An MDP-Based Winning Approach to RoboCup Soccer Simulation Challenge

Aijun Bai

Aug 16, 2016

UC Berkeley

Outline

The Problem

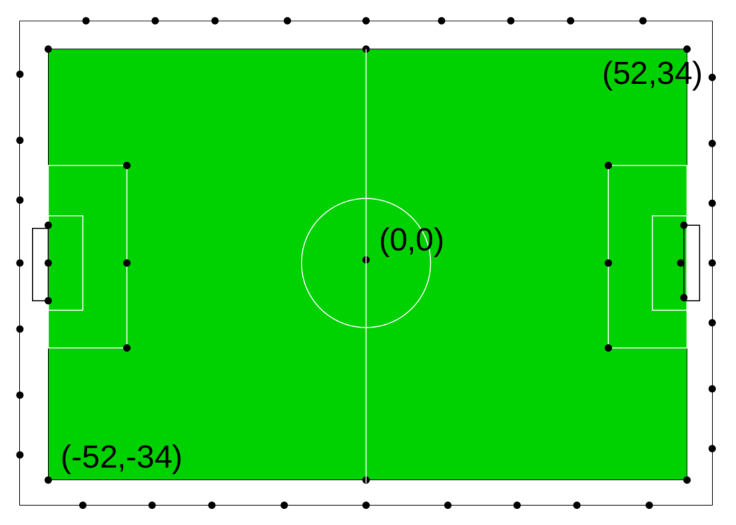
RoboCup Soccer Simulation 2D

- Simulated soccer game
- Server/Client fashion
  - Server: the simulated environment
  - Clients: 11 players and one coach for each team
- In each cycle (100 ms)
  - Server sends local observations to each client
  - Clients receive observations, update internal world models and send actions to the server
- Around 6,000 cycles ($\approx$ 10 mins)

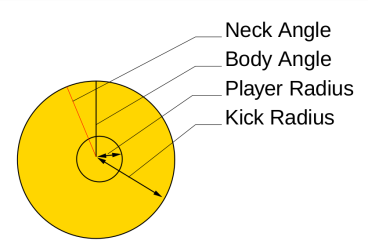
What Makes RoboCup 2D Interesting/Challenging?

- Key Features:
  - Abstractions made by the simulator
  - High-level planning, learning and cooperation
  - No need to handle robot hardware issues
- Key Challenges:
  - Fully distributed multi-agent stochastic system
  - Continuous state, observation and action spaces

The Soccer Field

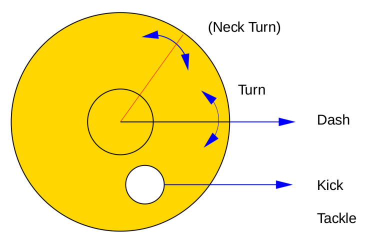


Player and Ball States



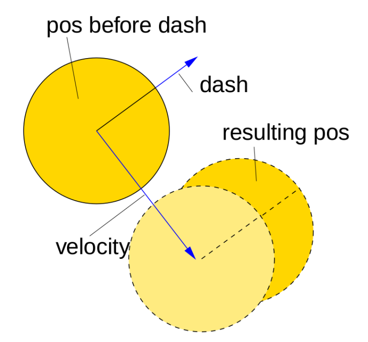
- Player
  - Position, Velocity, Body Angle, Neck Angle, Stamina, . . .
  - Maximal Speed, Kick Radius, Stamina Recovery, . . .
- Ball
  - Position, Velocity

Primitive Actions



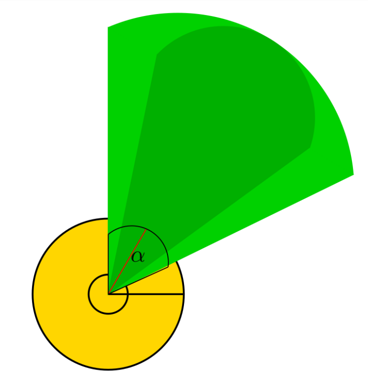
- Parameterized actions
  - $Dash(dir, power)$
  - $TurnBody(angle)$
  - $TurnNeck(angle)$
  - $Kick(dir, power)$
  - $Tackle(dir)$
  - $Catch(dir)$ [for goalie]

The Physics



- $Dash(dir, power)$
  - Moves the player
  - Exposed to noise
  - Costs some stamina
    - If stamina is too low: can not move at full speed

The View Model



- Relative noisy information
- Limited view angle
- Sensitive view distance

RoboCup 2D in Action

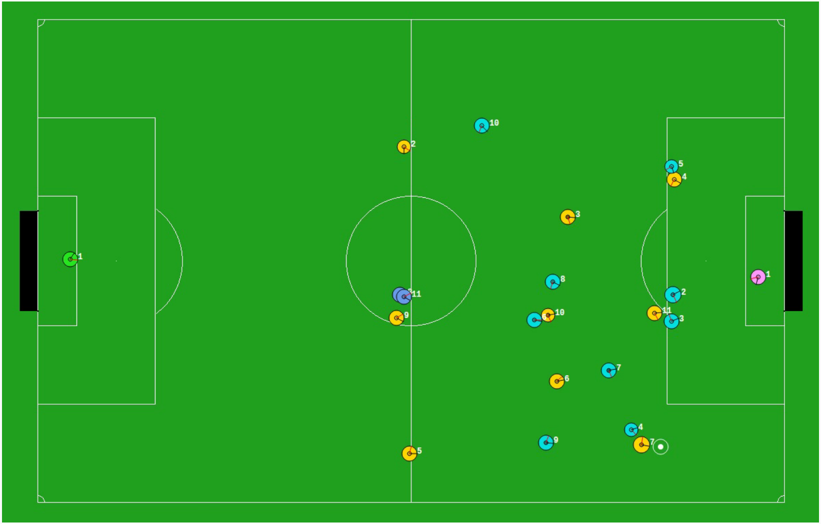


Figure 1: WrightEagle (my team) v.s. Helios (from Japan)

The RoboCup 2D Competition

- Earliest league since 1997
- 20 teams per year from different countries/universities
- Two rounds of group tournament, followed by an elimination
- More information: `https://en.wikipedia.org/wiki/RoboCup_2D_Soccer_Simulation_League`

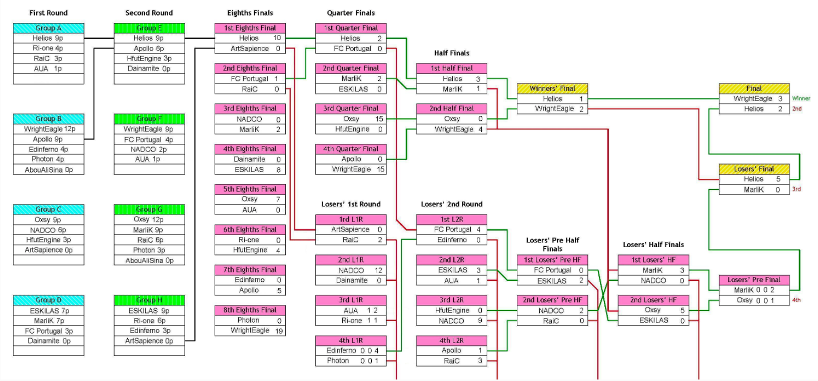
A Running Competition of RoboCup 2013

Our Achievements

WrightEagle team (from Univ. of Sci. & Tech. of China):

- 6 world champions: 2006, 2009, 2011, 2013, 2014 and 2015
- A little bit about myself:
  - Got Bachelor and PhD both in CS from USTC
  - Have been working on RoboCup 2D since 2005
  - Have been the main contributor since 2009
- More information: `http://www.wrighteagle.org/2d/`

The Path to Champion of RoboCup 2011



The Approach

The Winning Approach

Key components of WrightEagle:

- Markov decision process (MDP) formulation
- Belief state update (Bai et al., 2012a,c)
- Hierarchical decomposition (Bai et al., 2012a,b, 2013b, 2015)
- State abstraction (Bai et al., 2016)
- Monte-Carlo simulation (Bai et al., 2013a, 2014)
- Rationality assumption

Markov Decision Processes

- MDP models uncertainty:
  1. State space: $S = \left\{s_1, s_2, \ldots, s_{|S|}\right\}$
  2. Action space: $A = \left\{a_1, a_2, \ldots, a_{|A|}\right\}$
  3. Transition function: $T(s' \mid s, a) \to [0, 1]$
  4. Reward function: $R(s, a) \to \mathbb{R}$
- Policy: $\pi : S \to A$
- Value function: $V^\pi(s_0) = \mathbb{E}\left[\sum_{t \geq 0} \gamma^t R(s_i, \pi(s_i))\right]$
- Bellman optimality:

$$V^*(s) = \max_{a \in A} \left\{ R(s, a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^*(s') \right\} \tag{1}$$

- Optimal policy:

$$\pi^*(s) = \operatorname*{argmax}_{a \in A} V^*(s) \tag{2}$$

Partially Observable MDPs

- POMDP extends MDP to partially observable domains:
    1. Observation space: $O = \left\{o_1, o_2, \ldots, o_{|O|}\right\}$
    2. Observation function: $\Omega(o \mid a, s) \rightarrow [0, 1]$
- History: $h = (a_0, o_1, a_1, o_2, \ldots a_{t-1}, o_t)$
- Belief state: $b(s) = \Pr(s \mid b_0, h)$
- Belief space: $\mathcal{B} = \{b\}$
- Policy: $\pi : \mathcal{B} \rightarrow A$

Basic Framework

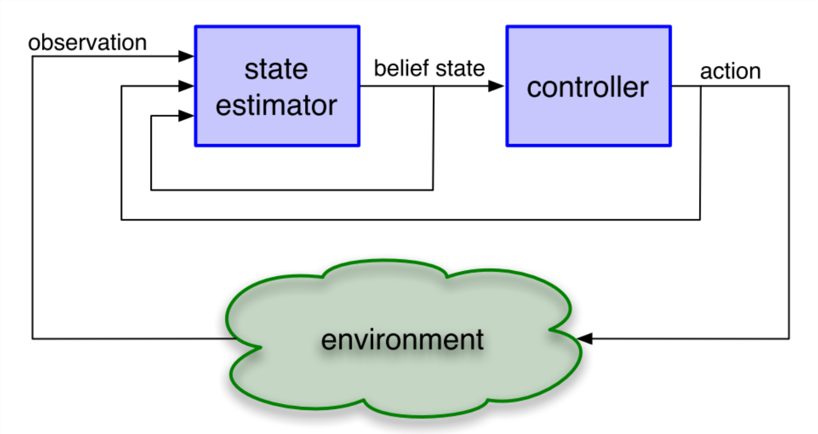


Figure 2: Agent & environment

Belief Update via Particle Filtering

- Particle filter based self-localization and multi-object tracking
- Use expected state to estimate the world state, consistent with an MDP formulation

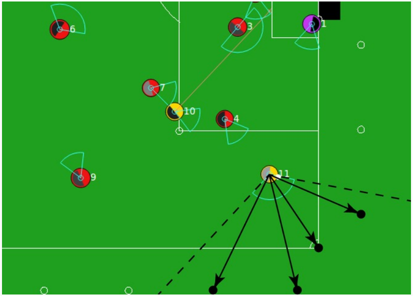


Figure 3: Localization

Belief State Visualization

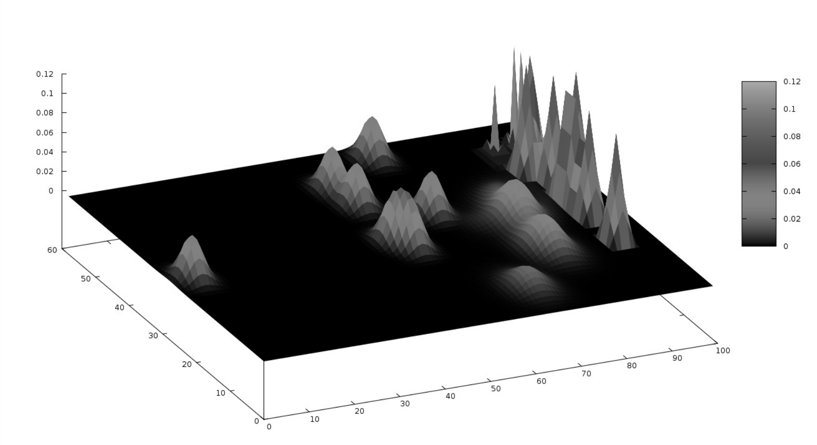


Figure 4: Belief state in terms of player position distributions

Hierarchical Online Planning

- Rule-based system:

  PlanAttack() {

  ...

  if *should_shoot* **then**

    ⌊ **return** PlanShoot()

  else if *should_pass* **then**

    ⌊ **return** PlanPass()

  else

    ⌊ **return** PlanDrrible()

  ...

  }

- Hierarchical planning:

  PlanAttack() {

  ...

  shoot $\leftarrow$ PlanShoot()

  pass $\leftarrow$ PlanPass()

  dribble $\leftarrow$ PlanDrrible()

  ...

  return $\max\{$shoot, pass, dribble, $\ldots\}$

  }

Hierarchical Decomposition

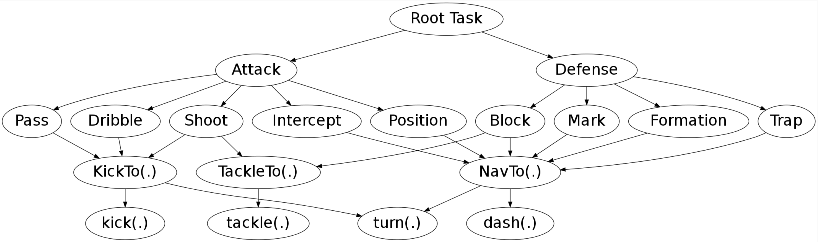


Figure 5: MAXQ based hierarchical decomposition in WrightEagle

MAXQ Value Function Decomposition

- Value function V^* of π^* satisfies

$$V^*(i,s) = \begin{cases} R(s,i) & \text{if } M_i \text{ is primitive} \\ \max_{a \in A_i} Q^*(i,s,a) & \text{otherwise} \end{cases} \tag{3}$$

$$Q^*(i,s,a) = V^*(a,s) + C^*(i,s,a) \tag{4}$$

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

- π^* satisfies

$$\pi_i^*(s) = \operatorname*{argmax}_{a \in A_i} Q^*(i,s,a) \tag{6}$$

Value Function Decomposition in WrightEagle

$$Q^*(\text{Root}, \boldsymbol{s}, \text{Attack}) = V^*(\text{Attack}, \boldsymbol{s}) + \sum_{s'} P_t(s' \mid \boldsymbol{s}, \text{Attack}) V^*(\text{Root}, \boldsymbol{s}'), \tag{7}$$

$$V^*(\text{Root}, \boldsymbol{s}) = \max\{Q^*(\text{Root}, \boldsymbol{s}, \text{Attack}), Q^*(\text{Root}, \boldsymbol{s}, \text{Defense})\}, \tag{8}$$

$$\begin{aligned} V^*(\text{Attack}, \boldsymbol{s}) = \max\{&Q^*(\text{Attack}, \boldsymbol{s}, \text{Pass}), Q^*(\text{Attack}, \boldsymbol{s}, \text{Dribble}), Q^*(\text{Attack}, \boldsymbol{s}, \text{Shoot}), \\ &Q^*(\text{Attack}, \boldsymbol{s}, \text{Intercept}), Q^*(\text{Attack}, \boldsymbol{s}, \text{Position})\}, \end{aligned} \tag{9}$$

$$Q^*(\text{Attack}, \boldsymbol{s}, \text{Pass}) = V^*(\text{Pass}, \boldsymbol{s}) + \sum_{s'} P_t(s' \mid \boldsymbol{s}, \text{Pass}) V^*(\text{Attack}, \boldsymbol{s}'), \tag{10}$$

$$Q^*(\text{Attack}, \boldsymbol{s}, \text{Intercept}) = V^*(\text{Intercept}, \boldsymbol{s}) + \sum_{s'} P_t(s' \mid \boldsymbol{s}, \text{Intercept}) V^*(\text{Attack}, \boldsymbol{s}'), \tag{11}$$

$$V^*(\text{Pass}, \boldsymbol{s}) = \max_{\text{position } p} Q^*(\text{Pass}, \boldsymbol{s}, \text{KickTo}(p)), \tag{12}$$

$$V^*(\text{Intercept}, \boldsymbol{s}) = \max_{\text{position } p} Q^*(\text{Intercept}, \boldsymbol{s}, \text{NavTo}(p)), \tag{13}$$

$$Q^*(\text{Pass}, \boldsymbol{s}, \text{KickTo}(p)) = V^*(\text{KickTo}(p), \boldsymbol{s}) + \sum_{s'} P_t(s' \mid \boldsymbol{s}, \text{KickTo}(p)) V^*(\text{Pass}, \boldsymbol{s}'), \tag{14}$$

$$Q^*(\text{Intercept}, \boldsymbol{s}, \text{NavTo}(p)) = V^*(\text{NavTo}(p), \boldsymbol{s}) + \sum_{s'} P_t(s' \mid \boldsymbol{s}, \text{NavTo}(p)) V^*(\text{Intercept}, \boldsymbol{s}'), \tag{15}$$

$$V^*(\text{KickTo}(p), \boldsymbol{s}) = \max_{\text{power } a,\ \text{angle } \theta} Q^*(\text{KickTo}(p), \boldsymbol{s}, \text{kick}(a, \theta)), \tag{16}$$

$$V^*(\text{NavTo}(p), \boldsymbol{s}) = \max_{\text{power } a,\ \text{angle } \theta} Q^*(\text{NavTo}(p), \boldsymbol{s}, \text{dash}(a, \theta)), \tag{17}$$

$$Q^*(\text{KickTo}(p), \boldsymbol{s}, \text{kick}(a, \theta)) = R(s, \text{kick}(a, \theta)) + \sum_{s'} P_t(s' \mid \boldsymbol{s}, \text{kick}(a, \theta)) V^*(\text{KickTo}(p), \boldsymbol{s}'), \tag{18}$$

MAXQ based Online Planning: MAXQ-OP

- Approximate $\Pr(s', N \mid s, a)$ either online or offline
- For non-primitive subtasks

$$V^*(i,s) \approx \max_{a \in A_i} \left\{ V^*(a,s) + \sum_{s'} \Pr(s' \mid s,a) V^*(i,s') \right\} \quad (19)$$

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

$$V(i,s,d) \approx \begin{cases} H(i,s) & \text{if } d[i] \geq D[i] \\ \max_{a \in A_i}\{V(a,s,d)+ & \\ \sum_{s'} \Pr(s' \mid s,a) V(i,s',d[i] \leftarrow d[i]+1)\} & \text{otherwise} \end{cases} \quad (20)$$

- Call $V(0,s,[0,0,\dots,0])$ to find the value of s in task M_0

MAXQ-OP in WrightEagle

- Task evaluation over hierarchy
  - Value function decomposition
- Terminating distribution approximation
  - Success and failure probabilities
- Search based (Monte Carlo) planning with pruning
- Heuristic evaluation

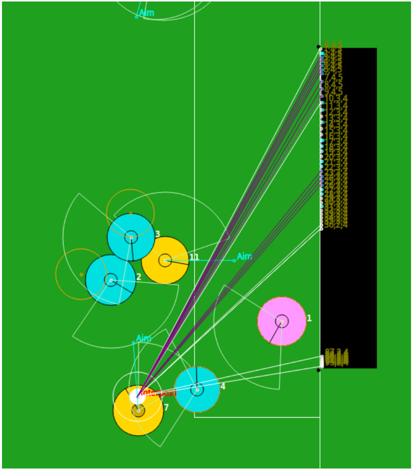


Figure 6: Search in shoot

Hierarchical Planning for Pass Behavior

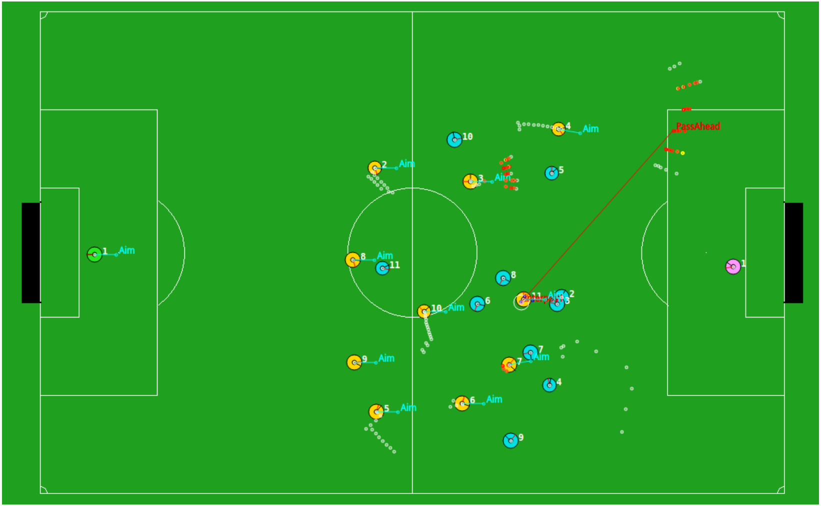


Figure 7: Hierarchical planning for pass behavior

- Transitions as explicit distributions $\Pr(s' \mid s, a)$ are not available
- Sampling rules $s' \sim \Pr(s' \mid s, a)$ are clearly defined by the simulator
- Monte-Carlo tree search w/ state abstraction
- Low-level skills: $NavTo$, $KickTo$, $\ldots$

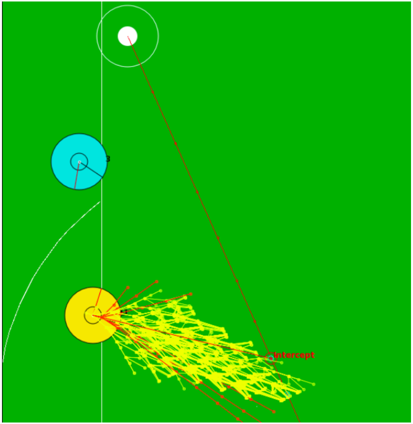


Figure 8: Search tree in *NavTo*

Terminating Distribution Estimation

- $\Pr(s' \mid s, a)$
  - $\Pr(success \mid s, Shoot)$
  - $\Pr(success \mid s, Pass)$
  - $\Pr(success \mid s, Intercept)$
    - $\Delta t = t_b - t_p$
    - $p \approx f(\Delta t)$

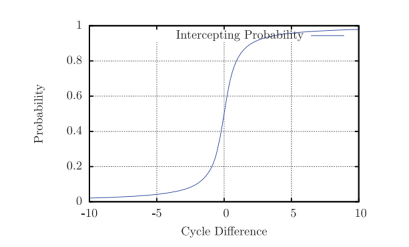


Figure 9: Intercepting probability

Heuristic Evaluation

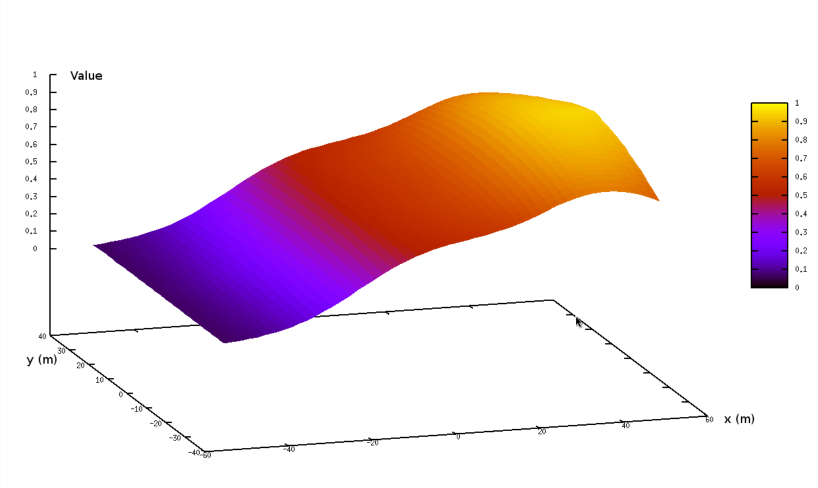


Figure 10: A heuristic function used in defense behaviors

The Results

RoboCup Competitions

- Six *world champions*
- Most successful team (according to Wikipedia)

Competitions	Games	Points	Goals	Win	Draw	Lost	Average Points	Average Goals
RoboCup 2005	19	47	84 : 16	15	2	2	2.47	4.42 : 0.84
RoboCup 2006	14	38	57 : 6	12	2	0	2.71	4.07 : 0.43
RoboCup 2007	14	34	125 : 9	11	1	2	2.42	8.92 : 0.64
RoboCup 2008	16	40	74 : 18	13	1	2	2.50	4.63 : 1.13
RoboCup 2009	14	36	81 : 17	12	0	2	2.57	5.79 : 1.21
RoboCup 2010	13	33	123 : 7	11	0	2	2.54	9.47 : 0.54
RoboCup 2011	12	36	151 : 3	12	0	0	3.00	12.6 : 0.25
RoboCup 2012	21	58	104 : 18	19	1	1	2.76	4.95 : 0.86
RoboCup 2013	19	53	104 : 9	17	2	0	2.79	5.47 : 0.47

Figure 11: Historical results of WrightEagle from RoboCup 2005 to 2013

Related Publications

- IJCAI (Bai et al., 2016)
- NIPS (Bai et al., 2013a)
- ICAPS (Bai et al., 2014; Zhang et al., 2015)
- AAMAS (Bai et al., 2012b)
- RoboCup Symposium (Bai et al., 2012a, 2013b)
- ACM Transactions (Bai et al., 2015)

Open-Sourced Codes

- WrightEagle Base: `https://github.com/wrighteagle2d/wrighteaglebase`
- MAXQ-OP: `https://github.com/aijunbai/maxq-op`
- Hierarchical Planning: `https://github.com/aijunbai/hplanning`
- Multi-Agent Reinforcement Learning: `https://github.com/aijunbai/keepaway`
- Particle Filtering over Sets: `https://github.com/aijunbai/pfs`

Summary

- RoboCup soccer simulation 2d domain
  - Fully-distributed multi-agent stochastic system
  - Continuous state, observation and action spaces
- WrightEagle soccer simulation team
  - Markov decision process formulation
  - Hierarchical decomposition
  - MAXQ based online planning

Thank you!

References

Bai, A., Chen, X., MacAlpine, P., Urieli, D., Barrett, S., & Stone, P. (2012a). Wright Eagle and UT Austin Villa: RoboCup 2011 simulation league champions. In T. Roefer, N. M. Mayer, J. Savage, & U. Saranli (Eds.) *RoboCup-2011: Robot Soccer World Cup XV*, vol. 7416 of *Lecture Notes in Artificial Intelligence*. Berlin: Springer Verlag.

Bai, A., Srivastava, S., & Russell, S. J. (2016). Markovian state and action abstractions for MDPs via hierarchical MCTS. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, (pp. 3029–3039). URL `http://www.ijcai.org/Abstract/16/430`

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

References II

Bai, A., Wu, F., & Chen, X. (2013a). Bayesian mixture modelling and inference based Thompson sampling in Monte-Carlo tree search. In *Advances in Neural Information Processing Systems 26*, (pp. 1646–1654).

Bai, A., Wu, F., & Chen, X. (2013b). Towards a principled solution to simulated robot soccer. In X. Chen, P. Stone, L. E. Sucar, & T. V. der Zant (Eds.) *RoboCup-2012: Robot Soccer World Cup XVI*, vol. 7500 of *Lecture Notes in Artificial Intelligence*. Berlin: Springer Verlag.

Bai, A., Wu, F., & Chen, X. (2015). Online planning for large markov decision processes with hierarchical decomposition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *6*(4), 45.

Bai, A., Wu, F., Zhang, Z., & Chen, X. (2014). Thompson sampling based Monte-Carlo planning in POMDPs. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS 2014)*. Portsmouth, United States.

Bai, A., Zhang, H., Lu, G., Jiang, M., & Chen, X. (2012c). WrightEagle 2D soccer simulation team description 2012. In *RoboCup Soccer Simulation 2D Competition, Mexico City, Mexico*.

References III

Zhang, Z., Hsu, D., Lee, W. S., Lim, Z. W., & Bai, A. (2015). PLEASE: palm leaf search for POMDPs with large observation spaces. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015.*, (pp. 249–258).