Beyond Cost-to-go Estimates in Situated Temporal Planning

Andrew Coles,[1] Shahaf S. Shperberg,[2] Erez Karpas,[3]
Solomon Eyal Shimony,[2] Wheeler Ruml[4]

[1]King's College London, UK
[2]Ben-Gurion University, Israel
[3]Technion — Israel Institute of Technology, Israel
[4]University of New Hampshire, USA

Situated Temporal Planning

'planning while the clock ticks', 'time-aware planning'

Example: planning a route involving a bus ride

- 'take 10:00 bus' action expires at 10:00
  - subtree of plans becomes invalid
  - consider only if sufficient time to complete plan
- exploring 'take 9:47 bus' action can invalidate 10:00 action
  - searching under multiple nodes means less time for each
- plan expiration time uncertain until plan is complete
  - but completion effort and final feasibility also uncertain
- which plans to explore?

Situated Temporal Planning

'planning while the clock ticks', 'time-aware planning'

Example: planning a route involving a bus ride

- 'take 10:00 bus' action expires at 10:00
  - subtree of plans becomes invalid
  - consider only if sufficient time to complete plan
- exploring 'take 9:47 bus' action can invalidate 10:00 action
  - searching under multiple nodes means less time for each
- plan expiration time uncertain until plan is complete
  - but completion effort and final feasibility also uncertain
- which plans to explore?

Previous planner (ICAPS-18) just pruned.
Recent work (AAAI-19) showed problem is hard.

'planning while the clock ticks', 'time-aware planning'

Example: planning a route involving a bus ride

- 'take 10:00 bus' action expires at 10:00
  - subtree of plans becomes invalid
  - consider only if sufficient time to complete plan
- exploring 'take 9:47 bus' action can invalidate 10:00 action
  - searching under multiple nodes means less time for each
- plan expiration time uncertain until plan is complete
  - but completion effort and final feasibility also uncertain
- which plans to explore?

Previous planner (ICAPS-18) just pruned.
Recent work (AAAI-19) showed problem is hard.

We implement an approximate hack and find it can work.

Previous work: the Time-Predictive Planner (ICAPS-18)

based on OPTIC (Benton, Coles, and Coles, ICAPS-12)
uses STN to track time flexibly

- encodes external events as TILs
- constrains actions to happen after now
- prunes infeasible nodes
- estimates if plan can be completed in time (temporal RPG)
- two open lists, prefers complete-able

Previous work: the Time-Predictive Planner (ICAPS-18)

based on OPTIC (Benton, Coles, and Coles, ICAPS-12)
uses STN to track time flexibly

- encodes external events as TILs
- constrains actions to happen after now
- prunes infeasible nodes
- estimates if plan can be completed in time (temporal RPG)
- two open lists, prefers complete-able

better than OPTIC assuming a fixed planning time

but used usual cost-based search order!

Allocating Effort when Actions Expire (AE2, AAAI-19)

n partial plans/nodes/processes to share CPU, discrete time

For each process i, given

termination CDF $M_i(t)$ = probability i requires CPU time $\leq t$
like heuristic for effort required

success probability P_i = probability i results in solution
without considering time found

deadline CDF $D_i(t)$ = probability i expires before wall time t
not certain until solution is complete

Find schedule for processes that

- maximizes probability of finding a solution
- that is still valid when found

can be formulated as an MDP (see paper)

Analysis of the AE2 MDP (AAAI-19)

policy = time allocation = time-aware planning strategy

Theorem. *With known deadlines, there exists a linear contiguous policy that is an optimal solution.*

Theorem. *Finding the optimal (linear contiguous) policy for the case of known deadlines is NP-hard.*

Implies that solving the full AE2 MDP is NP-hard.

Theorem. *With known deadlines and diminishing logarithm of returns, optimal policy can be computed in polynomial time. (algorithm given)*

A Greedy Algorithm for AE2 (AAAI-19)

$$\begin{aligned} m_i(t) &= \text{probability } i \text{ completes after } t \text{ units of computation} \\ &= M_i(t) - M_i(t-1) \\ f_i(t) &= \text{probability } i \text{ succeeds after } t \text{ units of computation} \\ &= P_i \sum_{t'=0}^{t} m_i(t')(1 - D_i(t')) \\ e_i &= \text{`most effective' computation time for } i \\ &= \operatorname*{argmin}_t \frac{\log(1 - f_i(t))}{t} \end{aligned}$$

Greedy algorithm: prioritize soonest deadline and greatest improvement per unit computation

$$\text{maximize } Q(i) = \frac{\alpha}{E[D_i]} - \frac{\log(1 - f_i(e_i))}{e_i}$$

tested on standalone AE2 problems

New Work

A Modified Greedy Algorithm for Use in Planning

original greedy: maximize $Q(i) = \frac{\alpha}{E[D_i]} - \frac{\log(1 - f_i(e_i))}{e_i}$

but don't have M, P, and D distributions for f_i and e_i
new modified approach:

- estimate $E[D_i]$ using slack in temporal RPG
  - time before current plan + relaxed plan must start
- approximate e_i with estimated remaining search time under i
  - estimated search distance times expansion delay
- replace $-\log(1 - f_i(e_i))$ with $E[D_i] - e_i$
  - slack beyond expected planning time

New greedy algorithm: prioritize soonest deadline and greatest planning slack

new: maximize $\hat{Q}(i) = \frac{\alpha}{\max(E[D_i], t_{10})} + \frac{\max(0, E[D_i] - e_i)}{e_i}$

Experimental Results

42 Robocup Logistics League problems
Time-Predictive Planner (ICAPS-18) with different search orders

		$\hat{Q}(n)$ with $\alpha =$					
	$h(n)$	-10^4	-1	0	0.1	1	10^4
number solved	21	27	29	29	29	**30**	**30**

most failures were missed deadlines

Conclusion

Summary

Planning while time passes is extra hard!

- just formalizing the problem is non-trivial
- metareasoning must be cheap

A greedy approach can perform well!

- even if highly simplified and approximated
- for problems with deadlines, searching on time beats cost!

Further directions

- more benchmarks
- consider solution cost


UNIVERSITY of NEW HAMPSHIRE
©2011 Mike Ross, UNH Photographic Services

Backup Slides

Solving the AE2 MDP

State space exponential in n.

Solving the AE2 MDP

State space exponential in n.

Restricted cases:

1. Linear policies (no feedback): (1, 1, 2, 1, 1, 3, ...)
2. Linear contiguous policies: (1, 1, 1, 2, 2, 3, 3, 3, ...)
3. Known deadlines

Solving the AE2 MDP

State space exponential in n.

Restricted cases:

1. Linear policies (no feedback): (1, 1, 2, 1, 1, 3, ...)
2. Linear contiguous policies: (1, 1, 1, 2, 2, 3, 3, 3, ...)
3. Known deadlines

Good news:

Theorem. *With known deadlines, there exists a linear contiguous policy that is an optimal solution.*

Solving the AE2 MDP

State space exponential in n.

Restricted cases:

1. Linear policies (no feedback): (1, 1, 2, 1, 1, 3, ...)
2. Linear contiguous policies: (1, 1, 1, 2, 2, 3, 3, 3, ...)
3. Known deadlines

Good news:

Theorem. *With known deadlines, there exists a linear contiguous policy that is an optimal solution.*

Bad news:

Theorem. *Finding the optimal (linear contiguous) policy for the case of known deadlines is NP-hard.*

Implies that solving the full AE2 MDP is NP-hard.

Solving the AE2 MDP

State space exponential in n.

Restricted cases:

1. Linear policies (no feedback): (1, 1, 2, 1, 1, 3, ...)
2. Linear contiguous policies: (1, 1, 1, 2, 2, 3, 3, 3, ...)
3. Known deadlines

Good news:

Theorem. *With known deadlines, there exists a linear contiguous policy that is an optimal solution.*

Bad news:

Theorem. *Finding the optimal (linear contiguous) policy for the case of known deadlines is NP-hard.*

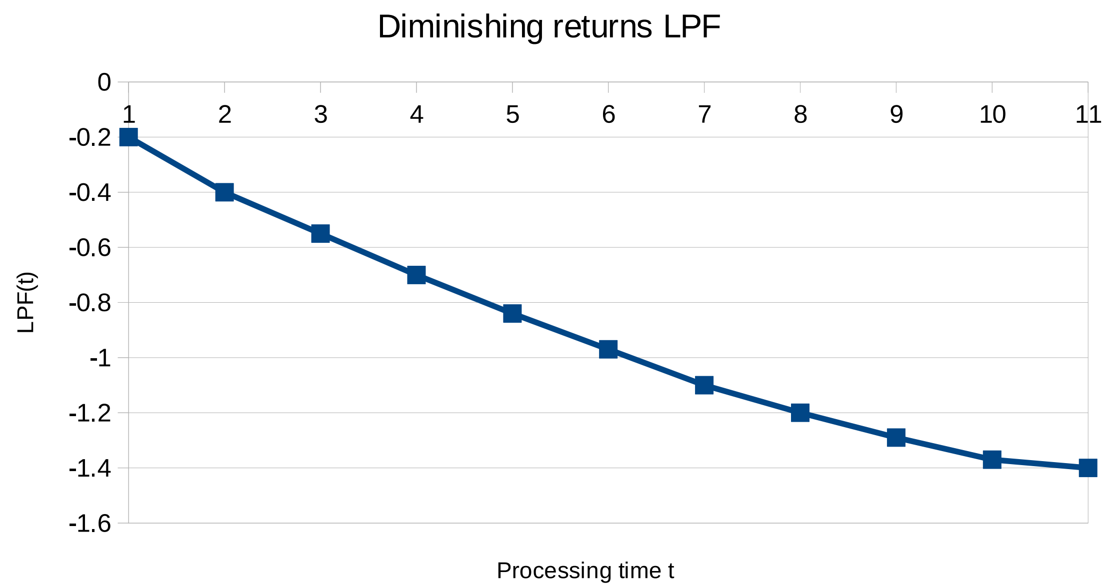
Implies that solving the full AE2 MDP is NP-hard.

However...

Diminishing Returns

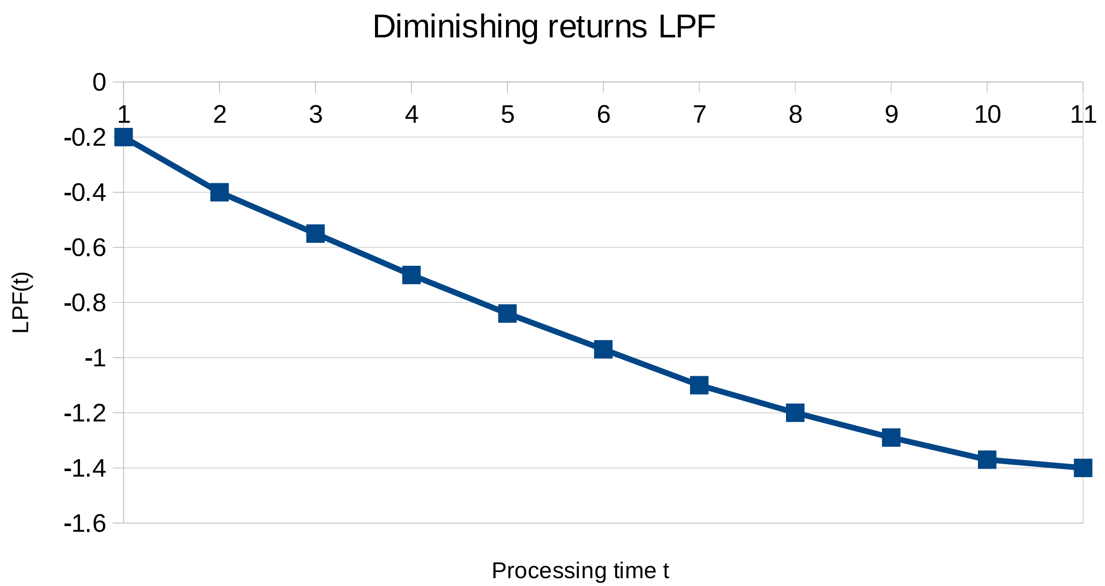
log probability i still running: $LPR_i(t)$

diminishing returns: $\frac{d(LPR_i(t))}{dt}$ is non-decreasing (B&D, 1994)



log probability i still running: $LPR_i(t)$

diminishing returns: $\frac{d(LPR_i(t))}{dt}$ is non-decreasing (B&D, 1994)



Good news:

Theorem. *With known deadlines and diminishing logarithm of returns, optimal policy can be computed in polynomial time.*

Four Types of Algorithms

Optimal: solve MDP directly

Simple Heuristics: run ‘most promising’ until failure; round robin; random

DiminishingReturns: optimal for DR

Greedy: inspired by DR, basically at each step select most likely to succeed

metric: probability a non-expired solution is found

Experimental Set-up

synthetic $M_i(t), P_i, D_i(t)$

- distributions: exponential (diminishing returns!), normal, uniform
- tried range of parameters

temporal planning problems

- OPTIC planner (as in ICAPS-18) on Robocup Logistics League
- search trees used to generate snapshots

known and unknown deadlines

Results with Known Deadlines

dist	n	Greedy	DR	MP
B	2	0.61	0.67	**0.70**
	5	0.72	**0.82**	0.61
	10	0.60	**0.88**	0.71
	100	0.81	**0.99**	0.91
N	2	**0.56**	0.45	0.33
	5	**0.83**	0.72	0.27
	10	**0.93**	0.41	0.09
	100	**1.00**	0.70	0.23
U	2	0.61	**0.65**	0.50
	5	**0.90**	0.88	0.75
	10	**0.98**	**0.98**	0.66
	100	**1.00**	**1.00**	0.80
P	2	0.72	**0.79**	0.01
	5	0.78	**0.81**	0.79
	10	**1.00**	0.87	0.99
	100	**1.00**	0.91	0.86
avg		**0.82**	0.78	0.58

simple 'Most Promising' not so good

Results with Known Deadlines

dist	n	Greedy	DR	MP
B	2	0.61	0.67	**0.70**
	5	0.72	**0.82**	0.61
	10	0.60	**0.88**	0.71
	100	0.81	**0.99**	0.91
N	2	**0.56**	0.45	0.33
	5	**0.83**	0.72	0.27
	10	**0.93**	0.41	0.09
	100	**1.00**	0.70	0.23
U	2	0.61	**0.65**	0.50
	5	**0.90**	0.88	0.75
	10	**0.98**	**0.98**	0.66
	100	**1.00**	**1.00**	0.80
P	2	0.72	**0.79**	0.01
	5	0.78	**0.81**	0.79
	10	**1.00**	0.87	0.99
	100	**1.00**	0.91	0.86
avg		**0.82**	0.78	0.58

simple 'Most Promising' not so good

DR optimal for DR, okay with known deadline

Results with Known Deadlines

dist	n	Greedy	DR	MP
B	2	0.61	0.67	**0.70**
	5	0.72	**0.82**	0.61
	10	0.60	**0.88**	0.71
	100	0.81	**0.99**	0.91
N	2	**0.56**	0.45	0.33
	5	**0.83**	0.72	0.27
	10	**0.93**	0.41	0.09
	100	**1.00**	0.70	0.23
U	2	0.61	**0.65**	0.50
	5	**0.90**	0.88	0.75
	10	**0.98**	**0.98**	0.66
	100	**1.00**	**1.00**	0.80
P	2	0.72	**0.79**	0.01
	5	0.78	**0.81**	0.79
	10	**1.00**	0.87	0.99
	100	**1.00**	0.91	0.86
	avg	**0.82**	0.78	0.58

simple 'Most Promising' not so good
DR optimal for DR, okay with known deadline
Greedy quite respectable

Results with Unknown Deadlines

dist	n	Greedy	DR	MP
B	2	0.61	0.35	**0.64**
	5	**0.65**	0.36	0.63
	10	**0.70**	0.45	0.66
	100	**0.70**	0.44	0.65
N	2	**0.63**	0.37	0.20
	5	**0.70**	0.35	0.09
	10	**0.65**	0.30	0.15
	100	**0.76**	0.32	0.06
U	2	**0.68**	0.39	0.53
	5	**0.70**	0.43	0.57
	10	**0.78**	0.46	0.59
	100	**0.86**	0.52	0.59
P	2	**0.61**	0.24	0.46
	5	**0.90**	0.54	0.45
	10	**0.90**	0.32	0.62
	100	**0.85**	0.77	0.38
avg		**0.73**	0.41	0.45

DR poor for unknown deadlines

Results with Unknown Deadlines

dist	n	Greedy	DR	MP
B	2	0.61	0.35	**0.64**
	5	**0.65**	0.36	0.63
	10	**0.70**	0.45	0.66
	100	**0.70**	0.44	0.65
N	2	**0.63**	0.37	0.20
	5	**0.70**	0.35	0.09
	10	**0.65**	0.30	0.15
	100	**0.76**	0.32	0.06
U	2	**0.68**	0.39	0.53
	5	**0.70**	0.43	0.57
	10	**0.78**	0.46	0.59
	100	**0.86**	0.52	0.59
P	2	**0.61**	0.24	0.46
	5	**0.90**	0.54	0.45
	10	**0.90**	0.32	0.62
	100	**0.85**	0.77	0.38
avg		**0.73**	0.41	0.45

DR poor for unknown deadlines
Greedy still respectable