AI planning for nonlinear optimal control

Applications to switched systems

Lucian Buşoniu

Technical University of Cluj-Napoca, Romania

IFAC CESCIT, 6 June 2018

Part I

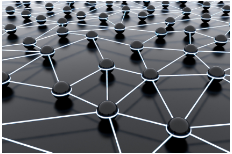
Introduction. Single-agent problems

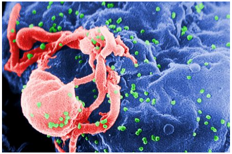
Overall theme

AI-based control of complex systems

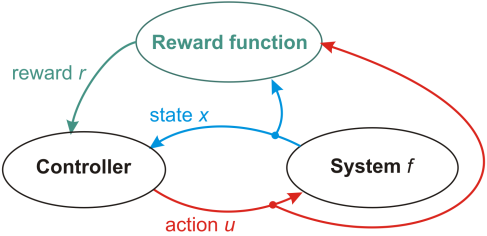
Complexity: general nonlinearity, stochastic dynamics, unknown behavior, distributed structure . . .

Applications: robotics, control, medicine, . . .





Setting: Deterministic Markov decision process

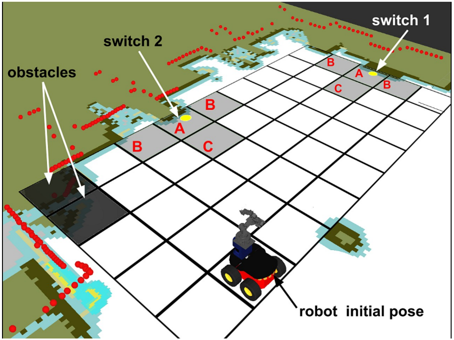


- At step k, controller measures states x, applies actions u
- System: dynamics $x_{k+1} = f(x_k, u_k)$
- Performance: reward function $r_{k+1} = \rho(x_k, u_k)$
- **Objective**: apply actions so as to maximize return

$$\sum_{k=0}^{\infty} \gamma^k r_{k+1}$$

with discount factor $\gamma \in (0, 1)$

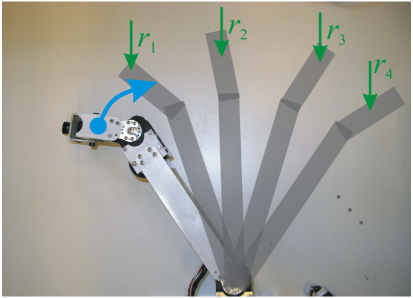
Example: Domestic robot



Domestic robot ensures light switches are off
Abstractization to high-level control (physical actions implemented by low-level controllers)

- States: grid coordinates, switch states
- Actions: movements NSEW, toggling switch
- Rewards: when switches toggled on→off

Example: Robot arm



Low-level control

- States: link angles and angular velocities
- Actions: motor voltages
- Rewards: e.g. to reach a desired state, minus the squared distance to that state

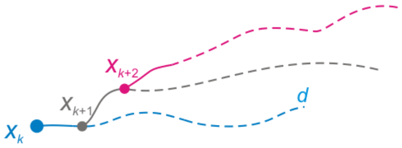
Example: Power-assisted wheelchair (Autonomad, T.M. Guerra, G. Feng)

- Hybrid power source: human and battery
- Objective: perform driving task, optimizing assistance to:
  - (i) attain desired user fatigue level
  - (ii) minimize battery usage
- Challenge: **unknown human dynamics** in the loop

Online planning idea

At each step, use a model to solve problem locally:

1. Explore action sequences from current state, to find a near-optimal sequence
2. Apply first action of this sequence, and repeat



- A type of receding-horizon model-predictive control
- Extension of classical planning / tree search (A*, B*, AO*)

Advantages of OP

- **Near-optimality guarantees** depending on computation *n* and complexity κ of the problem:

$$\text{error} = \mathrm{O}(\text{function}(n, \kappa))$$

(Munos, 2014)

- ...for general nonlinear dynamics and rewards

Talk structure

Online, optimistic planning (OP) in:

- Single-agent problems
  - algorithm
  - analysis
  - application to switched systems

- Adversarial, two-agent problems
  - algorithm
  - analysis
  - application to dual switched systems

Setting

Assumptions

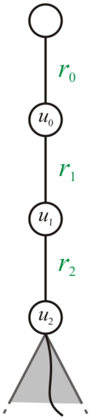
- Finite, discrete action space $U = \{u^1, \dots, u^M\}$
- Bounded reward function $\rho(x, u) \in [0, 1], \forall x, u$

Denote current step by 0 (by convention). Then:

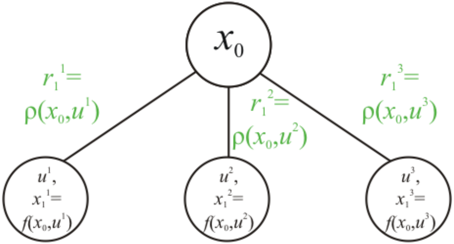
- Infinite action sequences: $\boldsymbol{u}_\infty = (u_0, u_1, \dots)$
- Solve $v^* = \sup_{\boldsymbol{u}_\infty} v(\boldsymbol{u}_\infty) := \sum_{k=0}^{\infty} \gamma^k r_{k+1}$

Setting: Values

- Finite sequence $\boldsymbol{u}_d = (u_0, \dots, u_{d-1})$
- $\ell(\boldsymbol{u}_d) = \sum_{k=0}^{d-1} \gamma^k \rho(x_k, u_k)$, **lower bound** on returns of $\boldsymbol{u}_\infty$ starting with $\boldsymbol{u}_d$
- $b(\boldsymbol{u}_d) = \ell(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma}$, diameter **optimistic upper bound** on the returns
- $v(\boldsymbol{u}_d) = \sup_{\boldsymbol{u}_\infty \text{ st. w. } \boldsymbol{u}_d} v(\boldsymbol{u}_\infty)$ value of applying $\boldsymbol{u}_d$ and then acting optimally



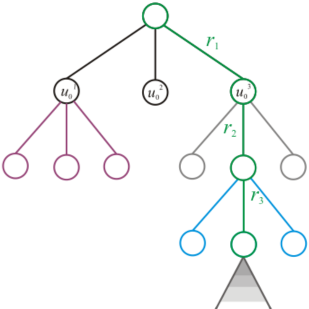
Tree structure



- Each tree node has the meaning of state
- One child for each action,
  each transition associated with a reward

Optimistic planning for deterministic systems (OPD)

initialize empty sequence $\boldsymbol{u}_0$
for $t = 1$ to n **do**
select **optimistic** leaf sequence $\boldsymbol{u}_t^{\dagger}$, maximizing b
expand $\boldsymbol{u}_t^{\dagger}$: children for all actions, setting ℓ and b
end for
return $\boldsymbol{u}_d^*$ maximizing ℓ, and maximal ℓ^*, b^*



(Hren & Munos, 2008)

Relation to reinforcement learning

RL solves MDPs without using a model, by learning

A deeper relation:

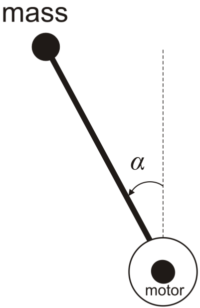
At one state, RL exploration modeled as multi-armed bandit:

- Discrete actions = arms with unknown, stochastic rewards
- Pull arms to learn, so that after *n* pulls,
  the optimal arm has been pulled the most
- Good idea: **optimism in the face of uncertainty**
  – pull arm with best upper confidence bounds

Relation to reinforcement learning (cont'd)

- In OP, the model is known, but the optimal sequence is not, because rewards only known up to depth d
- Sample transitions, so that after n expansions, sequence is close to optimal
- **Optimism in the face of uncertainty**: assume maximal rewards of 1 beyond depth d

Example: Inverted pendulum

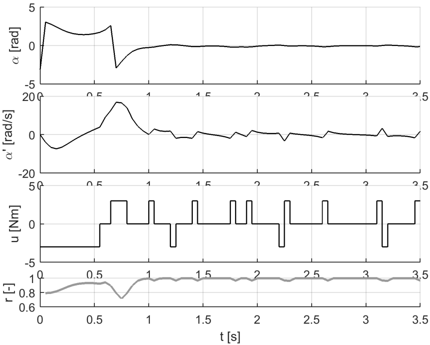


- $x = [\text{angle } \alpha, \text{ velocity } \dot{\alpha}]^\top$
- u = voltage
- $\rho(x, u) = -x^\top Q x - u^\top R u$, normalized to $[0, 1]$
- Discount factor $\gamma = 0.98$

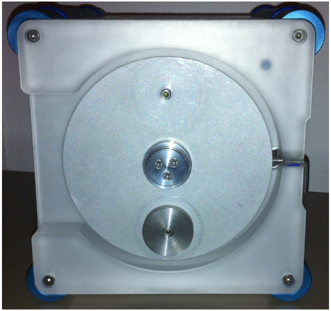
- Objective: stabilize pointing up
- Insufficient torque $\Rightarrow$ swing-up required

Example: Real-time demo

Swingup in simulation:



Real-time demo:



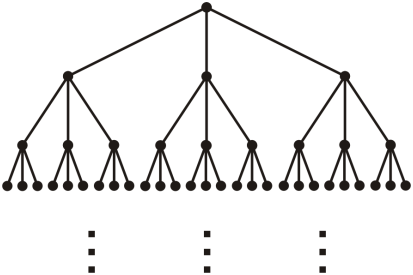
Near-optimality vs. depth

Theorem

1. OPD returns a sequence $\boldsymbol{u}_d^*$ so that $v(\boldsymbol{u}_d^*)$ and the optimal value v^* are both in $[\ell^*, b^*]$
2. The near-optimality gap $b^* - \ell^* \leq \frac{\gamma^{d^*}}{1-\gamma}$ where d^* is the deepest expanded

Case 1: All paths optimal

Take a tree where all rewards are 1:



$b(\boldsymbol{u}_d) = \frac{1}{1-\gamma}, \ \forall \boldsymbol{u}_d \ \Rightarrow$ OPD expands uniformly, breadth-first

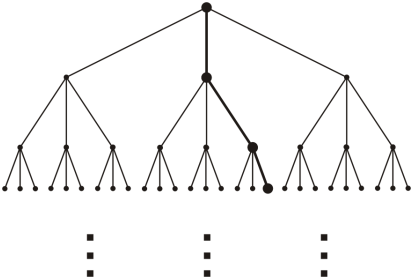
So to expand all nodes down to depth d, we must spend:

$$n = \sum_{i=0}^{d} M^i = \frac{M^{d+1} - 1}{M - 1}$$

and the depth grows slowly with budget n

Case 2: One path optimal

Take a tree where rewards are 1 only along a single path (thick line), and 0 everywhere else:



$b(\boldsymbol{u}_d) = \frac{1}{1-\gamma}$ only on optimal path, $\frac{\gamma^d}{1-\gamma}$ elsewhere
$\Rightarrow$ OPD expands only the optimal path

So to expand down to depth d, we must spend only $n = d$, and the depth grows fast with n

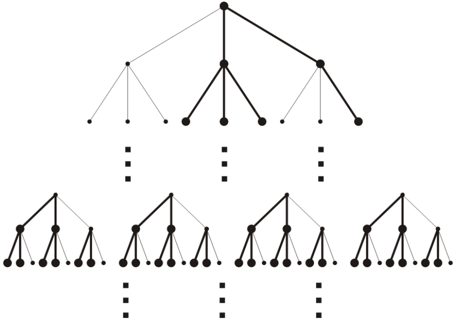
General case: Branching factor

- Algorithm only expands in near-optimal subtree:

$$\mathcal{T}^* = \left\{ \boldsymbol{u}_d \,\middle|\, v^* - v(\boldsymbol{u}_d) \leq \frac{\gamma^d}{1-\gamma} \right\}$$

- Define $\kappa \in [1, M]$ = asymptotic branching factor of $\mathcal{T}^*$:
  problem complexity measure

E.g. $\kappa = 2$, $M = 3$:



Depth vs. budget n

To reach depth d in tree with branching factor κ,
we must expand $n = \mathrm{O}(\kappa^d)$ nodes

$$\Rightarrow \quad d^* = \Omega\left(\frac{\log n}{\log \kappa}\right)$$

Final guarantee: Near-optimality vs. budget

Theorem

3. The near-optimality gap is:

$$b^* - \ell^* \leq \frac{\gamma^{d^*}}{1-\gamma} = \begin{cases} \mathrm{O}(n^{-\frac{\log 1/\gamma}{\log \kappa}}) & \text{if } \kappa > 1 \\ \mathrm{O}(\gamma^{cn}) & \text{if } \kappa = 1 \end{cases}$$

- Generality paid by exponential computation $n = \mathrm{O}(\kappa^d)$
- But κ can be small in interesting problems!

(Hren & Munos, 2008)

Setting

- Switched system $x_{k+1} = f(x_k, u_k)$,
  where now u has the meaning of **mode**
- Stage cost $g(x_k, u_k)$
- Cost function of infinite mode sequence:

$$J(\boldsymbol{u}_\infty) = \sum_{k=0}^{\infty} \gamma^k g(x_k, u_k)$$

with discount factor $\gamma \in (0, 1)$

(Automatica 2017)

Motivation

Open challenge

- Optimal control of nonlinear switched systems

(see survey of Zhu & Antsaklis, 2015)

Optimistic planning offers:

- General nonlinear modes
- Sequence design
- Certification bounds

...but without stability guarantees

Problem statement

- Optimal control, PO: Find $\underline{J} = \inf_{\boldsymbol{u}_\infty} J(\boldsymbol{u}_\infty)$ and corresponding sequence
- Worst-case switches, PW: Find $\overline{J} = \sup_{\boldsymbol{u}_\infty} J(\boldsymbol{u}_\infty)$ and corresponding sequence

Assumption

Bounded stage costs $g(x, u) \in [0, 1], \forall x, u$

Direct application of OP

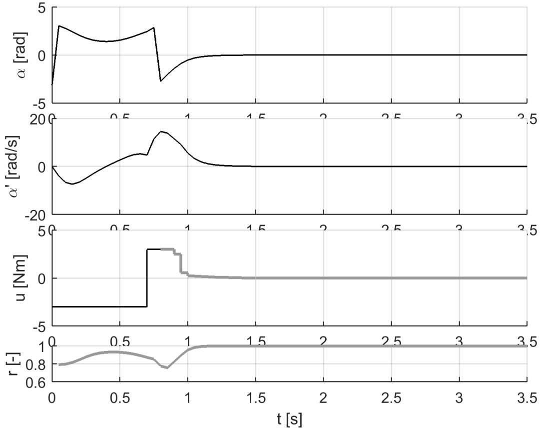
To solve PO, take rewards $\underline{\rho} = 1 - g$
To solve PW, take rewards $\overline{\rho} = g$

Corollary

4. In PO, cost of sequence returned and optimal cost $\underline{J}$ are in $[\frac{1}{1-\gamma} - b^*, \frac{1}{1-\gamma} - \ell^*]$, and the gap is $\mathrm{O}(n^{-\frac{\log 1/\gamma}{\log \underline{\kappa}}})$.
5. In PW, cost of sequence returned and worst-case cost $\overline{J}$ are in $[\ell^*, b^*]$, and the gap is $\mathrm{O}(n^{-\frac{\log 1/\gamma}{\log \overline{\kappa}}})$.

Inverted pendulum simulation

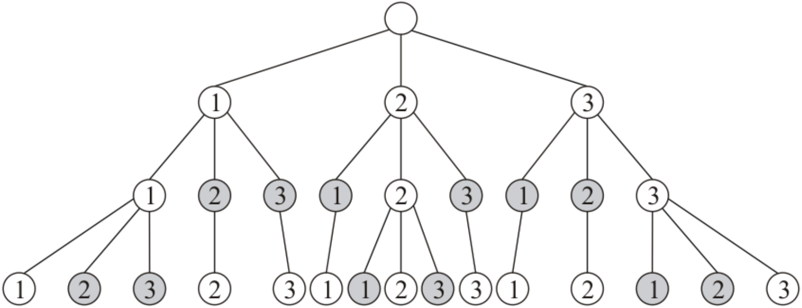
Zero action replaced by PD control mode:



Minimum dwell time

- **Minimum dwell time** δ (number of steps between switches) often required due to e.g. fundamental properties, practical actuator limitations

$\Rightarrow$ Only explore sequences ensuring dwell time δ



Algorithm: OPδ

initialize $\boldsymbol{u}_0$
for $i = 1$ to computational budget n **do**
select optimistic leaf sequence $\boldsymbol{u}_d^\dagger$, maximizing b
expand $\boldsymbol{u}_d^\dagger$:
if last mode in $\boldsymbol{u}_d^\dagger$ was active $< \delta$ steps **then**
create single child, continuing same action
else
create all children
end if
end for
return $\boldsymbol{u}_d^*$ maximizing ℓ, and maximal ℓ^*, b^*

Near-optimality vs. depth

Notation: subscript δ = constrained to obey the dwell time

Theorem

1. OPδ returns a sequence $\boldsymbol{u}_d^*$ so that $v_\delta(\boldsymbol{u}_d)$ and v_δ^* are both in $[\ell^*, b^*]$
2. Near-optimality gap $b^* - \ell^* \leq \frac{\gamma^{d^*}}{1-\gamma}$ where d^* is the deepest expanded

Complexity measure

- Algorithm only expands in constrained near-optimal subtree:

$$\mathcal{T}_\delta^* = \left\{ \boldsymbol{u}_d \text{ constrained} \;\middle|\; v_\delta^* - v_\delta(\boldsymbol{u}_d) \leq \frac{\gamma^d}{1-\gamma} \right\}$$

- Define $K \in [1, M\delta]$ = the smallest number so that $\left|\mathcal{T}_{d,\delta}^*\right| = \mathrm{O}(K^{d/\delta})$;
problem complexity measure
- Problem is simpler when K is smaller; intuitive meaning less clear than branching factor κ

Near-optimality vs. budget

To reach depth d, we expand $n = O(K^{d/\delta})$ nodes
$\Rightarrow$ largest depth $d^* = \Omega(\delta \frac{\log n}{\log K})$

Theorem (cont'd)

3. Near-optimality gap is:

$$b^* - \ell^* \leq \frac{\gamma^{d^*}}{1-\gamma} = \begin{cases} O(n^{-\delta \frac{\log 1/\gamma}{\log K}}) & \text{if } K > 1 \\ O(\gamma^{cn}) & \text{if } K = 1 \end{cases}$$

Comparison between OPδ and OP

- Take largest values of $K = M\delta$, $\kappa = M$ (most difficult problem)

$\Rightarrow$ Gaps are $\mathrm{O}(n^{-\delta \frac{\log 1/\gamma}{\log M\delta}})$ and $\mathrm{O}(n^{-\frac{\log 1/\gamma}{\log M}})$

- Since $\delta \frac{\log 1/\gamma}{\log M\delta} > \frac{\log 1/\gamma}{\log M}$, OP$\delta$ converges faster; due to OPδ exploring smaller, constrained tree
- However, the relationship will vary with the problem

Solving PO and PW with dwell time

Corollary

4. In PO, cost of sequence returned and optimal cost $\underline{J}_\delta$ are in $[\frac{1}{1-\gamma} - b^*, \frac{1}{1-\gamma} - \ell^*]$, and the gap is $O(n^{-\delta \frac{\log 1/\gamma}{\log \underline{K}}})$.
5. In PW, cost of sequence returned and worst-case cost $\overline{J}_\delta$ are in $[\ell^*, b^*]$, and the gap is $O(n^{-\delta \frac{\log 1/\gamma}{\log \overline{K}}})$.

Part II

Adversarial problems

Adversarial problem

- Look for "our" actions u that maximize return assuming opponent takes actions w to minimize it
- Two-player competitive games, robust control, etc.

Setting

- Maximizer & minimizer agents,
  with actions $u \in U$ and $w \in W$; $|U| = M_u, |W| = M_w$
- They alternately take an infinite sequence of actions:

$$(u_0, w_0, u_1, w_1, \dots) =: (z_0, z_1, z_2, \dots) = \boldsymbol{z}_\infty$$

- Dynamics $x_{d+1} = f(x_d, z_d)$, rewards $\rho(x_d, z_d)$
- Finite sequence $\boldsymbol{z}_d = (z_0, \dots, z_{d-1})$

Objective

Infinite-horizon value of sequence $\boldsymbol{z}_\infty$:

$$v(\boldsymbol{z}_\infty) := \sum_{d=0}^{\infty} \gamma^d \rho(x_d, z_d).$$

Objective: **discounted minimax-optimal solution:**

$$v^* := \max_{u_0} \min_{w_0} \cdots \max_{u_k} \min_{w_k} \cdots \quad v(\boldsymbol{z}_\infty)$$

Setting: Assumptions

Assumptions

- Both agents have discrete actions
- The rewards $\rho(x, z)$ are in $[0, 1]$ for all $x \in X, z \in U \cup W$.

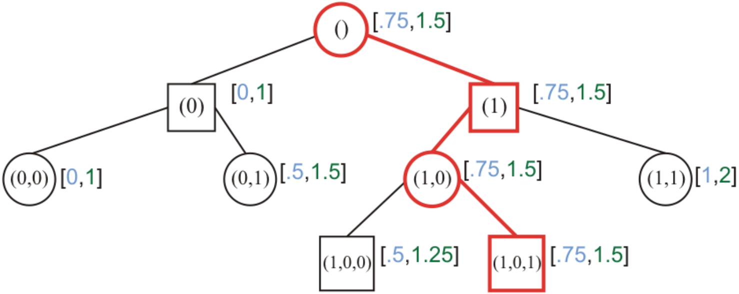
$\Rightarrow$ lower & upper bounds on all sequences $\boldsymbol{z}_\infty$ starting with $\boldsymbol{z}_d$:

$$\ell(\boldsymbol{z}_d) = \sum_{j=0}^{d-1} \gamma^j \rho(x_j, z_j), \quad b(\boldsymbol{z}_d) = \ell(\boldsymbol{z}_d) + \frac{\gamma^d}{1-\gamma}$$

where $\frac{\gamma^d}{1-\gamma}$ is the diameter, as before

Optimistic minimax search

OMS expands tree of possible minmax sequences, using lower and upper bounds on node values



Application of **classical, best-first B* search** to infinite-horizon problems (Berliner 1979)

Optimistic minimax search (cont'd)

for $t = 1, \dots, n$ **do**

propagate lower & upper bounds L, B at each node:

$$L(\boldsymbol{z}) \leftarrow \begin{cases} \ell(\boldsymbol{z}), & \text{if } \boldsymbol{z} \text{ leaf} \\ \max / \min_{\boldsymbol{z}' \in \text{children}(\boldsymbol{z})} L(\boldsymbol{z}'), & \text{otherwise} \end{cases}$$

$$B(\boldsymbol{z}) \leftarrow \begin{cases} b(\boldsymbol{z}), & \text{if } \boldsymbol{z} \text{ leaf} \\ \max / \min_{\boldsymbol{z}' \in \text{children}(\boldsymbol{z})} B(\boldsymbol{z}'), & \text{otherwise} \end{cases}$$

choose node to expand: $\boldsymbol{z} \leftarrow$ root, and while not leaf:

$$\boldsymbol{z} \leftarrow \begin{cases} \arg\max_{\boldsymbol{z}' \in \text{children}(\boldsymbol{z})} B(\boldsymbol{z}'), & \text{if } \boldsymbol{z} \text{ max node} \\ \arg\min_{\boldsymbol{z}' \in \text{children}(\boldsymbol{z})} L(\boldsymbol{z}'), & \text{if } \boldsymbol{z} \text{ min node} \end{cases}$$

expand $\boldsymbol{z}$

end for

output a **maximum-depth** expanded node $\boldsymbol{z}^*$

Example: HIV treatment

- 6 states:

  T_1, T_2 – healthy target cells per ml (types 1 & 2)
  T_1^t, T_2^t – infected target cells per ml (types 1 & 2)
  V – free virus copies per ml
  E – immune response cells per ml

- $M_u = 2$ actions u_1, u_2: application of RTI and PI drugs
  Unpredictable drug effectiveness among $M_w = 2$ levels

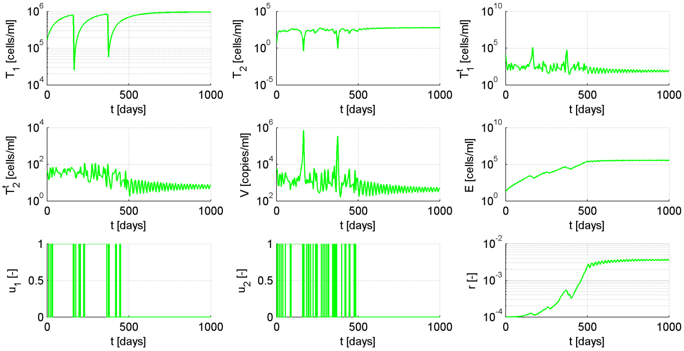
Goal: Starting from high level of infection x_0, optimally switch drugs on and off to:

1. maximize immune response
2. minimize virus load
3. minimize drug use

$$r = c_E E - c_V V - c_1 \epsilon_1 - c_2 \epsilon_2$$

HIV: OMS results

Effectiveness conservatively treated as opponent
Budget of $n = 4000$ node expansions



Infection eventually controlled without drugs

Near-optimality vs. diameter

For finite sequence $\boldsymbol{z}$, let $v(\boldsymbol{z})$ be the minimax-optimal value among sequences starting with $\boldsymbol{z}$

1. If d^* is the largest depth expanded, the solution $\boldsymbol{z}^*$ returned by OMS satisfies:

$$|v^* - v(\boldsymbol{z}^*)| \leq \frac{\gamma^{d^*}}{1-\gamma}$$

Explored tree

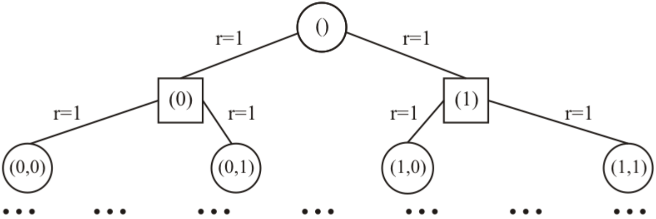
- Algorithm only expands nodes in the subtree:

$$\mathcal{T}^* := \left\{ \boldsymbol{z}_d \,\middle|\, \left| v^* - v(\boldsymbol{z}') \right| \leq \frac{\gamma^d}{1-\gamma}, \forall \boldsymbol{z}' \text{ on path from root to } \boldsymbol{z}_d \right\}$$

- Intuition: From the information available down to node $\boldsymbol{z}_d$ (interval of values of width $\frac{\gamma^d}{1-\gamma}$), cannot decide whether the node is (not) optimal. So it must be explored.

Example where the full tree is explored

- All rewards equal to 1, $v^* = \frac{1}{1-\gamma}$
- All solutions have value v^*, so $\mathcal{T}^*$ is the full tree
- $\left|\mathcal{T}_d^*\right| = (M_u M_w)^{d/2}$, branching factor $\kappa = \sqrt{M_u M_w}$



General case: Branching factor

- Let $\kappa \in [1, \sqrt{M_u M_w}]$ = asymptotic branching factor of $\mathcal{T}^*$
- Problem simpler when κ smaller

Depth vs. budget n

To reach depth d in tree with branching factor κ,
we must expand $n = \mathrm{O}(\kappa^d)$ nodes

$$\Rightarrow \quad d^* = \Omega\left(\frac{\log n}{\log \kappa}\right)$$

Final guarantee: Near-optimality vs. budget

Theorem

2. Given budget n, we have:

$$|v^* - v(\boldsymbol{z}^*)| \leq \frac{\gamma^{d^*}}{1-\gamma} = \begin{cases} O(n^{-\frac{\log 1/\gamma}{\log \kappa}}) & \text{if } \kappa > 1 \\ O(\gamma^{cn}) & \text{if } \kappa = 1 \end{cases}$$

(ADPRL 2014)

- Faster convergence when κ smaller (simpler problem)
- Exponential convergence when $\kappa = 1$

Setting

- Actions u, w now have the meaning of switching signals, u controlled, w uncontrolled: **dual switched system**

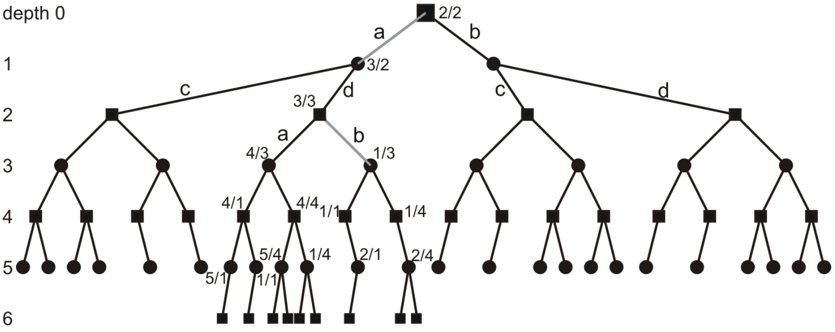
  (Bolzern et al., 2014)

- Signals respectively obey minimum dwell times δ_u, δ_w
- Notation: subscript δ = constrained to obey dwell times
- If $\delta_u = \delta_w = 1$, problem reduces to standard min-max and OMS directly applies

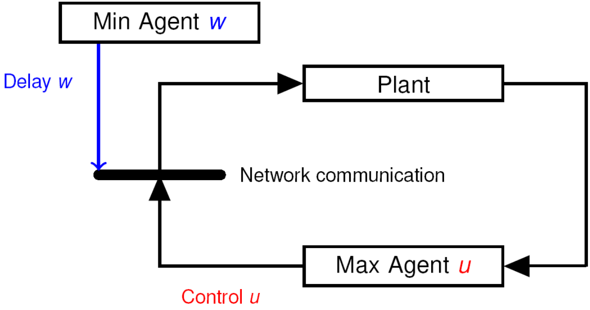
OMSδ for dual switched systems

OMSδ algorithm: mostly the same as OMS,
but when node does not satisfy dwell time condition,
only the child keeping the action constant is created

Example constrained tree for $\delta_u = \delta_w = 2$:



Switched control over delayed network



- Max action = controlled "mode"
  e.g. constant action or low-level controller
- Min action = network delay (multiple of sampling time)

Quanser rotational pendulum

System:

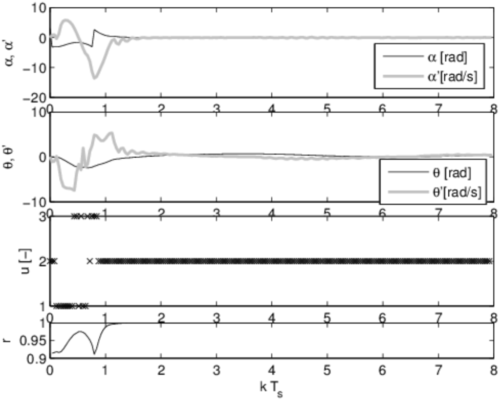
- x = rod angle α, base angle θ, angular velocities
- input ω = voltage
- Sampling time $T_s = 0.04$

Goal: swing up & stabilize pointing up:

- Reward $-x^\top Qx - \omega^\top R\omega$, normalized to $[0, 1]$
- Discount factor $\gamma = \sqrt{0.95}$

Results

- $M_u = 3$: #1 constant -6 V, #3 constant 6 V, #2 a stabilizing mode $\omega = Kx$ computed with LQR
- $M_w = 2$: 0 or 1-step delay



Near-optimality vs. depth

Similar to OMS

1. If d^* is the largest depth expanded, the solution $\widehat{\boldsymbol{z}}$ returned by OMSδ satisfies:

$$\left| v_\delta^* - v_\delta(\widehat{\boldsymbol{z}}) \right| \leq \frac{\gamma^{d^*}}{1-\gamma}$$

Complexity measure

Different from OMS, generalizes OPδ

- At depth d, algorithm only expands in the subtree:

$$\mathcal{T}^*_{\delta,d} := \Big\{ \boldsymbol{z}_d \;\Big|\; \boldsymbol{z}_d \text{ obeys dwell time conditions },$$
$$\left| v^*_\delta - v_\delta(\boldsymbol{z}') \right| \leq \frac{\gamma^d}{1-\gamma}, \forall \boldsymbol{z}' \text{ on path from root to } \boldsymbol{z}_d \Big\}$$

- Let $\delta = \min\{\delta_u, \delta_w\}$, $M = \max\{M_u, M_w\}$. Define $K \in [1, \delta M]$ the smallest positive number so that

$$\left| \mathcal{T}^*_{\delta,d} \right| = \mathrm{O}(K^{d/\delta})$$

Near-optimality vs. budget

Theorem

- Given budget n, we have:

$$\left| v_\delta^* - v_\delta(\widehat{\boldsymbol{z}}) \right| \leq \begin{cases} \mathrm{O}(n^{-\delta \frac{\log 1/\gamma}{\log K}}) & \text{if } K > 1 \\ \mathrm{O}(\gamma^{cn}) & \text{if } K = 1 \end{cases}$$

(ACC 2017)

Comparison between OMSδ and OMS

- Just like in the single-agent case, when exploring the full trees, OMSδ converges faster than OMS, since its constrained tree is smaller
- However, the relationship will vary with the problem

Outlook

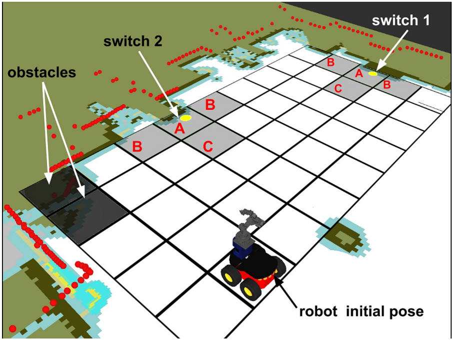
Summary

- Optimistic planning for general nonlinear systems, with performance guarantees
- Natural application to switched systems

Outlook

- Combination with learning
- Continuous and hybrid actions
- Stochastic uncontrolled mode w

Stochastic-case planner for partially-observable MDPs



- Domestic robot makes sure all switches are off
- `NSEW` actions change position on grid,
  `flip` action succeeds stochastically
- Switch states observed incorrectly with certain probas
- Low-level SLAM and control

(IROS 2016)

References

- Textbook: Munos, *From Bandits to Monte Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning*, Foundations and Trends in Machine Learning 7, 2014.
- Hren, Munos, *OP of deterministic systems*, EWRL 2008.
- Zhu, Antsaklis, *Optimal control of switched hybrid systems: A brief survey*, Discrete Event Dynamic Systems 25, 2015.
- Berliner, *The B* Search Algorithm: A Best First Proof Procedure*, Artificial Intelligence 1979.
- Bolzern, Colaneri, Nicolao, *Design of stabilizing strategies for dual switching stochastic-deterministic linear systems*, IFAC World Congress 2014.

- Busoniu, Daafouz, Bragagnolo, Moraresc, *Planning for optimal control and performance certification in nonlinear systems with controlled or uncontrolled switches*, Automatica 78, 2017.
- Busoniu, Munos, Pall, *An Analysis of Optimistic, Best-First Search for Minimax Sequential Decision Making*, ADPRL 2014.
- Ben Rejeb, Busoniu, Moraresc, Daafouz, *Near-Optimal Control of Nonlinear Switched Systems with Non-Cooperative Switching Rules*, ACC 2017.
- Pall, Tamas, Busoniu, *An Analysis and Home Assistance Application of Online AEMS2 Planning*, IROS 2016.