

Designing a Reinforcement Learning-based Adaptive AI for Large-Scale Strategy Games

Charles Madeira¹, Vincent Corruble¹ and Geber Ramalho²

¹Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie (Paris 6)
4 Place Jussieu
72252 Paris Cedex 05 France
{charles.madeira@lip6.fr, vincent.corruble@lip6.fr}

²Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851, Cidade Universitária
50732-970 Recife, PE Brazil
glr@cin.ufpe.br

Abstract

This paper investigates the challenges posed by the application of reinforcement learning to large-scale strategy games. In this context, we present steps and techniques which synthesize new ideas with state-of-the-art techniques from several areas of machine learning in a novel integrated learning approach for this kind of games. The performance of the approach is demonstrated on the task of learning valuable game strategies for a commercial wargame.

Introduction

Large-scale strategy games are characterized by hundreds of units that interact in a large and stochastic environment to achieve a long-term common goal. These units cannot determine with certainty what effects their decisions have with respect to the goal of the group. That is why considering the potential situations a unit could encounter, and specifying unit behavior adequately in advance, is a very hard problem (Buro 2003).

Despite these difficulties, actions of automated opponents in these games are very often selected by applying a rule base or a related technique (Rabin 2002; Nareyek 2004). Moreover, rule-based systems have no adaptation capability, which is a drawback for large-scale strategy games, where any strategic shortcoming can soon be exploited by the opponent (Corruble 2000; Schaeffer and van den Herik 2002; Nareyek 2004). Consequently, an important goal of the game AI, i.e., to continuously challenge the user at his/her level, can be easily missed, jeopardizing the game replay value.

Moreover, it is widely recognized in the AI community that one of the most important features of strategic decision-making is the capability to adapt and learn. Nevertheless, the high complexity found in large-scale strategy games (for instance, the combinatorial explosion in our case study leads to huge state and action spaces in the order of 10^{1900} and 10^{200} respectively) still constitutes

important theoretical and practical challenging issues to state-of-the-art machine learning algorithms.

In this paper, we try to answer important and hard questions: Is machine learning applicable to large-scale strategy games? Can it lead to an AI with higher performances? Can this be done in a reasonable¹ time? In this context, we develop some new ideas and combine them with existing techniques into a novel integrated learning approach based on reinforcement learning (RL). We investigate and propose innovative solutions for the following key issues: (1) structure of the decision-making problem; (2) abstraction of state and action spaces; (3) acquisition of valuable training data; (4) learning in stages; and (5) generalization of game strategies. It is the unified and simultaneous treatment of all these issues that lets us produce an efficient approach, which we successfully apply to a wargame, John Tiller's *Battleground*TM (Talonsoft).

RL to Large-Scale Strategy Games

Reinforcement learning is a general approach for learning sequential decision-making strategies through trial-and-error interactions with a stochastic and unknown environment (Sutton and Barto 1998).

RL distinguishes itself from other learning techniques by important characteristics: (1) there is no supervisor; (2) it is based on the principle of trial-and-error; and (3) it is guided by an anticipated estimate reward that agrees to desired results (a fixed goal, without requiring a direct description of the agent behavior). Hence, RL is very useful in the situations where effective behavioral strategies are unknown or not easily implemented. Consequently, it is a quite promising approach for large-scale strategy games.

Moreover, RL algorithms have already obtained significant practical results. One of the most impressive and well known applications of RL is TD-Gammon, in which a very successful game strategy was learned

¹ Reasonable here needs to be interpreted in two different contexts: (1) the one of the learning algorithm chosen in terms of convergence, and (2) the one of the game developer in terms of development cycle.

automatically for the game of Backgammon (Tesauro 2002).

However, the success of TD-Gammon cannot be easily repeated in large-scale strategy games. Although the high level of complexity found in Backgammon (state space in the order of 10^{20} and action space in the order of 10^2), it is dwarfed by the one found in the kind of games we are interested in here.

In order to illustrate the complexity of a large-scale strategy game, we take a look at our case study: Battleground™. Battleground™ is a turn-based commercial wargame that simulates a confrontation between two armies at historical battlefields. Its scenarios model units which move on maps composed of hexagons.

Taking into account a simple scenario of Battleground™, the allied army contains 128 units, the enemy army contains 102 units, and 3 objectives are placed on a map composed of 700 hexagons. The combinatory explosion of the possible situations leads to a huge state space in the order of 10^{1887} . Moreover, if we consider this decision-making problem with a centralized perspective, the complexity grows exponentially with the number of units involved (Guestrin et al. 2003). Then, this would lead to a maneuver action space in the order of 10^{231} and a combat action space in the order of 10^{131} for the allied army.

In order to tackle the curse of dimensionality, our approach, detailed in the next section, follows a direction adopted in the last years by the field of RL: the use of prior domain knowledge may allow much faster learning.

An Integrated Learning Approach for Large-Scale Strategy Games

Structuring the Decision-Making Problem

From an agent perspective, a decision-making architecture may be organized as an agent society where agents have specific roles to play. Given that societies in large-scale strategy games are often armies, we can easily identify that a hierarchical structure of command and control is a natural organization where the agents' decision-making is carried out at several levels, ranging from strategic to tactical decisions.

In this context, different levels of the hierarchy may correspond to different granularities of the state and action spaces in order to decrease the learning complexity for each specialized agent subgroup. Moreover, hierarchies can provide coordination mechanisms where agents interact in order to work as a team.

Abstracting State and Action Spaces

In strategic and tactical levels of operation, spatial information provides the most important context for analysis of sensed data (Grindle et al. 2004). This crucial information can be obtained by techniques of abstraction or terrain analysis. In this context, understanding the

terrain is especially useful because tactical locations can be determined, such as ideal locations for scouting parties, best line of sight/fire and also the ability to hide troops and equipment.

Following these ideas, we designed an algorithm for the automatic generation of abstracted state and action spaces, as detailed in (Madeira, Corruble, and Ramalho 2005).

Acquiring Valuable Training Data

Our approach generates experience for learning by using a "hand-made" (non-learning) AI agent to train against. For our application, a basic AI agent composed mainly of tactical (or low-level) actions can be built relatively easily, designed by using techniques such as rule bases. Then, we let our system (called the *learning AI*) play and learn against this pre-built AI (called *bootstrap AI*). After learning against the bootstrap AI, the self-play option can be considered to further improve the learnt strategy.

Learning in Stages

Learning a good strategy for all levels of a hierarchy at once remains a very complex task to tackle as a result of the extremely high dependence of the global performance on the performances of individual levels of the hierarchy (Stone 2000). For this reason, we use a bootstrap mechanism by again taking advantage of the hierarchical decomposition. This mechanism articulates some form of incremental learning by letting the learning AI take only partial control over its hierarchy, where the bootstrap AI takes control of the subordinate levels of the learning AI hierarchy, in addition to the full control of the opponent hierarchy. This allows cooperation between the learning AI and the bootstrap AI, leaving only a small part of the decision-making process to be learnt at any given time. This is an important point of our approach because it allows learning in stages (top-down approach) with increasingly refined levels of specificity and detail.

Generalizing Game Strategies

Finally, our approach uses parameterized function approximators in order to generalize between similar situations and actions. This is crucial to be able to make interesting decisions in the situations which one has never encountered before. However, function approximation in the context of RL is more difficult than in the classical supervised learning setting (Ratitch and Precup 2002) and still present important difficulties when applied in complex domains. Despite this, in practice, the use of function approximators has already proven quite successful (Tesauro 2002).

Experiments and Results

We used a complex scenario of Battleground™, representing a battle between French and Russian armies in 1812. We configured our system (the learning AI) to

control the decision-making at high level of the French army giving strategic orders to a subordinate level, while the Battleground™ existing AI (the bootstrap AI) (1) controls the Russian army, and (2) follows French orders flowing down the hierarchy from the level controlled by the learning AI. Commander's decision-making is implemented in a fully distributed fashion, allowing the use of different techniques for coordinating the agents.

We implemented the gradient-descent Sarsa(λ) algorithm (Sutton and Barto 1998) combined with a function approximator (multilayer artificial neural network (NN) with back-propagation of errors). We used two fully-connected cascade feed-forward architectures with 80-hidden neurons as NN: (1) *NN LAI 1*: a unique neural network with 64 inputs (abstracted state representation) and 45 outputs (abstracted action space); and (2) *NN LAI 45*: 45 neural networks with 64 inputs and 1 output each (a neural network for each action).

The reward function is computed as the change in the game score between the previous and current game turns for all commanders.

The application of our terrain analysis algorithm generated 11 key locations and 11 zones on the map. Moreover, an abstracted state representation composed of 64 continuous variables was built (see Figure 1). After abstraction, the complexity of the scenario was reduced to a state space in the order of 10^{82} , and an action space in the order of 10^5 at corps decision-making level.

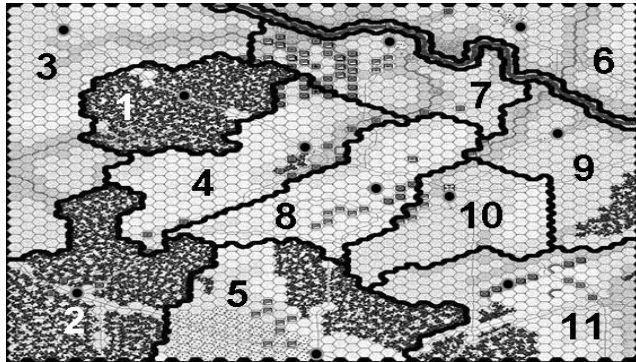


Figure 1. Abstraction of state and action spaces

Each LAI architecture implemented was trained for 10000 games with learning parameter decaying of 10% every 250 episodes and constant exploration rate. We compared our learning agents with other architectures (random agents, the commercial agents, and a human – the first author of this paper – who we consider as an average player), and evaluated all architectures by playing against the commercial agents (Russian army). All architectures used the same configuration as the learning agents, i.e., controlling French corps commanders.

The results indicate that our system has made interesting progress after only a few thousand learning games with an important improvement in average score (see Figure 2). The results place our learning agents close to the average score of the human player.

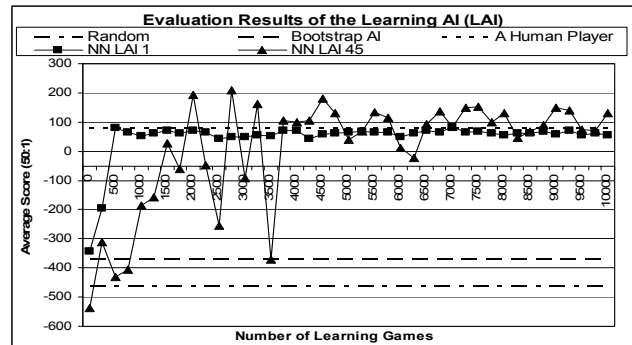


Figure 2. Progress evaluation of the function approximators

Conclusion and Future Work

In this paper, we introduced a novel integrated learning approach based on RL for large-scale strategy games. Our approach synthesizes new ideas with state-of-the-art techniques from several areas of machine learning.

We evaluated the approach on the Battleground™ simulator and obtained very satisfactory results since it outperformed by far the commercial AI. Furthermore, we demonstrated that RL can work in a reasonable time when combined with other techniques. We consider therefore our approach to be a promising direction in the pursuit of high quality game AI for large-scale strategy games.

References

- Buro, M. 2003. Real-Time Strategy Games: A new AI Research Challenge. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.
- Corruble V. 2000. AI approaches to developing strategies for wargame type simulations. AAAI Fall Symposium on Simulating Human Agents. Cape Cod, USA.
- Grindle, C., Lewis, M., Glington, R., Giampapa, J., Owens, S., and Sycara, K. 2004. Automating Terrain Analysis: Algorithms for Intelligence Preparation of the Battlefield. In Proceedings of the Human Factors and Ergonomics Society, Santa Monica, CA.
- Guestrin, C., Koller, D., Gearhart, C., and Kanodia, N. 2003. Generalizing Plans to New Environments in Relational MDPs. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico.
- Madeira, C., Corruble, V., and Ramalho, G. 2005. Generating Adequate Representations for Learning from Interaction in Complex Multiagent Simulations. In Proceedings of the IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology, 512-515, Compiègne, France.
- Nareyek, A. 2004. AI in Computer Games. *ACM Queue*, 1(10).
- Rabin, S. 2002. *AI Game Programming Wisdom*. Charles River.
- Ratitch, B., and Precup, D. 2002. Characterizing Markov Decision Process. In Proceedings of the Thirteenth European Conference on Machine Learning, Finland.
- Schaeffer, J., and van den Herik, H. 2002. Games, computers, and artificial intelligence. *Artificial Intelligence*, 134(1-2):1-8.
- Stone, P. 2000. *Layered Learning in Multi-agent Systems: A Winning Approach to Robotic Soccer*. MIT Press.
- Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.
- Tesauro, G. 2002. Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, 134(1-2), 181-199.