# Scaling Multiagent Markov Decision Processes (Extended Abstract)

Scott Proper
Oregon State University
Corvallis, OR 97331-3202, USA
proper@eecs.oregonstate.edu

## 1. THREE CURSES OF DIMENSIONALITY

Markov Decision Processes (MDPs) have proved to be useful and general models of optimal decision-making in uncertain domains. However, approaches to solving MDP's using reinforcement learning that depend on storing the optimal value function and action models as tables do not scale to large state-spaces. Three computational obstacles prevent the use of standard approaches when dealing with problems with many variables. First, the state space (and time required for convergence) grows exponentially in the number of variables. This makes computing the value function impractical or impossible in terms of both memory and time. Second, the space of possible actions is exponential in the number of agents, so even one-step look-ahead search is computationally expensive. Lastly, exact computation of the expected value of the next state is slow, as the number of possible future states is exponential in the number of variables. These three obstacles are referred to as the three "curses of dimensionality".

Much prior work exists on the topic of scaling reinforcement learning to large state spaces. Many state abstraction and function approximation techniques exist. These techniques are a result of the desire to reduce the number of parameters used to represent the value function, and thus reduce memory requirements and time to converge. In addition to such techniques, methods to incorporate prior knowledge can be successful in speeding up convergence.

In [4] I addressed the three curses of dimensionality, providing solutions to each. To solve the problem of exploding state space, I introduced a kind of function approximation called "tabular linear functions". To solve action space explosion, I used a hill climbing technique over the action search space. To solve the problem of computing the expected value of the next state, I introduced ASH-learning, which is a model-based average reward algorithm that uses afterstates to reduce the number of future states it is necessary to examine.

## 2. ASSIGNMENT-BASED DECOMPOSITION

A common approach to dealing with issues of scaling is to take advantage of domain-specific structure. Consider the setting of cooperative multiagent reinforcement learning, where the agents are trying to cooperate to maximize a global reward signal. The structure of such multiagent domains can be taken advantage of by decomposing the states and actions.

In my thesis I propose a new technique for dealing with scaling issues; in particular, I consider the problem of coordinating multiple agents that share a common reward function through a centralized controller. Many domains can be decomposed into a set of weakly coupled agents, where each agent needs to know only limited information about the others. This allows significant scaling by limiting the amount of global information and facilitates local decision-making. I demonstrate how to implement these techniques using a variety of common value iteration-based reinforcement learning techniques, including model-free Q-learning and model-based methods.

Rather than addressing separate solutions to each of the three curses of dimensionality, I propose a single technique for decomposing certain reinforcement learning problems such that all the curses of dimensionality are addressed. In my thesis, I consider a problem of multiple agents and multiple tasks, where the agents are to be assigned to tasks in an optimal fashion. I call these problems multiagent assignment MDPs. Given an assignment, the agents might work almost independently of each other. However, the assignment can potentially change opportunistically. I also show that the optimal value function even in the simplest of such assignment tasks is not expressible as a coordination graph. The difficulty is enforcing conditions such as assigning at most two agents to each task to get a reward.

I present a new assignment-based decomposition [5] approach where the action-selection step is split into two levels. At the top level agents are assigned to tasks and at the lower level the tasks are performed by the teams with minimal dynamic coordination. This is similar to the hierarchical multiagent reinforcement learning of [3], except that I learn a value function only at the lower level and use search to optimize the higher level. My approach thus scales much better since it is not necessary to store an exponentially large value function at the top level.

I will also show how assignment-based decomposition may be expanded and scaled to solve difficult problems, with many agents and tasks. Fast search methods (such as those based on hill climbing or bipartite matching algorithms) are useful here as the space of possible assignments grows very large as the number of agents and tasks increases. In addition, I will show how using transfer learning and generalization techniques will allow a policy learned on only a few agents or tasks may scale to many agents and tasks.

## 3. COORDINATION GRAPHS

When decomposing the states and actions of cooperative agents, the issue of coordination of agent actions presents itself. Recent work using coordination graphs between agents has been shown to be successful here [1, 2]. The nodes of the graph represent agents and the arcs between them represent potential interactions between them. The long-term value of a joint action over all agents is expressed as a sum of all the interaction terms, where each such term is based on the actions and states of two agents. Bayesian network inference algorithms such as variable elimination and belief propagation have been adapted to finding the best joint action that maximizes the total reward.

Unfortunately, in many domains, coordination graphs are not static but change dynamically based on the states and actions of the agents. The approaches based on coordination graphs are adapted to dynamic state-based coordination [1, 2]. For example, in the approach of [2], a set of rules dictate which agent should coordinate with whom, and the value of a state is based on the current coordination graph.

I will demonstrate a technique for combining coordination graphs and assignment-based decomposition by adding a context-sensitive coordination graph at the lower level of the assignment-based decomposition. Doing this allows us some advantages over using either technique alone through separation of concerns. First, consideration of details such as collision avoidance can be delegated to lower levels, freeing the top level to focus on assignment decisions. Second, the coordination graph at the lower level can take advantage of knowing the assignment when making coordination decisions. Third, since the lower level value functions are used in making the higher level assignment decisions, collision information is indirectly percolated to the assignment level.

## 4. RELATIONAL TEMPLATES

In [4] I introduced a new description of a function approximation method called “Tabular Linear Functions” (TLFs). TLFs are a means of combining tables and linear functions in such a way as to preserve some of the best qualities of both. I will take this reseach further, describing how to expand and apply TLFs to a relational setting to create a function approximation method I call “Relational Templates”. The use of relational templates greatly expands the kinds of domains that TLFs may be applied to.

I will also show how the use of relational templates facilitates transfer learning and the ability to generalize across multiple domains. Relational templates make be easily re-used across different (similar) domains. Also, parameters learned on one domain may often be transferred or generalized to multiple similar domains. I will show how to combine relational templates with assignment-based decomposition to easily scale a complex multiagent domain from few to many agents and tasks.

## 5. BIPARTITE SEARCH

Assignment-based decomposition solves many of the three curses of dimensionality, but introduces a new curse of it’s own: how to scale the assignment search problem as the number of agents increases? With many agents and tasks, there are correspondingly many possible assignments. In [5], I describe three simple methods for search: exhaustive search, sequential greedy assignment, and swap-based hillclimbing. All of these methods have trade off solution speed and solution quality. I will introduce a new, more sophisticated approximate search technique for solving the assignment search problem: iterated bipartite assignment search. This search algorithm quickly provides a high-quality approximation of the true optimal assignment, allowing assignment-based decomposition to scale to much larger numbers of agents and tasks.

## 6. PRELIMINARY RESULTS

I have implemented assignment-based decomposition successfully on many domains, including product delivery domains, multiagent predator-prey domains, and real time strategy (RTS) game simulations. For this latter domain, I implemented a simple RTS game simulation on a 10x10 gridworld. Agents vary in number from 3-12 archers or infantry, and may face off against up to 4 enemy “tasks”, either towers, knights, or ballista. These enemy units are more powerful than friendly units, and thus agents must coordinate in teams of up to three in order to destroy the enemy. Units are described by attributes such as location, hit points, damage, range, and mobility. I used a total reward version of ASH-learning [4] and assignment-based decomposition to solve this domain. Rewards were either +1 for a kill, -1 for a death, and -.1 per time step. As may be seen on this preliminary result in Figure 1, assignment-based decomposition greatly outperforms “flat” ASH-learning. Not only that, flat ASH-learning requires seven times as much CPU time to complete a single run.

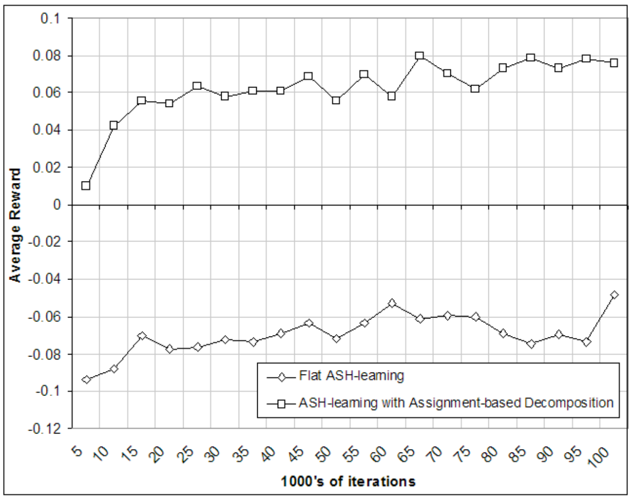

**Figure 1: Comparison of flat vs. assignment-based decomposition in 6 agent vs. 2 task RTS domain.**

## 7. REFERENCES

[1] C. Guestrin, S. Venkataraman, and D. Koller. Context specific multiagent coordination and planning with factored MDPs. In *AAAI ’02: Proceedings of the 8th National Conference on Artificial Intelligence*, pages 253–259, Edmonton, Canada, July 2002.

[2] J. R. Kok and N. A. Vlassis. Sparse Cooperative Q-learning. In R. Greiner and D. Schuurmans, editors, *ICML ’04: Proceedings of the 21st International Conference on Machine Learning*, pages 481–488, Banff, Canada, July 2004. ACM.

[3] R. Makar, S. Mahadevan, and M. Ghavamzadeh. Hierarchical multi-agent reinforcement learning. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 246–253, Montreal, Canada, 2001. ACM Press.

[4] S. Proper and P. Tadepalli. Scaling model-based average-reward reinforcement learning for product delivery. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *ECML ’06: Proceedings of the 17th European Conference on Machine Learning*, volume 4212 of *Lecture Notes in Computer Science*, pages 735–742. Springer, 2006.

[5] S. Proper and P. Tadepalli. Solving multiagent assignment markov decision processes. In *AAMAS ’09: Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (to be published)*, 2009.