*IGI PUBLISHING* **ITJ 4729**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.igi-global.com

This paper appears in the publication, International Journal of Agent Technologies and Systems, Volume 1, Issue 2
edited by **Goran Trajkovski © 2009, IGI Global**

# Reinforcement Learning with Reward Shaping and Mixed Resolution Function Approximation

*Marek Grzes, University of York, UK*

*Daniel Kudenko, University of York, UK*

## ABSTRACT

*A crucial trade-off is involved in the design process when function approximation is used in reinforcement learning. Ideally the chosen representation should allow representing as close as possible an approximation of the value function. However, the more expressive the representation the more training data is needed because the space of candidate hypotheses is bigger. A less expressive representation has a smaller hypotheses space and a good candidate can be found faster. The core idea of this paper is the use of a mixed resolution function approximation, that is, the use of a less expressive function approximation to provide useful guidance during learning, and the use of a more expressive function approximation to obtain a final result of high quality. A major question is how to combine the two representations. Two approaches are proposed and evaluated empirically.* [Article copies are available for purchase from InfoSci-on-Demand.com]

*Keywords:* *CMAC; Function Approximation; Heuristics; Hypotheses Space; Mixed Representation; Reinforcement Learning; Reward Shaping*

## INTRODUCTION

Reinforcement learning (RL) agents learn how to act given observations of the world. They execute actions which have some impact on the environment and the environment subsequently provides numerical feedback which can be used to guide the learning process. The RL agents use this information to find a policy which maximises the accumulated reward. A policy determines which action should be taken in a given state and is usually represented as a value function $Q(s,a)$ which estimates 'how good' it is to execute action $a$ in state $s$. The value function can be interpreted in terms of the expected reward which can be obtained when action $a$ will be chosen in state $s$ and the given policy followed thereafter (Sutton & Barto, 1998).

In contrast to supervised learning, RL agents are not given instructive feedback on what the best decision in a particular situation is. This leads to the *temporal credit assignment* problem, that is, the problem of determining which part of the behaviour deserves the reward (Sutton, 1984). To address this issue, the iterative approach to RL applies back-propagation of the value function in the state space. Because this is a delayed, iterative technique, it usually leads to a slow convergence especially when the state space is huge. In fact, the state space grows exponentially with each variable added to the encoding of the environment when the Markov property needs to be preserved (Sutton & Barto, 1998).

When the state space is huge the tabular representation of the value function with a separate entry for each state or state-action pair becomes infeasible for two reasons. Firstly, memory requirements are high. Secondly, there is no knowledge transfer between similar states and a vast number of states needs to be updated many times. The concept of value function approximation (FA) has been successfully used in reinforcement learning (Sutton, 1996) to deal with huge or infinite (e.g., due to continuous variables) state spaces. It is a supervised learning approach which aims at approximating the value function across the entire state space. It maps values of state variables to the value function of the corresponding state.

A crucial tradeoff is involved in the design process when function approximation is used. Ideally the chosen representation should allow representing as close as possible an approximation of the value function. However, the more expressive the representation the more training data is needed because the space of candidate hypotheses is bigger (Mitchell, 1997). A less expressive representation has a smaller hypotheses space and a good candidate can be found faster. Even though such a solution may not be particularly effective in terms of the asymptotic performance, the fact that it converges faster makes it useful when applied to approximating the value function in RL. Specifically, a less expressive function approximation results in a broader generalisation and more distant states will be treated as similar and the value function in this representation can be propagated faster. The core idea of this article is the use of a mixed resolution function approximation, that is, the use of less expressive FA to provide useful guidance during learning and the use of more expressive FA to obtain a final result of high quality. A major question is how to combine the two representations. The most straightforward way is to use two resolutions at the same time. A more sophisticated algorithm can be obtained with the application of reward shaping. The shaping reward can be extracted from a less expressive (abstract) layer and used to guide more expressive (ground) learning.

To sum up: in the article we propose combining more and less expressive function approximation, and three potential configurations are proposed and evaluated:

- the combination of less and more expressive representations in one approximation of the value function,
- the use of less expressive function approximation to learn the potential for reward shaping which is used to shape the reward of learning with desired resolution at the ground level,
- the synergy of the previous two, that is, learning potential from less expressive approximation and using it to guide learning which combines less and more expressive resolution in one FA at the ground level.

Our analysis of these ideas is based on tile coding (Lin & Kim, 1991) which is commonly used for FA in RL. The proposed extensions to RL are however of general applicability and can be used with different methods of function approximation (especially those which use basis functions with *local* support; see, for example, Munos and Moore (2002) for more details).

The rest of the article is organised as follows. In the next section function approximation with tile coding and also reward shaping are introduced. Learning with mixed resolution tile coding is presented in Section 3 and the