Action Selection for MDPs: Anytime AO* vs. UCT

Blai Bonet[1] and Hector Geffner[2]

[1]Universidad Simón Bolívar
[2]ICREA & Universitat Pompeu Fabra

AAAI, Toronto, Canada, July 2012

Online MDP Planning and UCT

Offline infinite-horizon MDP planning is unlikely to scale up to very large spaces

Online planning is more promising; it's just the selection of action to do in current state s

Selection can be done by solving **finite-horizon** version of MDP, rooted at s, with horizon H

Due to time constraints, such methods use **anytime optimal** finite-horizon MDP algorithms

UCT is one method, popular after success in Go [Gelly & Silver, 2007]

Why is UCT Successful?

UCT is a Monte-Carlo Tree Search method [Kocsis & Szepesvári, 2006]

Success of UCT is typically attributed to:

- adaptive Monte-Carlo sampling; i.e. Monte-Carlo simulations that become more and more focused as time goes by
- yet, RTDP [Barto et al., 1995] is also adaptive and anytime optimal, but not as popular or successful apparently
- another possible explanation is that UCT is anytime optimal with **arbitraty base policies**; RTDP needs **admissible heuristics**

Question: Can we develop a heuristic search algorithm for finite-horizon MDPs that is **anytime optimal** using **base policies** rather than **admissible heuristics**?

Anytime AO*

Anytime AO* is simple variant of AO* that is **anytime optimal** even with **non-admissible** heuristics, such as **rollouts** of base policies

Anytime A* [Hansen & Zhou, 2007] is variant of A* that is anytime optimal for OR graphs even with non-admissible heuristic

Main trick in Anytime A* is to **not stop** after first solution, but return best solution so far and continue search with nodes in OPEN

This trick doesn't work for AO*, but another one does:

- select **tip node to expand** that is **not** part of best partial solution graph with some probability (exploration)
- terminate when no tip is left to expand (in best partial graph or not)

Anytime AO* **seems competitive** with UCT in challenging tasks

Rest of the Talk

- MDPs: finite and infinite horizon, and action selection
- Finite-horizon MDPs as Acyclic AND/OR Graphs
- AO*
- UCT
- Anytime AO*
- Experiments
- Summary and Future Work

Markov Decision Processes

Fully observable, stochastic models, characterized by:

- state space S and actions A; $A(s)$ is set of applicable actions at s
- initial state s_0 and goal states G
- transition probabilities $P(s'|s,a)$ for every $s, s' \in S$ and $a \in A(s)$
- positive costs $c(s,a)$ for $s \in S$ and $a \in A(s)$, except goals where $P(s|s,a) = 1$ and $c(s,a) = 0$ for every $s \in G, a \in A$

Finite-Horizon MDP (FH-MDP) characterized by:

- same elements for MDPs
- time horizon H
- policies for FH-MDP are non-stationary (i.e. depend on time)

Action Selection in MDPs

Main Task: given state s and horizon H, select action to apply at s by only looking at most H steps into the future

- Given s and H, the MDP is converted into a Finite-Horizon MDP with initial state $s_0 = s$ and horizon H
- FH-MDP corresponds to an implicit AND/OR tree

FH-MDPs as Acyclic AND/OR Graphs

For initial state s_0 and lookahead H, implicit graph given by:

- root node is (s_0, H)
- terminal nodes are (s, d) where $d = 0$ or s is terminal in MDP
- children of non-terminal (s, d) are AND-nodes (a, s, d) for $a \in A(s)$
- children of (a, s, d) are nodes $(s', d-1)$ such that $P(s'|s, a) > 0$

Solutions are subgraphs T such that

- the root (s_0, H) belongs to T
- for each non-terminal (s, d) in T, **exactly** one child (a, s, d) is in T
- for each AND-node (a, s, d), **all** its children $(s', d-1)$ belong to T

The cost of T is computed by **backward induction**, propagating the values at the leaves upwards to the root which gives the cost of T

Best Lookahead Action

Definition

Given state s_0 and lookahead H, a **best action** *for s (wrt H) is the action that leads to the unique child of the root (s_0, H) in an* **optimal solution** *T^* of the* **implicit** *AND/OR graph*

Thus, need to solve the implicit AND/OR graph:

1. AO* [Nilsson, 1980]
2. UCT [Kocsis & Szepesvári, 2006]
3. Anytime AO*

AO* for Implicit AND/OR Graphs

AO* **explicates** implicit graph incrementally, one node at a time:

- G is **explicit graph**, initially contains just root node
- G^* is **best partial solution graph**:
  - G^* is optimal solution of G on the assumption that tips of G are terminal nodes whose value is given by heuristic h

Algorithm

1. Initially, $G = G^*$ and consists only of root node
2. Iteratively, a non-terminal leaf is **selected** from G^*:
   - the leaf is expanded
   - values of the children are set with $h(\cdot)$,
   - values are propagated upwards while recomputing G^*
3. Terminate as soon as G^* becomes a **complete graph**; i.e., it has no non-terminal leaves

UCT

UCT also maintains **explicit graph** G that expands incrementally

But, node selection procedure follows **path in explicit graph** with **UCB criteria** which balances exploration and exploitation, sampling next state after an action stochastically

First node generated that is not in explicit graph G, added to graph with value obtained by **rollout of best policy** from node

Values propagated upwards in G by **Monte-Carlo updates** (averages), rather than DP updates as in AO* or RTDP

No termination conditition

Anytime AO*

Two small changes to AO* algorithm for:

1. handling non-admissible heuristics
2. handling random (sampled) heuristics as rollouts of base policies

First change:

- select with prob. p non-terminal tip node **IN** best partial graph G^*; else, select non-terminal tip in explicit graph G that is **OUT** of G^*
- Anytime AO* **terminates** when no such tip exists in either graph

Second change:

- when using random heuristics, such as rollouts, **re-sample** $h(s, d)$ value every time that the value of tip (s, d) is needed to make a **DP update**, and use **average over sampled values**

Anytime AO*: Properties

Theorem (Optimality)

Given enough time, Anytime AO selects best action* **independently of admissibility** *of heuristic because it terminates when the implicit AND/OR tree has been* **fully explicated**

Theorem (Complexity)

The complexity of Anytime AO is no worse than the complexity of AO* because in the worst case, AO* expands (explicates) the whole implicit tree*

Choice of Tip Nodes

Intuition: select tip that has **biggest potential** to cause a change in best partial graph

Discriminant: $\Delta(n) =$ "change in the value of n for **causing** a change in best partial graph"

Theorem

$\Delta(n)$ *can be computed for every node by a complete graph traversal on* G *(see paper for details)*

Choose tip n that **minimizes** $|\Delta(n)|$: tips in IN have positive Δ-value; tips in OUT have negative Δ-value

Anytime AO* with this choice of tips is called **AOT**

Experiments

Experiments over several domains, comparing:

- UCT
- AOT with base policies and heuristics
- RTDP

Domains:

- Canadian Traveller Problem (CTP)
  - compared w/ **state-of-the-art domain-specific** UCT
  - compared w/ own implementation of UCT and RTDP
- Sailing and Racetracks
  - compared w/ own implementation of UCT
  - compared w/ own implementation of RTDP

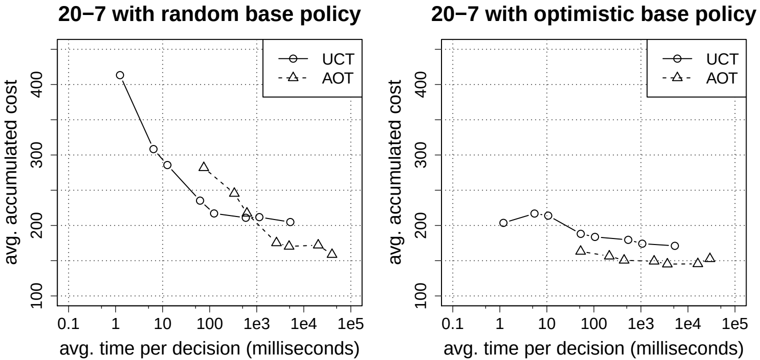
Focus: quality vs. average time per decision (ATD)

CTP: AOT vs. State-of-the-Art UCT

		br. factor		UCT–CTP		optimistic base policy		
prob.	P(bad)	avg	max	UCTB	UCTO	direct	UCT	AOT
20-1	17.9	13.5	128	210.7 ± 7	169.0 ± 6	191.8 ± 0	180.7 ± 3	$\mathbf{163.8 \pm 2}$
20-2	9.5	15.7	64	176.4 ± 4	$\mathbf{148.9 \pm 3}$	202.7 ± 0	160.8 ± 2	156.4 ± 1
20-3	14.3	15.2	128	150.7 ± 7	$\mathbf{132.5 \pm 6}$	142.1 ± 0	144.3 ± 3	133.8 ± 2
20-4	78.6	11.4	64	264.8 ± 9	235.2 ± 7	267.9 ± 0	238.3 ± 3	$\mathbf{233.4 \pm 3}$
20-5	20.4	15.0	64	123.2 ± 7	111.3 ± 5	163.1 ± 0	123.9 ± 3	$\mathbf{109.4 \pm 2}$
20-6	14.4	13.9	64	165.4 ± 6	$\mathbf{133.1 \pm 3}$	193.5 ± 0	167.8 ± 2	135.5 ± 1
20-7	8.4	14.3	128	191.6 ± 6	148.2 ± 4	171.3 ± 0	174.1 ± 2	$\mathbf{145.1 \pm 1}$
20-8	23.3	15.0	64	160.1 ± 7	$\mathbf{134.5 \pm 5}$	167.9 ± 0	152.3 ± 3	135.9 ± 2
20-9	33.0	14.6	128	235.2 ± 6	173.9 ± 4	212.8 ± 0	185.2 ± 2	$\mathbf{173.3 \pm 1}$
20-10	12.1	15.3	64	180.8 ± 7	167.0 ± 5	173.2 ± 0	178.5 ± 3	$\mathbf{166.4 \pm 2}$
Total				1858.9	1553.6	1886.3	1705.9	**1553.0**

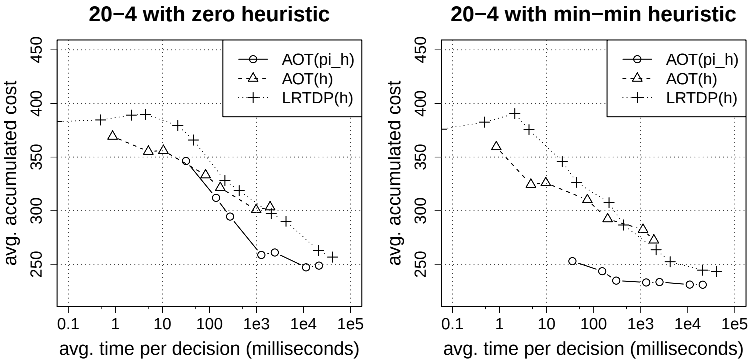
- data for UCT–CTP taken from [Eyerich, Keller & Helmert, 2010]
- each figure is average over 1,000 runs
- UCT run for 10,000 iterations, AOT for 1,000 iterations

CTP: Quality Profile



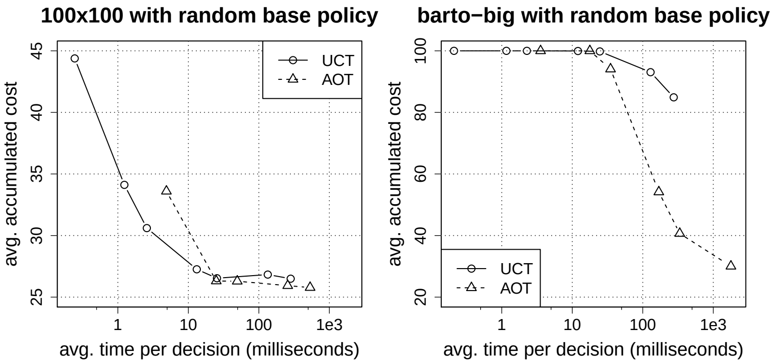
- each point is average over 1,000 runs
- UCT iterations: 10, 50, 100, 500, 1K, 5K, 10K and 50K
- AOT iterations: 10, 50, 100, 500, 1K, 5K and 10K
- ATD calculated globally: total time / total # decisions

CTP: Heuristics vs. Policies vs. RTDP



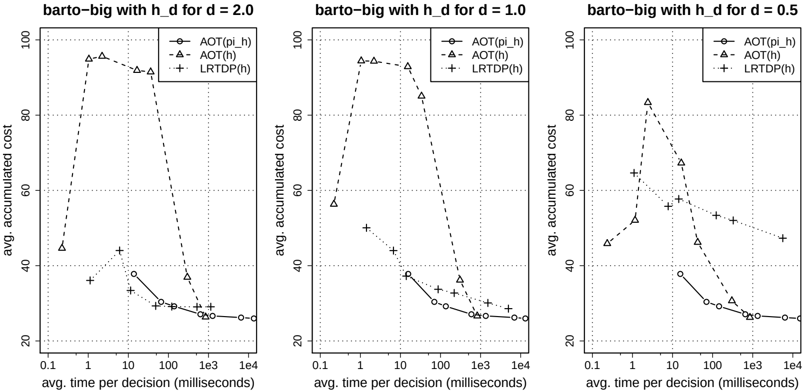
- two heuristics: zero and min-min, and policies greedy wrt heuristic
- algorithms: AOT(h), AOT(π_h), LRTDP(h)
- each figure is average over 1,000 runs

Sailing and Racetracks: Quality Profile



- each point is average over 1,000 runs
- UCT iterations: 10, 50, 100, 500, 1K, 5K and 10K
- AOT iterations: 10, 50, 100, 500, 1K, and 5K

Racetracks: Heuristics vs. Policies vs. RTDP



- heuristics: $h = d \times h_{\min}$ for $d = 2, 1$, and 0.5
- algorithms: AOT(h), AOT(π_h), LRTDP(h)
- each figure is average over 1,000 runs

Summary and Future Work

- UCT success seems to follow from combination of non-exhaustive search methods with ability to use informed base policies

- Anytime AO*, aimed at capturing both of these features in standard heuristic search **model-based** framework, compares well with UCT

- Results help to bridge the gap between MCTS methods and anytime heuristic search methods

- RTDP does better than expected in these domains;
  [see AAAI-12 paper by Kolobov, Mausam & Weld]