Solving Time-dependent Markov Decision Processes

Emmanuel Rachelson
Intelligent Systems Lab
Technical Univ. of Crete

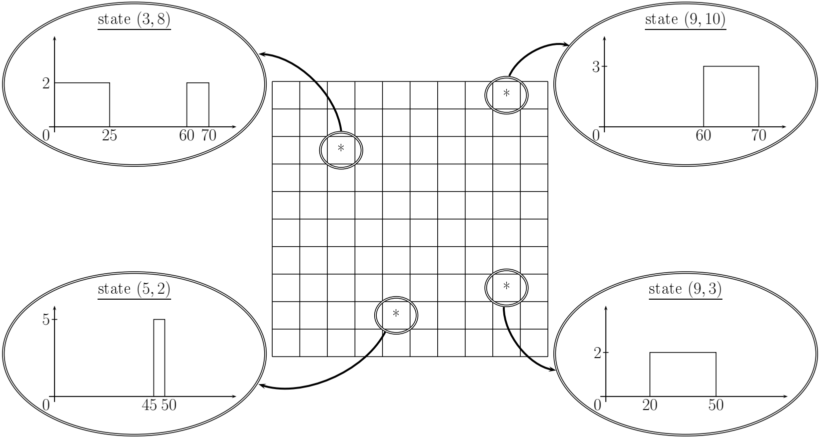
Patrick Fabiani
ONERA Toulouse

Frederick Garcia
INRA Toulouse

September 19th, 2009

UAV patrol mission

UAV patrol mission



Outline

1. **Time-dependent MDPs**

2. **Value iteration in practice:** $TiMDP_{poly}$

3. **Experiments**

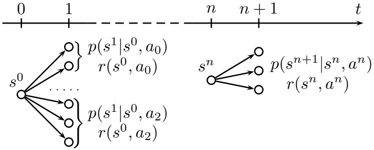
Modeling background

Sequential decision under probabilistic uncertainty:

Markov Decision Process

Tuple $\langle S, A, p, r, T \rangle$
Markovian transition model $p(s'|s, a)$
Reward model $r(s, a)$
T is a set of timed decision epochs $\{0, 1, \ldots, H\}$

Infinite (unbounded) horizon: $H \longrightarrow \infty$



Optimal policies for MDPs

Value of a sequence of actions

$$\forall (a_n) \in A^{\mathbb{N}}, V^{(a_n)}(s) = E\left(\sum_{\delta=0}^{\infty} \gamma^{\delta} r(s^{\delta}, a_{\delta}) \right)$$

Stationary, deterministic, Markovian policy

$$\mathscr{D} = \left\{ \pi \ : \left\{ \begin{array}{ccc} S & \rightarrow & A \\ s & \mapsto & \pi(s) = a \end{array} \right\} \right.$$

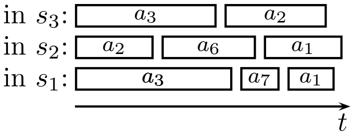
Optimality equation

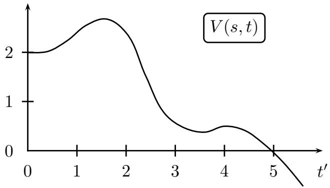
$$V^*(s) = \max_{\pi \in \mathscr{D}} V^{\pi}(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V^*(s') \right\}$$

What are we looking for?

One way of considering the UAV patrol problem consists in saying that we search for

policies and value functions which depend on time.

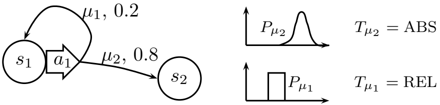




Time-dependent MDPs

Definition (TiMDP, [Boyan and Littman, 2001])

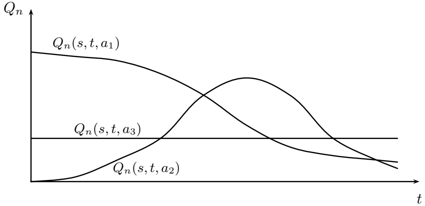
Tuple $\langle S, A, M, L, R, K \rangle$
M Set of outcomes $\mu = (s'_\mu, T_\mu, P_\mu)$
$L(\mu|s,t,a)$ Probability of triggering outcome μ
$R(\mu,t,t') = r_{\mu,t}(t) + r_{\mu,\tau}(t'-t) + r_{\mu,t'}(t')$



Boyan, J. A. and Littman, M. L. (2001).
Exact Solutions to Time Dependent MDPs.
Advances in Neural Information Processing Systems, 13:1026–1032.

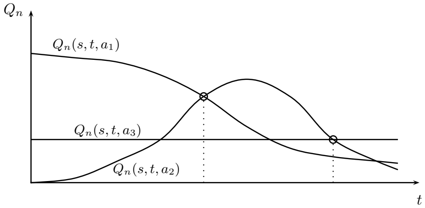
TiMDP dynamic programming equation

$$
\begin{aligned}
Q(s,t,a) &= \sum_{\mu \in M} L(\mu | s,t,a) \cdot U(\mu,t) \\
U(\mu,t) &= \begin{cases} \int_{-\infty}^{\infty} P_\mu(t')[R(\mu,t,t') + V(s'_\mu,t')]dt' & \text{if } T_\mu = \text{ABS} \\ \int_{-\infty}^{\infty} P_\mu(t'-t)[R(\mu,t,t') + V(s'_\mu,t')]dt' & \text{if } T_\mu = \text{REL} \end{cases}
\end{aligned}
$$



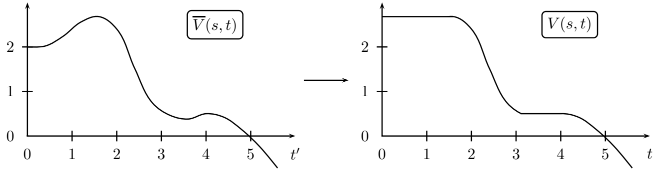
TiMDP dynamic programming equation

$$
\begin{aligned}
\overline{V}(s,t) &= \max_{a \in A} Q(s,t,a) \\
Q(s,t,a) &= \sum_{\mu \in M} L(\mu | s,t,a) \cdot U(\mu,t) \\
U(\mu,t) &= \begin{cases} \int_{-\infty}^{\infty} P_\mu(t')[R(\mu,t,t') + V(s'_\mu,t')]dt' & \text{if } T_\mu = \text{ABS} \\ \int_{-\infty}^{\infty} P_\mu(t'-t)[R(\mu,t,t') + V(s'_\mu,t')]dt' & \text{if } T_\mu = \text{REL} \end{cases}
\end{aligned}
$$



TiMDP dynamic programming equation

$$
\begin{aligned}
V(s,t) &= \sup_{t' \geq t} \left(\int_t^{t'} K(s,\theta) d\theta + \overline{V}(s,t') \right) \\
\overline{V}(s,t) &= \max_{a \in A} Q(s,t,a) \\
Q(s,t,a) &= \sum_{\mu \in M} L(\mu | s,t,a) \cdot U(\mu,t) \\
U(\mu,t) &= \begin{cases} \int_{-\infty}^{\infty} P_\mu(t')[R(\mu,t,t') + V(s'_\mu,t')]dt' & \text{if } T_\mu = \text{ABS} \\ \int_{-\infty}^{\infty} P_\mu(t'-t)[R(\mu,t,t') + V(s'_\mu,t')]dt' & \text{if } T_\mu = \text{REL} \end{cases}
\end{aligned}
$$



Optimality equation?

Is this DP equation an optimality equation for TiMDPs?
If yes, corresponding to which criterion?

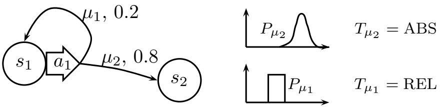
Rachelson, E., Garcia, F., and Fabiani, P. (2008).
Extending the Bellman Equation for MDP to Continuous Actions and Continuous Time in the Discounted Case.
In *International Symposium on Artificial Intelligence and Mathematics*.

Yes,
with as total reward criterion
and specific hypotheses on the transition and reward models.

Value Iteration for TiMDPs

Solving TiMDPs $\leftrightarrow$ solving the optimality equation.

Solving TiMDPs

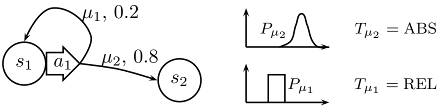


Value iteration Bellman backups for TiMDPs can be performed exactly if:

- $L(\mu|s,t,a)$ piecewise constant
- $R(\mu,t,t') = r_{\mu,t}(t) + r_{\mu,\tau}(t'-t) + r_{\mu,t'}(t')$
- $r_{\mu,t}(t), r_{\mu,\tau}(\tau), r_{\mu,t'}(t')$ piecewise linear
- $P_\mu(t')$, $P_\mu(t'-t)$ discrete distributions

Then $V^*(s,t)$ is piecewise linear.

Solving TiMDPs



- What about other, more expressive functions?
- How does this theoretical result scale to practical resolution?

Extending exact resolution

Piecewise polynomial (PWP) models: L, P_μ, $r_i \in \mathscr{P}_n$.

Degree evolution

$$\left.\begin{array}{l} P_\mu \in \mathscr{D}\mathscr{P}_A \\ r_i, V_0 \in \mathscr{P}_B \\ L \in \mathscr{P}_C \end{array}\right\} \Rightarrow d^\circ(V_n) = B + n(A + C + 1)$$

Extending exact resolution

Piecewise polynomial (PWP) models: L, P_μ, $r_i \in \mathscr{P}_n$.

Degree evolution

$$\left.\begin{array}{l} P_\mu \in \mathscr{D}\mathscr{P}_A \\ r_i, V_0 \in \mathscr{P}_B \\ L \in \mathscr{P}_C \end{array}\right\} \Rightarrow d^\circ(V_n) = B + n(A + C + 1)$$

Stability $\Leftrightarrow A + C = -1$.

Extending exact resolution

Piecewise polynomial (PWP) models: L, P_μ, $r_i \in \mathscr{P}_n$.

Degree evolution

$$\left.\begin{array}{l} P_\mu \in \mathscr{D}\mathscr{P}_A \\ r_i, V_0 \in \mathscr{P}_B \\ L \in \mathscr{P}_C \end{array}\right\} \Rightarrow d^\circ(V_n) = B + n(A + C + 1)$$

$$\text{Stability} \Leftrightarrow A + C = -1.$$

Exact resolution conditions

Degree stability + exact analytical computations: $\left\{\begin{array}{l} P_\mu \in \mathscr{D}\mathscr{P}_{-1} \\ r_i \in \mathscr{P}_4 \\ L \in \mathscr{P}_0 \end{array}\right.$

Extending exact resolution

Piecewise polynomial (PWP) models: L, P_μ, $r_i \in \mathscr{P}_n$.

Degree evolution

$$\left.\begin{array}{l} P_\mu \in \mathscr{D}\mathscr{P}_A \\ r_i, V_0 \in \mathscr{P}_B \\ L \in \mathscr{P}_C \end{array}\right\} \Rightarrow d^\circ(V_n) = B + n(A + C + 1)$$

Stability $\Leftrightarrow A + C = -1$.

Exact resolution conditions

Degree stability + exact analytical computations: $\left\{\begin{array}{l} P_\mu \in \mathscr{D}\mathscr{P}_{-1} \\ r_i \in \mathscr{P}_4 \\ L \in \mathscr{P}_0 \end{array}\right.$

If $B > 4$: approximate root finding.

Extending exact resolution

Piecewise polynomial (PWP) models: L, P_μ, $r_i \in \mathscr{P}_n$.

Degree evolution

$$\left.\begin{array}{l} P_\mu \in \mathscr{D}\mathscr{P}_A \\ r_i, V_0 \in \mathscr{P}_B \\ L \in \mathscr{P}_C \end{array}\right\} \Rightarrow d^\circ(V_n) = B + n(A + C + 1)$$

Stability $\Leftrightarrow A + C = -1$.

Exact resolution conditions

Degree stability + exact analytical computations: $\left\{\begin{array}{l} P_\mu \in \mathscr{D}\mathscr{P}_{-1} \\ r_i \in \mathscr{P}_4 \\ L \in \mathscr{P}_0 \end{array}\right.$

If $A + C > 0$: projection scheme of V_n on $\mathscr{P}_B$.

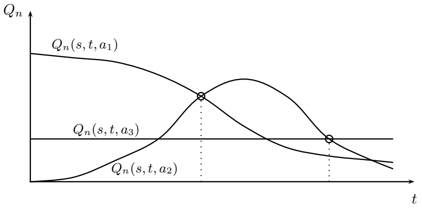
And in practice?

Experimental result

The number of definition intervals in V_n grows with n and does not necessarily converge.

$\Rightarrow$ numerical problems occur before $\|V_n - V_{n-1}\| < \varepsilon$.

e.g. $\overline{V}$ calculation:



And in practice?

Experimental result

The number of definition intervals in V_n grows with n and does not necessarily converge.

$\Rightarrow$ numerical problems occur before $\|V_n - V_{n-1}\| < \varepsilon$.

$\longrightarrow$ general case: approximate resolution by piecewise polynomial **interval simplification** for the value function.

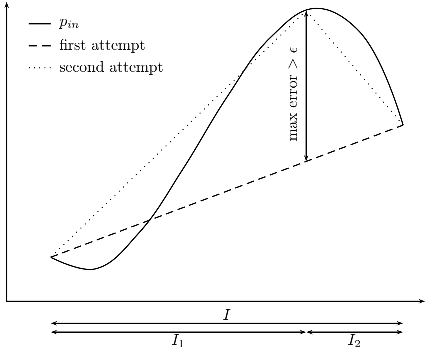
$$\text{Approximation} \begin{array}{l} \nearrow \quad \text{degree reduction} \\ \searrow \quad \text{interval simplification} \end{array}$$

TiMDP$_{poly}$: Approximate Value Iteration on TiMDPs

TiMDP$_{poly}$ polynomial approximation

$p_{out} = \mathrm{poly\_approx}(p_{in}, [l, u], \varepsilon, B)$

Two phases: incremental refinement and simplification.

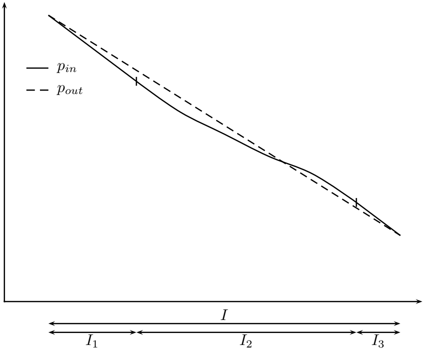


TiMDP$_{poly}$: Approximate Value Iteration on TiMDPs

TiMDP$_{poly}$ polynomial approximation

$p_{out} = \text{poly\_approx}(p_{in}, [l, u], \varepsilon, B)$

Two phases: incremental refinement and simplification.



TiMDP$_{poly}$: Approximate Value Iteration on TiMDPs

TiMDP$_{poly}$ polynomial approximation

$p_{out} = \text{poly\_approx}(p_{in}, [l, u], \varepsilon, B)$
Two phases: incremental refinement and simplification.

Properties

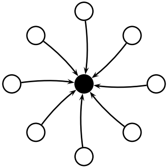
- $p_{out} \in \mathscr{P}_B$
- $\|p_{in} - p_{out}\|_\infty \leq \varepsilon$
- suboptimal number of intervals
- good complexity compromise

TiMDP$_{poly}$: Approximate Value Iteration on TiMDPs

Prioritized Sweeping.

Leveraging the computational effort by
ordering Bellman backups

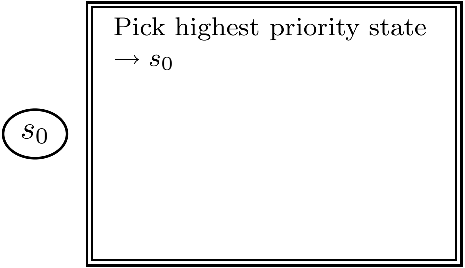
Perform Bellman backups in states with the largest value function change.



Moore, A. W. and Atkeson, C. G. (1993).
Prioritized Sweeping: Reinforcement Learning with Less Data and Less Real Time.
Machine Learning Journal, 13(1):103–105.

TiMDP$_{poly}$: Approximate Value Iteration on TiMDPs

Adapting Prioritized Sweeping to TiMDPs.



TiMDP$_{poly}$: Approximate Value Iteration on TiMDPs

Adapting Prioritized Sweeping to TiMDPs.

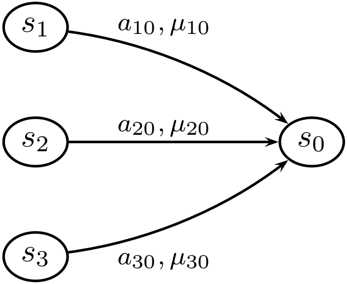
`update` $\overline{V}(s_0, t)$
`update` $V(s_0, t)$
poly_approx $(V(s_0, t))$

s_0

Pick highest priority state
$\rightarrow s_0$
Bellman backup
$\rightarrow V(s_0, t)$

TiMDP$_{poly}$: Approximate Value Iteration on TiMDPs

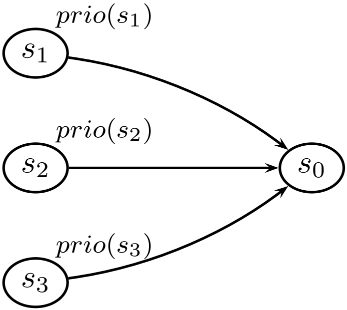
Adapting Prioritized Sweeping to TiMDPs.



Pick highest priority state
$\rightarrow s_0$
Bellman backup
$\rightarrow V(s_0, t)$
Update Q values
$\rightarrow Q(s, t, a)$

TiMDP$_{poly}$: Approximate Value Iteration on TiMDPs

Adapting Prioritized Sweeping to TiMDPs.



Pick highest priority state
$\rightarrow s_0$
Bellman backup
$\rightarrow V(s_0, t)$
Update Q values
$\rightarrow Q(s, t, a)$
Update priorities
$\rightarrow prio(s) = \|Q - Q_{old}\|_\infty$

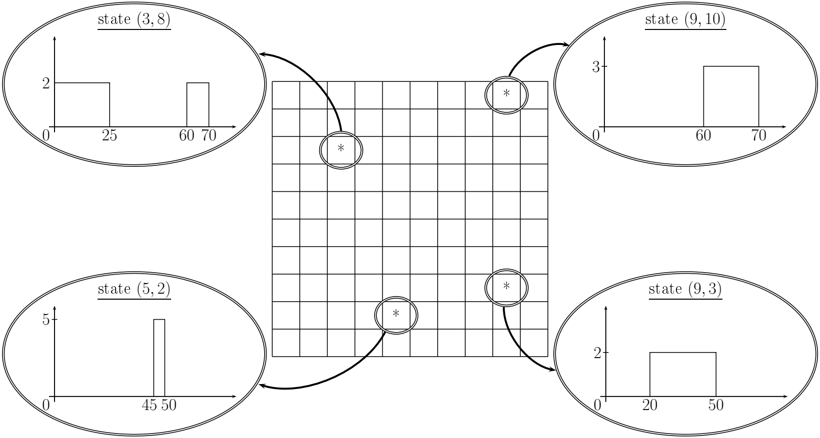
TiMDP$_{poly}$

TiMDP$_{poly}$ in a nutshell

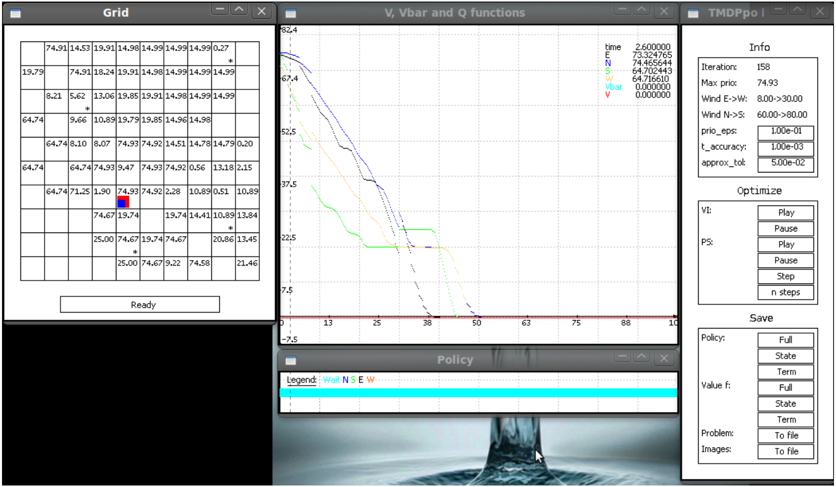
$$TiMDP_{poly}: \left\{ \begin{array}{l} \text{Analytical polynomial calculations} \\ L_\infty\text{-bounded error projection} \\ \text{Prioritized Sweeping for TiMDPs} \end{array} \right.$$

- Analytical operations: option for representing continuous quantities.
- Approximation makes resolution possible.
- Asynchronous VI makes it faster.

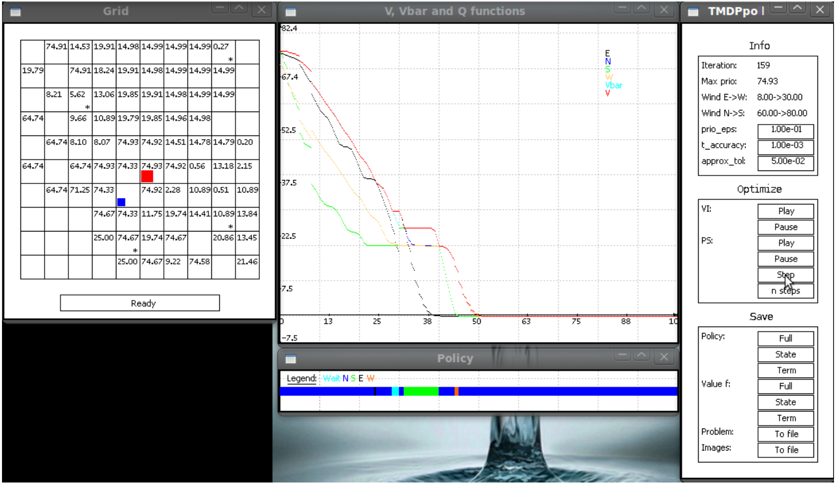
The UAV patrol problem



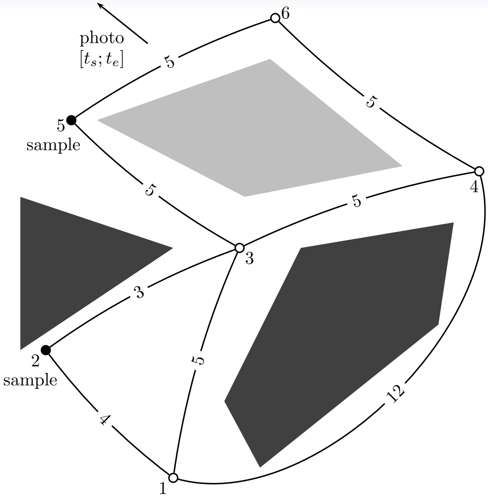
The UAV patrol problem



The UAV patrol problem

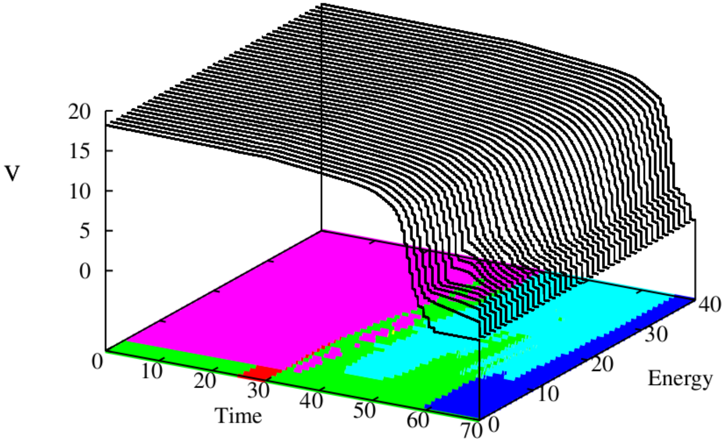


A Mars rover problem



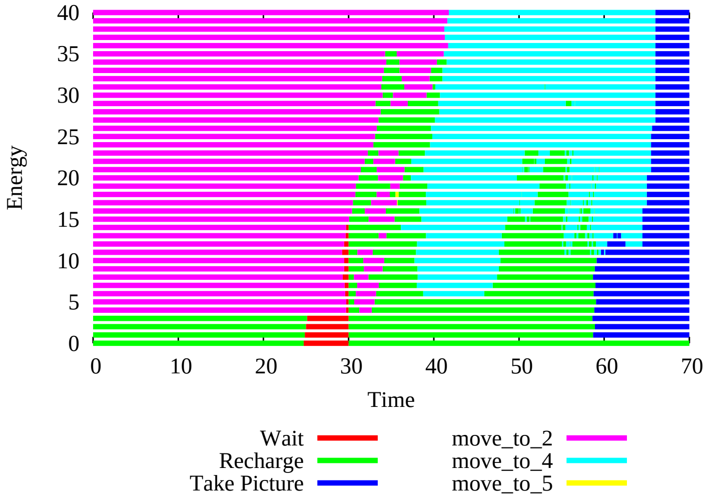
Mars rover policy

V and π in $p = 3$ when no goals have been completed yet.



Mars rover policy

π in $p = 3$ when no goals have been completed yet — 2D view.



Related work and differences

Representation issues and formal resolution

- [Feng et al., 2004] extends the [Boyan and Littman, 2001] idea to continuous state spaces with discrete transition models and uses kd-trees for storing partitions.
- [Li and Littman, 2005] extends to continuous state space MDPs and PW constant functions illustrating the need for simplification.
- *TiMDP*$_{poly}$ extends to PWP representations in the one-dimensional case with direct generalization to continuous state spaces.
- *TiMDP*$_{poly}$ keeps the specific *wait* action of TiMDPs.

Related work and differences

Dynamic Programming

[Boyan and Littman, 2001, Feng et al., 2004, Li and Littman, 2005]
$\longrightarrow$ finite horizon optimization

Optimality equation analysis:

$TiMDP_{poly}$ $\longrightarrow$ infinite horizon, asynchronous optimization.

Conclusion

- We exploit previous results about observable time in MDPs [Rachelson et al., 2008] to provide better understanding of TiMDPs
- *TiMDP*$_{poly}$: an improved VI algorithm for solving TiMDPs with
    - Analytical Bellman backups
    - L_∞-bounded value function approximation
    - Asynchronous dynamic programming

Perspectives

- Generalization to continuous state space MDPs
  Rectangular partitions? Kuhn triangulations?
- Spline theory tools.
- Continuous action parameter optimization.
- Prioritizing $prio(s) \longrightarrow prio(s, I)$.

Thank you for your attention!

Boyan, J. A., and Littman, M. L.
2001.
Exact Solutions to Time Dependent MDPs.
Advances in Neural Information Processing Systems 13:1026–1032.

Li, L., and Littman, M. L.
2005.
Lazy Approximation for Solving Continuous Finite-Horizon MDPs.
In *National Conference on Artificial Intelligence*.

Feng, Z.; Dearden, R.; Meuleau, N.; and Washington, R.
2004.
Dynamic Programming for Structured Continuous Markov Decision Problems.
In *Conference on Uncertainty in Artificial Intelligence*.

Rachelson, E., Garcia, F., and Fabiani, P.
2008.
Extending the Bellman Equation for MDP to Continuous Actions and Continuous Time in the Discounted Case.
In *International Symposium on Artificial Intelligence and Mathematics*.