Towards a Principled Solution to Simulated Robot Soccer

Aijun Bai, Feng Wu and Xiaoping Chen

WrightEagle 2D Soccer Simulation Team,
University of Science and Technology of China

Jun 24, 2012

Outline

The RoboCup 2D Domain

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

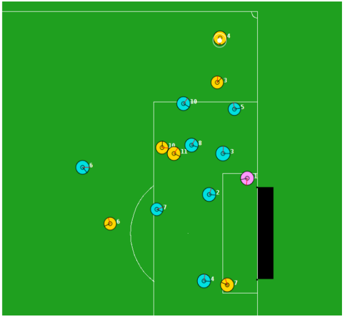

Figure: RoboCup 2D

Our Work

- A MAXQ-OP [1] approach to automated planning in the RoboCup 2D domain
- Key contributions:
  - Overall framework for exploiting the MAXQ hierarchies online
  - Approximation methods for computing the *completion function*
- WrightEagle - 3 champions and 5 runners-up since 2005

MDP Framework

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

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 goal-directed MDPs**
  - $\gamma = 1$
  - Stochastic shortest path problems

MAXQ Hierarchical Decomposition

- Decompose a given MDP into a set of sub-MDPs [2]
  - $M = \{M_0, M_1, \cdots, M_n\}$
  - $M_i = \{T_i, A_i, R_i\}$
    - 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
- Hierarchical policy
  - $\pi = \{\pi_0, \pi_1, \cdots, \pi_n\}$
  - *Recursively optimal policy* π^*
  - **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^*
- Inapplicable for large domains
- Intractable for online algorithms

Completion Function Approximation (Cont.)

- Recall that $\gamma = 1$ in our settings
- A distribution over terminal states

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

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

- 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* [3]

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

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 depth array d and heuristic evaluation function $H(i, s)$

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

- The main structure of MAXQ-OP

Comparing to Traditional Online Search Algorithms

- Traditional Online Search Algorithms
  - Search only in state space
  - Search path:
    $R(s_1, a_1) + R(s_2, a_2) + \cdots + R(s_{n-1}, a_{n-1}) + H(s_n)$
- MAXQ-OP Algorithm
  - Search in task hierarchy and state space
  - Search path: $V(s_1, t_1) + V(s_2, t_2) + \cdots + V(s_n, t_n)$, where
    $V(s, t) = R(s, a) + R(s', a') + \cdots + H(t, s'^{\cdots\prime})$
- Intuitively, MAXQ-OP jumps faster in state space

MAXQ Task Graph in WrightEagle

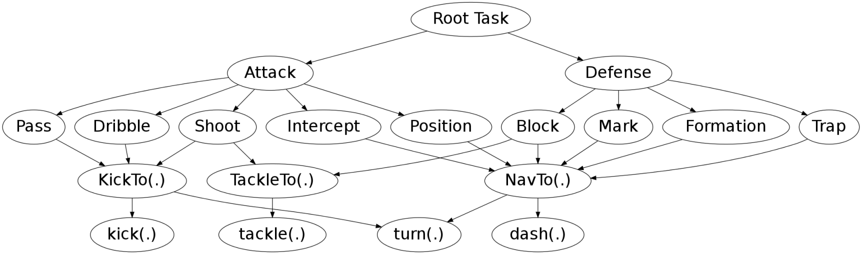

Figure: MAXQ task graph in WrightEagle

Implementation Details

- Involves some pre-defined components
  - Prior terminating distribution
  - Heuristic search methods
  - Heuristic evaluation function
- Gives a decision-theoretical based principled solution to automated planning in RoboCup 2D

Fixed Scene Test and Full Game Test

- Outperforms HAND-CODED and RANDOM algorithms
- Outperforms 4 best 2D teams: BrainStormers08, Helios10, Helios11 and Oxsy11
- Key advantage of MAXQ-OP: provide a formal framework for conducting the search process over task hierarchy

Conclusion

- MAXQ-OP: a principled solution to large stochastic domains
  - MAXQ hierarchical decomposition
  - Online planning methods
- Continuously developed in WrightEagle, reaching outstanding performances in RoboCup competitions
- Showing its potential of being a principled solution to Simulated Robot Soccer

References

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

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.