

Approximate Policies for Time Dependent MDPs



Emmanuel Rachelson
ONERA-DCSD, Toulouse, France



Continuous Time and MDPs

Continuous Time Markov Processes [1]

CTMDP and Semi-MDPs:

- uncertain continuous transition time
- time-homogeneous (stationary)
→ no time-dependency

Criteria: discounted, average.

Optimization: Turns into a discrete time MDP.

[1] M.L. Puterman. *Markov Decision Processes*. John Wiley & Sons, Inc, 1994.

Asynchronous events : GSMDP[2]

Builds on the GSMP framework:

- stochastic clock with each event
- events rush to trigger
→ composite process of concurrent SMDPs

Criterion: discounted.

Optimization: Approximation using continuous phase-type distribution and conversion to a CT-MDP.

[2] H.L.S. Younes and R.G. Simmons. Solving Generalized Semi-Markov Decision Processes using Continuous Phase-type Distributions. In *AAAI*, 2004.

Concurrent actions: CoMDP[3]

Similar to multi-agent MDPs:

- integer valued durations
- concurrent actions
→ execution of non-mutex action combinations

Criterion: discounted, total.

Optimization: RTDP (simulation based value iteration) algorithms.

[3] Mausam and D. Weld. Concurrent probabilistic temporal planning. In *ICAPS*, 2005.

Time as a resource

Stochastic Shortest Path problems:

- absorbing goal states
- eg. Mars rover benchmark [4]

Algorithms:

- HAO*
- ALP algorithms
- Feng et al. continuous structured MDPs
- ...

[4] J. Bresina, R. Dearden, N. Meuleau, D. Smith, and R. Washington. Planning under Continuous Time and Resource Uncertainty: a Challenge for AI. In *UAI*, 2002.

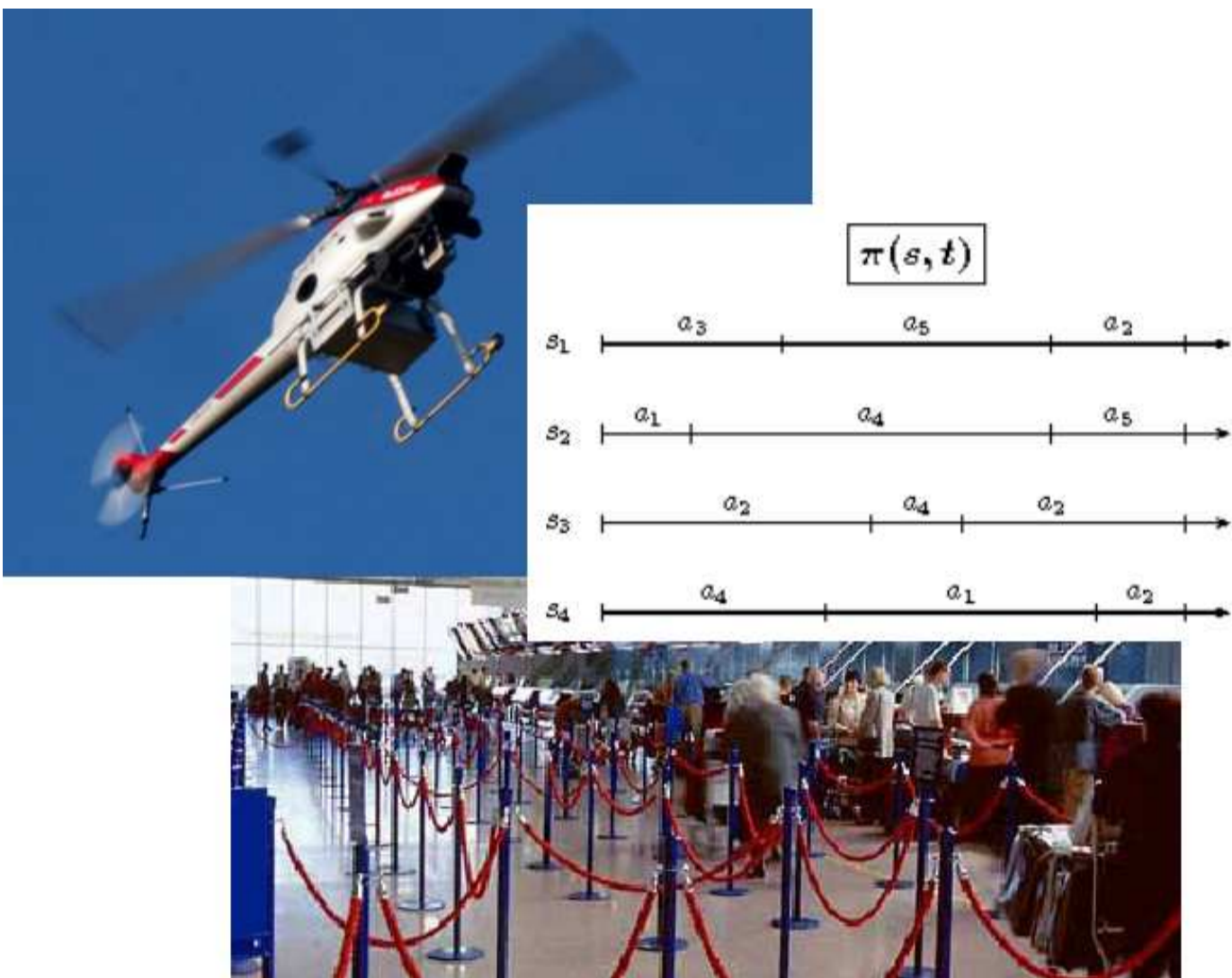
Our problem

Fully observable MDPs with:

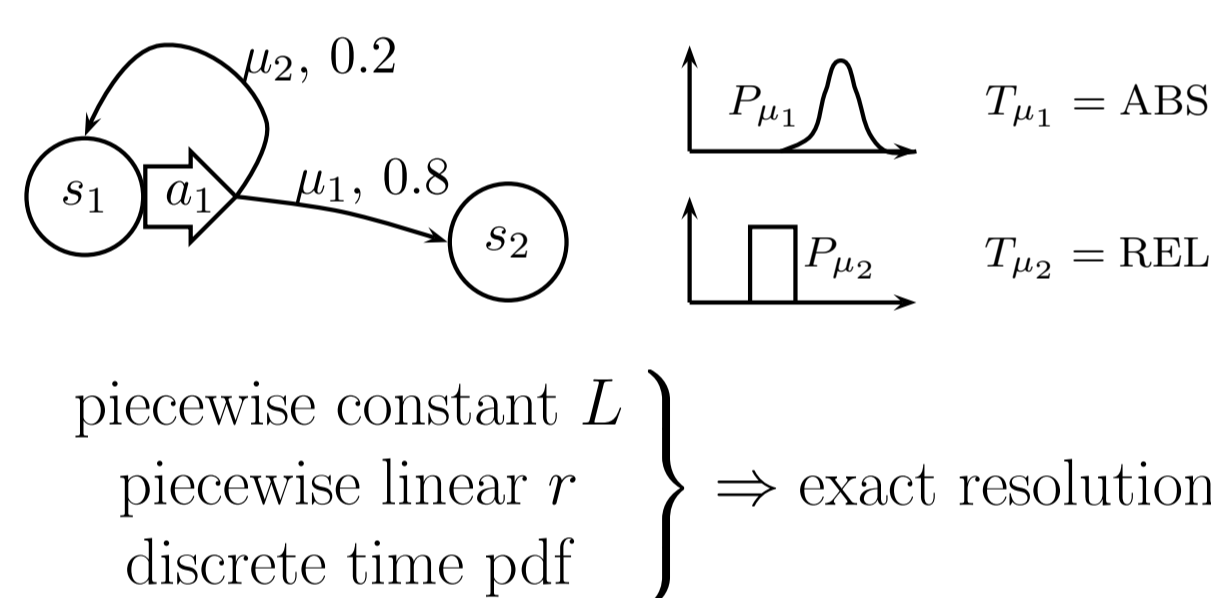
- *continuous time*
- *time-dependent* dynamics (unstationary problems)
→ We look for policies defined as *timelines*.

In a first step: non-absorbing goal-states and no knowledge of initial state.

Examples: subway traffic control, airport queues, forest fire monitoring, ...



TMDP [5]



[5] J.A. Boyan and M.L. Littman. Exact Solutions to Time-Dependent MDPs. In *NIPS*, 2001.

Our contributions: *TMDPpoly*

Generalization of the TMDP results to piecewise polynomial functions and distributions with exact and approximate resolution.

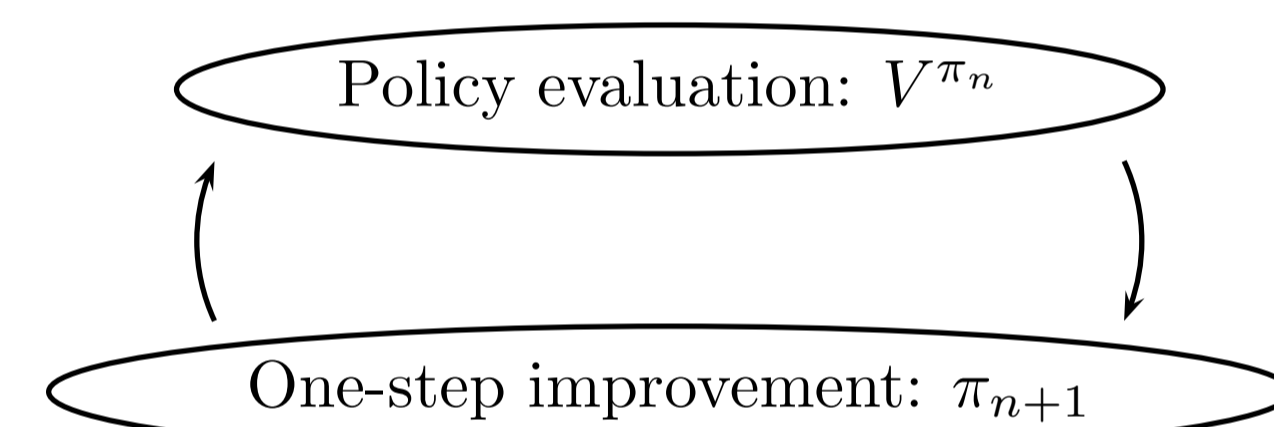
XMDP

A framework for expressing parametric actions in MDPs, such as “wait(τ)”. We proved the existence of Bellman equations in the discounted case.

ATPI

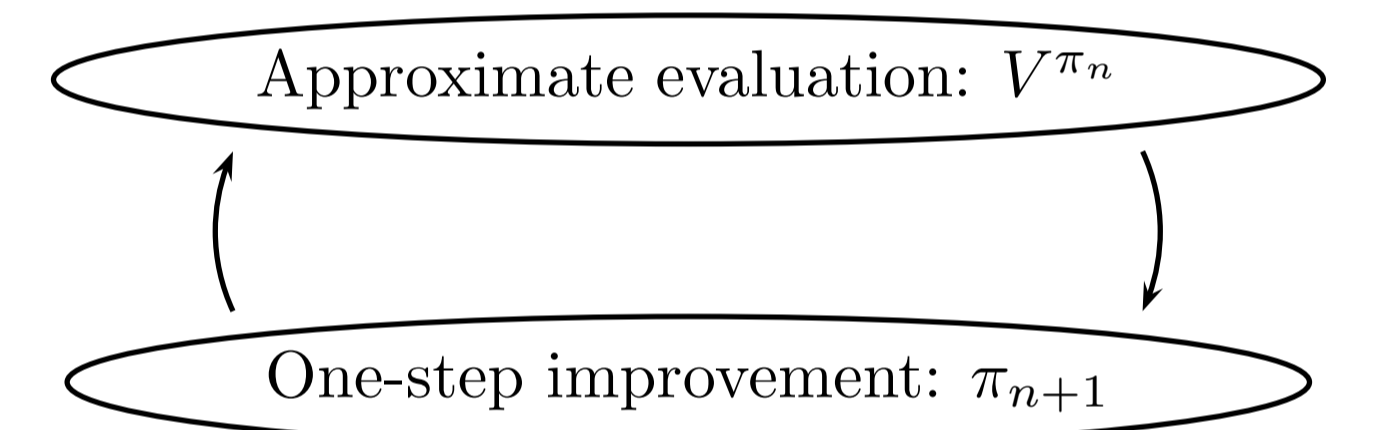
Policy Iteration

Init: π_0



Approximate Policy Iteration

Init: π_0

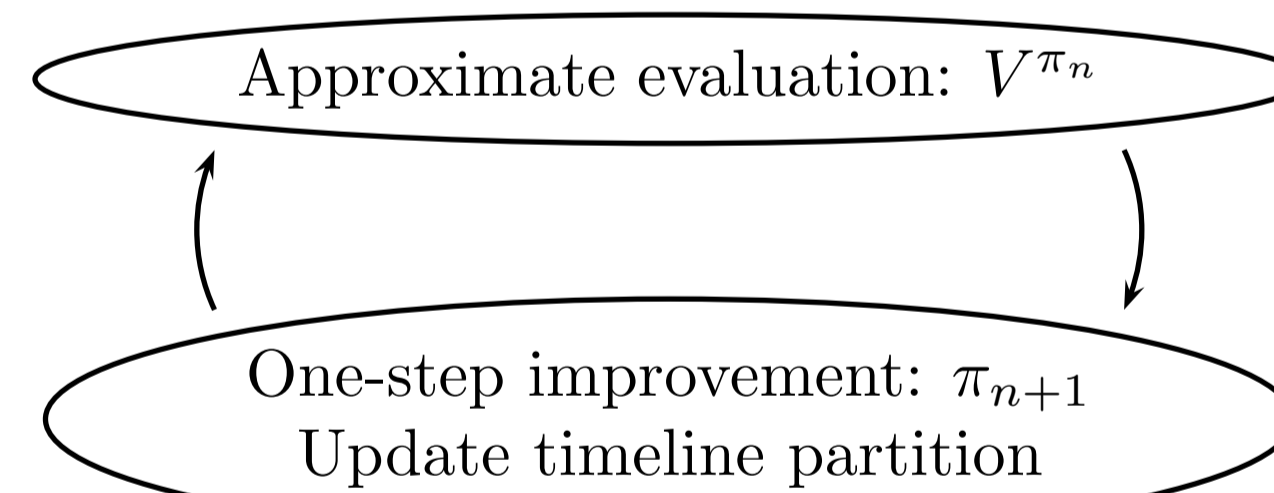


Warning: convergence issues !

Approximate Temporal Policy Iteration

Idea: find simultaneously the timeline partition and the actions to perform

Init: π_0



Different algorithms can be used for each step. For the first step, examples are: piecewise constant or polynomial approximations, linear programming on feature functions, etc. For the second step: Bellman error maximization, sampling, etc.

ATPI using TMDP approximation

Problem formulation

We suppose we have a generic problem formulated as follows:

- State space: $S \times t$
- Action space: A
- Transition model:
 $p(s', t' | s, t, a) = P(s' | s, t, a) \cdot f(t' | s, t, a, s')$
- Reward model: $r(s, t, a)$

General idea: iteratively construct the timelines using a TMDP approximation of the model at each step for evaluation and Bellman error calculation.

We use the following operators:

- $PC^{\pi_n}(\cdot)$: uses π_n 's time partitions to build a piecewise constant function with the argument function.
- $sample(\cdot)$: samples a continuous pdf in non zero values in order to build a discrete pdf.
- $BE_s(V)$: Calculates the one-step improvement of π_n in s using V and the general continuous model, and the date t_s where Bellman error ϵ_s was the greatest.

Algorithm

```

/* Initialization */
pi_{n+1} ← pi_0
associate each (s', a, s) with one or several mu
repeat
    pi_n ← pi_{n+1}
    /* TMDP approximation */
    foreach s in S do
        (t_s, epsilon_s, a_s(t)) ← BE_s(V^{pi_n})
        P_mu(t' - t) = sample(F(t'|s, t, pi_n(s, t), s'))
    end
    /* V^{pi_n} calculation */
    solve (within epsilon-optimality) V^{pi_n} = L^{pi_n} V^{pi_n}
    /* timelines and policy update */
    foreach s in S do
        (t_s, epsilon_s, a_s(t)) ← BE_s(V^{pi_n})
        if epsilon_s > epsilon then
            timeline(s) ← timeline(s) ∪ {t_s}
            pi_{n+1}(s, timeline(s)) ← a_s(t)
        end
    end
until pi_{n+1} = pi_n
    
```

Other ATPI versions

- Piecewise constant approximation and discrete MDP resolution: first proposed in [6]. Relies on approximation for discretization. Issue: more adapted for replenishable resources (some versions of the algorithm allow reverse time)
- Linear programming on a family of feature functions: not explored yet.

[6] E. Rachelson, P. Fabiani, J.-L. Farges, F. Teichteil & F. Garcia. Une approche du traitement du temps dans le cadre MDP : trois méthodes de découpage de la droite temporelle. In *JFPDA*, 2006.

Online ATPI ?

Idea: Only evaluate and update the policy in relevant states using heuristic search guided by the initial policy. RTDP-like selection of states for updates.

→ Simulation-based Policy Iteration

Issue: Convergence not guaranteed.