MAXQ-OP Based Hierarchical Online Planning

Aijun Bai, Feng Wu and Xiaoping Chen

School of Compute Science and Technology,
University of Science and Technology of China

Apr 11, 2013

Outline

1. Introduction
2. Background
3. MAXQ-OP Framework
4. Experiment: The Taxi Domain
5. Case Study: The RoboCup 2D Domain
6. Conclusions

Our Work

- A MAXQ-OP [1] approach to hierarchical planning in large stochastic domains
- Key contributions:
  - Overall framework for exploiting the MAXQ hierarchies online
  - Approximation methods for computing the *completion function*

MDP Framework

- An expressive model for planning under uncertainty
- 4-tuple $< S, A, T, R >$:
  - State space: $S = \left\{s_1, s_2, \cdots, s_{|S|}\right\}$
  - Action space: $A = \left\{a_1, a_2, \cdots, a_{|A|}\right\}$
  - Transition function: $T(s'|s, a) \to [0, 1]$
  - Reward function: $R(s, a) \to \mathbf{R}$

MDP Framework (Cont.)

- Policy: $\pi(s) \rightarrow A$
- Value Function: $V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, a, s') V^{\pi}(s')$
- Optimal Policy: π^* with highest value for each state
- Solving an MDP equals finding the optimal policy
- **Concentrate on undiscounted and goal-directed MDPs**
    - $\gamma = 1$
    - Stochastic shortest path problems

MAXQ Hierarchical Decomposition

- Decompose a given MDP into a set of sub-MDPs [3]
  - $M = \{M_0, M_1, \cdots, M_n\}$
  - $M_i = \{T_i, A_i, R_i\}$
    - Terminate predicate T_i - give active states and subgoals
    - Available actions A_i - primitive or macro actions
    - Pseudo-reward function R_i - optional local version of rewards
  - Solving M_0 solves the original MDP M

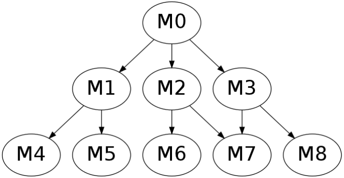

Figure 1: MAXQ task graph

MAXQ Hierarchical Decomposition (Cont.)

- Hierarchical policy
  - $\pi = \{\pi_0, \pi_1, \cdots, \pi_n\}$
    - An assignment of policies to each individual subtask
  - Exist a *Recursively optimal policy* π^*
    - Each subtask is optimal given the policies of its descendants
    - Reach a kind of local optimality
  - **MAXQ-OP approximately finds π^* online in real-time!**

Recursively Optimal Policy

- Value function V^* of π^* satisfies

$$V^*(i,s) = \begin{cases} R(s,i) & \text{if } M_i \text{ is primitive} \\ \max_{a\in A_i} Q^*(i,s,a) & \text{otherwise} \end{cases} \tag{1}$$

$$Q^*(i,s,a) = V^*(a,s) + C^*(i,s,a) \tag{2}$$

$$C^*(i,s,a) = \sum_{s',N} \gamma^N P(s',N|s,a) V^*(i,s') \tag{3}$$

- π^* satisfies

$$\pi_i^*(s) = \operatorname*{argmax}_{a\in A_i} Q^*(i,s,a) \tag{4}$$

Completion Function Approximation

- Completion function

$$C^*(i,s,a) = \sum_{s',N} \gamma^N P(s',N|s,a) V^*(i,s') \tag{5}$$

$$\begin{aligned} P(s',N|s,a) \quad &= \textstyle\sum_{\langle s,s_1,\ldots,s_{N-1}\rangle} P(s_1|s,\pi_a^*(s)) \cdot P(s_2|s_1,\pi_a^*(s_1)) \\ &\qquad\cdots P(s'|s_{N-1},\pi_a^*(s_{N-1})). \end{aligned} \tag{6}$$

- $\langle s, s_1, \ldots, s_{N-1}\rangle$ is a path from s to s' by following π^*
- Can be completely solved offline by exhausted full searches
  - Inapplicable for large domains
  - Intractable for online algorithms

Completion Function Approximation (Cont.)

- Recall that $\gamma = 1$ in our settings
- Introduce terminating distribution

$$P(s'|s,a) = \sum_{N} P(s', N|s,a) \tag{7}$$

- Rewrite complete function as

$$C^*(i,s,a) = \sum_{s'} P(s'|s,a) V^*(i,s') \tag{8}$$

- Use a prior distribution $D_i(s'|s,a)$ to approximate $P(s'|s,a)$
- Draw states from $D_i(s'|s,a)$ by *importance sampling* [4]

$$C^*(i,s,a) \approx \frac{1}{|\tilde{G}_a|} \sum_{s' \in \tilde{G}_a} V^*(i,s') \tag{9}$$

Main Structure of MAXQ-OP

- For non-primitive subtasks

$$V^*(i,s) \approx \max_{a \in A_i}\{V^*(a,s) + \frac{1}{|\tilde{G}_a|}\sum_{s' \in \tilde{G}_a} V^*(i,s')\} \tag{10}$$

- Introduce search depth array d, maximal search depth array D and heuristic evaluation functions $H(i,s)$

$$V^*(i,s,d) \approx \begin{cases} H(i,s) & \text{if } d[i] \geq D[i] \\ \max_{a \in A_i}\{V^*(a,s,d) + \\ \frac{1}{|\tilde{G}_a|}\sum_{s' \in \tilde{G}_a} V^*(i,s',d[i] \leftarrow d[i]+1)\} & \text{otherwise} \end{cases} \tag{11}$$

- The main structure of MAXQ-OP

Comparing to Traditional Online Search Algorithms

- Traditional online search algorithms
  - Search only in state space
  - Search path:

$$R(s_1, a_1) + R(s_2, a_2) + \cdots + R(s_{n-1}, a_{n-1}) + H(s_n) \tag{12}$$

- MAXQ-OP algorithm
  - Search both in task hierarchy and state space
  - Search path:

$$V(s_1, t_1) + V(s_2, t_2) + \cdots + V(s_n, t_n), \tag{13}$$

  where

$$V(s, t) = R(s, a) + R(s', a') + \cdots + R(s'^{\cdots\prime}, a'^{\cdots\prime}) + H(t, s'^{\cdots\prime\prime}) \tag{14}$$

- Intuitively, MAXQ-OP can search much deeper given appropriate heuristic evaluations over the task hierarchy

The Taxi Domain

- States: $25 \times 5 \times 4 = 400$
  - Taxi location: (x, y)
  - Passenger location: R, Y, B, G and In
  - Destination location: R, Y, B, G
- Actions: 6
  - North, South, East, West
  - Pickup, Putdown

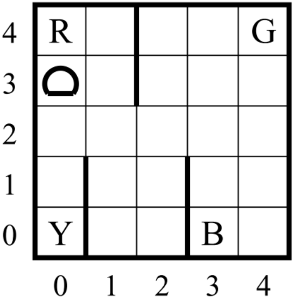

Figure 2: Taxi domain

Empirical Results

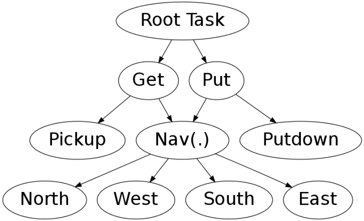

Figure 3: Task graph for Taxi

Table 1: Empirical results in the Taxi domain

Algorithm	**Trials**	**Average Rewards***	**Offline Time**	**Online Time**
MAXQ-OP	1000	3.93 ± 0.16	-	0.20 ± 0.16 ms
R-MAXQ	100	3.25 ± 0.50	1200 ± 50 episodes	-
MAXQ-Q	100	0.0 ± 0.50	1600 episodes	-

* The upper bound of Average Rewards is 4.01 ± 0.15 averaged over 1000 trials.

The RoboCup 2D Domain

- Key feature: Abstraction
- Key challenges:
  - Fully distributed
  - Multi-agent
  - Stochastic
  - Continuous:
    - State space
    - Action space
    - Observation space

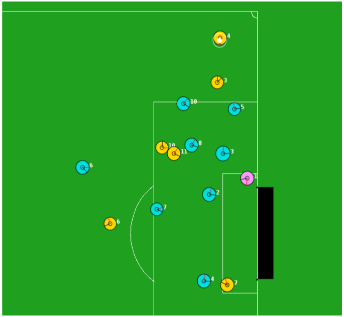

Figure 4: RoboCup 2D

MAXQ Task Graph in WrightEagle

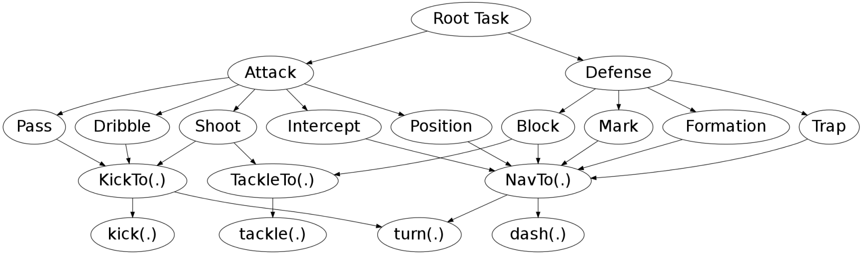

Figure 5: Task graph in WrightEagle

Implementation Details

- Some necessary pre-defined components
  - Prior terminating distributions
  - Heuristic search methods
  - Heuristic evaluation functions
- Provide a decision-theoretical based principled solution to automated planning in the RoboCup 2D domain [2]

Team Performance

- RoboCup annual competitions: Has been keeping in top-2 places (3 champions and 5 runners-up) since 2005
- Key advantage of MAXQ-OP: provide a formal framework for conducting the search process over task hierarchies

Conclusions

- MAXQ-OP: a principled solution to automated planning in large stochastic domains
  - Online planning
  - Hierarchical decomposition
  - Heuristic and approximation techniques
- Can find a near-optimal policy online in the Taxi domain
- Continuously developed in WrightEagle, reaching outstanding performances in RoboCup competitions
- Demonstrate the soundness and stability of MAXQ-OP for solving large MDPs given pre-defined task hierarchies

References

A. Bai, F. Wu, and X. Chen.
Online planning for large MDPs with MAXQ decomposition.
In *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems*, Valencia, Spain, June 2012.

A. Bai, F. Wu, and X. Chen.
Towards a principled solution to simulated robot soccer.
In X. Chen, P. Stone, L. E. Sucar, and T. V. der Zant, editors, *RoboCup-2012: Robot Soccer World Cup XVI*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, 2013.

T. G. Dietterich.
Hierarchical reinforcement learning with the MAXQ value function decomposition.
Journal of Machine Learning Research, 13(1):63, May 1999.

S. Thrun, D. Fox, W. Burgard, and F. Dellaert.
Robust monte carlo localization for mobile robots.
Artificial intelligence, 128(1-2):99–141, 2001.