Anticipatory On-line Planning

Ethan Burns[1], J. Benton[2], Wheeler Ruml[1], Sungwook Yoon[3], and Minh Do[4]

1 University of New Hampshire

2 Arizona State University

3 parc Palo Alto Research Center

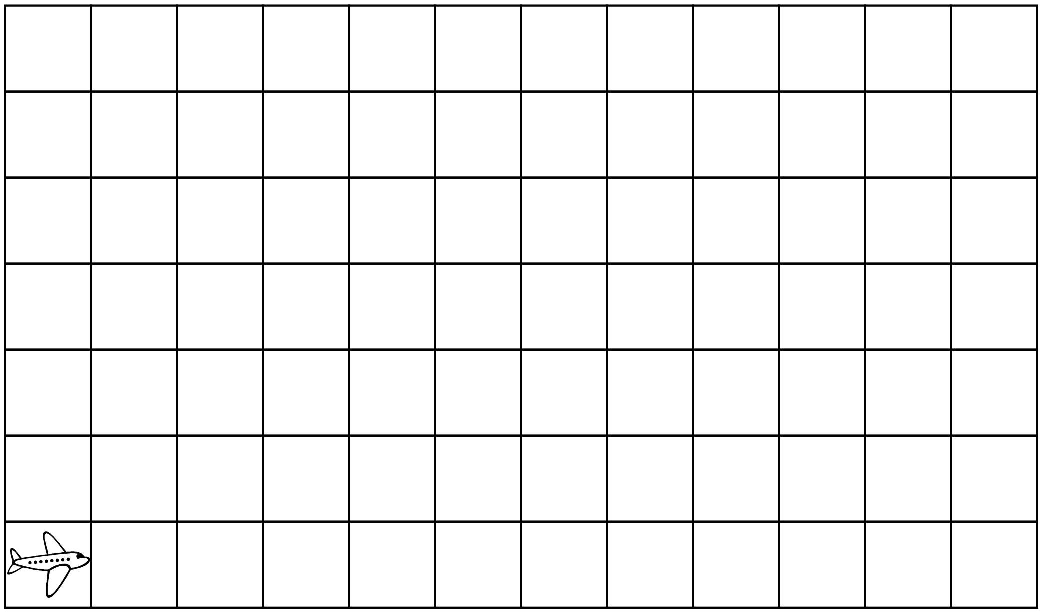
4 Stinger Ghaffarian Technologies

[Thanks to NSF, DARPA, ONR, and ARL for support]

An On-line Planning Problem

Consider a UAV fulfilling observation requests:

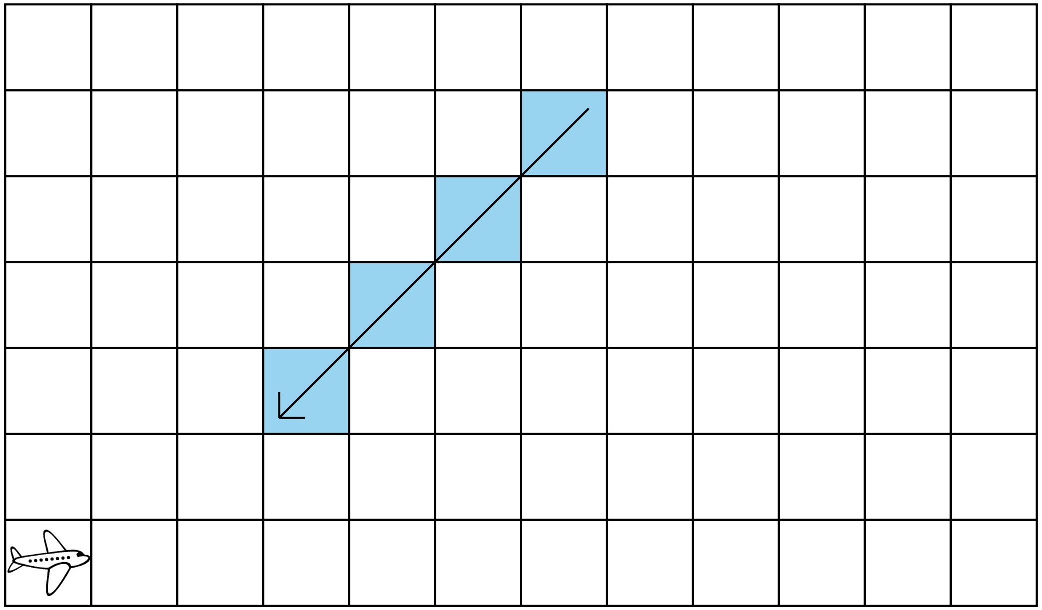
- Observation requests (start–end locations) arrive over time
  - Requests draw from known distribution
- Minimize time to service observations
  - Re-planning may reduce cost.



An On-line Planning Problem

Consider a UAV fulfilling observation requests:

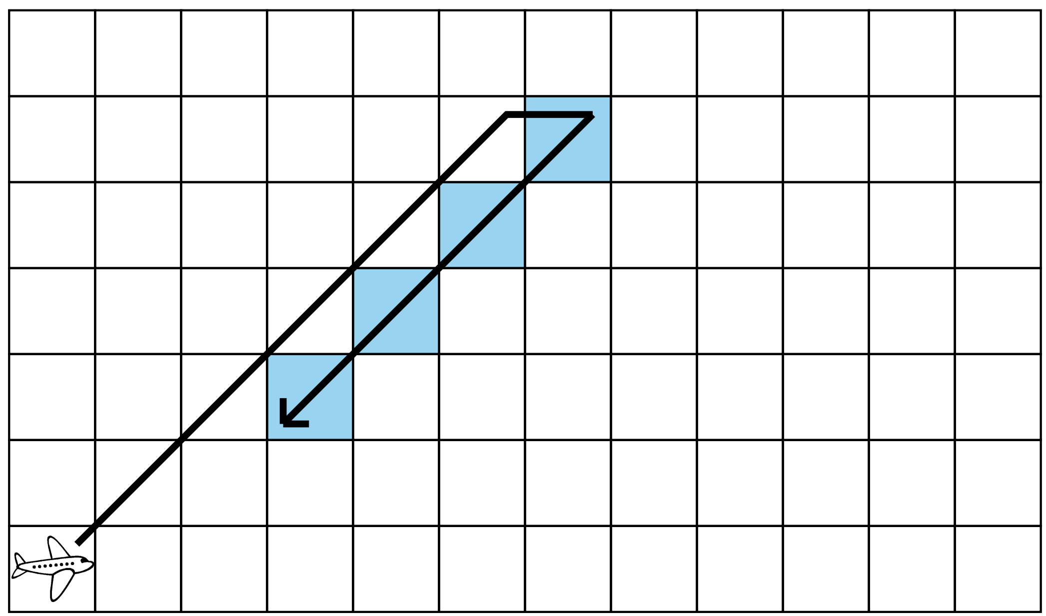
- Observation requests (start–end locations) arrive over time
  - Requests draw from known distribution
- Minimize time to service observations
  - Re-planning may reduce cost.



An On-line Planning Problem

Consider a UAV fulfilling observation requests:

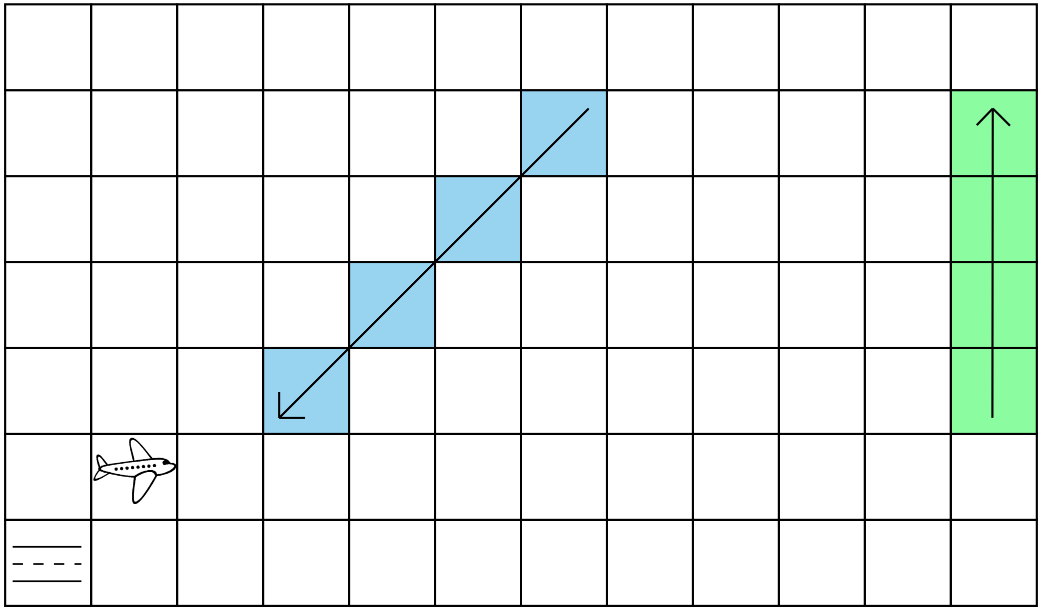
- Observation requests (start–end locations) arrive over time
  - Requests draw from known distribution
- Minimize time to service observations
  - Re-planning may reduce cost.



An On-line Planning Problem

Consider a UAV fulfilling observation requests:

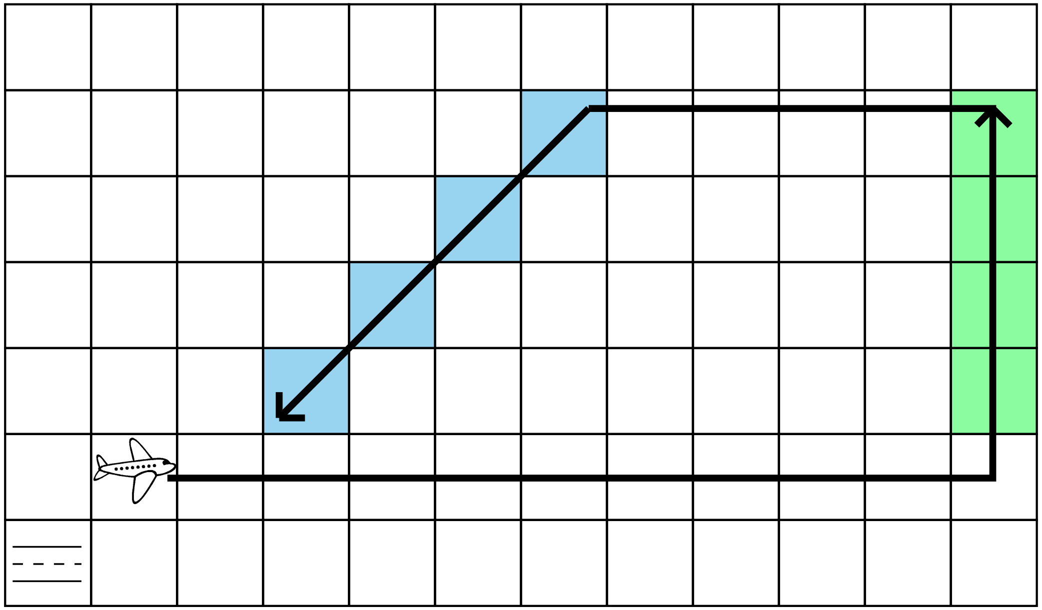
- Observation requests (start–end locations) arrive over time
  - Requests draw from known distribution
- Minimize time to service observations
  - Re-planning may reduce cost.



An On-line Planning Problem

Consider a UAV fulfilling observation requests:

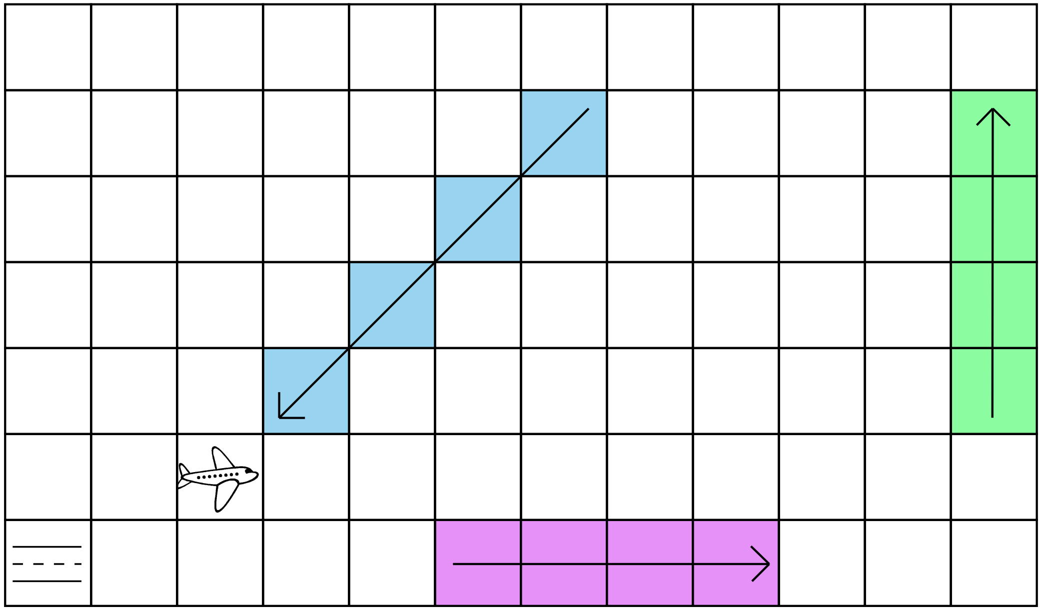
- Observation requests (start–end locations) arrive over time
  - Requests draw from known distribution
- Minimize time to service observations
  - Re-planning may reduce cost.



An On-line Planning Problem

Consider a UAV fulfilling observation requests:

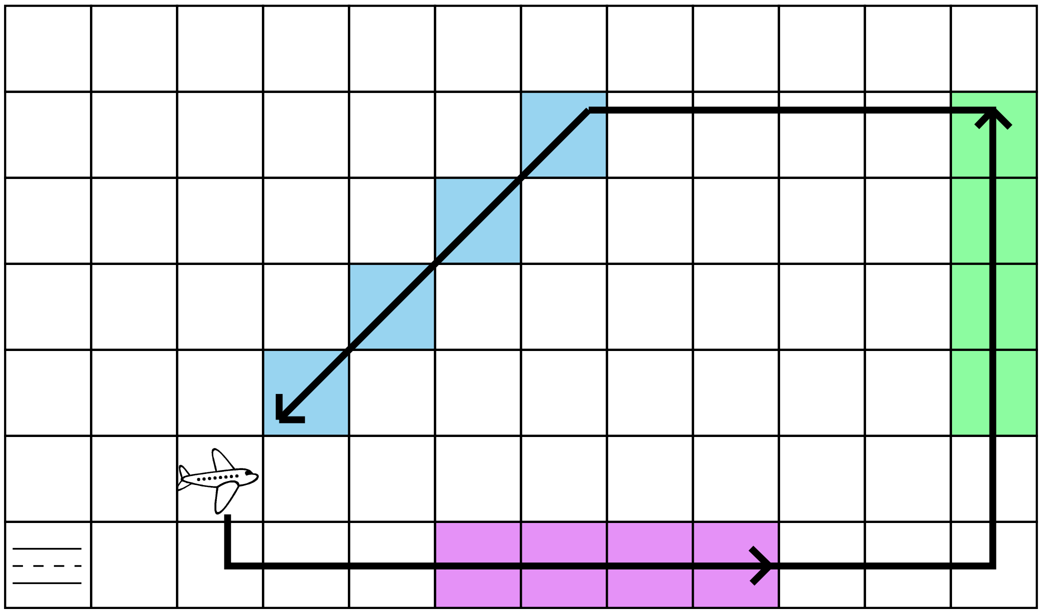
- Observation requests (start–end locations) arrive over time
  - Requests draw from known distribution
- Minimize time to service observations
  - Re-planning may reduce cost.



An On-line Planning Problem

Consider a UAV fulfilling observation requests:

- Observation requests (start–end locations) arrive over time
  - Requests draw from known distribution
- Minimize time to service observations
  - Re-planning may reduce cost.



Example Continued

Observations:

- Life is more than one goal—others will come
- "Planner" can run, even before goal arrives

Other examples:

- Life (e.g., insurance)
- Satellite planning
- Taxi/ambulance dispatching
- Manufacturing
- PARC printer

Opportunity:

- Estimate of future goals

Contributions

- Simple and clean formalization of on-line planning
    - *Exposes key issues, yet very approachable*
- Show that hindsight optimization applies easily
    - *Not just for probabilistic planning any more!*
- Show that simple HOP is better than "reactive" planning
    - *Many avenues for future work*

Formalization

On-line Continual Planning Problems

Like classical planning:

- Action effects are deterministic

but, on-line:

- Goals arrive stochastically, distribution is known

Like MDP:

- Minimize total cost, approximate over a fixed horizon, H
- When evaluating actions:

$$V_H^*(s_1) = \min_{a_1,\ldots,a_H} \mathop{E}_{s_2,\ldots,s_H} \left[\sum_{i=1}^{H} C(s_i, a_i) \right]$$

- State incorporates unachieved goals

Simple and clean formulation of on-line planning

A Simple Approach: Reactive Planning

1. Plan for the current goals
2. Execute the plan until the goals change
3. Repeat

A Simple Approach: Reactive Planning

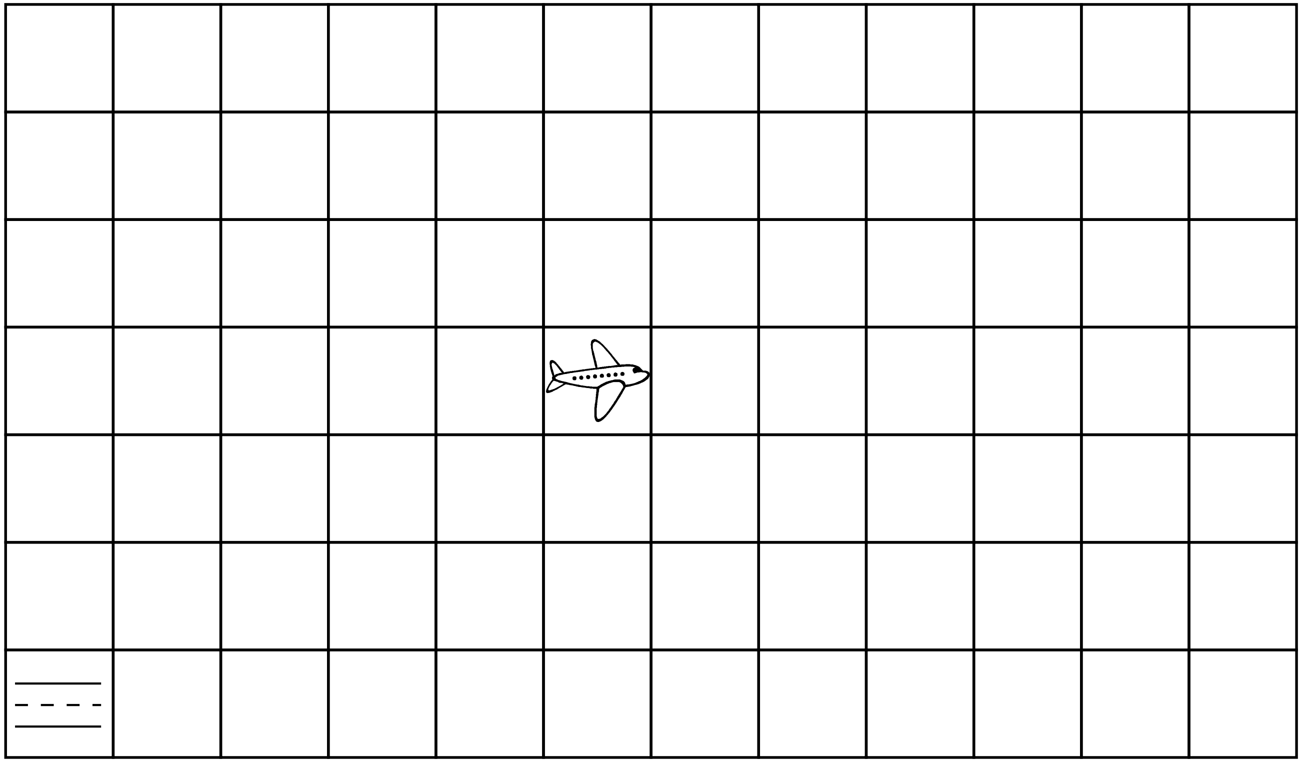
1. Plan for the current goals
2. Execute the plan until the goals change
3. Repeat

Doesn't take advantage of knowledge about future goals

Hindsight Optimization

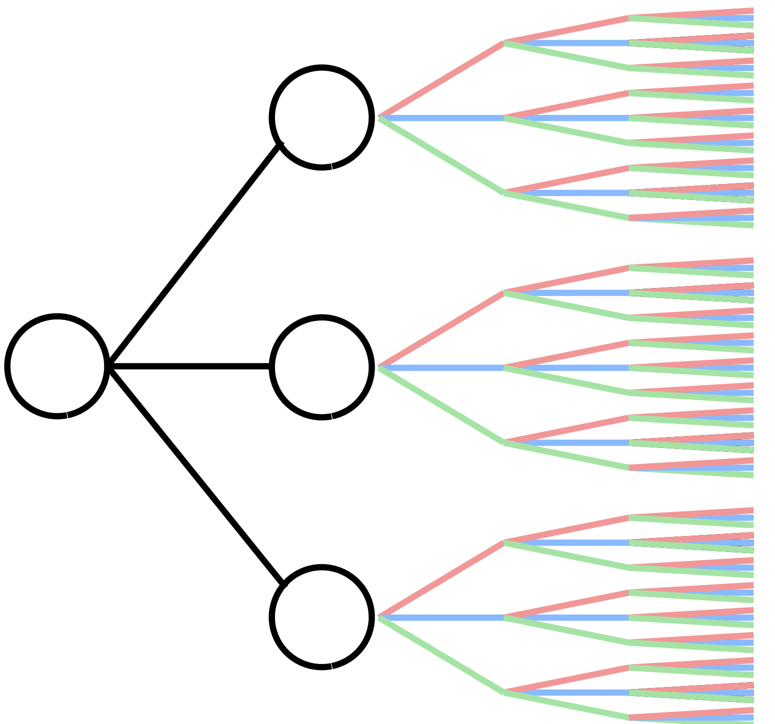
A Better Approach: Anticipatory On-line Planning

- Consider possible future goals when planning actions



Solve for all possible future goals

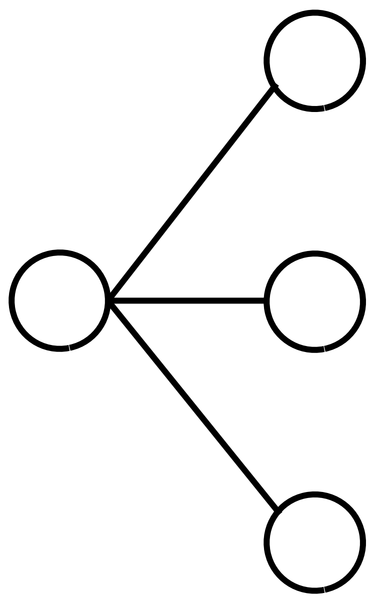
$$V_H^*(s_1) = \min_{a_1,\ldots,a_H} \underset{s_2,\ldots,s_H}{E} \left[\sum_{i=1}^{H} C(s_i, a_i) \right]$$



A 3x3 UAV with ≤ 3 requests has tens of millions of states.

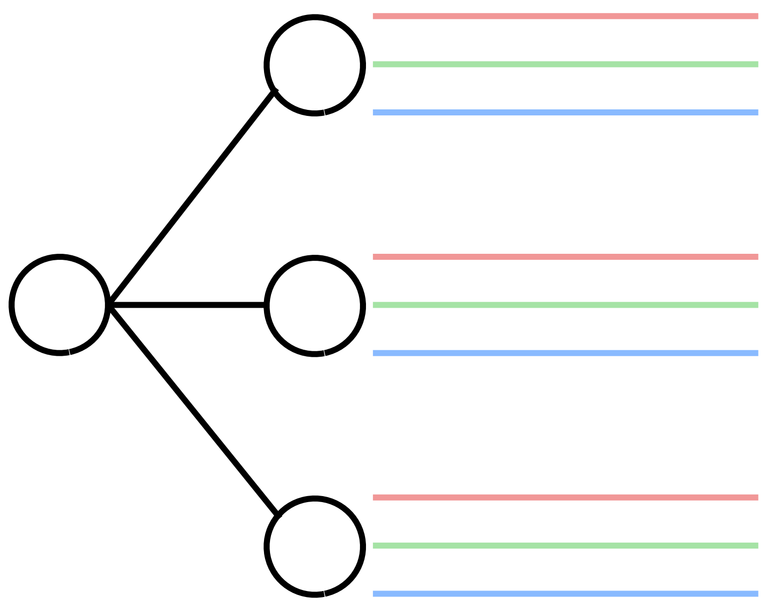
Hindsight optimization – solve a sample of future goals

$$\hat{V}_H(s_1) = \underset{s_2,\ldots,s_H}{E}\left[\min_{a_1,\ldots,a_H}\sum_{i=1}^{H} C(s_i, a_i)\right]$$



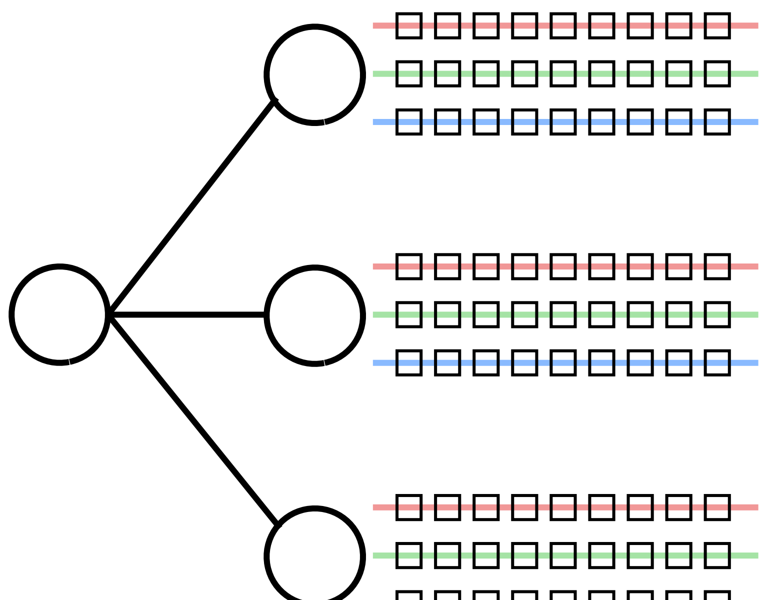
Hindsight optimization – solve a sample of future goals

$$\hat{V}_H(s_1) = \underset{s_2,\ldots,s_H}{E}\left[\min_{a_1,\ldots,a_H}\sum_{i=1}^{H} C(s_i,a_i)\right]$$



Hindsight optimization – solve a sample of future goals

$$\hat{V}_H(s_1) = \underset{s_2,\ldots,s_H}{E} \underbrace{\left[\min_{a_1,\ldots,a_H} \sum_{i=1}^{H} C(s_i, a_i) \right]}_{\text{Deterministic Planning}}$$



1. Sample future goal arrivals
2. For each action
   Evaluate mean plan cost over sampled futures
3. Take the best action
4. Repeat

Hindsight Optimization (continued)

1. Sample future goal arrivals
2. For each action

   Evaluate mean plan cost over sampled futures
3. Take the best action
4. Repeat

Simple, and uses knowledge about goal arrivals

Other Applications

HOP has been applied to:

- On-line scheduling (Chong, Givan, Chang 2000)
- Stochastic integer programs (Mercier and van Hentenrych 2007)
- Probabilistic planning (Yoon et al. 2008)

Now:

- On-line Planning

Experiments

How do we compare different techniques?

Goal: gain the most reward

How do we know our technique is doing well?

How do we compare different techniques?

Goal: gain the most reward

How do we know our technique is doing well?

- How does it compare to optimal?
  - Imagine we have an oracle that knows the future
- How does it compare to a simple planner?
  - Greedy: evaluate cost-to-go heuristic on each successor
  - Go to the state with the lowest heuristic value

How do we compare different techniques?

Goal: gain the most reward

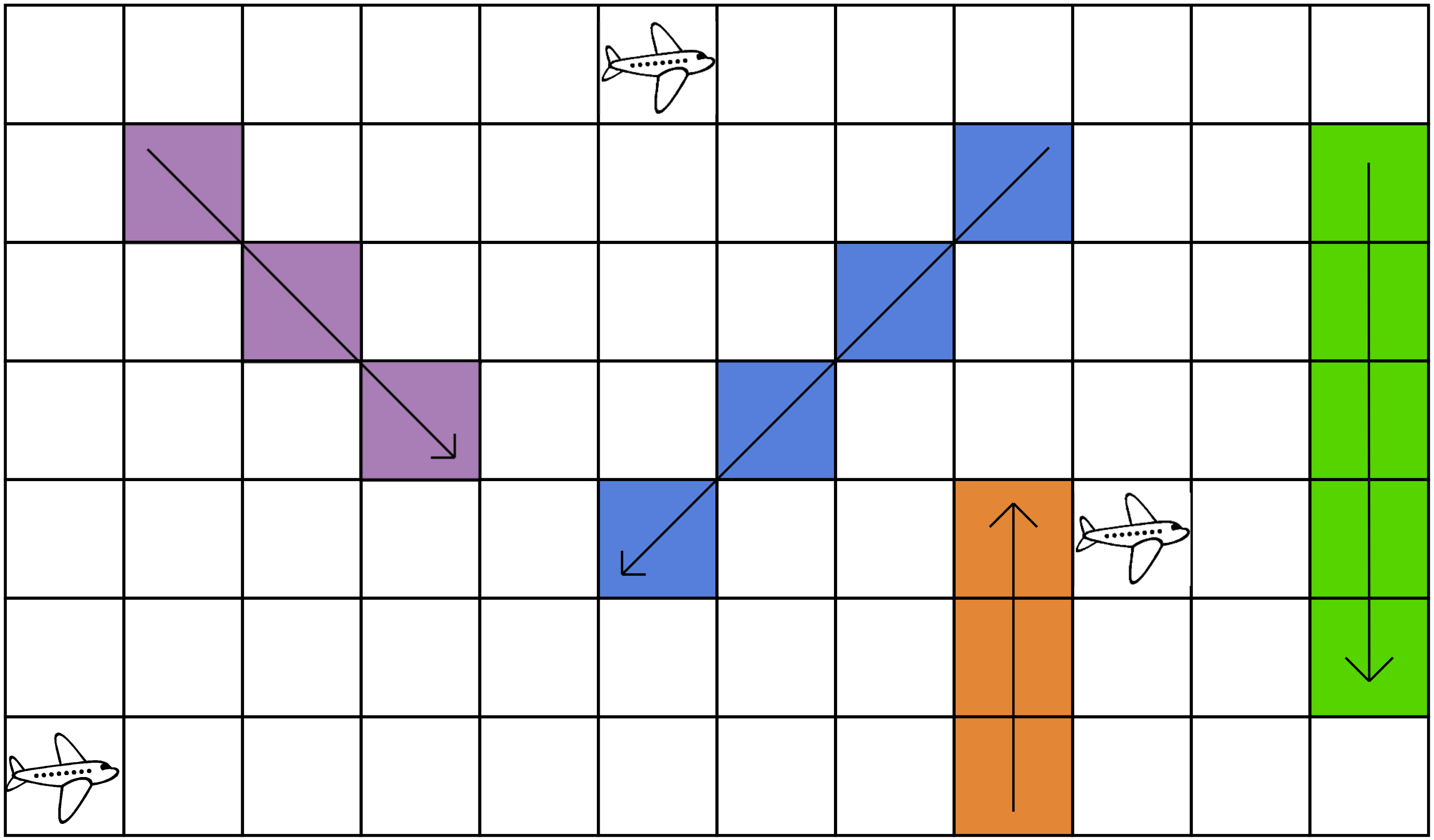
How do we know our technique is doing well?

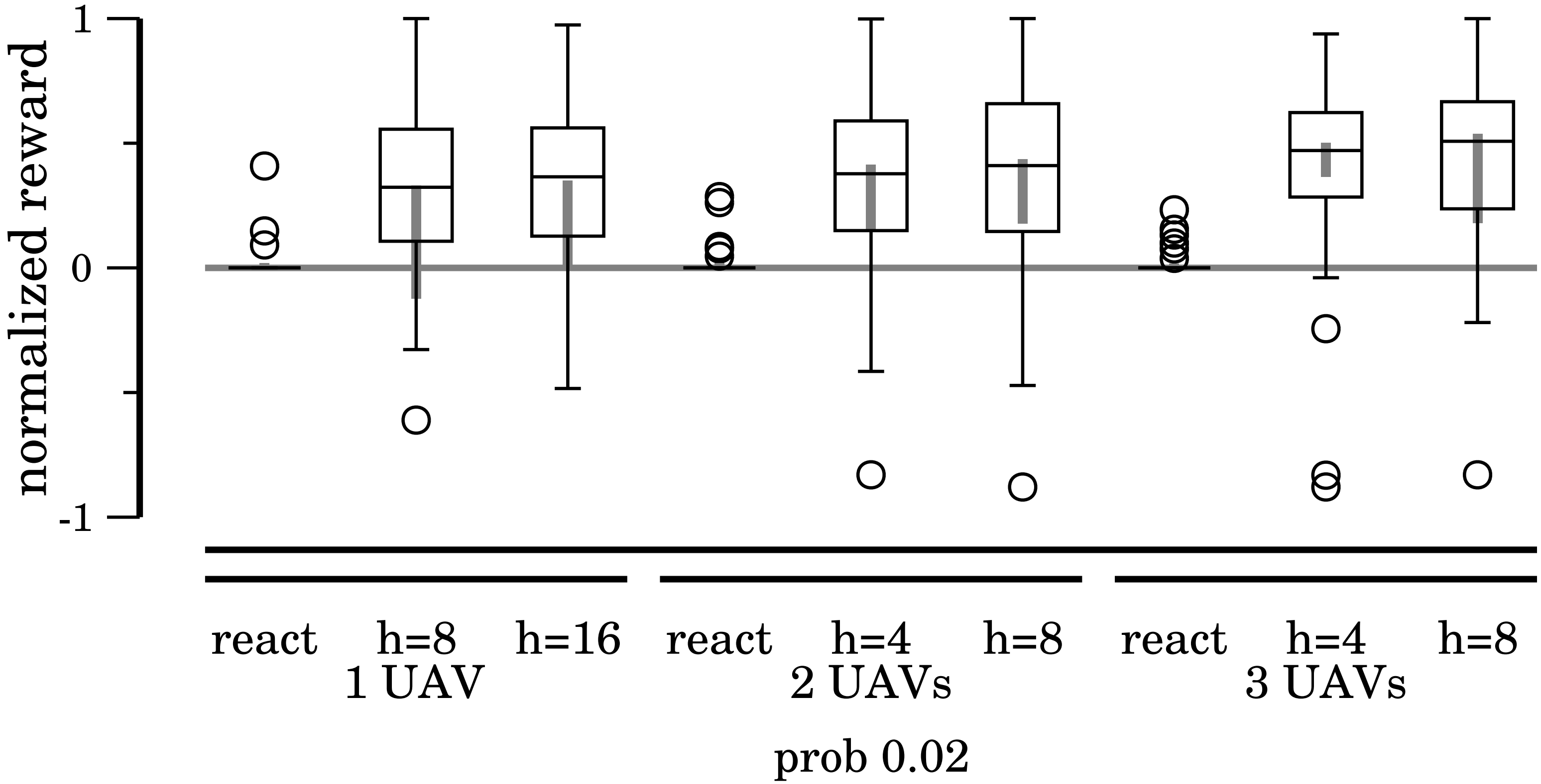
- How does it compare to optimal?
  - Imagine we have an oracle that knows the future
- How does it compare to a simple planner?
  - Greedy: evaluate cost-to-go heuristic on each successor
  - Go to the state with the lowest heuristic value

Normalized reward:

reward normalized between optimal (1) and greedy (0)

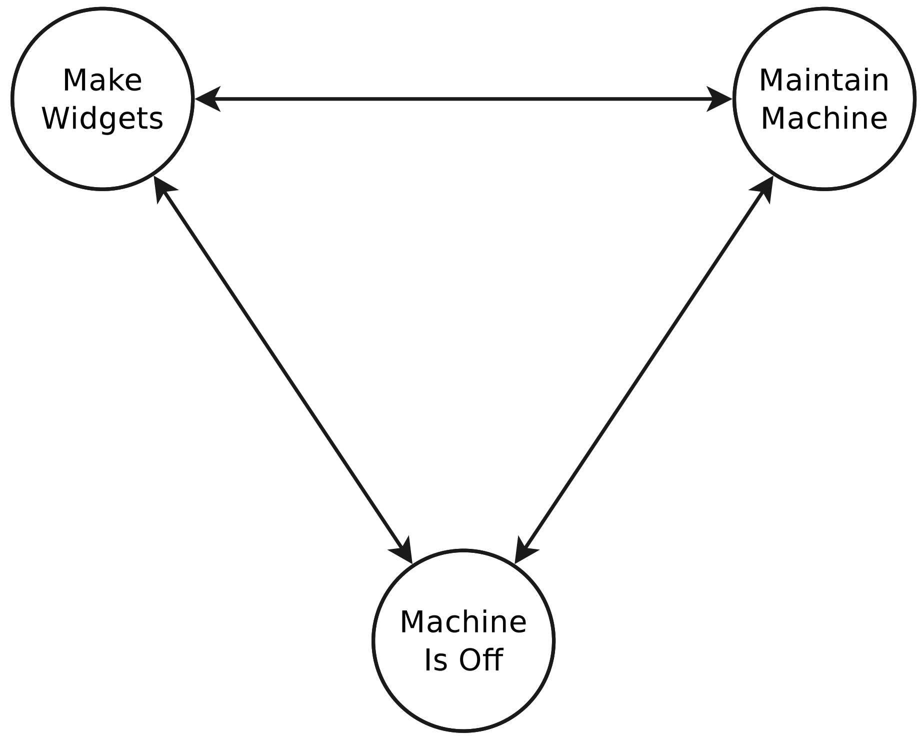
UAV Domain



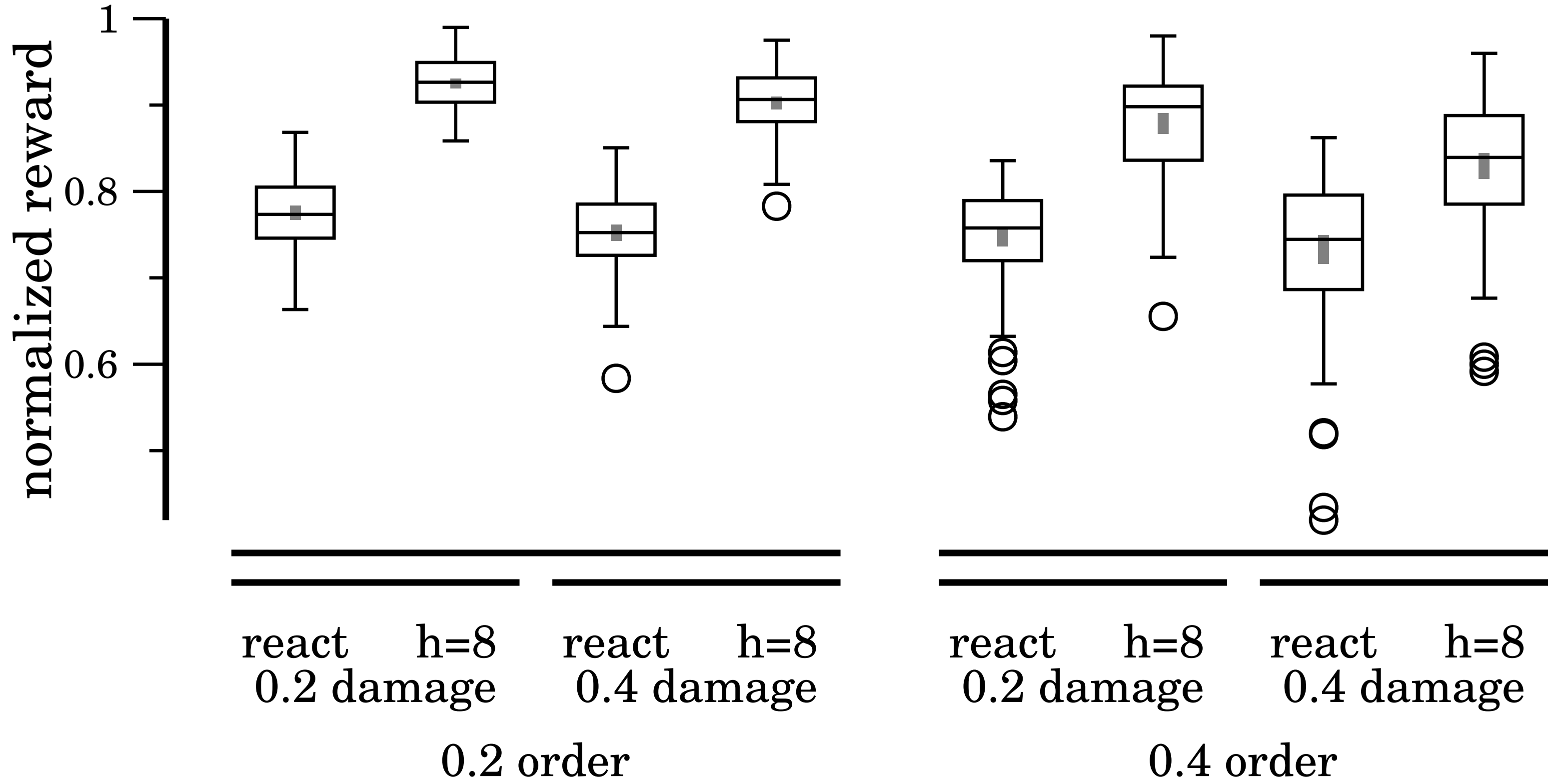


Reactive ∼ greedy, HOP gives the most reward

Manufacturing Domain



Manufacturing Results



HOP is close to optimal and gives the most reward

Pros and Cons

Pros:

- HOP simple to implement (just need deterministic planner)
- Better than the simple reactive approach
- Better than MDP solvers for these problems
  - 3x3 UAV: LRTDP 100x slower and worse than greedy

Cons:

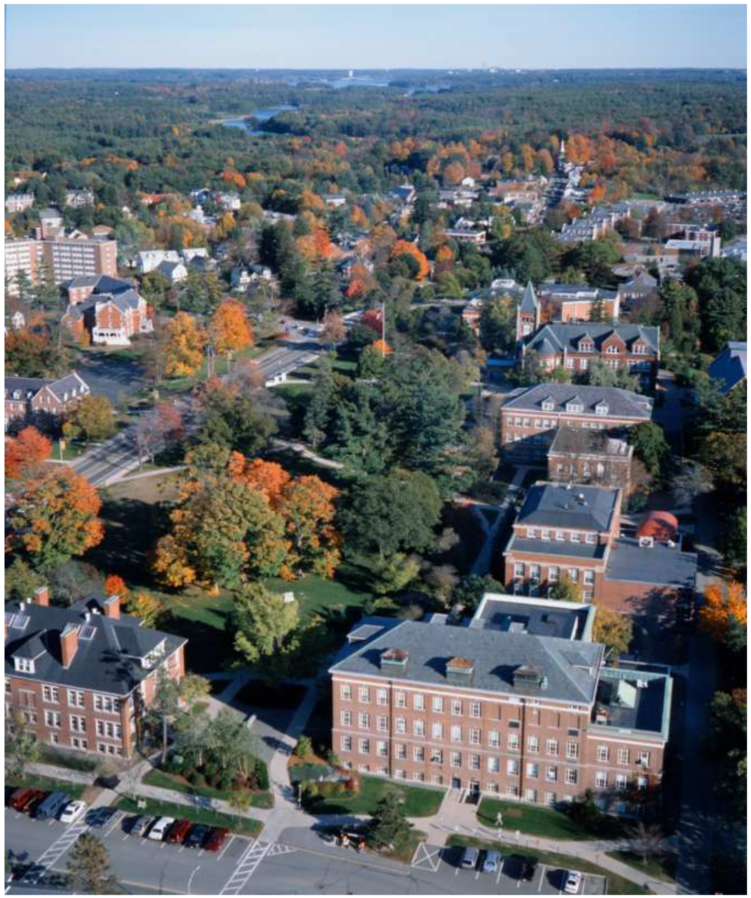
- HOP slower than reactive
  - Finds many plans instead of just one
  - Reactive: ~ 0 seconds, HOP: 0.002–10 seconds
  - But, see (Yoon et al. ICAPS 2010)

Conclusion

- Many problems are on-line continual planning problems
- We can take advantage of the known goal distribution
- Hindsight optimization is simple and works well

The University of New Hampshire

Tell your students to apply to grad school in CS at UNH!



- friendly faculty
- funding
- individual attention
- beautiful campus
- low cost of living
- easy access to Boston, White Mountains
- strong in AI, infoviz, networking, systems