Allocating Planning Effort when Actions Expire

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

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

Planning While the Clock Ticks

'situated temporal planning', '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 also uncertain
- which plans to explore?

Planning While the Clock Ticks

'situated temporal planning', '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 also uncertain
- which plans to explore?

We formalize and analyze this problem.

Allocating Effort when Actions Expire (AE2)

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

Analysis

The AE2 MDP

AE2 as MDP, policy = time-aware planning strategy

States: $\langle T, T_1, \ldots, T_n \rangle$ where
- T is wall clock time
- T_i is time allocated so far to process i ($\perp$ = failed)
- terminal states: SUCCESS and FAIL

Reward: 1 in SUCCESS, 0 elsewhere

Actions: a_i allocates one time unit to i

Transitions: derive probabilities from $M_i(T_i), P_i, D_i(T)$
- increment T and T_i unless terminated,
- if failed, $T_i = \perp$ and FAIL if all $\perp$.
- otherwise, SUCCESS.

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

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.*

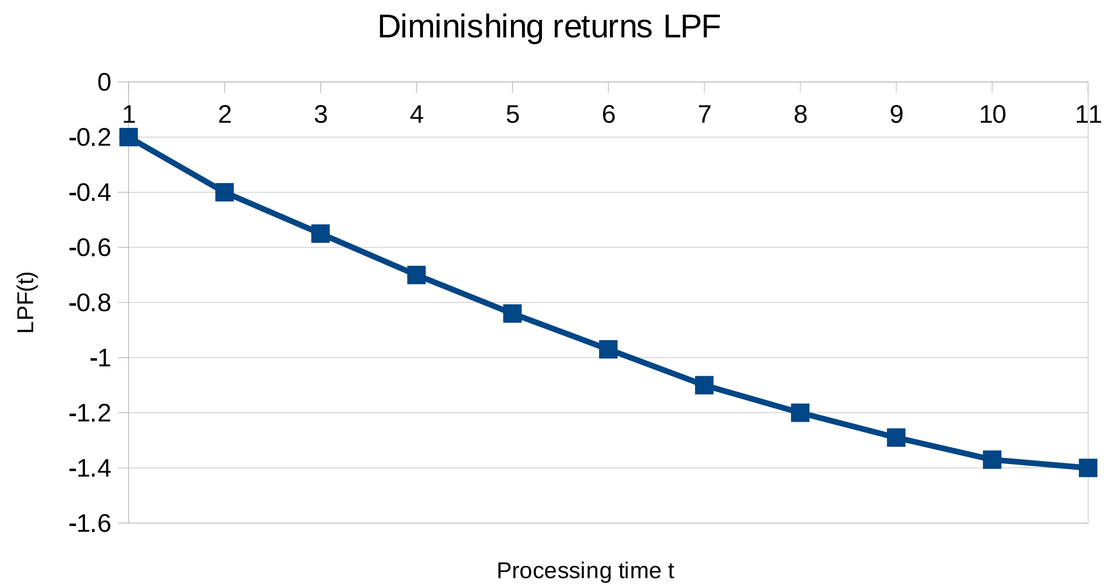
Implies that solving the full AE2 MDP is NP-hard.

However...

Diminishing Returns

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

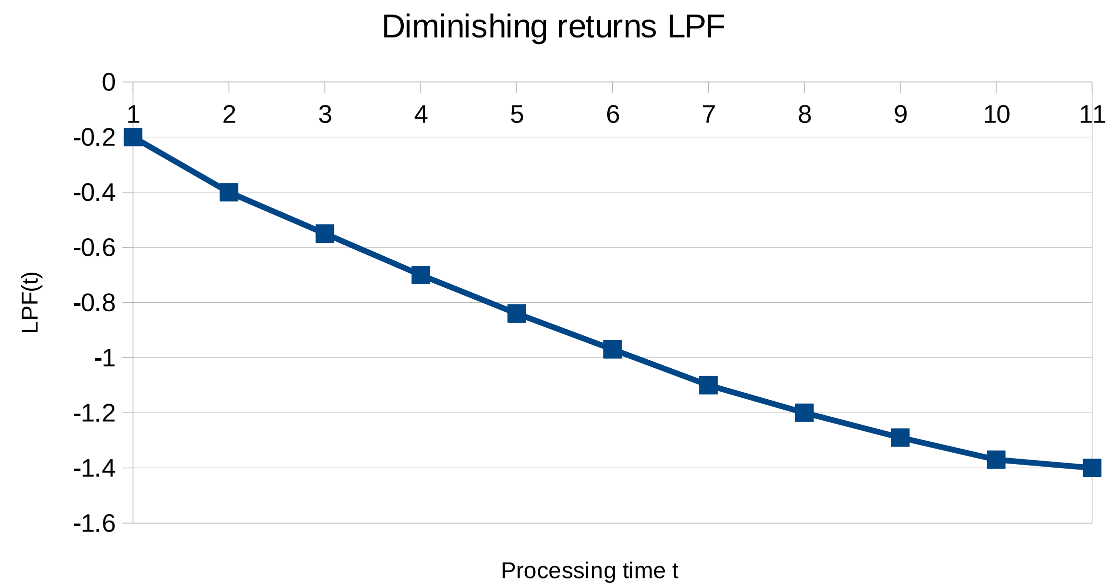
diminishing returns: $\frac{d(LPR_i(t))}{dt}$ is non-decreasing (B&D, 1994)



Diminishing Returns

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.*

Algorithms

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

Conclusion

Summary

Planning while time passes is extra hard!

- benefits from deliberation scheduling
- AE2 captures the most basic form of the problem
- NP-hard to solve except in restricted cases

A greedy approach can perform well

- both random problems and planner search trees
- reasonable runtime

Further directions

- integrate into a planner
- solution cost