# Dimensions of Complexity of Intelligent Agents

David Poole and Alan Mackworth

Department of Computer Science, University of British Columbia
`http://www.cs.ubc.ca/spider/poole/`
`http://www.cs.ubc.ca/spider/mack/`

**Abstract.** This paper aims to provide a framework for understanding the construction of intelligent agents. This is used to explain the history of AI, and provide a roadmap of future research. Research has progressed by making simplifying assumptions about the representations of the agents or about the environments the agents act in. In particular, we present a number of dimensions of simplifying assumptions that have been made. For each of these dimensions, there is a simplified case and progressively more complex cases. We argue that an intelligent agent needs the complex value in each of these dimensions (i.e., to simultaneously give up many simplifying assumptions). However these dimensions interact in complex ways. Much of the recent history can be seen as understanding the interaction of these dimensions.

## 1 Introduction

A symposium on practical cognitive agents and robots that celebrates 50 years of AI seems like a good venue for reflecting on the field; where it has come and where it is going. This paper is not a traditional research paper, but should be seen as a very high-level overview that gives a rough approximation to the field. From the viewpoint of this high level we can see a roadmap of where AI is going, and the many challenges that lie ahead of us. Our main aim is to get people to see the big picture and encourage work that combines many fields. While many people may see these sub-fields of AI as competing, we argue that there is a coherent framework that can sensibly combine the AI traditions.

## 2 The Dimensions of Complexity

Agents acting in environments have various ranges of complexity from thermostats to companies with multiple goals acting in competitive environments. There are a number of simplifying assumptions that have been made in designing agents. Each of these simplifying assumptions gives a dimensions of complexity of the design of intelligent agents. The dimensions specify which simplifying assumptions to adopt or relax; these dimensions typically have multiple levels going from the most simple to the most complicated. These dimensions have typically been considered separately, but need to be combined in order to build an intelligent agent.

In each of these dimensions there is the simple and the complex. While for any particular agent, you should use the appropriate assumptions, a really intelligent agent

needs to be complex in all of the dimensions. Making these dimensions explicit should help people to choose the appropriate technology for their domain, and lead researchers to explore more of the design space. Under the view here, very little of the design space of intelligent agents has been explored.

The dimensions in this paper are related to that of [1] but are different from the dimensions of Russell and Norvig [2]. For example, these is no dimension of discrete versus continuous, as we expect that we need both discrete and continuous variables in an intelligent agent. That is, assuming the world is discrete and assuming the world is continuous are both simplifying assumptions that don't reflect the complexity of the world our agents need to reason in. We also include logical agents, not as competition to probabilistic agents, but as complementary.

Much of the history of AI can be seen as starting from the simple and adding in complexity in one of these dimensions.

### 2.1 Modularity

**Modularity** is important for reducing complexity. It is apparent in the structure of the brain, has been discovered by computer scientists, and is an important part of any organization of any size [3].

Modularity is typically expressed in terms of a hierarchical decomposition. For example, the visual cortex and the eye of a human is a module that takes in light and outputs some simplified description of a scene. Large organizations have a hierarchical organization so that the top-level decision makers are not overwhelmed by details and do not need to micro-manage all details of the organization. Procedural abstraction and object-oriented programming in computer science are designed for modularity.

The first dimension is, whether the representation is

– **flat**: there is no organizational structure
– **hierarchical**: the agent is organized hierarchically in terms of interacting modules

This is the question is about whether the agent is reasoning at a single level of abstraction or at multiple levels of abstraction. In a flat representation, you choose one such level of abstraction and the agent reasons at that level. In a hierarchical representation, an agent reasons about multiple levels of abstraction.

For example, in taking a trip from your home to a conference location overseas, an agent needs to get from its home to an airport, fly to an airport near the destination, then get from the airport to the destination. The agent needs to make specific motor controls to do anything, but it can't just reason at the level of these motor controls, as there are too many possible sequences of controls and there is too much uncertainty about the specific motor controls it will do in the future. Each level of abstraction is correct in some sense, as all of them are used to get a working system.

Note that a flat representation will be either continuous or discrete; a hierarchical representation is typically a hybrid of continuous and discrete processes. It seems that the lowest levels are continuous and the highest levels are discrete, and their interface is due to discrete objects made of continuous parts or due to discrete events in continuous time.

As in each of these dimensions, you really want hierarchical decomposition, but in order to explore the other dimensions we often ignore the hierarchical structure and just assume a flat representation. Ignoring hierarchical decomposition is often fine for small or moderately sized problems, as it is for simple animals, small organizations or small to moderately sized computer programs.

## 2.2 Succinctness and Expressiveness

Much of the work on modern AI is about finding compact representations and exploiting that compactness for computational gains. There are good arguments to say that compactness is what enables understanding, and this is the basis of much of machine learning. The next dimension is the level of succinctness and expressiveness: how compact can you state the knowledge about a domain. Succinct descriptions allow for more expressiveness. While there are fine distinctions that can be made in terms of succinctness and expressiveness, at the topmost level, there seems to be three main levels of levels. The first is in terms of states, the second is in terms of features, and the third is in terms of objects and relations.

Each of the different ways the world could be to affect what an agent should do is called a **state**. We can split the state of the world into the memory state of the agent, and the environment state.

At the least succinct level, an agent can reason explicitly in terms of individually identified states. For example, a thermostat can easily be defined in terms of a few memory states and environment states, and the actions can be defined in terms of a finite state machine.

Instead of enumerating states, it is often easier to reason in terms of features or propositions. A state can be described in terms of **features**, where each feature has a value in every state, and a state corresponds to assigning a value to each feature. Thirty binary features can encode $2^{30} = 1,073,741,824$ states. It may be easier to specify and reason with thirty binary variables (or thirty propositions) than with over a billion states. It, however, usually isn't as simple as this as the features are typically interdependent and not all combinations of values to features are possible. Representations in terms of features are often called factored representations. Note that some questions, e.g., what logically follows, can be answered in linear time in the number of states, but is NP-complete in terms of features. While NP-complete may seem to be more complex than linear, linear in the number of states is the same as exponential in the number of features, and you can typically solve NP-complete problems much faster than exponential time (e.g., by exploiting structure).

When describing a complex world, the number of features depends on the number and type of individuals. For example, if an agent is enrolling students in courses, there may be a feature for every student-course pair where the student took the course that gives the grade of the student in the course (the real case is even more complicated if a student can take a course multiple times). Rather than having a $passed$ feature for every student-course pair that depends on the $grade$ feature for that pair, it may be easier to reason in terms of individual students and courses, and the relations $grade$ and $passed$. You can define rules for how $passed$ depends on $grade$ once, and apply it for each

student and course. Moreover this rule can be defined before you know of any of the individuals and so before you know any of the features.

Thus, instead of dealing with features or propositions, it is often more convenient and efficient to reason in terms of individuals and relationships. For example, one binary relation and 100 individuals can succinctly represents $100^2 = 10000$ propositions and $2^{10000}$ states. By reasoning in terms of relations and individuals, you can specify knowledge and reason about whole classes of individuals without ever enumerating the features or propositions, let alone the states. Often you even want to reason about infinite sets of individuals, for example, the set of all English sentences that could be meant by a sequence of sounds, which cannot be done at the state or feature level. Allowing infinitely many individuals increases the expressiveness.

The second dimension is thus whether the agent reasons in terms of

– **states**
– **features**
– **individuals and relations**

Some of the AI frameworks have been developed in terms of individuals and relations, some in terms of features and some have been developed in terms of states. For example, modern reinforcement learning [4] was developed in terms of states and has been extended to be feature-based, to the point now that no serious reinforcement learning is in terms of states. It is an open research problem to extend reinforcement learning to individuals and relations.

## 2.3 Planning Horizon

The next dimension is how far ahead the agent plans for. For example, when a dog is called to come, it needs to turn around to start running in order to get a reward in the future. It does not act only to get an immediate reward, but it also plausibly doesn't act for goals arbitrarily in the future.

How far the agent "looks into the future" when deciding what to do is called the **planning horizon**. That is, the planning horizon is how far ahead the agent considers the consequences of its actions. For completeness we include the static case where the agent isn't reasoning in time.

The next dimension has four values:

– A **static** domain is one where time isn't involved
– A **finite stage** problem is where the agent looks for a fixed finite number time steps ahead. For example, a doctor may have to treat a patient but may have time for some testing, and so there may be two stages, a testing stage and a treating stage, to plan for. In the degenerate case where an agent only looks one time step ahead, it is said to be greedy or myopic.
– An **indefinite stage** problem is when the agent looks ahead some finite, but not predetermined, number of steps ahead. For example, an agent who wants to get to some location may not know a priori how many steps it will take to get there.
– An **infinite stage** problem is one where the agent plans for going on forever. This is often called a process model. For example the stabilization module of a robot should plan on going on forever.

Note that the planning horizon interacts with the level of abstraction. For example, at one level of abstraction the dog may be getting an immediate reward (it comes and gets a treat), but at another level of abstraction it may be an indefinite stage: at the level of deciding where to place its paws, there may be a long time until it gets the reward.

## 2.4 Uncertainty

An agent could reason assuming there is no uncertainty or it could reason taking its uncertainty about the domain into consideration. This will be divided into two dimensions: one about the uncertainty about the effect of actions, and one about the uncertainty from sensing. Different domains can have different degrees of uncertainty in action effects or sensing.

In each dimension, you can either have

- no uncertainty
- disjunctive uncertainty, where you can state that some set of worlds are possible, but not specify which one actually holds
- probabilistic uncertainty, where as well as specifying which worlds are possible, you put a probability distribution over the worlds.

There are three main reasons for the extra power of probabilities. The first is that probabilities are what is needed for making decisions [5, 6]. The second reason is that probabilities are what can be obtained from data (Bayes rule specifies how to combine background knowledge with data). The third is that you don't need to make any a priori assumptions of what is possible and impossible, but you can use empirical data to make informed decisions. In many domains, you can't a priori rule out any possibilities.

**Sensing Uncertainty** In some cases you can observe the state of the world. For example, in some board games or on a factory floor, you may know the state of the world. In many other cases you only have some noisy perception of the state, and the best you can do is to have some probability distribution over the states based on what you perceive. For example, a medical doctor may only have a probability distribution over what diseases a patient may have based on observing the patient's symptoms.

The fourth dimension is whether the agent can determine the state from the observations:

- **Fully-observable** is where the agent knows the state of the world from the observations.
- **Partially-observable** is when the agent doesn't observe the state of the world, but there can be many states that possible given an observation. An agent that uses disjunction to state uncertainty can conjoin the observations to its knowledge. When it is using probabilities to measure uncertainty, an agent needs to condition on the observations.

**Effect Uncertainty** In some cases you know the effect of an action. That is, given a state and an action, you can predict the result of carrying out that action in that state. For example, you may be able to predict the effect of deleting a file given the state of your computer system. In many cases, it is difficult to predict the effect of an action, and the best you can do is to have a probability distribution over the effects. For example, you may not know the effect of calling your dog, even if you knew the state of the dog. But based on experience you have some idea of what it will do. You may even have some idea of what some other dog you have never seen will do; otherwise you will not have any reason to call a dog if you want it to come.

The dynamics can be:

– **Deterministic** when the state resulting from carrying out an action in state is determined from the action and the state
– **Stochastic** when there is uncertainty over the states resulting from executing a given action in a gives state. This either means a disjunction over possible resulting states or a probability over the resulting states.

### 2.5   Goals versus complex preferences

An agent may have a simple goal which is a state to be reached or a proposition to be true, such as to get someone a coffee (i.e., end up in a state where they have coffee). Other agents often have more complex preferences. For example, a medical doctor may be expected to to take into account suffering, life expectancy, quality of life, monetary costs (for the patient, the doctor and society), the ability to justify decisions in case a law suits, the short-term and long-term effects, and to trade these off when they conflict.

– **achievement goal** is a goal to achieve. This can be a complex logical formula.
– **complex preferences** that may involve tradeoffs between various desiderata, perhaps at different times.

For complex preferences, you can distinguish ordinal preferences where it is only the ordering of the outcomes that is important, and cardinal preferences where the actual values matter, for example if you are adding them or have expectations over them.

This preference models interacts with the other dimensions. In particular, when there is uncertainty, you typically want to reason about how bad the state is when you don't achieve a goal. There are standard arguments that agents should adopt a cardinal preference measure, known as utility, when dealing with preferences under uncertainty (see e.g., [6]).

### 2.6   Single or multiple agents

Reasoning about a single agent can be much easier than reasoning about multiple agents mainly because, in a multiple agent setting, an agent needs to reason strategically; the other agents may act to trick or manipulate it. The is particularly apparent when reasoning in partially observable environments or with simultaneous actions when it is often optimal to act randomly because other agents can exploit deterministic strategies. Even

when the agents are cooperating and have a common goal, the problem of coordination and communication makes multi-agent reasoning challenging, and it is still sometimes optimal to act stochastically.

The next dimension is:

- **single agent** is where an agent assumes that any other agents are part of the environment. This is a reasonable assumption if the other agents are not going to change what they do based on the agent's action.
- **multiple agent** reasoning is when the reasoning of other agents needs to be taken into account. This happens when there are multiple intelligent agents whose goals or preferences depend, in part, on what other agents do.

It turns out that reasoning in the presence of other agents is much more difficult if the agents act simultaneously or if the environment is partially observable.

### 2.7 Learning from experience

In some cases, you may have a good model of the agent and the environment, but often you don't have a good model, but need to use data from past experiences to help decide good actions. The next dimension is whether:

- **knowledge is given**.
- **knowledge is learned from data or past experience**. That is, the knowledge needed to decide what to do is not provided as part of the model.

Learning is a huge field in itself, but doesn't stand in isolation from the rest of AI. Often there is considerable background knowledge that is used to determine what data to collect, the structure of the data, and to design the resulting representations that need to be learned.

### 2.8 Perfect rationality versus bounded rationality

Sometimes an agent can derive what the best action is if it were given enough time, but often there are computational limitations that prevents the computer from being able to find the best action soon enough to be able to act on it. In some domains, it may not much use to take ten minutes to derive what the best thing to do ten minutes ago, when the agent has to act *now*. Often instead we need to trade off how long it takes to get a solution with how good the solution is; it may be better to find a reasonable solution quickly than find a better solution later, because the world will have changed during the computation. This is known as bounded rationality [3, 7, 8]. The next dimension is whether there is:

- **perfect rationality** where an agent reasons about the best action without taking into account its limited computational resources
- **bounded rationality** where an agent decides on the best action that it can find given its computational limitations.

Computational limitations include computation time, memory and numerical accuracy (caused by computers not representing real numbers exactly).

| | CP | MDPs | IDs | RL | POMDPs | GT |
|---|---|---|---|---|---|---|
| hierarchical | ✔ | | | | | |
| properties | ✔ | | ✔ | ✔ | | |
| relational | ✔ | | | | | |
| indefinite stage | ✔ | ✔ | | ✔ | ✔ | |
| stochastic dynamics | | ✔ | ✔ | ✔ | ✔ | ✔ |
| partially observable | | | ✔ | | ✔ | ✔ |
| values | | ✔ | ✔ | ✔ | ✔ | ✔ |
| dynamics not given | | | | ✔ | | |
| multiple agents | | | | | | ✔ |
| bounded rationality | | | | | | |

**CP** classical planning
**MDPs** Markov decision processes
**IDs** Influence Diagrams
**RL** Reinforcement Learning (classical formulation, such as Q-learning)
**POMDPs** Partially observable Markov decision processes
**GT** Game theory representations of strategic and extensive form of a game.

**Fig. 1.** Some classic representations in terms of the dimensions

## 3 Traditions in AI Agent research

In this section, we show how many of the traditions of agents in AI (and other disciplines) can be seen in terms of these dimensions.

Much of the work in AI can be seen as extending one of the base formalisms to cover more of the dimensions. The base formalisms we consider are the first-order predicate calculus, Markov decision processes and the extensive form of a game.

### 3.1 Logical AI

Logical AI has a long tradition of reasoning about agents, particularly in the planning community [9–11]. This work is based on the foundation of the first-order predicate calculus, which has the notion of objects and relations at the forefront. Logical AI had traditionally eschewed probabilities, even claiming they were "epistemologically inadequate" [12]. We venture to suggest that this is because logical AI had traditionally been dealing with objects and relations; and in some sense, combining this dimension with the other dimensions has been the most challenging. To make progress in some of the other dimensions, it seems as though you need to (at least initially) give up on reasoning with objects and relations.

The planning work has traditionally concentrated on indefinite horizon problems with goals. There has been considerable work on hierarchical planning (see e.g., [13]). There has also been considerable work on multi-agent reasoning in this tradition [14], which is quite distinct from the work based on game theory that is described below.

### 3.2 Decision-theoretic Planning and Reinforcement Learning

A different tradition in AI is based on the foundations of Markov decision processes (MDPs) [15]. An MDP is a state-based model with stochastic dynamics, infinite or indefinite state, and is fully observable.

The idea of decision theoretic planning [16] is to extend Markov decision processes to reason in terms of features. This factored MDP model has been extended to be relational [17, 18] and also to hierarchical domains [19]. The MDP model has also been extended to be partially observable in the POMDP model [20].

Reinforcement learning [4, 21] has been a core part of AI since Samuel's checkers player. In the 1980's there was considerable advance in reinforcement learning following the formalization in terms of Markov decision processes (and so defined in terms of states). In these frameworks, agents can learn what to do (sometimes, but not always, by learning a model). In terms of the dimensions, these methods allowed stochastic dynamics, but assumed full observability. They were infinite stage (which can also implement indefinite stage problems). Except for toy problems for which you can reason in terms of states, virtually all current reinforcement learning is in terms of features (see e.g., [22]), using some sort of function approximation over the features. There has been notable work extending reinforcement learning to be hierarchical [23], and for multiple agents [24].

### 3.3 Multi-agent Systems

There is also a large body of work that starts from reasoning with multiple agents, which is the subject of game theory [6]. There are two classic representations, namely the strategic form of a game or the extensive form of a game. Partial observability makes games much more difficult. With agents taking turns and full observability, the minimax algorithm (with $\alpha$-$\beta$-pruning) can solve or approximately solve quite large games (e.g., chess). With simultaneous moves or partial observability, the story is much more complicated as sometimes stochastic policies are optimal; you have to reason about how the other agents are reasoning about you.

There has been considerable work on factored representations of games, i.e., reasoning where some aspect of the game is described in terms of features (e.g., [25, 26]) and learning in games (e.g., [27]).

## 4 Difficulty in Combining Dimensions

Unfortunately these dimensions interact with each other in complex ways. Much of the work cited in the previous sections can be seen as extending one of the base formalisms (such as logic, MDPs or the extensive form of a game) to include other dimensions.

The partial observability dimensions is the most problematic. Partial observability makes muti-agent reasoning much more complicated, and also it makes the indefinite and infinite horizon problems more complicated (e.g, POMDPs are much more complicated to reason with than MDPs). It also makes effect uncertainty redundant, as stochastic effects can be modeled as deterministic actions with hidden variables.

The modularity dimension interacts with other dimensions in complicated ways. The levels of abstraction interact in complex ways with uncertainty: at one level of abstraction an action may be deterministic whereas at another level it may be stochastic. For example, in the partial observability dimension, consider the result of flying to Perth. At one level of abstraction you may know where you are (in Perth), at a lower level of abstraction, you may be quite lost and not know where you are on a map of the airport. At an even lower level of abstraction, e.g., at a level responsible for maintaining balance, you may know where you are: you are standing on the ground. At a higher level of abstraction you may be very unsure whether you have impressed your companion. The levels abstraction also interacts in complex ways with the succinctness dimension; some levels of abstraction may be simple enough to be able to reason in terms of a finite set of states, whereas other levels of abstraction may require reasoning about individuals and relations. In going to Perth, you may be able to reason about your location as a state, but at a lower level of abstraction, you may need to reason about your multiple pieces of luggage and how to collect them all before leaving the airport. And at an even lower level of abstraction, your balancing mechanism may only have a few states to distinguish.

Three of these dimensions promise to make reasoning simpler, although exactly how to do this remains problematic.

 – Simon [3] argues that any natural or artificial complex system must be hierarchical otherwise the complexity makes the design (by people or by nature) impossible.
 – Reasoning in terms of individuals and relations also promises to make systems simpler. Representing using objects and relations is more compact, and we would hope that we can exploit this complexity in reasoning, without mapping into features (or states). There has been a huge advance, particularly under the umbrella of "graphical models", that shows how representations in terms of features can be exploited for reasoning. Reasoning in terms of individuals and relations should be able to provide more computational leverage (e.g., by reasoning at a lifted level for logic [28] or for probabilities [29]).
 – Bounded rationality promises to simplify reasoning, by having principled reasons for approximation, but how to do this is even less well understood than the other dimensions.

How to deliver on these promised has been problematic, although there has been significant advances (e.g., [30]).

Note also that adding in individuals and relations, doesn't just mean adding probabilities to relations, as we also need to be concerned about existence uncertainty and identity uncertainty [31, 32], as well as combining the uncertainty with rich ontologies and reasoning about the creation and destruction of objects through time.

## 5   Next 50 years

This dimensional view points to where AI is going in the next 50 years. In essence we need to remove simplifying assumptions in all of these dimensions. This has proved to be very challenging.

Progress can be achieved through several paths, for example, hierarchical relational reinforcement learning in partially observable domains. An alternative may be starting from logical representations and adding uncertainty and combining it with inductive logic programming to learn the structure of the dynamics and the rewards. You could also imagine starting at POMDPs and adding learning and relations. Whether these end up at the same point or which one will lead to an intelligent system the quicker is an open problem. The point of this paper, is that no matter where we start from, we want to have the complex in *all* of the dimensions of this paper.

In some sense adding the relational has proved to be problematic. However, it isn't clear that the problem isn't as much sociological as technical. There are really sophisticated logic-based representations that haven't been extended into the other dimensions of this paper. In order to combine them with the other dimensions, they probably need to be simplified; we can only really understand the complex in terms of the simpler components. It isn't clear that the sophistication will survive the combination; the representations that will survive may not be sophisticated in all of the dimensions.

## 6  Conclusion

This paper has tried to place the history of AI in terms of a number of dimensions, where the dimensions are not alternatives, but where an intelligent system needs the complex value in all of these dimensions. Much of the history of AI can be seen as growing from the simplest of these dimensions. We are currently writing a textbook based on these ideas (the second edition of [33]), and we would like to create a discussion about the components of a really intelligent agent. Hopefully this discussion can also encourage people to work on the big picture as well as the narrow fields of AI.

## References

1. Mackworth, A.K.: On seeing robots. In Basu, A., Li, X., eds.: Computer Vision: Systems, Theory, and Applications. World Scientific Press, Singapore (1993) 1–13
2. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Second edn. Series in Artificial Intelligence. Prentice-Hall, Englewood Cliffs, NJ (2003)
3. Simon, H.: The Sciences of the Artificial. third edn. MIT Press, Cambridge, MA (1996)
4. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Canbridge, MA (1998)
5. Neumann, J.V., Morgenstern, O.: Theory of Games and Economic Behavior. Third edn. Princeton University Press, Princeton, NJ (1953)
6. Myerson, R.B.: Game Theory: Analysis of Conflict. Harvard University Press, Cambridge, MA (1991)
7. Horvitz, E.J.: Reasoning about beliefs and actions under computational resource constraints. In Kanal, L., Levitt, T., Lemmer, J., eds.: Uncertainty in Artificial Intelligence 3. Elsevier, New York (1989) 301–324
8. Russell, S., Wefald, E.: Do the Right Thing: Studies in Limited Rationality. MIT Press, Cambridge, MA (1991)
9. Allen, J., Hendler, J., Tate, A., eds.: Readings in Planning. Morgan Kaufmann, San Mateo, CA (1990)

10. McDermott, D.: Robot planning. AI Magazine **13**(2) (Summer 1992) 55–79
11. Weld, D.: Recent advances in ai planning. AI Magazine **20**(2) (1999)
12. McCarthy, J., Hayes, P.J.: Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, M., Michie, D., eds.: Machine Intelligence 4. Edinburgh University Press (1969) 463–502
13. Yang, Q.: Intelligent Planning: A Decomposition and Abstraction-Based Approach. Springer–Verlag, New York (1997)
14. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley and Sons, Chichester, England (2002)
15. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley and Sons, New York (1994)
16. Boutilier, C., Dean, T., Hanks, S.: Decision-theoretic planning: Structual assumptions and computational leverage. Journal of Artificial Intelligence Research **11** (1999) 1–94
17. Poole, D.: The independent choice logic for modelling multiple agents under uncertainty. Artificial Intelligence **94** (1997) 7–56.
18. Sanner, S., Boutilier, C.: Approximate linear programming for first-order mdps. In: Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI-05), Edinburgh (2005) 509–517
19. Guestrin, C., Gordon, G.: Distributed planning in hierarchical factored MDPs. In: Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI 2002), Edmonton (2002) 197–206
20. Kaelbling, L., Littman, M., Cassandra, A.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101** (1998) 99–134
21. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research **4** (May 1996) 237–285
22. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming. Athena Scientific, Belmont, Massachusetts (1996)
23. Dietterich, T.G.: Hierarchical reinforcement learning with the maxq value function decomposition. Journal of Artificial Intelligence Research **13** (2000) 227–303
24. Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence **136** (2002) 215–250
25. Kearns, M., Littman, M., Singh, S.: Graphical models for game theory. In: Proc. UAI-01. (2001)
26. Jiang, A.X., Leyton-Brown, K.: A polynomial-time algorithm for action-graph games. In: National Conference for Artificial Intelligence (AAAI-06). (2006)
27. Brafman, R.I., Tennenholtz, M.: A near-optimal polynomial time algorithm for learning in certain classes of stochastic games. Artificial Intelligence **121**(1-2) (2000) 31–47
28. Robinson, J.A.: A machine-oriented logic based on the resolution principle. Journal ACM **12**(1) (January 1965) 23–41
29. Poole, D.: First-order probabilistic inference. In: Proc. Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico (2003) 985–991
30. Guestrin, C., Koller, D., Gearhart, C., Kanodia, N.: Generalizing plans to new environments in relational MDPs. In: International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico, (2003)
31. Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D.L., , Kolobov, A.: Blog: Probabilistic models with unknown objects. In: Proc. IJCAI-05, Edinburgh (2005)
32. Carbonetto, P., Kisynski, J., de Freitas, N., Poole., D.: Nonparametric Bayesian logic. In: Proc. Twenty First Conf. on Uncertainty in Artificial Intelligence, Edinburgh (July 2005)
33. Poole, D., Mackworth, A., Goebel, R.: Computational Intelligence: A Logical Approach. Oxford University Press, New York (1998)