Generating Adequate Representations for Learning from Interaction in Complex Multiagent Simulations

Charles Madeira[1], Vincent Corruble[1] and Geber Ramalho[1,2]

[1]*Laboratoire d'Informatique de Paris 6*
Université Pierre et Marie Curie (Paris 6)
4 Place Jussieu
75252 Paris Cedex 05 France
{Charles.Madeira,Vincent.Corruble}@lip6.fr

[2]*Centro de Informática*
Universidade Federal de Pernambuco
Caixa Postal 7851, Cidade Universitária
50732-900 Recife, PE, Brazil
glr@cin.ufpe.br

Abstract

Wargames are an example of complex multiagent simulations for which, specifying agent behavior adequately in advance for all potential situations is not feasible. In this context, we have applied reinforcement learning as an adaptive approach to design strategies for these simulations. In this paper, we introduce our approach and focus on a novel algorithm for generating representations with adequate granularities for commanders of a military hierarchy.

1. Introduction

Many complex multiagent simulations such as wargames, that are the focus of our interest, are characterized by a stochastic environment, very large state and action spaces, and a large number of units that have to act in parallel to achieve a long-term common goal. In this context, considering all potential situations an agent could encounter, and specifying agent behavior adequately in advance, is difficult for classical artificial intelligence (AI) approaches. Moreover, fixed strategies are dangerous in wargames, as in most games, since any shortcomings can soon be exploited by the opponent.

Therefore, we propose using reinforcement learning (RL) [1] as an adaptive approach to the design of game strategies for these simulations. However, the high complexity encountered in wargames presents also important theoretical and practical challenges to RL state-of-art techniques and algorithms [1]. The complexity involved here concerns firstly the amount of time and data needed to get valuable knowledge on all combinations of agents states and actions. Hence, some kind of abstraction of the state and action spaces is essential to limit the amount of information that is available.

In this paper, we introduce our learning approach and focus especially on a novel algorithm for generating adequate representations in order to deal with the difficulties of applying RL to wargames. We adopt a military hierarchical structure of command and control as a natural organization to facilitate the communication between commanders. Then, a representation with an appropriate granularity is required as an abstraction of the low-level state and action spaces [4].

We have applied our learning approach to the design of an adaptive AI for a commercial wargame, John Tiller's Battleground™ series[1] (Talonsoft), for which we carried out experiments and have obtained very interesting results.

In the next section, we discuss the applicability of RL to wargames, and the generation of accurate representations for RL. Following this, we introduce our algorithm for generating representations. Finally, we present experiments applying our algorithm to a Battleground™ scenario and discuss ongoing work.

2. Background

In this section, we present some elements of background on reinforcement learning in wargames, and military terrain analysis.

2.1. Reinforcement Learning in Wargames

Reinforcement learning is a general approach for learning sequential decision-making strategies from interaction with a stochastic and unknown environment, based on long-term performance criteria [1]. A RL agent learns a behavior through trial-and-error interactions with the environment in order to maximize its cumulative reward over time. In the domain of strategy games, important RL practical results have already been achieved. One of the most impressive and well known is that obtained by the famous TD-Gammon application which

[1] Battleground™ is a turn-based commercial game that simulates a confrontation between two armies at historical battlefields. Its scenarios model units which move on historical maps composed of hexagons. Terrain features as well as units detailed characteristics are taken into account to simulate movement and combat.

was able to play Backgammon at strong master level, equaling the world's best human players [2]. Although the high level of complexity found in Backgammon (state space of approximately 10^{20} and action space of approximately 400), it is dwarfed by the one found in the kind of game we are interested in here.

Taking a centralized perspective, the complexity of decision-making in wargames grows exponentially with the number of units (if there are U units, each with A possible actions, the resulting branching factor is A^U). In addition, this decision-making requires the units to take into account multiple variables (terrain, troops, equipments, visibility, etc.). For instance, given a simple scenario of Battleground™ in which the friendly army contains 128 units and the map is composed of 700 hexagons, each unit may move, per turn, to 64 different locations on average (depending on the unit type and terrain) and the troops may combat 20 enemy units on average (depending on the troop type, weapons and visibility conditions). This would lead to a maneuver action space in the order of 64^{128} and a combat action space in the order of 20^{101}. The state space of this simple scenario is even much larger, since one must consider several parameters for each unit (friendly or enemy). Moreover, the state representation must include information concerning the terrain features. This leads to a state space in the order of 10^{1887}. Consequently, designing strategies and determining what effects decisions have towards the goal of the game is a hard problem [3].

Considering this, we focus on the fact that many complex systems can be decomposable into local subsystems, interacting only weakly between themselves [8]. This allows learning for each subsystem relatively independently. In wargames, we can easily identify that using the military hierarchical structure of command and control is a natural way to decompose the problem. In this structure, the ability for decision-making is concentrated in specific commanders at each level of the hierarchy. Once a commander has received an order, he is free to use his domain-specific knowledge and available local information to take his own decisions to satisfy the order. The main benefit is that it allows high-level commanders to take strategic decisions to achieve high-level goals, only considering information at an appropriate level of detail. Tactical decisions are only taken by lower-level commanders. In this context, taking appropriate decisions at strategic and tactical levels is largely dependent on the availability of accurate information about the environment. Accurate information can be obtained as an abstraction of the low-level representation [4].

2.2. Abstraction and Terrain Analysis

Knowledge about the military domain is needed in order to generate accurate representations at different granularities for each level of the hierarchy. In military operations, spatial (or terrain) information is of crucial importance to commander decision-making. Terrain provides important context for analysis of sensed data as well as for guiding the tasking of data collection features [7]. This important information can be obtained by techniques of abstraction known as terrain analysis.

Terrain analysis consists in interpreting natural and man-made features of a geographic area to determine their effects on military operations [7]. The analysis must support both strategic and tactical levels of operation. Strategic operation is considered to be high-level decision-making where commanders can obtain a broad overview of the battlefield. Tactical analysis provides the commander with a much more detailed view of specific areas-of-interest on the battlefield.

Qualitative spatial reasoning (QSR) techniques can help the process of terrain analysis in wargames [5]. The essence of qualitative reasoning is to make explicit the essential knowledge, finding ways to represent continuous properties of the world by discrete systems of symbols [6]. Hence, using appropriate reasoning techniques, one could make predictions, diagnose and explain the behavior of physical systems in a qualitative manner without recourse to an often intractable or perhaps unavailable quantitative model. By using this conceptually meaningful information, reasoning strategies are able to achieve results that are more human-like.

3. Generating Representations

In this section, we introduce our algorithm for generating adequate representations. This algorithm is a key component of our integrated learning approach. The approach develops some new ideas and combines them with state-of-the-art techniques, proposing innovative solutions for some key issues: (1) decomposition of the decision-making process by using the natural hierarchical structure of the domain; (2) abstraction of the state and action spaces by automatically generating adequate representations; (3) acquisition of valuable training data by adopting a particular bootstrap mechanism; and (4) generalization of the learnt experience by using function approximators on a problem of very high dimension.

The functional capability of our algorithm (see Figure 1) is a terrain analyzer to support commander decision-making in Battleground™. It abstracts low-level information about any scenario of Battleground™ to higher level (or topological) concepts. This abstraction takes inspiration from QSR techniques in order to find key tactical locations (villages, crossroads, bridges, top of hills, etc.) and key strategic zones (forests, rivers, hills, etc.) on the map. A key location is any position on the map whose control is likely to give distinct military advantage to the force that holds it. Such information, combined with information about possible enemy characteristics and force structure, provide measures of ease of movement of forces throughout the terrain.

Our algorithm takes into account some military features of a given scenario such as terrain topography, position of the units of each army and objectives affected. As a result, the map is partitioned into zones, each one representing different categories of mobility. Mobility is a key spatial constraint that allows calculating the ability of a unit to move across a specified terrain type. Then, description key variables (e.g., unit's quality, formation, position, visibility, strength, fatigue, mobility, etc.) that are considered essential for each specific level of the hierarchy are selected, allowing the generation of adequate state representations for applying RL.

Input:

- A scenario of the game (a map, position of the troops, and objectives)
- Key variables (leader and troop abilities)

Output:

- Key locations
- Partition of the map into zones
- A state representation composed of two main groups of information (variables describing troops and their environment)

Main procedure:

1st Step (Tactical analysis): Determining key locations (hexagons)

- Objectives, villages, forts, top of hills, crossroads, bridges, etc.

2nd Step (Tactical analysis): Partitioning the map into zones

- Rivers (as major obstacles) delimitate static zones
- Forests constitute initial zones that can be merged in the final of this step
- A progressive expansion of the key locations is made by adding neighboring hexagons until a full partitioning of the map into zones has been reached
- Neighboring zones can be merged depending on the hierarchy level

3rd Step (Building a representation): Troops and their environment descriptions are generated based on the results of the tactical analysis and input key variables

Figure 1. Sketch of algorithm for generating representations

The main procedure is composed of three steps. In the first one, a tactical analysis evaluates features of the terrain in order to determine key locations on the map that are also used as an abstraction of the action space. Each one of the key locations corresponds to a node of a graph which is associated to an identifier, a type, a coordinate on the map, and a list of identifiers of other connected nodes.

In the second step, firstly the terrain locations of type forest, orchard, and river are located on the map. Each hexagon of these types is associated to a preexistent zone of the same type if there is one adjacent, otherwise a new zone is created. Secondly, the key locations identified previously are used as seeds to be expanded progressively by adding neighboring hexagons until a full partitioning of the map into zones has been reached. The rate of expansion (number of hexagons absorbed by the zones) at each iteration is directly linked to the mobility rate of maneuver for each type of terrain. The hexagons of a given zone try to agglomerate their neighbors that do not yet belong to any zone. Each one of these free hexagons has a token for each existing zone. The tokens are calculated according to many geographical features (terrain types, elevation, etc. found between a hexagon of a zone and a free hexagon). The higher the token of a free hexagon is to a zone, the more quickly the hexagon is aggregated. The expansion of each zone normally finishes when encountering another zone or a border of the map. The main loop stops when no free hexagon remains.

In the last step, a state representation is built based on the results of the two first steps, by a projection onto a list of key variables. The structure of this final representation contains two main groups of data: (1) a relatively detailed description of the group of units for which an order is under consideration (commanders at all levels of the hierarchy see these low-level units organized in groups). The characteristics of each group are a summary of the characteristics of individual units; (2) a summarized description of their environment (in terms mainly of the repartition of friendly and enemy forces across all zones).

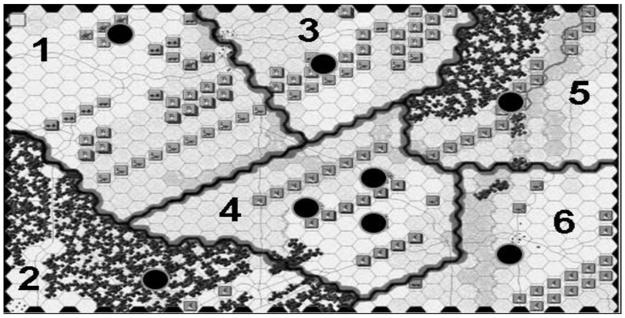
4. Experiments

In this section, we present experiments using our approach applied to Battleground™. In order to carry them out, we chose the game scenario introduced in the section 2.1 which represents different phases of the battle of *Borodino* between French and Russian armies near Moscow in 1812.

Considering the fact that the military hierarchical organization is useful for allowing abstraction of the complex state and action spaces in this game, it also opens itself to a mechanism of learning by levels of the hierarchy (in an incremental manner). Following this idea, we configured our system to learn the decision-making of each level of the hierarchy independently. In this context, we use as opponent to our "learning AI" another AI system (the commercial AI included in Battleground™) developed by wargame experts to perform this incremental learning mechanism.

In the following, we present the application of our algorithm to the higher level of the hierarchy. In this context, the learning process is carried out as follows: the commercial agents (a) control totally the Russian army, and (b) follow orders flowing down the hierarchy from the subordinate commanders for the French army. As a result, our system (the learning agents) controls only the decision-making at the level of the French army commander giving strategic orders to the subordinate level, composed of 3 subordinate commanders. In these experiments, the learning agents learn strategies for the 3 subordinate commanders, coordinating them implicitly.

As a result of the application of the algorithm, 8 key locations and 6 zones were obtained on the game map, and a state representation composed of 32 continuous variables was built (see Figure 2).



A Corps description (8 variables)

- Position (x, y) of the troop on the map (2 variables)
- Artillery, cavalry, and infantry strength levels (3 variables)
- Fatigue level, quality rating, and movement allowance (3 variables)

An Environment description (24 variables)

- Strength and fatigue levels of friendly and enemy units in each zone

Figure 2. Abstraction of the state and action spaces. It shows 8 key locations and 6 zones on the map, and a state representation composed of 32 variables

After abstraction, the combination of strategic order types and key locations leads to an action space of 33 possible actions for each of the three subordinate commanders, reducing from 2801^3 to 33^3 the complexity of the army commander decision-making. Moreover, the abstracted state representation reduced the original state space of the scenario approximately from the order of 10^{1887} at all levels to the order of 10^{82} at the higher level.

Our learning agents implement the gradient-descent Sarsa(λ) algorithm [1] combined with a multilayer artificial neural network trained by backpropagating temporal-difference errors. We compared them with other architectures (random agents, the commercial agents and an average human 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 only the decision-making level of the army commander. The results indicate that our system has made interesting progress in only 6000 learning episodes with an important improvement in average score (see Figure 3). The results place our agents close to the average score of the human player.

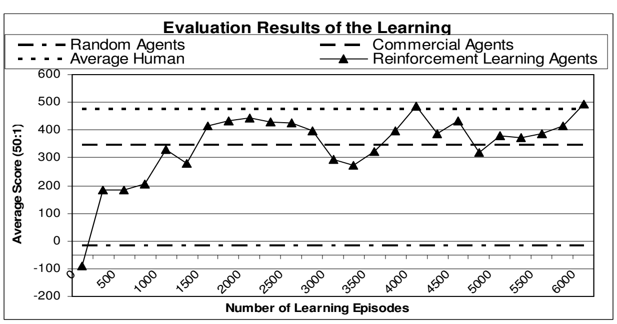


Figure 3. Progress evaluation of the RL agents in a total of 6000 episodes of learning (50 evaluations every 250 episodes)

5. Conclusion and Future Work

We have applied the RL approach on a very complex and original problem. In this paper, we introduced a novel algorithm for generating representations in order to deal with the applicability of RL to complex multiagent simulations such as wargames. The algorithm constructs adequate representations for aiding commanders of a military hierarchical structure, hence leading to tractable, though still large, state and action spaces.

The core of our algorithm takes some inspiration from QSR techniques, aiming for expressive spatial representations through a deep terrain analysis. These techniques exploit particular properties of terrain, being able to describe its essential configurations and features for allowing to construct more human-like representations. In this context, our system is able to learn a game strategy that can better use terrain in order to reach its goals.

We evaluated the algorithm on Battleground™ and obtained very satisfactory results since it outperformed by far the commercial agents and achieved the level of performance of a human player in a reasonable time.

6. References

[1] R.S. Sutton, A.G. Barto, *Reinforcement Learning, An Introduction*. MIT Press, Cambridge MA, 1998.
[2] G. Tesauro, "Temporal Difference Learning and TD-Gammon", *Communications of the ACM*, ACM Press, New York NY, 1995, 38(3) pp.58-68.
[3] M. Buro, "Real-Time Strategy Games: A new AI Research Challenge". *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, Mexico, 2003.
[4] J.-D. Zucker, N. Bredèche, and L. Saitta, "Abstracting Visual Percepts to Learn Concepts". *In Proceedings of the Symposium on Abstraction Reformulation and Approximation*, Canada, 2002, pp.256-273.
[5] K.D. Forbus, J.V. Mahoney, and K. Dill, "How qualitative spatial reasoning can improve strategy game AIs", *In Proceedings of AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, Stanford, CA, 2001.
[6] A.G. Cohn, and S.M. Hazarika, "Qualitative Spatial Representation and Reasoning: An Overview", *Fundamenta Informaticae*, 46(1-2), 2001, pp.1-29.
[7] C. Grindle, and M. Lewis, "Automating Terrain Analysis: Algorithms for Intelligence Preparation of the Battlefield", *In Proceeding of The 48th Meeting of Human Factors and Ergonomics Society*, New Orleans, Louisiana, 2004.
[8] H. A. Simon, *The Sciences of the Artificial*. MIT Press, Cambridge, MA, 1981.

Acknowledgements

Charles Madeira's doctorate work is funded by a scholarship from CAPES, Brazil within the framework of SMART-E's, a collaborative project between UFPE, Brazil, and UPMC, France, sponsored by CAPES-COFECUB.