



Master's thesis in  
Information- and Communication Technology

Title:  
**Playing Axis & Allies Revised Using Learning Automata**

Candidate:  
Gjermund Karlsen Lie

Supervisor:  
Ole-Christoffer Granmo, AUC.

### Introduction

The Artificial Intelligence (AI) of opponents in computer games in general, and in strategy games in particular, have been plagued with performance problems of many kinds since they first appeared. Not the least of these problems is the fact that their design schemes often base themselves on predefined ways to play the game, making these opponents predictable and dull to a seasoned player.

In this thesis, we propose using Learning Automata (LA) to create opponents that are able to adapt to any game situation and find a good response – much in the way a player would – by looking ahead in time to see what could happen in the game beyond the immediate next move. As a suitable environment for these LA, we have chosen the game Axis & Allies Revised. A turn-based war game emulating the second world war, it has many layers of complexity for the LA to struggle with: Multiple moves or attacks per turn, random outcome of combat, and highly complex rules. To play this game well, the artificial opponent would need not only to coordinate all his units into the best combined move each turn, but also to avoid performing moves in the present that it would be punished for during the next turns.

### Proposed Solution

We propose a two-step method for solving this problem: Multi-Tiered Decentralized Learning Automata. First, each unit will be assigned its own, independent LA. This in order to keep the amount of choices a single

LA has open to it at a manageable level, and to allow us to reward each unit individually.

Second, for each possible action that this unit can select in the next immediate turn, another independent LA will be assigned. This process can then be repeated until a sufficient depth into future moves has been achieved. Each tier of LA in this structure will receive its feedback not from its immediate surroundings - but from the status of the next LA down the tree, in order to give it feedback on the results of a set of choices beyond the immediate next turn.

Shown in figure 1 is an example of this structure with two tiers of automata. Each unit has its own tree of automata selecting actions individually, and each automata receives feedback from its children nodes.

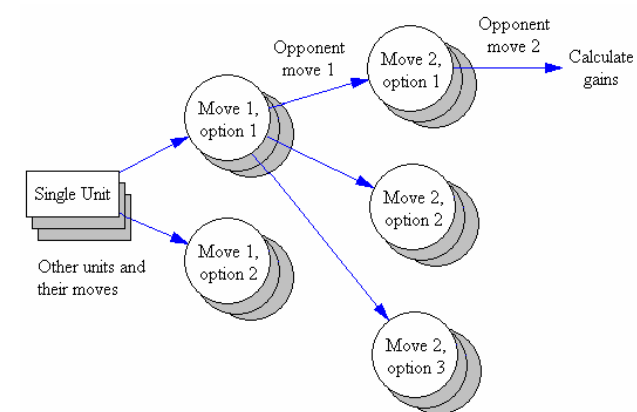


Figure 1: Example structure of Multi-Tiered Decentralized Automata

We lay the foundation for such a solution by implementing the method on a smaller scale, and by carefully testing its performance in a controlled environment. We find which approaches give the best

results, which can only perform under certain conditions, and which are suitable for expanding into larger scale.

The three types of LA chosen for our testing covers most schools of reinforcement learning: The Tsetlin Automata, with no persistent memory and a simple, state based structure. The Linear Reward Inaction (LRI) Automata, with its linear updating scheme. And finally the Bayesian Learning Automata (BLA), shaping conjugate distributions in order to determine the optimal action. Each has their own strengths and weaknesses.

### Results

Each of our automata was tested against scenarios designed to challenge the teams of LA both in terms of immediate reactivity and short-term proactivity. All our types of automata achieved at least some degree of success in each scenario, but the overall best performer was the BLA, with an overall average across scenarios of 94.7% runs converging to the optimal set of choices.

However, the time required to run a simulation of even a small scenario containing few units have proven too high for practical use. Attempts at making as few as three units cooperate with a search depth of three moves ahead against fixed opposition varied from 78 seconds for the fast but imprecise Tsetlin to an average of 440 seconds for the more precise LRI and BLA, using an average home computer.

Shown in figure 2 is the percentage of simulation runs in which each type of, or

combination of automata ended up choosing what had been evaluated as the optimal move in the scenario mentioned above.

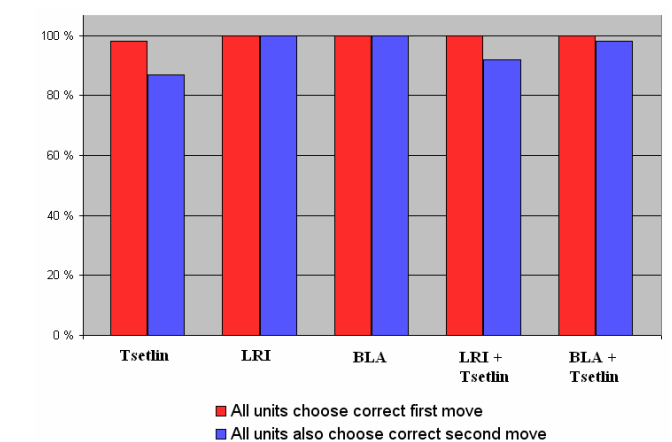


Figure 2: Example rate of convergence to correct movement combination.

### Conclusion

While LA may in fact have the potential to perform well in almost any type of scenario, our testing shows that it would still be impractical considering the time spent on deciding on a move. While the speed of decision making of our LA varies, so does its performance, even in our small scale testing. In addition, our most precise type of LA – the BLA – scales poorly as the amount of simulation runs required to converge increases, thus making its performance for larger scenarios lacking at best.

Nevertheless, we believe that our results should give some insight into the possibilities and benefits, both in terms of performance and design simplicity, of using LA as the decision maker for artificial players.