

# Advice Taking in Multiagent Reinforcement Learning

Michael Rovatsos and Alexandros Belesiotis  
School of Informatics  
The University of Edinburgh  
Edinburgh EH8 9LE, United Kingdom  
mrovatso@inf.ed.ac.uk

## ABSTRACT

This paper proposes the  $\beta$ -WoLF algorithm for multiagent reinforcement learning (MARL) that uses an additional “advice” signal to inform agents about mutually beneficial forms of behaviour.  $\beta$ -WoLF is an extension of the WoLF-PHC algorithm that allows agents to assess whether the advice obtained through this additional reward signal is (i) useful for the learning agent itself and (ii) currently being followed by other agents in the system. We report on experimental results obtained with this novel algorithm which indicate that it enables cooperation in scenarios in which the need to defend oneself against exploitation results in poor coordination using existing MARL algorithms.

## Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Multiagent Systems*

## Keywords

Multiagent reinforcement learning, communication

## 1. INTRODUCTION

In recent years, the problem of designing multiagent reinforcement learning (MARL) algorithms has received much attention (see [1, 3] for overviews) due to its challenging nature. As opposed to the single-agent case [4] (where the environment exhibits a stationary behaviour) MARL adds an element of *non-stationarity* to the original learning problem since opponents may be adaptive themselves, i.e. their future strategy may be any function of the history of previous system behaviour.

In this paper, we argue that *communication* about certain properties of agent behaviour can be used to tackle some of the fundamental problems of MARL algorithms, and as a first step toward exploiting this basic idea, we consider stochastic games in which an additional “advice” signal is available to agents that provides feedback about optimal joint actions. We present a novel algorithm called  $\beta$ -WoLF

based on WoLF-PHC [1] that enables to *autonomously* decide whether and to which degree they want to follow that advice based on two simple criteria: (1) advice will only be followed if it yields payoffs that are at least as high as an individually rational strategy (*rationality*), and (2) advice will only be followed if other agents are also following it (*mutuality*).

The remainder of this paper is structured as follows: Section 2 introduces the  $\beta$ -WoLF algorithm, section 3 reports on experimental results and section 4 concludes.

## 2. THE $\beta$ -WoLF ALGORITHM

We use the framework of stochastic games (SGs) [2] and extend it with additional reward signals that represent an information source external to the stochastic game itself. For this, we define  $n$ -player *stochastic games with advice*  $\langle n, S, A_1, \dots, A_n, T, R_1, \dots, R_n, W_1, \dots, W_n \rangle$  with states  $S$ , agent action sets  $A_i$  (resulting in a joint action space  $A = \times_{i=1}^n A_i$ ), transition model  $T$ , individual real-valued reward functions  $R_i$  and real-valued individual *advice functions*  $W_i : S \times A \rightarrow \mathbb{R}$  for each agent  $i$ . We assume that the agents’ goal is to learn a stationary (potentially stochastic) policy  $\pi_i : S \times A_i \rightarrow [0 : 1]$  that will maximise expected, discounted future payoff in terms of  $R_i$  *alone*, i.e. not taking  $W_i$  into account – in other words, obtaining advice does not directly affect the agent’s performance. This distinguishes our approach from work on the Collective Intelligence (COIN) framework [5]: we focus on respecting agent autonomy rather than attempting to *design* individual agent reward functions from a birds-eye point of view.

The way SGs with advice work is as follows: Agents observe state  $s \in S$ , execute action  $a_i \in A_i$ . Based on the resulting joint action  $a = (a_1, \dots, a_n) \in A$  and distribution  $\{T(s, a, s') | s' \in S\}$  the successor state  $s'$  is determined. Agent  $i$  receives its reward  $R_i(s, a)$  and the advice signals  $W_j(s, a)$  for all agents  $j \in \{1, \dots, n\}$ .<sup>1</sup> and the next iteration is initiated.

$\beta$ -WoLF essentially consists of a number of WoLF-PHC learning “modules” that learn optimal strategies for different sub-problems and a criterion for coordinating how these components are integrated by the agent to yield a single policy. More specifically, a  $\beta$ -WoLF-agent maintains the following data structures:

1. The *individual reward learner*: A WoLF-PHC learning algorithm used for maximising individual rewards, using a Q-table [6]  $Q(s, a_i)$ , updated using rewards  $R_i(s, a_i)$  for  $a_i \in A_i$ , and evolving a policy  $\pi_i(s, a_i)$ .
2. The *collective reward learner*: A Q-table is maintained for values  $Q'(s, a)$  where  $a \in A$  and updated using the

<sup>1</sup>We justify below why this is necessary.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS’07 May 14–18 2007, Honolulu, Hawaii, USA.  
Copyright 2007 ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

standard Q-update rule given rewards  $R_i(s, a)$  as in  $Q$ , to learn how useful *joint actions* are for the agent considering individual rewards  $R_i(s, a)$  as in  $Q$ .

3. *n individual advice learners*: One WoLF-PHC per agent  $j$  (including  $i$  itself) to simulate  $j$ 's learning process if it followed the external advice  $W_j$  (rather than its actual reward). We denote these Q-tables by  $V_j(s, a_j)$  for  $a_j \in A_j$  using update equation

$$V_j(s, a_j) \leftarrow (1 - \alpha)V_j(s, a_j) + \alpha(W_j(s, a_j) + \gamma \max_{a'_j} V_j(s', a'_j))$$

and the resulting advice-based strategy by  $\rho_j(s, a_j)$ .<sup>2</sup>

4. Using *advice factor*  $\beta \in [0 : 1]$  and an *advice learning rate*  $\delta_\beta \in (0 : 1]$  policy  $\sigma_i(s, a_i)$  is updated using

$$\sigma_i(s, a_i) = (1 - \beta)\pi_i(s, a_i) + \beta\rho_i(s, a_i)$$

adjusting  $\beta$  according to the following criterion:

$$\beta \leftarrow \begin{cases} \min\{1, \beta + \delta_\beta\} & \text{if } \sum_a \prod_j \rho_j(s, a_j) Q'(s, a) > \\ & \sum_{a_i} \pi_i(s, a_i) Q(s, a_i) \\ & \text{and } d|\bar{\sigma}_{-i}(s) - \rho_{-i}(s)|/dt < 0 \\ \max\{0, \beta - \delta_\beta\} & \text{else} \end{cases}$$

Here  $\bar{\sigma}_{-i}$  is the average (posterior) long-term strategy of the remaining agents maintained and updated in the same way as  $\pi_i$  in normal WoLF-PHC.

5. If  $\sum_a \prod_j \rho_j(s, a_j) Q'(s, a) > \sum_{a_i} \pi_i(s, a_i) Q(s, a_i)$ , act according to  $\rho_i$  for  $k$  iterations with probability  $\epsilon/2$  (for some  $\epsilon \in (0 : 1]$ ) and randomly with probability  $\epsilon/2$ . Else, choose a random action with probability  $\epsilon$ . With probability  $1 - \epsilon$  behave according to  $\sigma_i$ .

We discuss each step one by one: First, using normal WoLF-PHC learning, agent  $i$  updates tables  $Q$  and  $Q'$  (whose difference is that  $Q'$  learns estimates for joint, rather than individual actions) and adapts its (individual) strategy  $\pi_i$  while updating long-term average strategies  $\bar{\pi}_i/\bar{\sigma}_{-i}$ . Next, a purely advice-based WoLF-PHC learning process is simulated for *all* agents using tables  $V_j$  and individual agent actions  $a_j$ . The respective strategies  $\rho_j$  will learn to behave optimally “as if” the advice was the actual reward. The actual policy  $\sigma_i$  for agent  $i$  is a convex combination of the reward-based (individual) strategy  $\pi_i$  and the advice-based strategy  $\rho_i$  controlled by the advice factor  $\beta$ . To determine whether following advice is better than applying some locally optimal strategy, we compare the expected utility  $Q'(s, a)$  for agent  $i$  under joint action  $a$  considering the joint action probability resulting from  $\prod_j \rho_j(s, a_j)$  compared to the expected utility of its individually rational strategy  $\pi_i$  that has been evolved disregarding advice completely.<sup>3</sup> To determine whether others are also following the advice, we constantly check whether the distance between the average opponent policy and the advice-based policy  $|\bar{\sigma}_{-i}(s) - \rho_{-i}(s)|$  is decreasing over time, i.e. if the other agents are “approaching” the behaviour that is optimal according to the advice signal. In the simplest case (which we assume in our experiments below) checking this so-called *distance criterion* can be done by verifying whether the current value of this distance is smaller than its immediate predecessor. If both conditions apply,  $\beta$  is increased, and otherwise decreased while making sure that its value is bounded by 0 and 1. Thereby, the advice learning rate  $\delta_\beta$

<sup>2</sup>While this requires knowledge of all  $W_j$  signals by  $i$  each advice learner is only concerned with its *own* actions.

<sup>3</sup>Note that it is this inequality that necessitates maintaining an additional Q-table  $Q'$  for joint action values.

serves to provide some “inertia” in the process of adapting the degree to which advice is taken into account. Finally, as the criteria for rationality and mutuality of advice taking only verify whether *others* are following the advice, there is nothing that would ensure that some agent initiates this kind of behaviour if it is beneficial. Therefore, we require that with half the probability  $\epsilon$  that is used for  $\epsilon$ -greedy exploration, the agent follows a *purely* advice-led strategy for  $k$  steps if the advice is beneficial (using the same criterion as above).

### 3. EXPERIMENTAL RESULTS

We have evaluated the algorithm extensively in a number of two-player games out of which we can only report on results with Iterated Prisoner’s Dilemma (IPD) games here. The calculation of advice was based on the *social welfare*  $R_1(s, (a_1, a_2)) + R_2(s, (a_1, a_2))$  of each joint action, such that the “advice giver” acts as a “passive” RL agent and learns action values  $Q_g(s, a)$  for the global reward of joint action  $a$  in state  $s$  using ordinary Q-learning. To compute the advice given to the two agents, let

$$q_i(s, a) = \frac{Q_g(s, (a_i, a_{-i})) - \min_{a'_i} Q_g(s, (a'_i, a_{-i}))}{\sum_{a_i \in A_i} Q_g(s, (a_i, a_{-i})) - \min_{a'_i} Q_g(s, (a'_i, a_{-i}))}$$

if  $\sum_{a_i \in A_i} Q_g(s, (a_i, a_{-i})) - \min_{a'_i} Q_g(s, (a'_i, a_{-i})) > 0$  and  $q_i(s, a) = \frac{1}{|A_i|}$  else. This calculates the “relative cooperativeness” of each agent  $i$  compared to the most harmful action  $i$  might have performed (we use  $a_{-i}$  to refer to the joint action of all agents but  $i$ ). With this, we can calculate the advice for each agent as  $W_i(s, a) = q_i(s, a)Q_g(s, a)$ . We present results regarding joint action probabilities that result from the  $\sigma$ -distributions of the two agents (rather than *observed* joint action frequencies or average payoffs) to ignore the effects of exploration<sup>4</sup>, and average over 100 repeated simulations<sup>5</sup>; when referring to joint actions  $(X, Y)$  we assume that the  $\beta$ -WoLF agent is agent 1 (i.e. plays  $X$ ). For a single run, the results of  $\beta$ -WoLF in a self-play situation are shown in the top part of figure 1: initially, the agents learn the equilibrium strategy  $(D, D)$ , but the  $\beta$ -value (here shown for one of the two agents only) exhibits occasional “spikes” representing attempts to achieve joint advice following. Eventually the agents succeed in coordinating their attempts and very quickly switch to the Pareto efficient action  $(C, C)$  and while occasional deviation from it (due to exploration) implies that its probability is below one, the agents behave cooperatively most of the time. While we cannot give any guarantees for *when* this switching will occur, in our experiments all 100 simulations converged to an average probability  $(C, C)$ -probability of 1 within 5000 rounds.

Beyond self-play, we evaluated the performance of  $\beta$ -WoLF against other types of (fixed and adaptive) opponents. We ran IPD simulations in which the algorithm has to play against an ALL D (always defects), an ALL C (always cooperates), a TIT FOR TAT, and a “malicious” agent (behaves like  $\beta$ -WoLF for 1500 rounds and then switches to ALL D). Convergence to playing the best response  $D$  with probability 1 is achieved against ALL C and ALL D within little more than 40 rounds over 100 simulations on the average (not shown for lack of space), we have to be a bit more careful about TIT FOR TAT: Here, the  $\beta$ -WoLF agent cannot represent the opponent’s strategy (as it is not merely a

<sup>4</sup>keeping in mind that the cost of  $\epsilon$ -greedy exploration would have to be subtracted from the alleged cumulative payoff

<sup>5</sup>For all reported “convergence” results standard deviation between runs converged to zero.

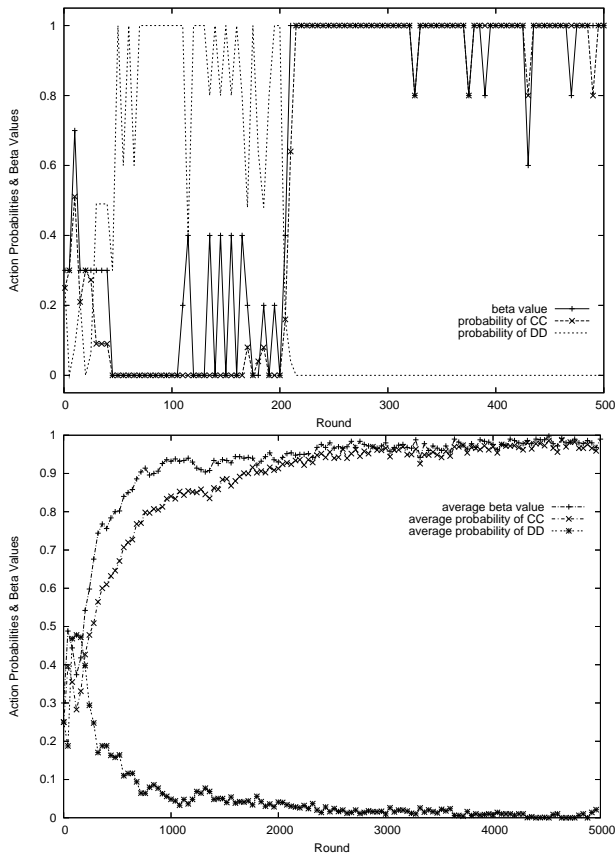


Figure 1: Self-play: single run (top), averaged over 100 runs (bottom)

probability distribution), but as the plot shows (which depicts the number of games out of 100 that have converged to  $(C, C)$  with probability  $> 0.9$ ) over 90% of all games converge to mutual cooperation. This is an important result which underlines that communicated advice has the capacity to act as a “gold standard” *regardless* of whether agents can explicitly model their opponents. Results with the malicious agent show that  $\beta$ -WoLF is able to recover from excessive reliance on advice: while being exploited initially (probability of  $(C, D)$ ), our agent later abandons the advice and switches to best-response behaviour. Unfortunately this may take very long in some cases – as the long average convergence time for 100 simulations indicates. This is due to the fact that (1) Q-tables have converged to (numerically) fairly high values by the time the malicious agent switches to  $D$ , (2) the learning rate has decreased by that time and (3) average strategies take very long to adapt to the “misconduct” at this stage.

#### 4. CONCLUSION

We presented a MARL algorithm for processing advice regarding mutually beneficial behaviour and for deciding *autonomously* whether or not to follow. Our experimental evaluation shows that this algorithm generates optimally coordinated behaviour in an example game in which achieving this is a highly non-trivial task for MARL algorithms.

This comes at the price of increased computational complexity. Also, advice-following will only work under certain, fairly strong, assumptions (such as, e.g., that the globally optimal behaviour can be represented as a convex combina-

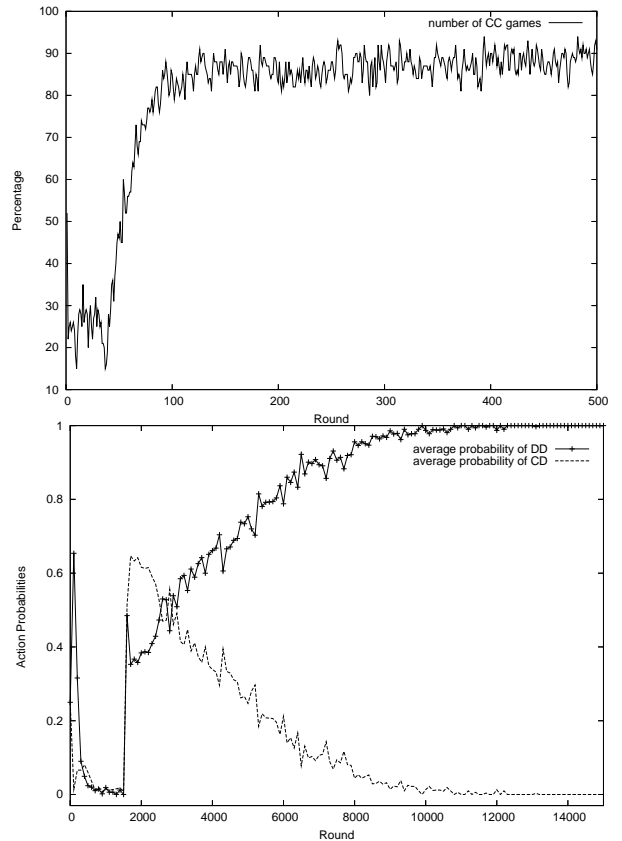


Figure 2: TIT FOR TAT (top), malicious (bottom)

tion of individually rational strategies and the strategy suggested by the advice signal). However, this shortcoming is alleviated by the fact that if all else fails,  $\beta$ -WoLF agents will resort to using their individual reward WoLF-PHC learning module and learn a simple best-response strategy – our additional machinery does not jeopardise the performance guarantees of a communication-free MARL.

In the future, we would like to establish formal properties of the algorithm (especially in terms of performance guarantees), and to exploit more complex forms of (potentially unreliable) communication for the development of advanced MARL algorithms.

#### 5. REFERENCES

- [1] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.
- [2] J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of ICML-98*, pages 242–250, 1998.
- [3] Y. Shoham, R. Powers, and T. Grenager. Multi-agent reinforcement learning: a critical survey. Technical report, Stanford University, 2003.
- [4] R. Sutton and A. Barto. *Reinforcement Learning. An Introduction*. The MIT Press/A Bradford Book, Cambridge, MA, 1998.
- [5] K. Tumer and D. H. Wolpert. Collective Intelligence and Braess’ Paradox. In *Proceedings of AAAI-00* pages 104–109, Austin, TX, 2000.
- [6] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.