

Discounted optimal control with continuous actions using optimistic planning

Lucian Buşoniu, Előd Páll, Rémi Munos

TUCluj, Romania, and Google DeepMind, UK
Contact: lucian@busoniu.net

6 July 2016, ACC, Boston



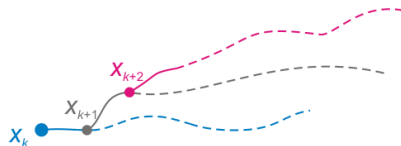
Setting

- Deterministic system $x_{k+1} = f(x_k, u_k)$
- Reward function $\rho(x_k, u_k)$
- Locally at state x_0 , find action sequence $\mathbf{u}_\infty = (u_0, u_1, \dots)$ to maximize discounted value:

$$v(\mathbf{u}_\infty) = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k)$$

where **discount factor** $\gamma \in [0, 1)$

- Find near-optimal sequence, repeat in receding horizon



Optimistic planning (OP): Main idea

initialize **set of all possible sequences**

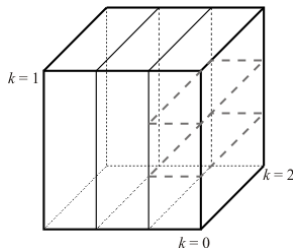
repeat

 select most promising, **optimistic** set

 refine selected set

until computation budget n exhausted

return sequence in best set



Bandit-based optimization; branch & bound if deterministic

Advantages of OP

- **Near-optimality guarantees** as a function of computation n and of complexity m of the problem:

$$\text{error} = O(g(n, m))$$

(Munos, 2014)

- ...for general nonlinear dynamics and rewards



Assumptions

- Rewards $r \in [0, 1]$
- Action space $U = [0, 1]$
(can be extended to compact multidimensional U)
- Lipschitz dynamics and rewards:

$$\|f(x, u) - f(x', u')\| \leq L_f(\|x - x'\| + |u - u'|)$$

$$|\rho(x, u) - \rho(x', u')| \leq L_\rho(\|x - x'\| + |u - u'|)$$

- $\gamma L_f < 1$: most restrictive

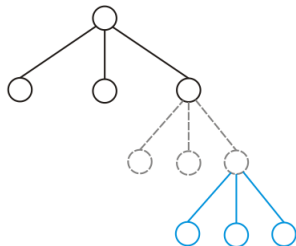
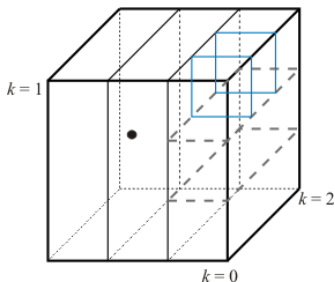


- 1 Introduction
- 2 OPC algorithm
- 3 Near-optimality analysis
- 4 Experiments
- 5 Conclusions



Search refinement

- Split U^∞ iteratively, leading to a tree of hyperboxes



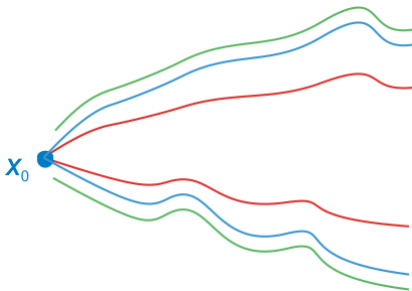
- Each box i only represents explicitly dimensions already split, $k = 0, \dots, K_i - 1$
- Box i has value $v(i) = \sum_{k=0}^{K_i-1} \gamma^k r_{i,k+1}$, rewards of center sequence

Lipschitz value function

- For any two **action sequences** $\mathbf{u}_\infty, \mathbf{u}'_\infty$:

$$|v(\mathbf{u}_\infty) - v(\mathbf{u}'_\infty)| \leq \frac{L_\rho}{1 - \gamma L_f} \sum_{k=0}^{\infty} \gamma^k |u_k - u'_k|$$

- Intuition: **states** (and so **rewards**) may diverge somewhat, but divergence controlled due to $\gamma L_f < 1$

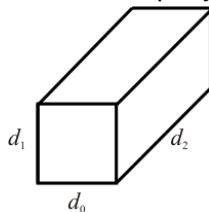


Box upper bound

- For any sequence \mathbf{u}_∞ in box i :

$$v(\mathbf{u}_\infty) \leq v(i) + \frac{\max\{1, L_\rho\}}{1 - \gamma L_f} \sum_{k=0}^{\infty} \gamma^k d_{i,k} := b(i)$$

- $d_{i,k}$ length of dimension k , 1 if not split yet



- $b(i)$ **b-value** of box i

Diameter and dimension selection

- **Diameter** $\delta(i) := \frac{\max\{1, L_\rho\}}{1-\gamma L_f} \sum_{k=0}^{\infty} \gamma^k d_{i,k}$
= uncertainty on values in the box
 - **Impact** of dimension k on uncertainty is $\gamma^k d_{i,k}$
- ⇒ when splitting a box, choose dimension with largest impact, to reduce uncertainty the most
- Always split into odd $M > 1/\gamma$ pieces



OPC algorithm

initialize tree with root box U^∞

while n not exhausted **do**

 select **optimistic** leaf box $i^\dagger = \arg \max_{i \in \mathcal{L}} b(i)$

 select **max-impact** dimension $k^\dagger = \arg \max_k \gamma^k d_{i^\dagger, k}$

 split i^\dagger along k^\dagger , creating M children on the tree

end while

return best center sequence seen, $i^* = \arg \max_i v(i)$



- 1 Introduction
- 2 OPC algorithm
- 3 Near-optimality analysis**
- 4 Experiments
- 5 Conclusions



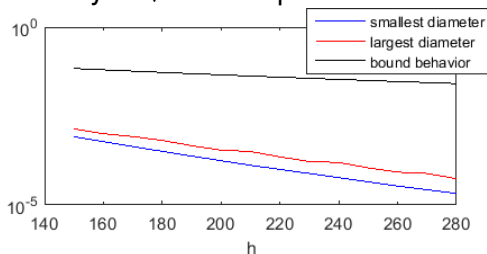
Diameter bound

Lemma

Given depth in tree h = total number of splits:

$$\delta(i) = \tilde{O}\left(\gamma \sqrt{2h \frac{\tau-1}{\tau^2}}\right), \text{ where } \tau = \left\lceil \frac{\log 1/M}{\log \gamma} \right\rceil$$

Diameters vary by the order of splits, but they all converge to 0 roughly exponentially in \sqrt{h} . Example:



Complexity measure

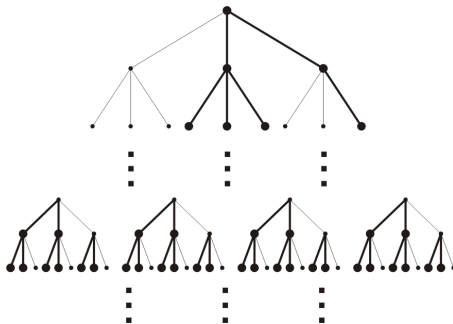
- OPC only expands in near-optimal subtree:

$$\mathcal{T}^* = \{i \in \mathcal{T} \mid v^* - v(i) \leq \delta(i)\}$$

(nodes that cannot be eliminated as suboptimal)

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

E.g. $m = 2, M = 3$:



Performance guarantee

Theorem

After spending n model calls, OPC suboptimality is:

$$v^* - v(i^*) = \begin{cases} \tilde{O}(\gamma \sqrt{\frac{2(\tau-1) \log n}{\tau^2 \log m}}), & \text{if } m > 1 \\ \tilde{O}(\gamma n^{1/4} b), & \text{if } m = 1 \end{cases}$$

- Convergence rate modulated by problem complexity m , faster when m smaller
- When $m = 1$, convergence is fast, with power $n^{1/4}$
- When $m > 1$, we pay for generality: exponential computation m^h to reach depth h



Related planning algorithms

- Inspired by **optimistic optimization (DOO & SOO)**
(Munos 2011)
- **HOLOP, HOOT**: fixed finite horizon
(Weinstein et al. 2012, Mansley et al. 2011)
- **Lipschitz planning (LP)**: different dimension selection, no guarantees
(Hren 2012)
- **SOOP**: no Lipschitz constants, no guarantees
(ADPRL 2013)



- 1 Introduction
- 2 OPC algorithm
- 3 Near-optimality analysis
- 4 Experiments**
- 5 Conclusions



Example: Quanser pendulum



System:

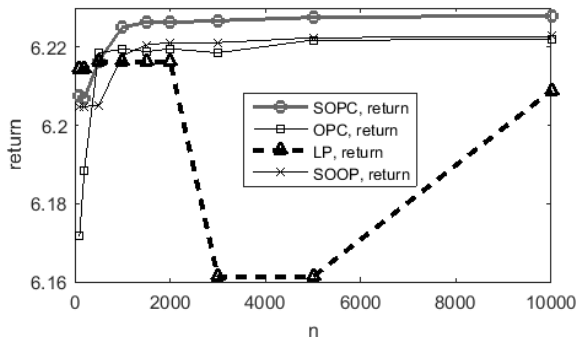
- $x =$ rod angle α , base angle θ , angular velocities
- $u =$ motor voltage $\in [-9, 9] \text{ V}$
- Sampling time $T_s = 0.05$

Goal: stabilize pointing up:

- $\rho = -\alpha^2 - \theta^2 - .005(\dot{\alpha}^2 + \dot{\theta}^2) - .05u^2$, normalized to $[0, 1]$
- Discount factor $\gamma = 0.85$
- Swingup required

OPC versus other planners

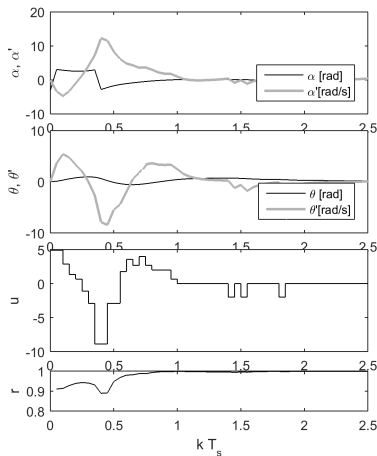
OPC with $L_f = L_\rho = 1.1$, tighter diameter formula
Parameters of algorithms optimized



⇒ OPC theoretically nice,
beaten by **simultaneous** algorithms in practice

Controlled trajectory (SOPC)

$n = 5000$ model calls; note adaptive discretization of control magnitude



Real-time control (SOPC)



Conclusions and next steps

Optimistic planning with continuous actions

- General nonlinear systems and cost functions
 - Guaranteed near-optimality and convergence rate
 - ... but beaten by simultaneous OPC in practice
- ⇒ Analyze SOPC
- + Eliminate assumption $\gamma L_f < 1$ using stability

Thank you!

