Learning Long-term Planning in Basketball Using Hierarchical Memory Networks

Stephan Zheng
Caltech
1200 E California Boulevard
Pasadena, CA
stzheng@caltech.edu

Yisong Yue
Caltech
1200 E California Boulevard
Pasadena, CA
yyue@caltech.edu

ABSTRACT

We study the problem of learning to plan spatiotemporal trajectories over long time horizons using expert demonstrations. For instance, in sports, agents often choose action sequences with long-term goals in mind, such as achieving a certain strategic position. Conventional policy learning approaches, such as those based on Markov decision processes, generally fail at learning cohesive long-term behavior in such high-dimensional state spaces, and are only effective when myopic planning leads to the desired behavior. The key difficulty is that such approaches use "shallow" planners that only learn a single state-action policy. We instead propose to learn a hierarchical planner that reasons about both long-term and short-term goals, which we instantiate as a hierarchical deep memory network. We showcase our approach in a case study on learning to imitate demonstrated basketball trajectories, and show that it generates significantly more realistic trajectories compared to non-hierarchical baselines as judged by professional sports analysts.

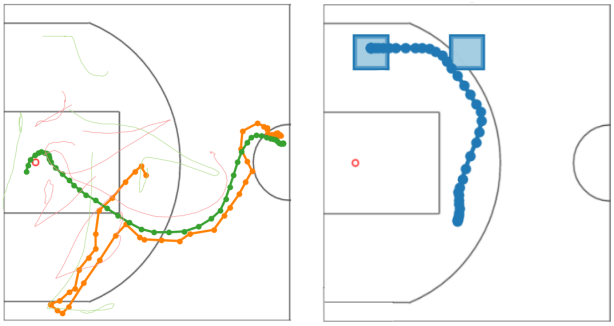
Keywords

Imitation learning. Spatiotemporal planning.

1. INTRODUCTION

Modeling long-term planning behavior is a key challenge in reinforcement learning and artificial intelligence. Consider a sports player choosing a movement trajectory to achieve a certain strategic position. The space of all such trajectories is prohibitively large, and precludes conventional approaches, such as those based on Markovian dynamics.

Many planning settings can be naturally modeled as requiring high-level, long-term *macro-goals*, which span time horizons much longer than the timescale of low-level *micro-actions* (cf. [1, 2]). A natural example for macro-micro planning behavior occurs in spatiotemporal games, such as basketball, where players execute complex trajectories. Each agent's micro-actions are to move around the court and if they have the ball, dribble, pass or shoot. These micro-actions operate at the millisecond scale, whereas their macro-goals, such as "maneuver behind these 2 defenders towards the basket", span multiple seconds. Figure 1 depicts an example from a professional basketball game, where the player must make a sequence of movements (micro-actions) in order to reach a specific location on the court (macro-goal).



(a) Two player (green) macro-goals: 1) pass the ball (orange) 2) move to the basket.

(b) Depicting two spatial macro-goals (blue boxes) as an agent moves to the top left.

Figure 1: Macro-goals from data and model predictions.

Intuitively, agents need to trade-off between short-term and long-term behavior: often sequences of individually reasonable micro-actions do not form a cohesive trajectory towards a macro-goal. For instance, in Figure 1 the player takes a highly non-linear trajectory towards his macro-goal of positioning near the basket. As such, conventional approaches are not well suited for these settings, as they generally use a single (low-level) planner, which is only successful when myopic planning leads to the desired behavior.

In this paper, we propose a novel class of *hierarchical policy models*, which we instantiate using deep memory networks (HPN), that can simultaneously reason about both macro-goals and micro-actions. Our model utilizes an attention mechanism through which the macro-planner guides the micro-planner. Our model is further distinguished from previous work on hierarchical planners by dynamically predicting macro-goals instead of following fixed goals, which gives additional flexibility to our model class that can be fitted to data (rather than having the macro-goals be hand-crafted). We demonstrate how to train HPNs to generate high quality basketball player trajectories, where quality is some global measure of the entire trajectory (e.g., realistic looking trajectories as in Figure 1). We showcase our approach in a case study on learning to imitate demonstrated behavior in professional basketball. Our primary result is that our ap-

KDD '16 June 16–19, 2013, San Francisco, CA, USA

DOI: 10.1145/1235

proach generates significantly more realistic player trajectories compared to non-hierarchical baselines, as judged by professional sports analysts. We also provide a comprehensive qualitative analysis of the behavior of the model.

2. RELATED WORK

The reinforcement learning community has largely focused on non-hierarchical, or "shallow", planners such as those based on Markovian or linear dynamics (cf. [4, 5, 2]). By and large, such policy classes are only shown to be effective when the optimal action can be found via short-term planning. Previous research has instead focused on issues such as how to perform effective exploration, plan over parameterized action spaces, or deal with non-convexity issues from using deep neural networks within the policy model. In contrast, we focus on developing hierarchical planners that can effectively generate realistic long-term plans in complex settings such as basketball gameplay.

The use of hierarchical models to decompose macro-goals from micro-actions is relatively common in the planning community (cf. [6, 1, 7]). For instance, the winning team in 2015 RoboCup Simulation Challenge [7] used a manually constructed hierarchical planner to solve MDPs with a set of fixed sub-tasks. In contrast, we study how one can *learn* a hierarchical planner from a large amount of expert demonstrations that can adapt its policy in non-Markovian environments with dynamic macro-goals.

3. LONG-TERM TRAJECTORY PLANNING

We first introduce some notation: We let $\mathcal{S}, \mathcal{A}$ denote the state, action space. Furthermore, let $\mathcal{G}$ be a macro-goal space. At time t, the full game state is $\mathbf{s}_t = \{s_t\}$ and each agent takes an action $a_t \in \mathcal{A}$. The history of events is $\mathbf{h}_{p,q} = \{(s_t, a_t)\}_{t\in[p,q]}$. Let $\pi(s, \mathbf{h})$ denote a policy that maps state and history to a distribution over actions $P^\pi(a|s, \mathbf{h})$.

Incorporating Macro-Goals. Our main modeling assumption is that the policy should *simultaneously optimizes behavior hierarchically on multiple well-separated timescales*. We consider two distinct timescales (*macro* and *micro*-level), although our approach could in principle be generalized to even more timescales. During an episode $[t_0, t_1]$, the policy executes a sequence of micro-actions (a_t) that leads to a macro-goal g_t. We do not assume that the start and end times t_0, t_1 are fixed, for instance, macro-goals can change before they are reached. We finally assume that macro-goals are relatively static on the timescale of the micro-actions, that is: $dg_t/dt \ll 1$. Figure 1b depicts an example of the agent with two unique macro-goals over a 50-frame trajectory. At every timestep t, the agent executes a micro-action a_t, while the macro-goals g_t change more slowly.

As they are unobservable, we model macro-goals as latent variables. Given the goal space $\mathcal{G}$ and the state-action space $\mathcal{S} \times \mathcal{A}$, we further assume that there exists a transfer function $m : \mathcal{G} \longrightarrow \mathcal{S} \times \mathcal{A}$ that describes how latent goals are mapped onto observable states and actions. Equivalently, one can model this map through an auxiliary transfer function: $P(s_t, a_t|g_t) = \int dm P(s_t, a_t|m)P(m|g_t)$, where m models an intermediate latent variable that is Markovian. The full posterior then becomes:

$$P(a_t|s_t, \mathbf{h}_{t-n,t}) = \int dm_t \int dg_t P(a_t|s_t, \mathbf{h}_{t-n,t})P(s_t, a_t|m_t) \times P(m_t|g_t)P(g_t|s_t, \mathbf{h}_{t-n,t}). \quad (1)$$

Figure 2 shows an instance of this two-level policy using two deep memory networks, which we discuss in Section 4. This hierarchical decomposition intuitively can be generalized to multiple scales l using a set of transfer functions m^l.

4. HIERARCHICAL MEMORY NETWORK

Figure 2 depicts a high-level overview of our hierarchical policy class for long-term spatiotemporal planning. When using continuous latent variables m, g, computing the posterior (1) exactly is intractable, and one would have to resort to approximation methods. Therefore, we discretize the state-action and latent spaces. In spatiotemporal settings, such as for basketball, a state $s_t^i \in \mathcal{S}$ can naturally be represented as a 1-hot occupancy vector of the basketball court. Goal states g are then sub-regions of the court that i wants to reach, defined at a coarser resolution than $\mathcal{S}$. With this choice, it is natural to instantiate the macro and micro-planners as convolutional recurrent neural networks, which can capture both predictive spatial patterns and non-Markovian temporal dynamics.

Attention mechanism for integrating macro-goals and micro-actions. We model the transfer function m as an *attention mechanism over the action space* $\mathcal{A}$, that is, a non-linear weight function on $\mathcal{A}$. In this case, fixing a macro-goal g, the policy π becomes

$$\pi(s_t|g) = m_g(\pi_{\text{micro}}(s_t)), \quad m_g(a) \equiv m_g \odot a, \quad (2)$$

where $\odot$ denotes the Hadamard product. Intuitively, this interaction captures the trade-off between the macro- and micro-planner. On the one hand, the micro-planner optimizes its policy π_{micro} for short-term optimality, On the other hand, the macro-planner π_{macro} and attention m_g can learn to attend to sequences of actions that lead to a macro-goal g and bias the agent i towards good long-term behavior.

Multi-stage learning. Given a training set D of sequences $(\mathbf{s}_t, \hat{a}_t)$, the optimal long-term planning policy can be learned by solving $\theta^* = \text{argmin}_\theta \sum_D \sum_{t=1}^T L_t(\mathbf{s}_t, \hat{a}_t; \theta)$. Given the hierarchical structure of the HPN, the objective function L_t decomposes as:

$$L_t(\mathbf{s}_t, \hat{a}_t; \theta) = L_{t,\text{macro}}(\mathbf{s}_t; g_t, \theta_{\text{macro}}) + L_{t,\text{micro}}(\mathbf{s}_t, \hat{a}_t; g_t, \theta) + R_t(\theta), \quad (3)$$

$$L_{t,\text{micro}}(\mathbf{s}_t, \hat{a}_t; \theta) = \hat{a}_t \log P\left(m_{g_t}(\theta_m) \odot \pi_{\text{micro}}(\mathbf{s}_t; \theta_{\text{micro}})|\mathbf{s}_t\right),$$

where $R_t(\theta)$ is a regularization for the model weights $\theta = \{\theta_{\text{macro}}, \theta_{\text{micro}}, \theta_m\}$. Although we have ground truth labels $\hat{a}_t^i$ for the observable micro-actions, in general we do not have labels for the latent macro-goals g_t that induce optimal long-term planning. As such, one would have to appeal to separate solution methods to minimize $L_{t,\text{macro}}(\mathbf{s}_t; g_t, \theta_{\text{macro}})$.

To reduce this complexity and given the non-convexity of (3), we instead follow a multi-stage learning approach combined with a set of *weak labels* $\hat{g}_t, \hat{m}_t$ for the macro-goals $g_t = \pi_{\text{macro}}(\mathbf{s}_t)$ and attention masks m_t. We assume access to such weak labels and only use them in the initial training phases. We first train the micro-planner π_{micro}, macro-planner π_{macro} and attention m_g individually through

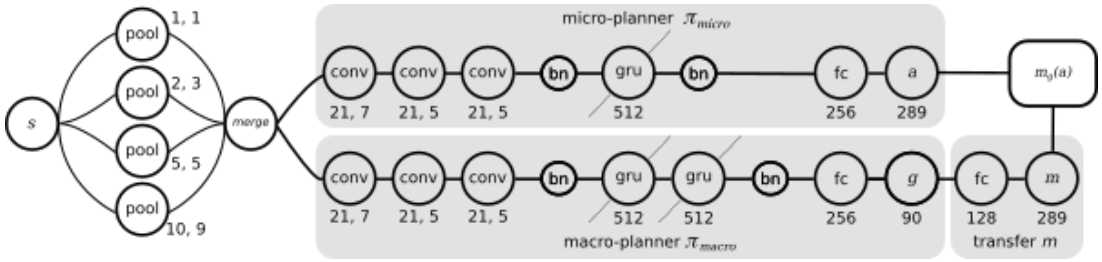


Figure 2: Network architecture and hyperparameters of the hierarchical policy network, displayed for a single timestep t. Max-pooling layers (numbers indicate kernel size) with unit stride upsample the sparse tracking sequence data $\boldsymbol{s}_t$. Both the micro and macro planners $\pi_{\text{micro}}, \pi_{\text{macro}}$ use a convolutional (kernel size, stride) and GRU memory (number of cells) stack to predict a_t and g_t. Batch-normalization (bn) is applied to stabilize training. The output attention mechanism m is implemented by 2 fully-connected layers (number of output units). Finally, the network predicts the final output $\pi(s_t) = m_g \odot \pi_{\text{micro}}(s_t)$.

cross-entropy minimization with the labels $\hat{a}_t^i, \hat{g}_t$ and $\hat{m}_t$. In the final stage we fine-tune the entire network on $L_{t,\text{micro}}$, using only the ground truth micro-action labels $\hat{a}_t^i$.

5. BASKETBALL BEHAVIOR CASE STUDY

We validated the HPN by learning a basketball movement policy that predicts the *instantaneous velocity* $v_t = \pi(s_t, \mathbf{s}_t)$, a challenging problem in the spatiotemporal setting.

Training data. We trained the HPN on a large dataset of tracking data from professional basketball games [3], consisting of *possessions*: sequences of tracking coordinates $s_t = (x_t, y_t)$ for each player, recorded at 25 Hz, where one team has continuous possession of the ball. For simplicity, we modeled every player identically using a single policy network. After pre-processing, we extracted 120,000 tracks for training and 20,000 as a holdout set.

Labels. We extracted micro-action labels $\hat{v}_t = s_{t+1} - s_t$ as 1-hot vectors in a grid of 17×17 unit cells. Additionally, we constructed a set of weak macro-labels $(\hat{g}_t, \hat{m}_t)$ by heuristically segmenting each track using its stationary points. The labels $\hat{g}_t$ were defined as the next stationary point. For $\hat{m}_t$, we used 1-hot velocity vectors $v_{t,\text{straight}}$ along the straight path from the player's location s_t to the macro-goal g_t.

Baselines. We compared the HPN against two natural baselines: a policy with only a micro-planner (with memory (GRU-CNN) and without (CNN)), and an HPN H-GRU-CNN-CC without an attention mechanism, but instead predicts using a feature concatenation of the micro and macro-planner.

Rollout evaluation. To evaluate the quality of our model, we generated rollouts $(s_t; \mathbf{h}_{0,r_0})$ with *burn-in period* r_0, These are generated by 1) feeding a ground truth sequence of states $\mathbf{h}_{0,r_0} = (\mathbf{s}_0, \ldots, \mathbf{s}_{r_0})$ to the policy network and 2) for $t > r_0$, iteratively predicting the next action a_t and updating the game-state $\mathbf{s}_t \to \mathbf{s}_{t+1}$, using ground truth locations for the other agents.

5.1 How Realistic are the Generated Tracks?

The most holistic way to evaluate the trajectory rollouts is via visual analysis: would a basketball expert find the rollouts by HPN more realistic than those by the baselines?

Visualization. Figure 3 depicts HPN and baseline rollouts. Every rollout consists of two parts: 1) a ground truth segment from a holdout sequence and 2) a continuation by either the HPN, baseline or ground truth. In particular, HPN tracks do not move towards macro-goals in unrealistic straight lines, but often take a curved route, indicating that the policy balances moving towards macro-goals with short-term responses to the current state (see also Figure 3e). In contrast, the baselines often generate more constrained behavior, such as moving in straight lines or remaining stationary for long periods of time.

Human preference study. Our primary empirical result is a preference study eliciting judgments on the relative quality of rollouts between HPN and baselines or ground truth. We recruited seven experts (professional sports analysts) and eight knowledgeable non-experts (e.g., college basketball players) as judges. Because all the learned policies perform better with a "burn-in" period, we first animated with the ground truth for 20 frames, and then extrapolated with a policy for 30 frames. During extrapolation, the other nine players do not animate. For each test case, the users were shown an animation of two rollout extrapolations of a specific player's movement: one generated by the HPN, the other by a baseline or ground truth. The judges then chose which rollout looked more realistic.

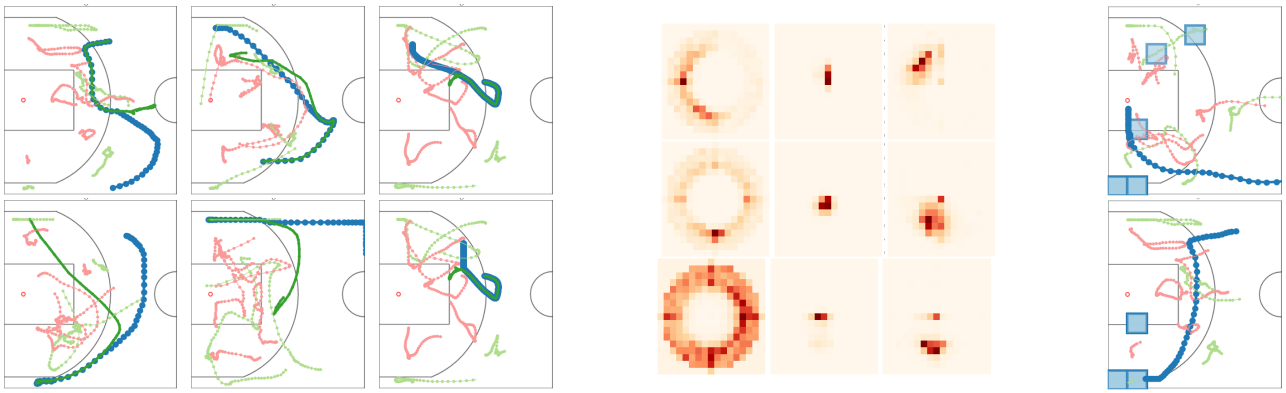
Table 1 shows the preference study results. We tested 25 scenarios (some corresponding to scenarios in Figure 3e). HPN won the vast majority of comparisons against the baselines using expert judges, with slightly weaker preference using non-expert judges. HPN was also competitive with the ground truth. This suggests that the HPN is able to generate high-quality player movement trajectories that are significant improvements upon baseline methods, and approach the ground truth quality for our extrapolation setting.

5.2 Macro and Micro-planner Integration

Our model integrates the macro- and micro-planner by converting the macro-goal into an attention mask on the micro-action output space, which intuitively guides the micro-planner towards the macro-goal. Figure 3d depicts how the macro-planner π_{macro} guides the micro-planner π_{micro} through the attention m, by attending to the direction in which the predicted macro-goal can be reached. Figure 3e depicts predicted macro-goals by HPN along with rollouts. In general, we see that the rollouts are guided towards the predicted macro-goals. However, we also observe that the HPN makes some inconsistent macro-goal predictions, which suggests there is still ample room for improvement.

6. LIMITATIONS AND FUTURE WORK

There are several notable limitations to our HPN model. First, we did not consider all aspects of basketball gameplay, such as passing and shooting. We also modeled all players using a single policy whereas in reality player behaviors vary (although the variability can be low-dimensional [3]). We also only modeled offensive players, and another inter-



(a) HPN rollouts *(b) HPN (top) and failure case (bottom)* *(c) HPN (top), baseline (bottom)* *(d) Attention masks m_g (left column), velocity $\pi_{\text{micro}}(\boldsymbol{s}_t)$ (middle) and prediction $m_g \odot \pi_{\text{micro}}(\boldsymbol{s}_t)$ (right). The center corresponds to 0 velocity.* *(e) Rollouts and predicted macro-goals (boxes).*

Figure 3: Rollouts generated by the HPN and baseline (columns a, b, c). Attention model (column d). Macro-goals (column e). ***Rollouts.*** *Each frame shows an offensive player (dark green), a rollout (blue) track that extrapolates after 20 frames, the offensive team (light green) and defenders (red). Note we do not show the ball, as we did not use semantic basketball features (i.e "currently has the ball") during training. The HPN rollouts do not memorize training tracks (column a) and display a variety of natural behavior, such as curving, moving towards macro-goals and making sharp turns. We also show a failure case (b, bottom), where the HPN behaves unnaturally by moving along a straight line off the right side of the court – which may be fixable by using additional game state information. For comparison, a hierarchical baseline without an attention model, produces a straight-line rollout (column c, bottom), whereas the HPN produces a more natural movement curve (column c, top).* ***Attention model.*** *Visualizing how the attention mask generated from the macro-planner interacts with the micro-planner π_{micro}. Row 1: the micro-planner π_{micro} decides to stay stationary, but the attention m_g goes to the left. The weighted result $m_g \odot \pi_{\text{micro}}(\boldsymbol{s}_t)$ is to move to the left, with a magnitude that is the average. Row 2: π_{micro} wants to go straight down, while m_g boosts the velocity so the agent bends to the bottom-left. Row 3: π_{micro} remains stationary, but m_g prefers to move in any direction. As a result, the agent moves down.* ***Macro-goals.*** *The HPN dynamically predicts macro-goals and guides the micro-planner in order to reach them. The HPN starts the rollout after 20 frames. Macro-goal box intensity corresponds to relative prediction frequency during the trajectory. The macro-goal predictions are stable over a large number of timesteps. The HPN sometimes predicts inconsistent macro-goals. For instance, in the bottom right frame, the agent moves to the top-left, but still predicts the macro-goal to be in the bottom-right sometimes.*

Model comparison	Experts		Non-Experts		All	
	W/T/L	Avg Gain	W/T/L	Avg Gain	W/T/L	Avg Gain
VS-CNN	21 / 0 / 4	0.68	15 / 9 / 1	0.56	21 / 0 / 4	0.68
VS-GRU-CNN	21 / 0 / 4	0.68	18 / 2 / 5	0.52	21 / 0 / 4	0.68
VS-H-GRU-CNN-CC	22 / 0 / 3	0.76	21 / 0 / 4	0.68	21 / 0 / 4	0.68
VS-GROUND TRUTH	11 / 0 / 14	-0.12	10 / 4 / 11	-0.04	11 / 0 / 14	-0.12

Table 1: Preference study results. We asked 7 basketball experts and 8 knowledgeable non-experts to judge the relative quality of policy rollouts. We compare HPN with ground truth and 3 baselines: a memory-less (CNN) and memory-full (GRUCNN) micro-planner and a hierarchical planner without attention (GRUCNN-CC). For each of 25 test cases, HPN wins if more judges preferred the HPN rollout over a competitor. Average gain is the average signed vote (1 for always preferring HPN , and -1 for never preferring). We see that the HPN is preferred over all baselines (all results against baselines are 95% statistically significant). Moreover, the HPN is competitive with ground truth, indicating that HPN generates realistic trajectories within our rollout setting.

esting direction is modeling defensive players and integrating adversarial reinforcement learning [8] into our approach. These issues limited the scope of the rollouts in our preference study, and it would be interesting to consider extended settings.

In order to focus on our HPN model class, we considered only the imitation learning setting. More broadly, many planning problems require collecting training data via exploration [5], which can be more challenging. One interesting scenario is having two adversarial policies learn to be strategic against each other through repeatedly game-play in a basketball simulator. Furthermore, in general it can be difficult to acquire the appropriate weak labels to initialize the macro-planner training.

**Team formation. Value based methods. **

7. REFERENCES

[1] Ruijie He, Emma Brunskill, and Nicholas Roy. PUMA: Planning Under Uncertainty with Macro-Actions. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, July 2010.

[2] Matthew Hausknecht and Peter Stone. Deep reinforcement learning in parameterized action space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016.

[3] Yisong Yue, Patrick Lucey, Peter Carr, Alina Bialkowski, and Iain Matthews. Learning Fine-Grained Spatial Models for Dynamic Sports Play Prediction. In *IEEE International Conference on Data Mining (ICDM)*, 2014.

[4] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, pages 1433–1438,

2008.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[6] Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1âĂŞ2):181–211, August 1999.

[7] Aijun Bai, Feng Wu, and Xiaoping Chen. Online planning for large markov decision processes with hierarchical decomposition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(4):45, 2015.

[8] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.