Optimistic Planning for Continuous-Action Deterministic Systems

L. Bușoniu, A. Daniels, R. Munos, R. Babuška
(lucian@busoniu.net)

JFPDA 2013, 1 July, Lille
Optimal control problem (deterministic MDP)

- **System**: dynamics $x_{k+1} = f(x_k, u_k)$
- **Performance**: reward function $r_{k+1} = \rho(x_k, u_k)$
- **Objective**: maximize discounted return $\sum_{k=0}^{\infty} \gamma^k r_{k+1}$
- **Motivation**: very general $f$ and $\rho$
Applications

Robotics, multi-agent systems, medicine, AI, economics etc.
Online planning

At each step $k$, solve local optimal control at state $x_k$:

- Infinite action sequences: $u_\infty = (u_k, u_{k+1}, \ldots) \in U^\infty$
- Optimization problem: $\sup_{u_\infty} v(u_\infty) (= \sum_{i=0}^{\infty} \gamma^i r_{k+1+i})$

1. Explore sequences from $x_k$, to find a near-optimal one $u$
2. Apply first action of $u$

**Focus:** Optimistic planning, deal with continuous actions
1 Background: Optimistic optimization

2 SOOP: Planning with continuous actions

3 Experiments & conclusions
DOO: Deterministic optimistic optimization

- Maximize $v : U \rightarrow \mathbb{R}$, Lipschitz: $|v(u) - v(u')| \leq \ell(u, u')$
- Input: hierarchical partitioning of $U$
- Always expand **optimistic** set, with largest upper bound: $b(U_i) = v(u_i) + \delta(U_i)$, diam. $\delta(U_i) = \sup_{u, u' \in U_i} \ell(u, u')$
- Until $n$ expansions exhausted

(Munos, 2011)
SOO: Simultaneous optimistic optimization

- What if $\ell / \delta$ unknown? (i.e., smoothness of $\nu$ unknown)
- Assume only: $\delta(U_j) \geq \delta(U_i)$ iff depth $d_j \leq d_i$ (total order)
- Expand all potentially optimistic sets $U_i$, for which:
  $\nu(u_i) \geq \nu(u_j)$ for all $j$ at smaller depths, $d_j \leq d_i$

(Munos, 2011)
1. Background: Optimistic optimization

2. SOOP: Planning with continuous actions

3. Experiments & conclusions
Assumptions

- Action space $U = [0, 1]$ (can be extended to compact multidimensional $U$)
- Rewards $r \in [0, 1]$
Partitioning

- Partition $U^\infty$ using iterative trisection (we no longer have a tree structure!)

- Each box $U_i$ represented by only initializing trisected dimensions, $k = 0, \ldots, K_i - 1$

- $\hat{v}(U_i) = \sum_{k=0}^{K_i-1} \gamma^k r_{k+1}$, rewards of center sequence
Challenges

- Challenge 1: $\ell$ (diameters $\delta$) **unknown**
  $\implies$ Use SOO – expand all potentially optimistic boxes

- Challenge 2: Total order on diameters **unavailable**
Partial order

**Definition**

- A box $U_j$ is partially greater than $U_i$ ($U_j \succeq U_i$) if it was trisected fewer (or as many) times along every dimension.

**Assumption**

- If $U_i \succeq U_j$, then diameters $\delta(U_i) \geq \delta(U_j)$.
Relaxed expansion criterion

Box $U_i$ is **potentially optimistic** if $\hat{\nu}(U_i) \geq \hat{\nu}(U_j), \forall U_j \succeq U_i$

- **Safe**: if a box is potentially optimistic, it is expanded
- **Conservative**: a box may be expanded even when not potentially optimistic:

  $\hat{\nu}(U_i) < \hat{\nu}(U_j)$ for some $\delta(U_j) \geq \delta(U_i)$

  but we cannot tell because $U_j \not\succeq U_i$
SOO for Planning: SOOP

**Input:** state $x_0$, budget of model calls $n$
create a single box $[0, 1]^\infty$

**loop** until budget exhausted
select potentially optimistic boxes:
$$\mathcal{O} = \{ U_i | \forall j \text{ so that } U_j \succeq U_i, \hat{v}(U_i) \geq \hat{v}(U_j) \}$$

**for** each box in $U_i \in \mathcal{O}$ **do**
trisect dimension $k$, creating 3 new boxes
remove old box $U_i$

**end for**

**end loop**

**Output:** sequence at center of best box, $\max_i \hat{v}(U_i)$
SOOP details

- Select dimension to trisect:
  \[ \arg \max_k (\alpha^k \cdot \text{size of box along dimension } k) \]
  – since early actions dominate performance
- \( \alpha \in (0, 1) \) is the only parameter of the algorithm
- Expansions take a varying number of model calls
Related work

- **Optimistic planning for deterministic systems (OPD):**
  discrete actions, DOO works
  
  (Hren & Munos 2008)

- **HOLOP, HOOT:** continuous actions, finite horizon
  
  (Weinstein et al. 2012, Mansley et al. 2011)

- **Lipschitz planning (LP):** continuous actions, $f, \rho$ assumed
  Lipschitz with known constants $\Rightarrow$ DOO works
  
  (Hren 2012)

- Also, **adaptive discretization** in global methods
  
  (Pazis & Lagoudakis, 2009)
1. Background: Optimistic optimization

2. SOOP: Planning with continuous actions

3. Experiments & conclusions
Underactuated pendulum swingup

Requires **continuous** actions & **long planning horizon** – SOOP dominates
Swingup example

**SOOP** (left) versus **OPD** (right), $n = 2500$ model calls
Robot arm (horizontal acrobot)

Discrete actions work well, so OPD cannot be outperformed – SOOP holds its ground, still better than LP, HOLOP
Conclusions

**SOOP** algorithm:
- Searches for infinite-horizon, continuous action sequences
- No knowledge about system smoothness
- Competitive in all tested problems

Next step: Near-optimality analysis

Thank you!