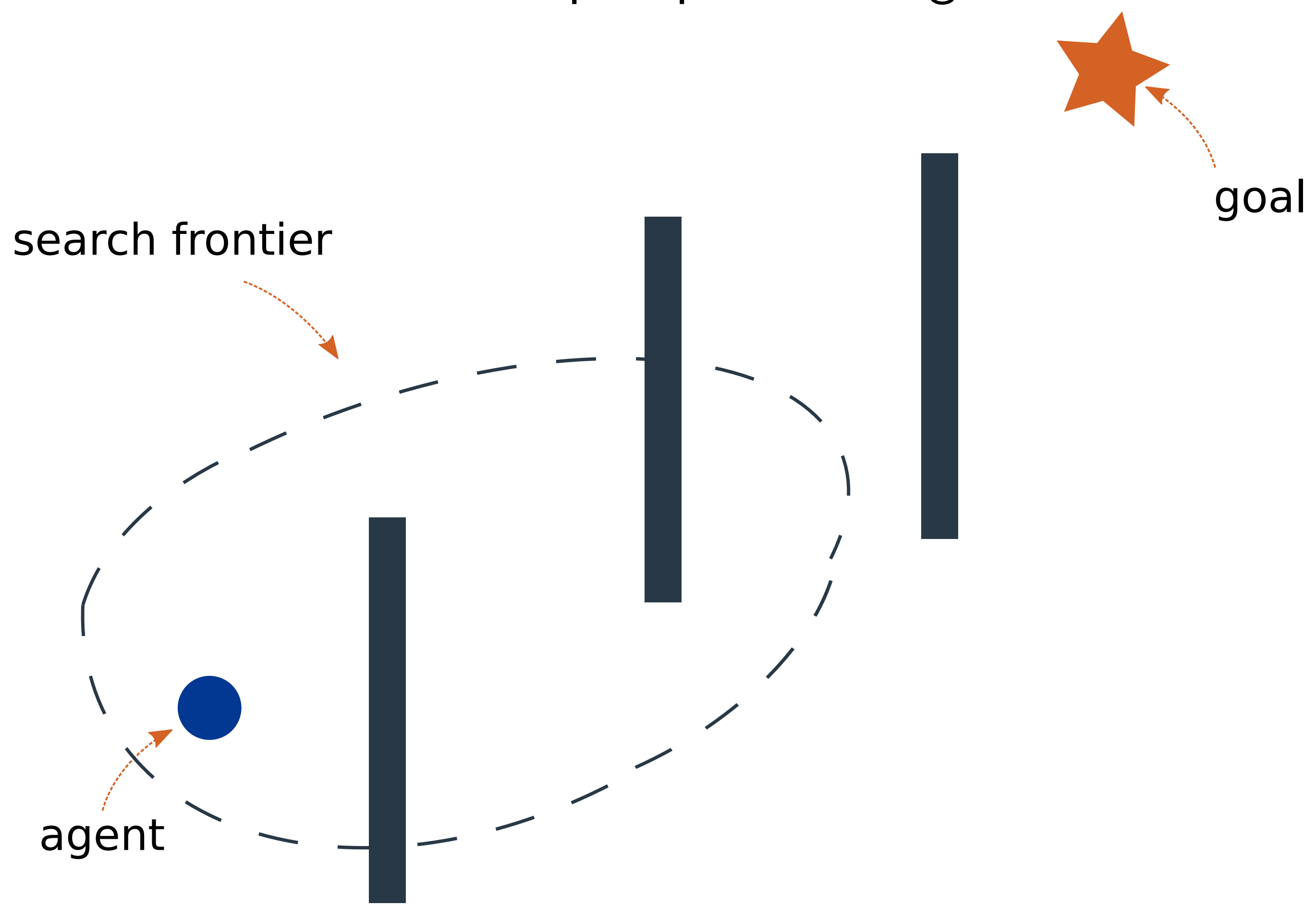
What is Real-time Heuristic Search?

An example: path finding

agent performs search for a bounded time

What is Real-time Heuristic Search?

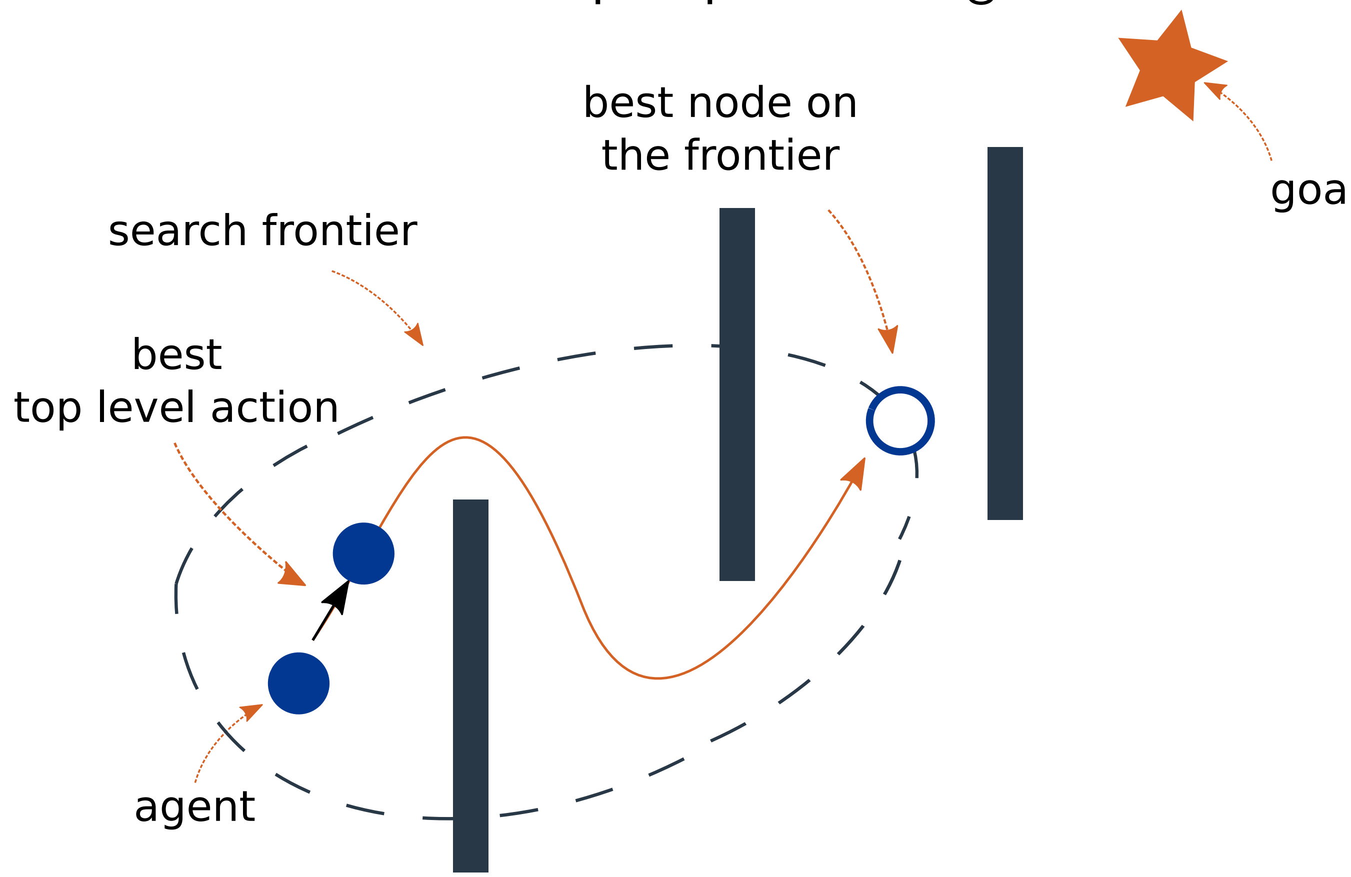
An example: path finding



agent performs search for a bounded time

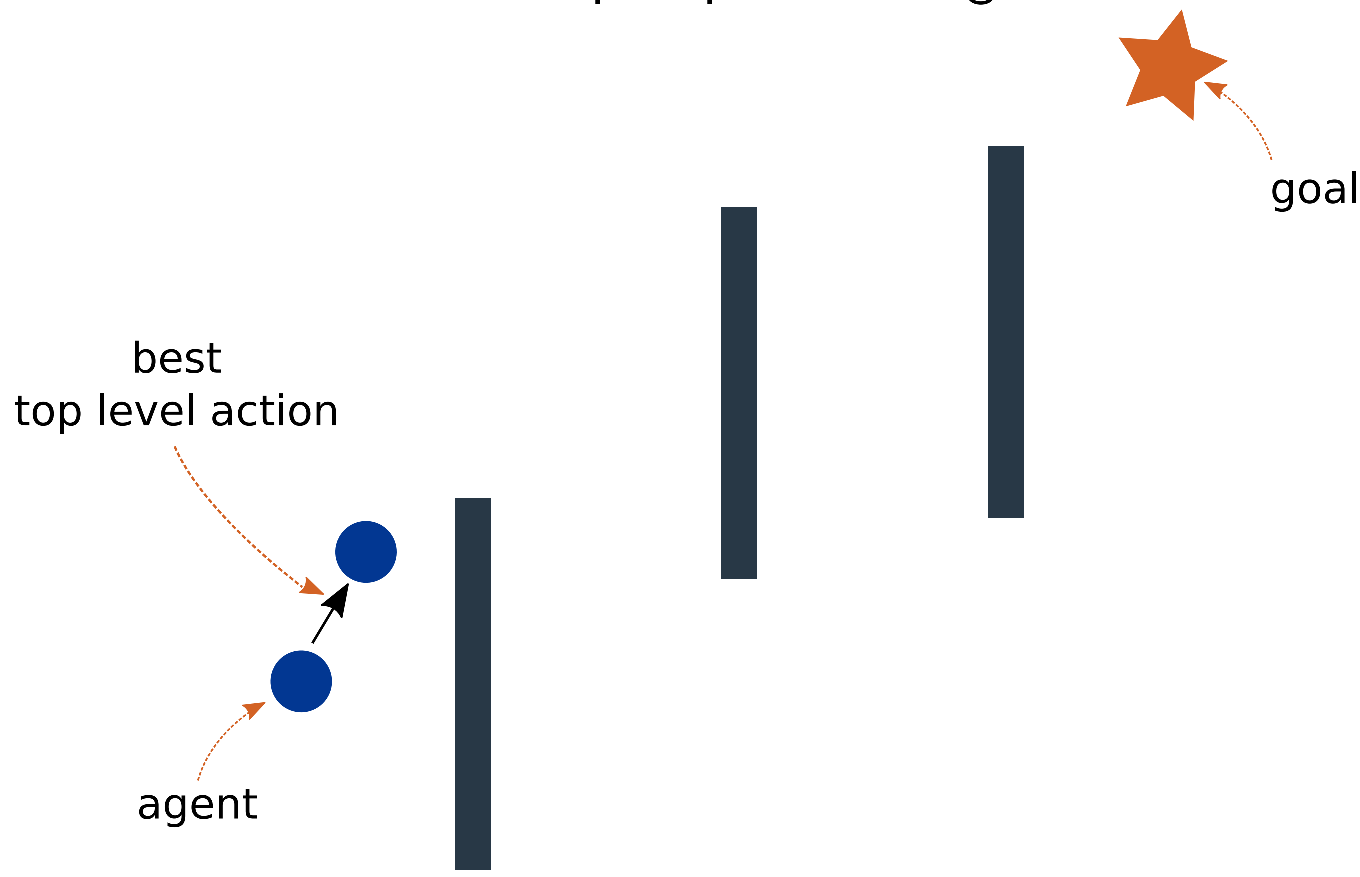
What is Real-time Heuristic Search?

An example: path finding



agent commits to best action and executes

An example: path finding



agent commits to best action and executes

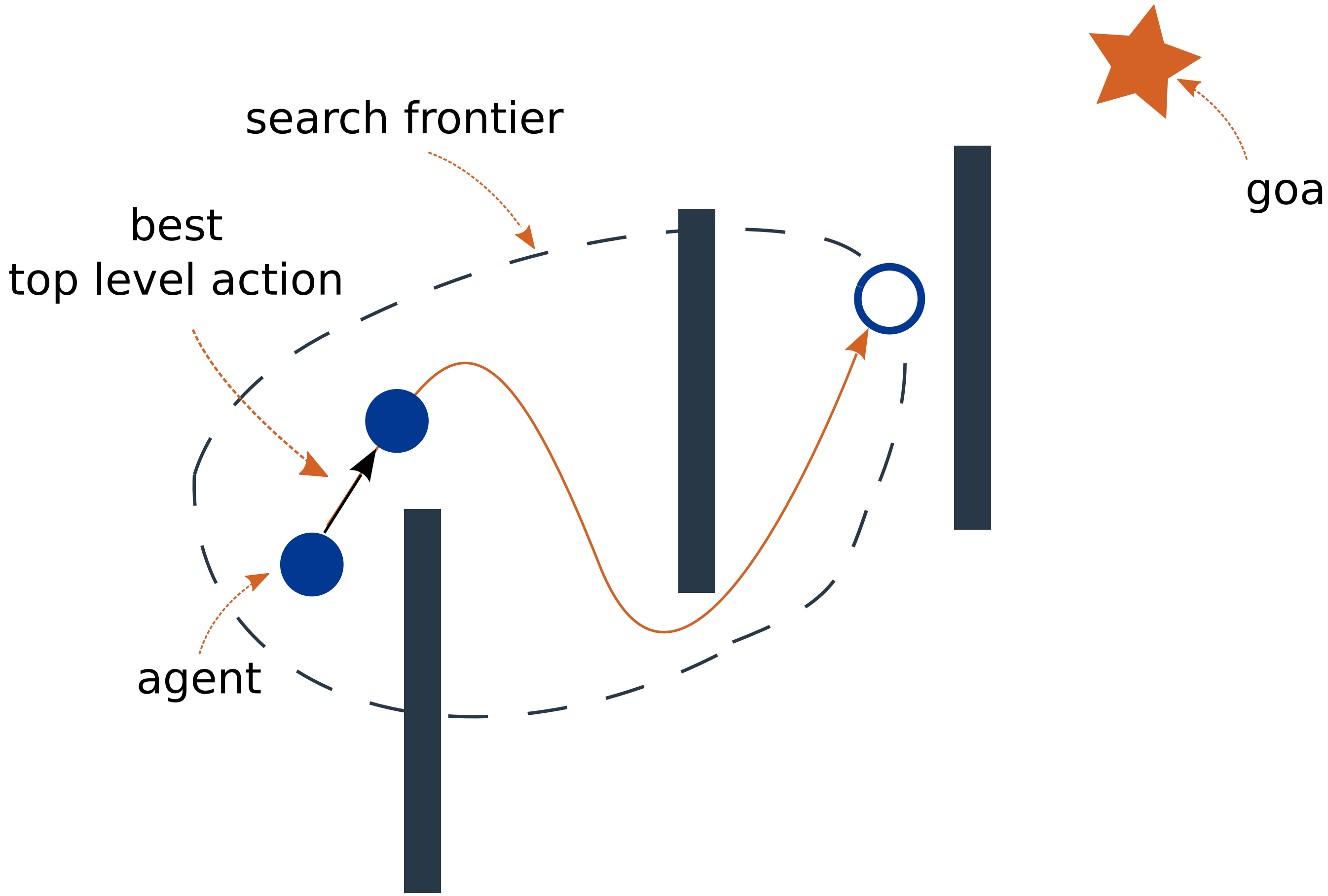
What is Real-time Heuristic Search?

An example: path finding

agent commits to best action and executes

What is Real-time Heuristic Search?

An example: path finding



online planning: interleaving search and action execution

“receding horizon control”

Motivation for Real-time Heuristic Search

Real-time heuristic search:

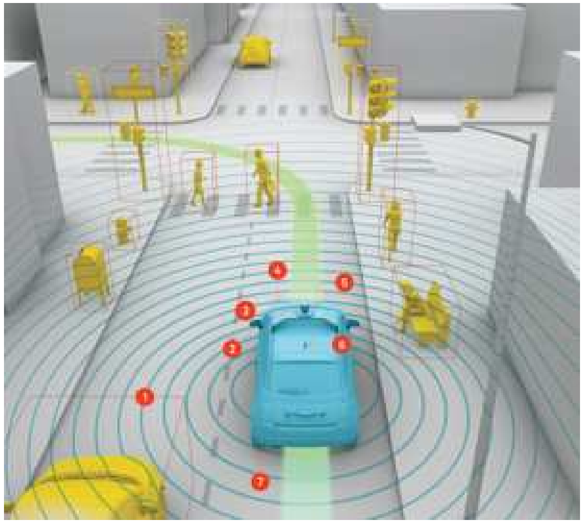
return the next action within a time bound

Applications:

interacting with humans

dynamic environment

- autonomous vehicle
  inaccurate sensor
  update model online



Real-time Search as Decision-making Under Uncertainty: The Nancy Framework

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)

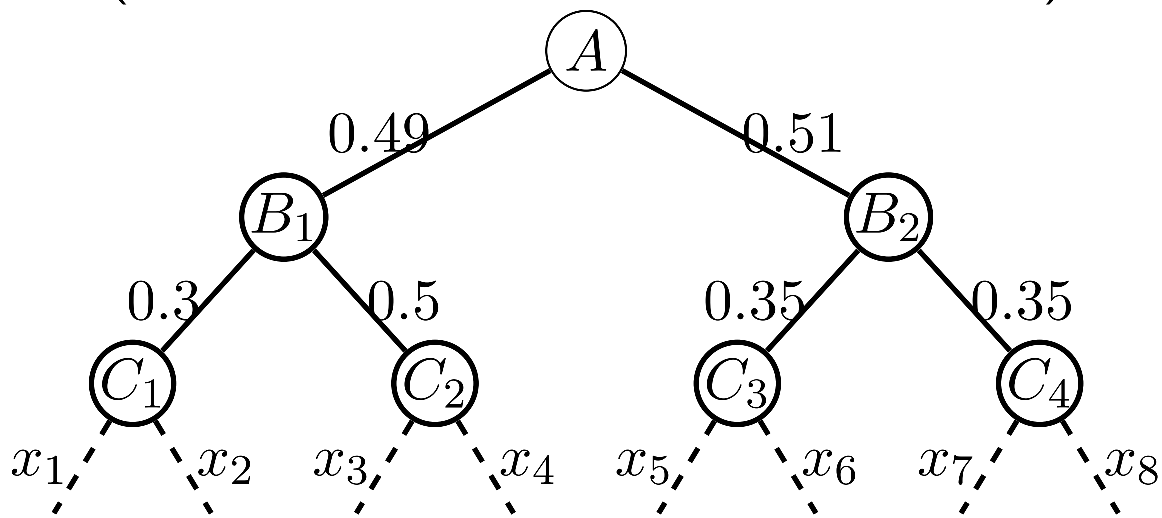
repeat until at a goal

derived from offline search, but optimal for online?

Decision-making Phase: A Troublesome Example

Should an agent at A move to B_1 or B_2?

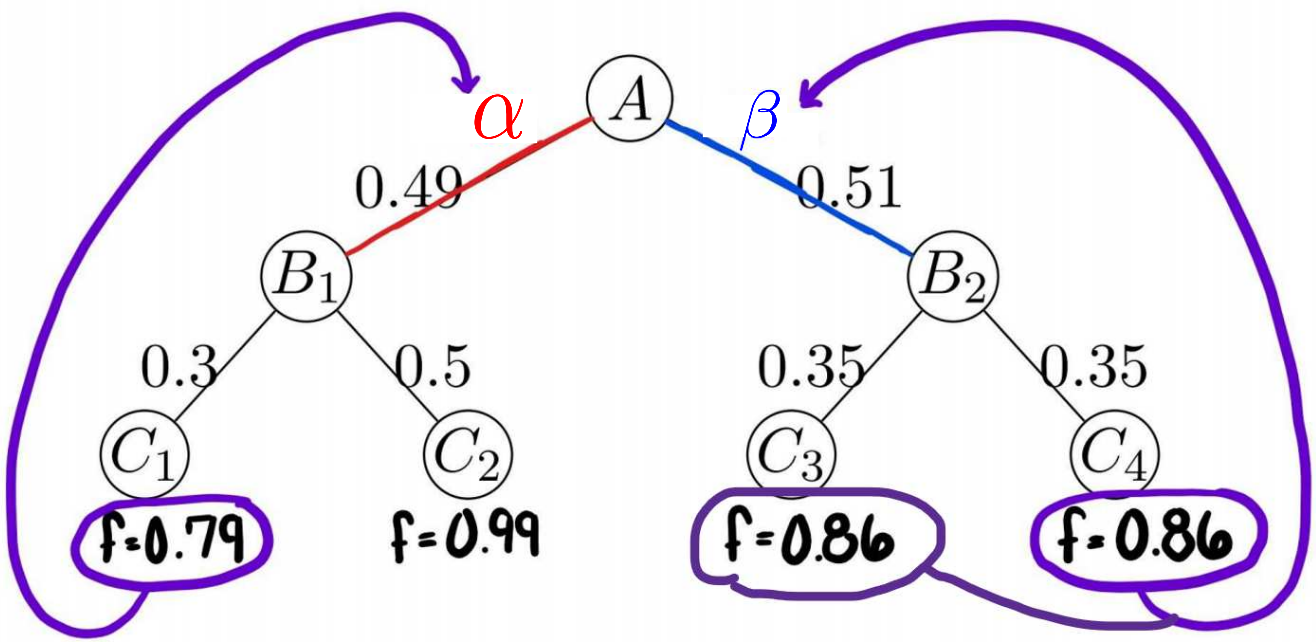
(x_i are unknown but i.i.d. uniform 0-1)



random tree domain (Pemberton & Korf 1995)

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

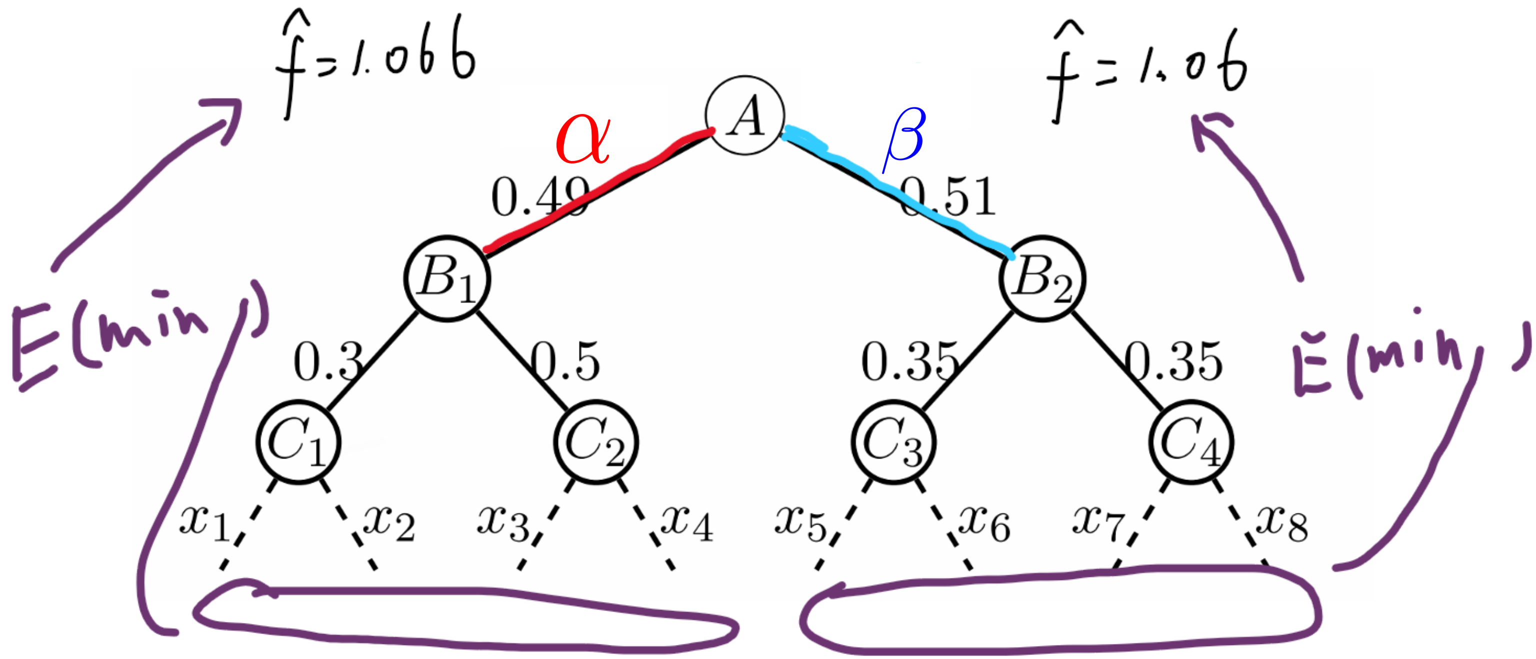
Should an agent at A move to B_1 or B_2?
(x_i are unknown but i.i.d. uniform 0-1)



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

Decision-making Phase: A Troublesome Example

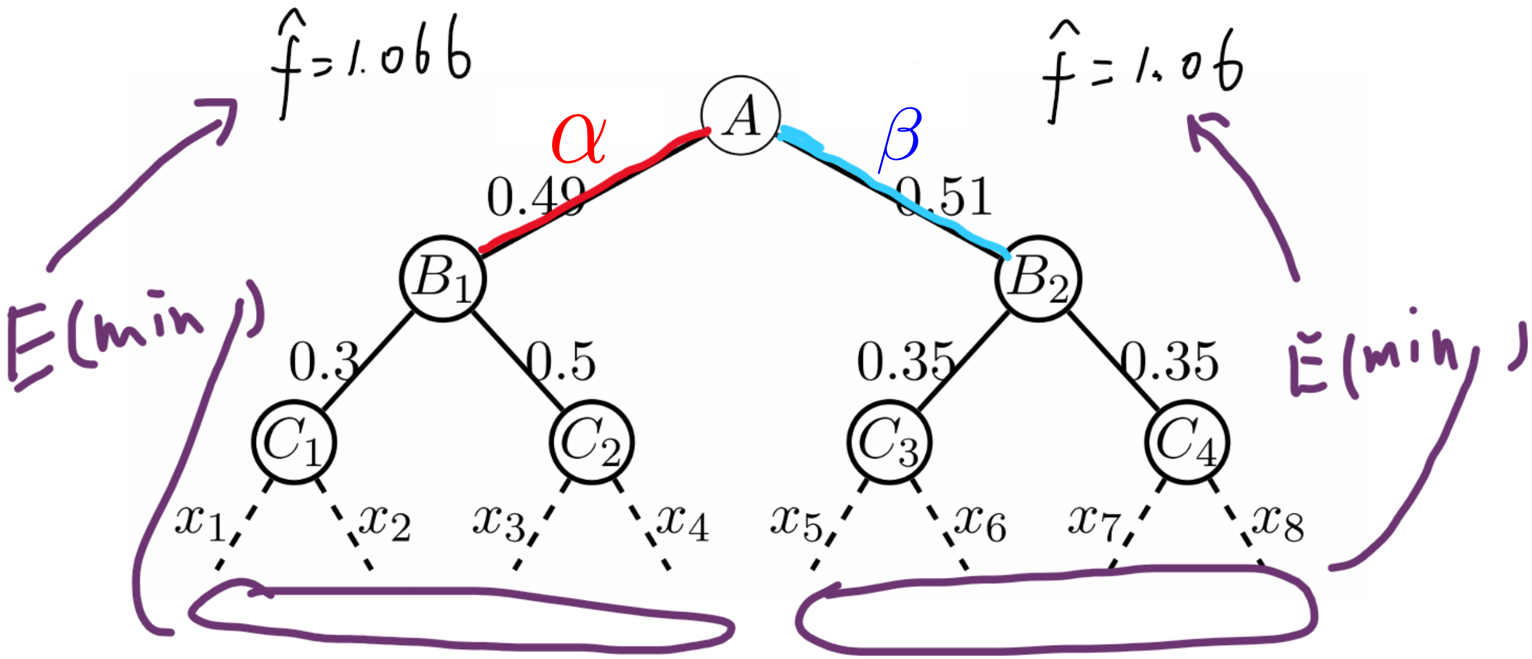
Should an agent at A move to B_1 or B_2?
(x_i are unknown but i.i.d. uniform 0-1)



$\hat{f}$ is expected plan cost

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)

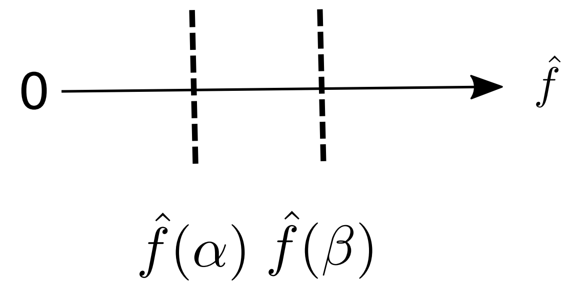


$\hat{f}$ is expected plan cost

f is not the answer: should minimize expected value!

plan under time pressure $\rightarrow$ bounded rationality

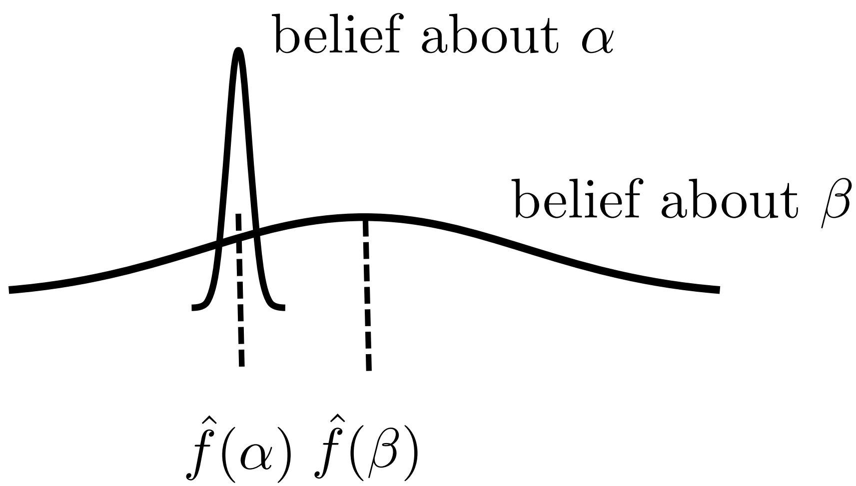
Lookahead Phase: A Troublesome Example



$\hat{f}$ 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 [1]**:**

want to maximize value of information

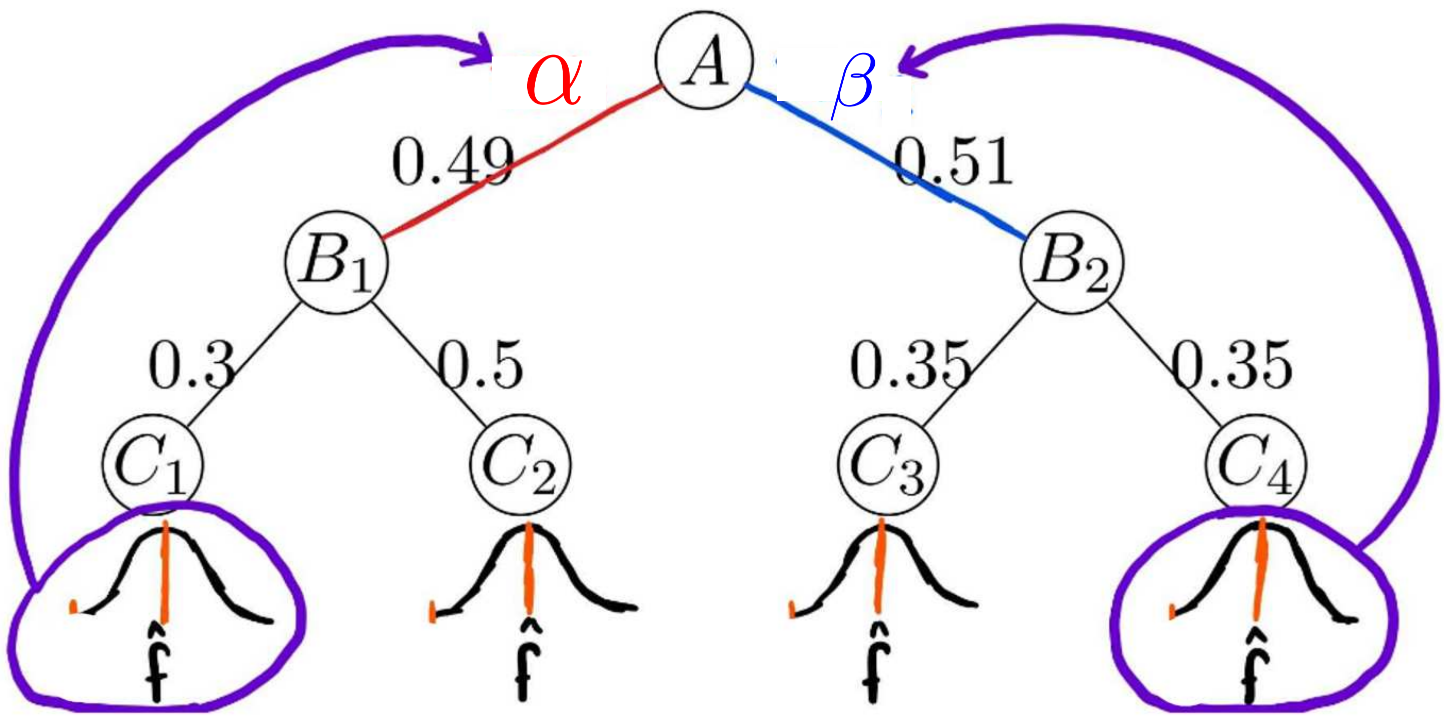
expand nodes which minimize expected regret

relies on belief of values

choose expansions that decrease uncertainty in beliefs

[1] Real-time Planning as Decision-making Under Uncertainty, Andrew Mitchell, Wheeler Ruml, Fabian Spaniol, Joerg Hoffmann, and Marek Petrik, AAAI, 2019.

Backup Rules: Nancy



Nancy:

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

conveys an entire belief distribution

How to Form The Belief Distribution?

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

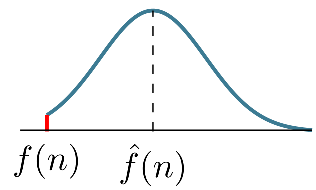
But where do beliefs come from?

How to Form The Belief Distribution?

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

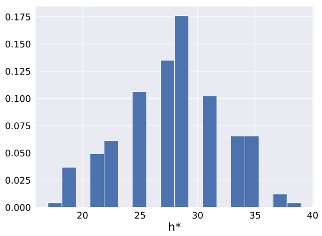
But where do beliefs come from?

Nancy:



truncated Gaussian based on $\hat{f}$ and f,
few parameters allows online learning

Data-Driven Nancy[2]:



expressive histogram,
many parameters requires offline learning

[2] Beliefs We Can Believe In: Replacing Assumptions with Data in Real-Time Search, Maximilian Fickert, Tianyi Gu, Leonhard Staut, Wheeler Ruml, Joerg Hoffmann, and Marek Petrik, AAAI, 2020.

Mean Solution Cost on Planning Domains

Domain	Lookahead	LSS-LRTA*	Nancy	Nancy (DD)
Blocksw.	100	46	**33**	38
	300	36	**30**	34
	1000	30	32	**27**
Transport	100	631	615	**496**
	300	519	559	**485**
	1000	499	567	**422**
Transport (unit-cost)	100	48	40	**31**
	300	47	**30**	34
	1000	35	29	**27**
Elevators (unit-cost)	100	50	**35**	39
	300	32	**29**	30
	1000	34	27	**26**

Both version of Nancy outperform conventional approach!

Conclusions

Summary

Examples of using distribution to guide search:

- real-time planning: Nancy, DDNancy
- suboptimal search: XES (IJCAI-21 paper 994)
- robotics: BEAST (IROS-17)

Summary

Examples of using distribution to guide search:

- real-time planning: Nancy, DDNancy
- suboptimal search: XES (IJCAI-21 paper 994)
- robotics: BEAST (IROS-17)

Exciting time in AI!

- Planning, RL, ML, Robotics

Examples of using distribution to guide search:

- real-time planning: Nancy, DDNancy
- suboptimal search: XES (IJCAI-21 paper 994)
- robotics: BEAST (IROS-17)

Exciting time in AI!

- Planning, RL, ML, Robotics

Much work needs to be done!

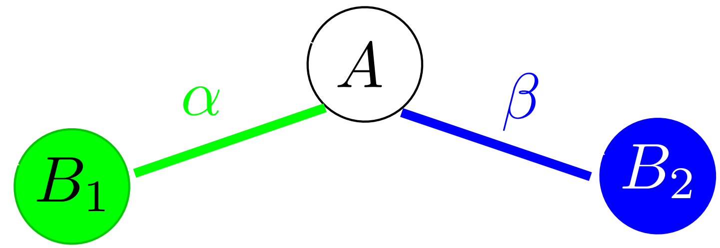
- data-driven + planning
- statistics + model-based approach

Questions?

Back-up Slides

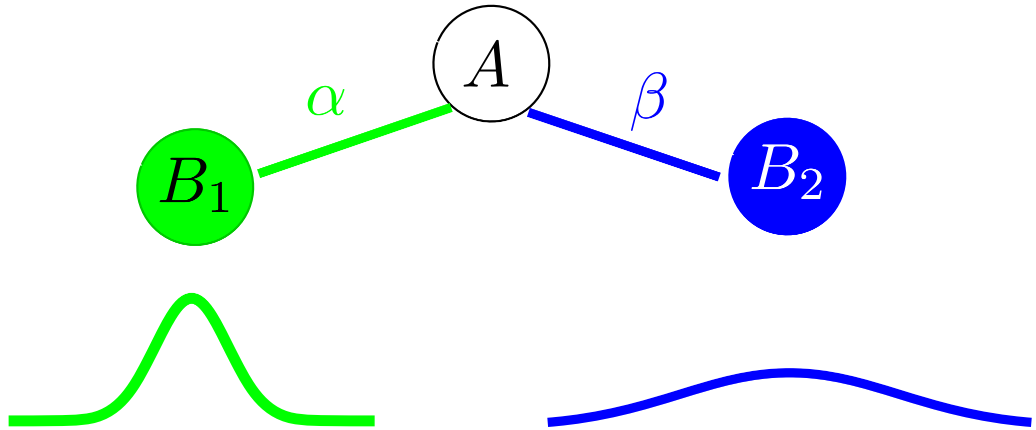
Risk-based Lookahead Example

expand under α or β?



Risk-based Lookahead Example

expand under α or β?

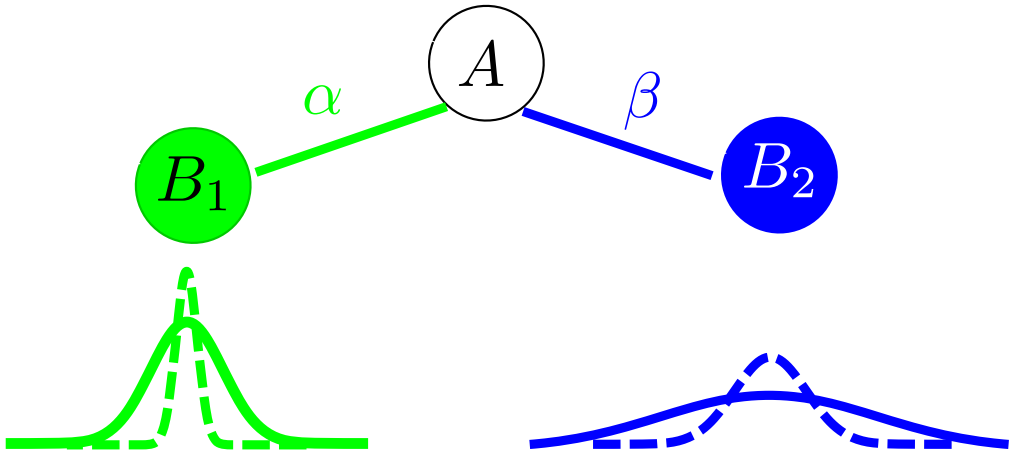


need 2 things:

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

Risk-based Lookahead Example

expand under α or β?

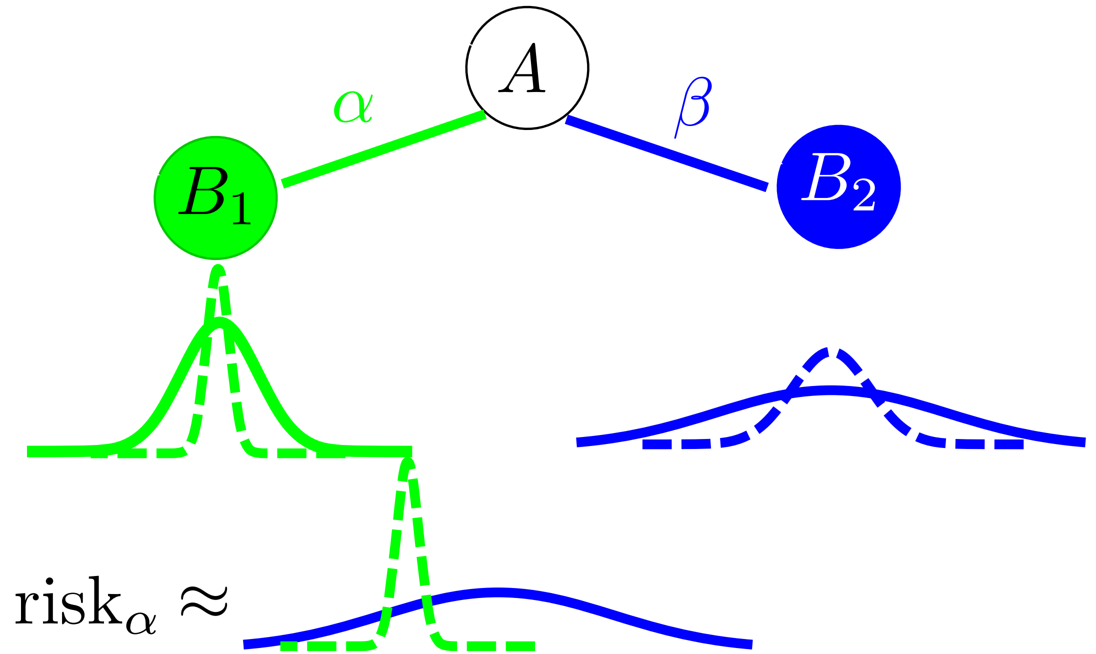


need 2 things:

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

Risk-based Lookahead Example

expand under α or β?

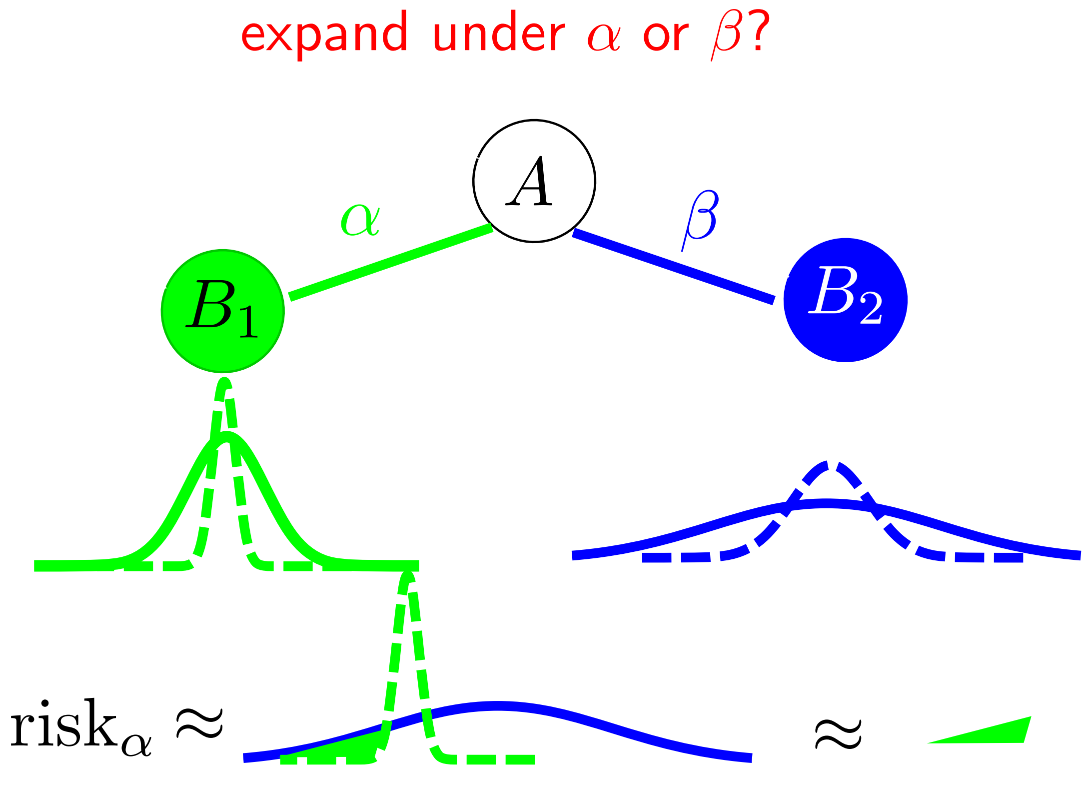


Risk: expected regret if a suboptimal action is selected

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

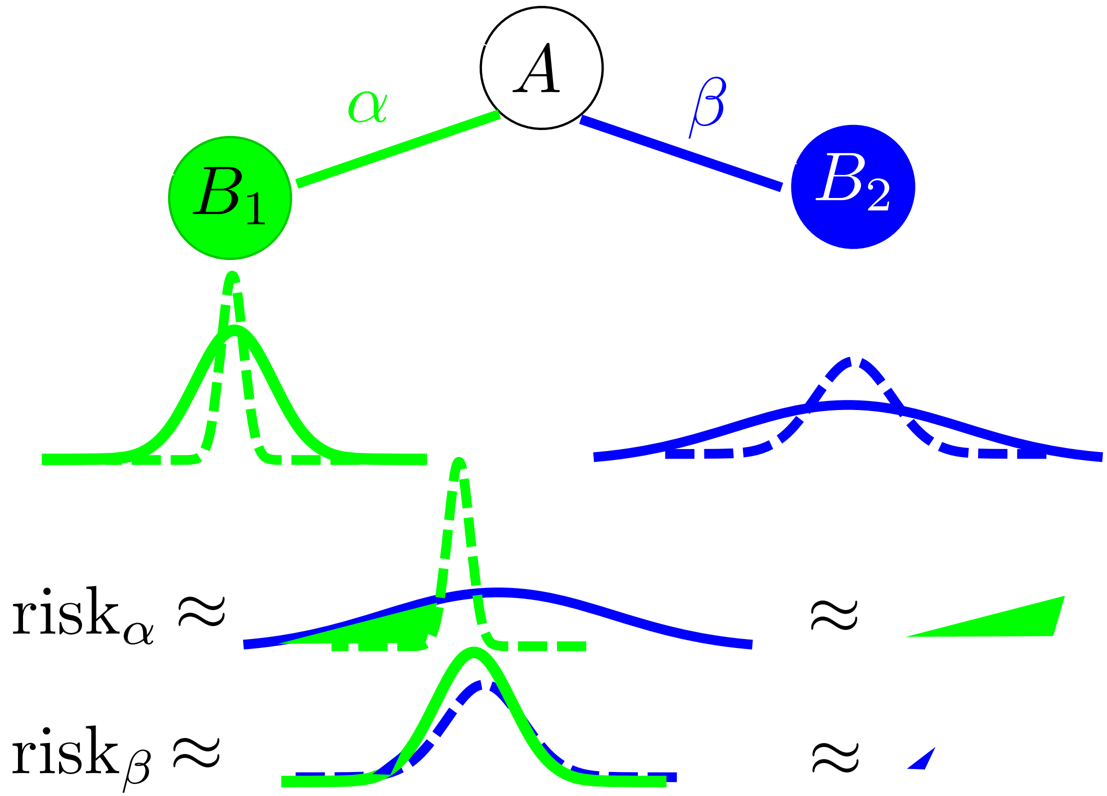


Risk: expected regret if a suboptimal action is selected
α is TLA with lowest expected value, other is β

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

Risk-based Lookahead Example

expand under α or β?



expand under the TLA that minimizes risk!

expand under β!

Completeness proof

Lemma 1 *Under assumptions of goal-awareness and finite state space, if a real-time search algorithm is incomplete, it must have a circulating set* $S_\circ$.

Lemma 5 *Under our assumptions, a reasonable real-time search algorithm cannot have a circulating set.*

Theorem 1 *Under our assumptions, a reasonable real-time search algorithm will eventually reach a goal.*

Lemma 7 *Nancy is a reasonable real-time search algorithm.*

Lemma 8 *LSS-LRTA* is a reasonable real-time search algorithm.*

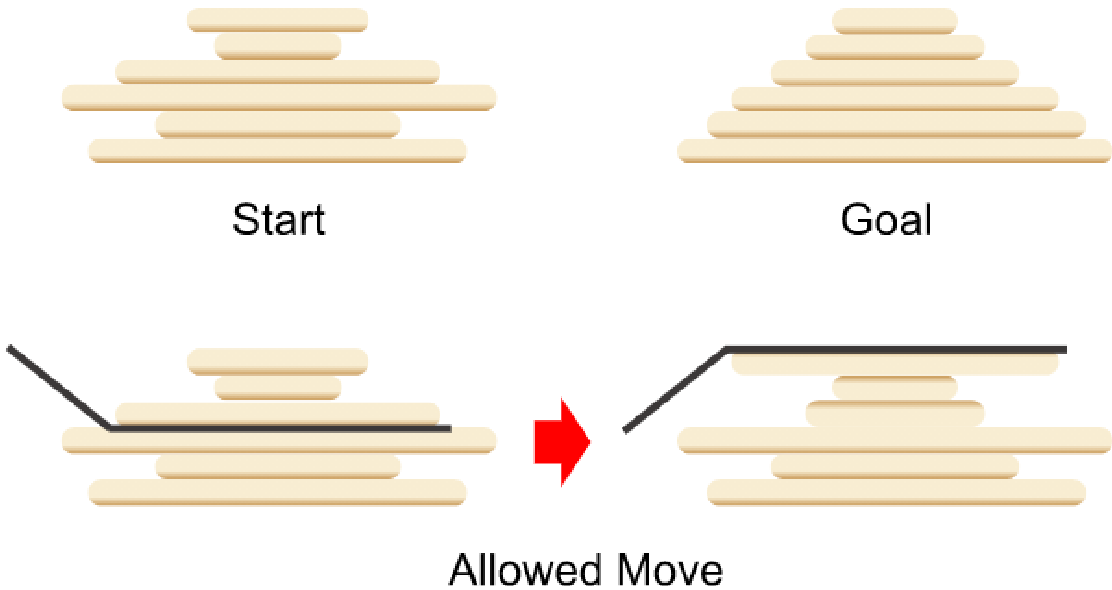
This proof applies to any LSS-LRTA*-style algorithm: no longer need heuristic consistency!

Search Domains

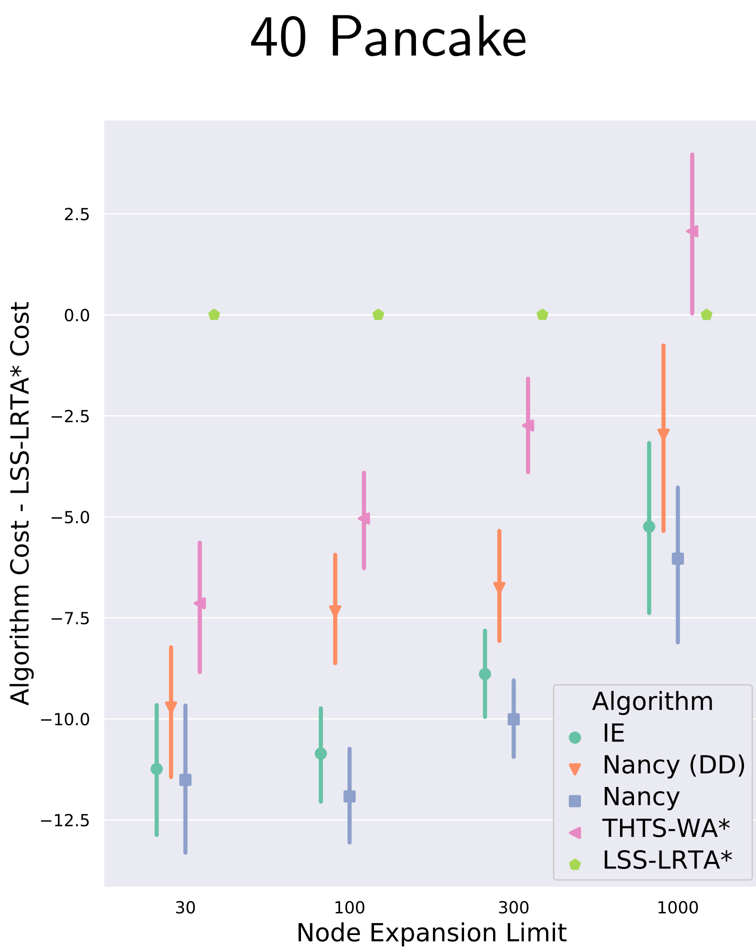
- sliding tile puzzle
  - uniform, heavy ,inverse
- pancake puzzle
  - different size
- racetrack
  - reminiscent of autonomous driving

Comparison to IE and MCTS on Classic Search Domains

pancake puzzle



Comparison to IE and MCTS on Classic Search Domains



Nancy outperforms conventional approaches and MCTS[3]

[3] Real-time Planning as Data-driven Decision-making, Maximilian Fickert, Tianyi Gu, Leonhard Staut, Sai Lekyang, Wheeler Ruml, Joerg Hoffmann, and Marek Petrik, Bridging the Gap Between AI Planning and Reinforcement Learning (PRL), 2020.