The Knowledge Engineering Review, Vol. 00:0, 1–24. © 2004, Cambridge University Press
DOI: 10.1017/S000000000000000 Printed in the United Kingdom

Plan-Based Reward Shaping for Multi-Agent Reinforcement Learning

SAM DEVLIN and DANIEL KUDENKO

Department of Computer Science, University of York, UK
E-mail: sam.devlin@york.ac.uk,daniel.kudenko@york.ac.uk

Abstract

Recent theoretical results have justified the use of potential-based reward shaping as a way to improve the performance of multi-agent reinforcement learning (MARL). However, the question remains of how to generate a useful potential function.

Previous research demonstrated the use of STRIPS operator knowledge to automatically generate a potential function for single-agent reinforcement learning. Following up on this work, we investigate the use of STRIPS planning knowledge in the context of MARL.

Our results show that a potential function based on joint or individual plan knowledge can significantly improve MARL performance compared with no shaping. In addition, we investigate the limitations of individual plan knowledge as a source of reward shaping in cases where the combination of individual agent plans causes conflict.

1 Introduction

Using reinforcement learning agents in multi-agent systems (MAS) is often considered impractical due to an exponential increase in the state space with each additional agent. Whilst assuming other agents' actions to be part of the environment can save from having to calculate the value of all joint-policies, the time taken to learn a suitable policy can become impractical as the environment now appears stochastic.

One method, explored extensively in the single-agent literature, to reduce the time to convergence is reward shaping. Reward shaping is the process of providing prior knowledge to an agent through additional rewards. These rewards help direct an agent's exploration, minimising the number of sub-optimal steps it takes and so directing it towards the optimal policy quicker.

Recent work has justified the use of these methods in multi-agent reinforcement learning and so now our interest shifts towards how to encode knowledge commonly available. Previous research, again from the single-agent literature, translated knowledge encoded as STRIPS operators into a potential function for reward shaping (Grześ & Kudenko 2008*b*). In this paper we will discuss our attempts to use this approach in MAS with either coordinated plans made together or greedy plans made individually. Both are beneficial to agents but the former more so. However, planning together will not always be possible in practice and, therefore, we also present a subsequent investigation into how to overcome conflicted knowledge in individual plans.

The next section begins by introducing the relevant background material and existing work in multi-agent reinforcement learning, reward shaping and planning. Section 3 goes on then to describe our novel combination of these tools. The bulk of experimentation and analysis is in Sections 4, 5 and 6. Finally, in the closing section we conclude with remarks on the outcomes of this study and relevant future directions.

2 Background

In this section we introduce all relevant existing work upon which this investigation is based.

2.1 Multi-Agent Reinforcement Learning

Reinforcement learning is a paradigm which allows agents to learn by reward and punishment from interactions with the environment (Sutton & Barto 1998). The numeric feedback received from the environment is used to improve the agent's actions. The majority of work in the area of reinforcement learning applies a Markov Decision Process (MDP) as a mathematical model (Puterman 1994).

An MDP is a tuple $\langle S, A, T, R\rangle$, where S is the state space, A is the action space, $T(s, a, s') = Pr(s'|s, a)$ is the probability that action a in state s will lead to state s', and $R(s, a, s')$ is the immediate reward r received when action a taken in state s results in a transition to state s'. The problem of solving an MDP is to find a policy (i.e., mapping from states to actions) which maximises the accumulated reward. When the environment dynamics (transition probabilities and reward function) are available, this task can be solved using dynamic programming (Bertsekas 2007).

When the environment dynamics are not available, as with most real problem domains, dynamic programming cannot be used. However, the concept of an iterative approach remains the backbone of the majority of reinforcement learning algorithms. These algorithms apply so called temporal-difference updates to propagate information about values of states, $V(s)$, or state-action pairs, $Q(s, a)$ (Sutton 1984). These updates are based on the difference of the two temporally different estimates of a particular state or state-action value. The SARSA algorithm is such a method (Sutton & Barto 1998). After each real transition, $(s, a) \rightarrow (s', r)$, in the environment, it updates state-action values by the formula:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)] \tag{1}$$

where α is the rate of learning and γ is the discount factor. It modifies the value of taking action a in state s, when after executing this action the environment returned reward r, moved to a new state s', and action a' was chosen in state s'.

It is important whilst learning in an environment to balance exploration of new state-action pairs with exploitation of those which are already known to receive high rewards. A common method of doing so is $\epsilon-$greedy. When using this method the agent explores, with probability ϵ, by choosing a random action or exploits its current knowledge, with probability $1 - \epsilon$, by choosing the highest value action for the current state (Sutton & Barto 1998).

Temporal-difference algorithms, such as SARSA, only update the single latest state-action pair. In environments where rewards are sparse, many episodes may be required for the true value of a policy to propagate sufficiently. To speed up this process, a method known as eligibility traces keeps a record of previous state-action pairs that have occurred and are therefore eligible for update when a reward is received. The eligibility of the latest state-action pair is set to 1 and all other state-action pairs' eligibility is multiplied by λ (where $\lambda \leq 1$). When an action is completed all state-action pairs are updated by the temporal difference multiplied by their eligibility and so Q-values propagate quicker (Sutton & Barto 1998).

Applications of reinforcement learning to MAS typically take one of two approaches; multiple individual learners or joint action learners (Claus & Boutilier 1998). The latter is a group of multi-agent specific algorithms designed to consider the existence of other agents. The former is the deployment of multiple agents each using a single-agent reinforcement learning algorithm.

Multiple individual learners assume any other agents to be a part of the environment and so, as the others simultaneously learn, the environment appears to be dynamic as the probability of transition when taking action a in state s changes over time. To overcome the appearance of a dynamic environment, joint action learners were developed that extend their value function to consider for each state the value of each possible combination of actions by all agents.

Learning by joint action, however, breaks a common fundamental concept of MAS. Specifically, each agent in a MAS is self-motivated and so may not consent to the broadcasting of their action choices as required by joint action learners. Furthermore, the consideration of the joint action causes an exponential increase in the number of values that must be calculated with each additional agent added to the system. For these reasons, this work will focus on multiple individual learners and not joint action learners. However, it is expected that the application of these approaches to joint action learners would have similar benefits.

Typically, reinforcement learning agents, whether alone or sharing an environment, are deployed with no prior knowledge. The assumption is that the developer has no knowledge of how the agent(s) should behave. However, more often than not, this is not the case. As a group we are interested in knowledge-based reinforcement learning, an area where this assumption is removed and informed agents can benefit from prior knowledge. One common method of imparting knowledge to a reinforcement learning agent is reward shaping, a topic we will discuss in more detail in the next subsection.

2.2 Multi-Agent and Plan-Based Reward Shaping

The idea of reward shaping is to provide an additional reward representative of prior knowledge to reduce the number of suboptimal actions made and so reduce the time needed to learn (Ng et al. 1999, Randløv & Alstrom 1998). This concept can be represented by the following formula for the SARSA algorithm:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + F(s,s') + \gamma Q(s',a') - Q(s,a)] \tag{2}$$

where $F(s,s')$ is the general form of any state-based shaping reward.

Even though reward shaping has been powerful in many experiments it quickly became apparent that, when used improperly, it can change the optimal policy (Randløv & Alstrom 1998). To deal with such problems, potential-based reward shaping was proposed (Ng et al. 1999) as the difference of some potential function Φ defined over a source state s and a destination state s':

$$F(s,s') = \gamma\Phi(s') - \Phi(s) \tag{3}$$

where γ must be the same discount factor as used in the agent's update rule (see Equation 1).

Ng et al. (Ng et al. 1999) proved that potential-based reward shaping, defined according to Equation 3, does not alter the optimal policy of a single agent in both infinite- and finite- state MDPs.

However, in multi-agent reinforcement learning the goal is no longer the single agent's optimal policy. Instead some compromise must be made and so agents are typically designed instead to learn a Nash equilibrium (Nash 1951, Shoham et al. 2007). For such problem domains, it has been proven that the Nash equilibria of a MAS are not altered by any number of agents receiving additional rewards provided they are of the form given in Equation 3 (Devlin & Kudenko 2011).

Recent theoretical work has extended both the single-agent guarantee of policy invariance and the multi-agent guarantee of consistent Nash equilibria to cases where the potential function is dynamic (Devlin & Kudenko 2012).

Reward shaping is typically implemented bespoke for each new environment using domain-specific heuristic knowledge (Babes et al. 2008, Devlin et al. 2011, Randløv & Alstrom 1998) but some attempts have been made to automate (Grześ & Kudenko 2008*a*, Marthi 2007) and semi-automate (Grześ & Kudenko 2008*b*) the encoding of knowledge into a reward signal. Automating the process requires no previous knowledge and can be applied generally to any problem domain. The results are typically better than without shaping but less than agents shaped by prior knowledge. Semi-automated methods require prior knowledge to be put in but then automate the transformation of this knowledge into a potential function.

Plan-based reward shaping, an established semi-automated method in single-agent reinforcement learning, uses a STRIPS planner to generate high-level plans. These plans are encoded into a potential function where states later in the plan receive a higher potential than those lower or not in the plan. This potential function is then used by potential-based reward shaping to encourage the agent to follow the plan without altering the agent's goal. The process of learning the low-level actions necessary to execute a high-level plan is significantly easier than learning the low-level actions to maximise reward in an unknown environment and so with this knowledge agents tend to learn the optimal policy quicker. Furthermore, as many developers are already familiar with STRIPS planners, the process of implementing potential-based reward shaping is now more accessible and less domain specific. (Grześ & Kudenko 2008*b*)

In this investigation we explore how multi-agent planning, introduced in the following subsection, can be combined with this semi-automatic method of reward shaping.

2.3 Multi-Agent Planning

The generation of multi-agent plans can occur within one centralised agent or spread amongst a number of agents (Rosenschein 1982, Ziparo 2005).

The centralised approach benefits from full observation making it able to, where possible, satisfy all agents' goals without conflict. However, much like joint-action learning, this approach requires sharing of information, such as goals and abilities, that agents in a MAS often will not want to share.

The alternative approach, allowing each agent to make their own plans, will tend to generate conflicting plans. Many methods of coordination have been attempted including, amongst others, social laws (Shoham & Tennenholtz 1995), negotiation (Ziparo 2005) and contingency planning (Peot & Smith 1992) but still this remains an ongoing area of active research (De Weerdt et al. 2005).

In the next section we will discuss how plans generated by both of these methods can be used with plan-based reward shaping to aid multiple individual learners.

3 Multi-Agent, Plan-Based Reward Shaping

Based on the two opposing methods of multi-agent planning, centralised and decentralised, we propose two methods of extending plan-based reward shaping to multi-agent reinforcement learning.

The first, joint-plan based reward shaping, employs the concept of centralised planning and so generates where possible plans without conflict. This shaping is expected to outperform the alternative but may not be possible in competitive environments where agents are unwilling to cooperate.

Alternatively, individual-plan-based reward shaping, requires no cooperation as each agent plans as if it is alone in the environment.

Unfortunately, the application of individual-plan-based reward shaping to multi-agent problem domains is not as simple in practice as it may seem. The knowledge given by multiple individual plans will often be conflicted and agents may need to deviate significantly from this prior knowledge when acting in their common environment. Our aim is to allow them to do so. Reward shaping only encourages a path of exploration, it does not enforce a joint-policy. Therefore, it may be possible that reinforcement learning agents, given conflicted plans initially, can learn to overcome their conflicts and eventually follow coordinated policies.

For both methods, the STRIPS plan of each agent is translated into a list of states so that, whilst acting, an agent's current state can be compared to all plan steps. The potential of the agent's current state then becomes:

$$\Phi(s) = \omega * CurrentStepInPlan \tag{4}$$

where ω is a scaling factor and $CurrentStepInPlan$ is the corresponding state in the state-based representation of the agent's plan (for example see Listing 5).

If the current state is not in the state-based representation of the agent's plan, then the potential used is that of the last state experienced that was in the plan. This was implemented in the original work to not discourage exploration off of the plan and is now more relevant as we know that, in the case of individual plans, strict adherence to the plan by every agent will not be possible. This feature of the potential function makes plan-based reward shaping an instance of dynamic potential-based reward shaping (Devlin & Kudenko 2012).

Finally, to preserve the theoretical guarantees of PBRS in episodic problem domains, the potential of all goal/final states is set to zero. These potentials are then used as in Equation 3 to calculate the additional reward given to the agent.

In the next section we will introduce a problem domain and the specific implementations of both our proposed methods in that domain.

4 Initial Study

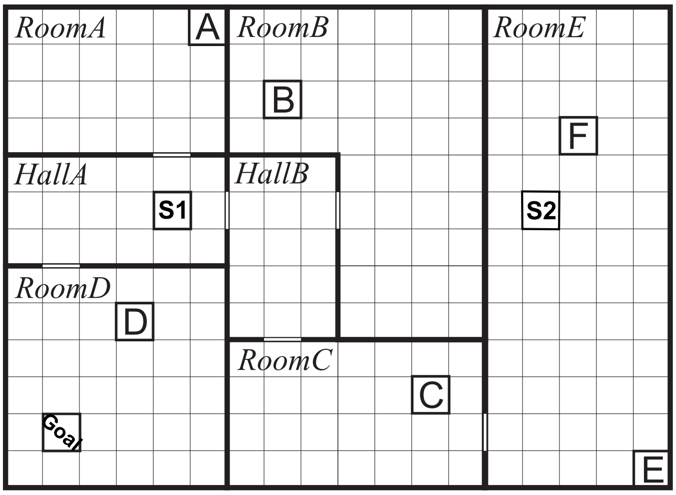


Figure 1: Multi-Agent, Flag-Collecting Problem Domain

Our chosen problem for this study is a flag collecting task in a discrete, grid-world domain with two agents attempting to collect six flags spread across seven rooms. An overview of this world is illustrated in Figure 1 with the goal location labeled as such, each agent's starting location labeled Si where i is their unique id and the remaining labeled grid locations being flags and their unique id.

At each time step an agent can move up, down, left or right and will deterministically complete their move provided they do not collide with a wall or the other agent. Once an agent reaches the goal state their episode is over regardless of the number of flags collected. The entire episode is completed when both agents reach the goal state. At this time both agents receive a reward equal to one hundred times the number of flags they have collected in combination. No other rewards are given at any other time. To encourage the agents to learn short paths, the discount factor γ is set to less than one.[1]

[1] Experiments with a negative reward on each time step and $\gamma = 1$ made no significant change in the behaviour of the agents.

Additionally, as each agent can only perceive its own location and the flags it has already picked up, the problem is a DEC-POMDP.

Given this domain, the plans of agent 1 and agent 2 with joint-plan based reward shaping are documented in Listings 1 and 2. It is important to note that these plans are coordinated with no conflicting actions.

Listing 1: Joint-Plan for Agent 1 Starting in HallA

```
MOVE( hallA ,roomA)
TAKE( flagA ,roomA)
MOVE(roomA, hallA )
MOVE( hallA , hallB )
MOVE( hallB ,roomB)
TAKE( flagB ,roomB)
MOVE(roomB, hallB )
MOVE( hallB , hallA )
MOVE( hallA ,roomD)
```

Listing 2: Joint-Plan for Agent 2 Starting in RoomE

```
TAKE( flagF ,roomE)
TAKE( flagE ,roomE)
MOVE(roomE,roomC)
TAKE( flagC ,roomC)
MOVE(roomC, hallB )
MOVE( hallB , hallA )
MOVE( hallA ,roomD)
TAKE( flagD ,roomD)
```

Alternatively, Listings 3 and 4 document the plans used to shape agent 1 and agent 2 respectively when receiving individual-plan-based reward shaping. However, now both plans cannot be completed as each intends to collect all flags. How, or if, the agents can learn to overcome this conflicting knowledge is the focus of this investigation.

Listing 3: Individual Plan for Agent 1 Starting in HallA

```
MOVE( hallA , hallB )
MOVE( hallB ,roomC)
TAKE( flagC ,roomC)
MOVE(roomC,roomE)
TAKE( flagE ,roomE)
TAKE( flagF ,roomE)
MOVE(roomE,roomC)
MOVE(roomC, hallB )
MOVE( hallB ,roomB)
TAKE( flagB ,roomB)
MOVE(roomB, hallB )
MOVE( hallB , hallA )
MOVE( hallA ,roomA)
TAKE( flagA ,roomA)
MOVE(roomA, hallA )
MOVE( hallA ,roomD)
TAKE( flagD ,roomD)
```

Listing 4: Individual Plan for Agent 2 Starting in RoomE

```
TAKE( flagF ,roomE)
TAKE( flagE ,roomE)
MOVE(roomE,roomC)
TAKE( flagC ,roomC)
MOVE(roomC, hallB )
MOVE( hallB ,roomB)
TAKE( flagB ,roomB)
MOVE(roomB, hallB )
MOVE( hallB , hallA )
MOVE( hallA ,roomA)
TAKE( flagA ,roomA)
MOVE(roomA, hallA )
MOVE( hallA ,roomD)
TAKE( flagD ,roomD)
```

As mentioned in Section 3, these plans must be translated into state-based knowledge. Listing 5 shows this transformation for the joint-plan starting in hallA (listed in Listing 1) and the corresponding value of ω.

In all our experiments, regardless of knowledge used, we have set the scaling factor ω so that the maximum potential of a state is the maximum reward of the environment. As the scaling factor affects how likely the agent is to follow the heuristic knowledge (Grześ 2010), maintaining a constant maximum across all heuristics compared ensures a fair comparison. For environments with an unknown maximum reward the scaling factor ω can be set experimentally or based on the designer's confidence in the heuristic.

Listing 5: State-Based Joint-Plan for Agent 1 Starting in HallA

```
0  robot-in_hallA
1  robot-in_roomA
2  robot-in_roomA  taken_flagA
3  robot-in_hallA  taken_flagA
4  robot-in_hallB  taken_flagA
5  robot-in_roomB  taken_flagA
6  robot-in_roomB  taken_flagA  taken_flagB
7  robot-in_hallB  taken_flagA  taken_flagB
8  robot-in_hallA  taken_flagA  taken_flagB
9  robot-in_roomD  taken_flagA  taken_flagB
```

$\omega = MaxReward/NumStepsInPlan = 600/9$

For comparison, we have implemented a team of agents with no prior knowledge/shaping and a team with the domain-specific knowledge that collecting flags is beneficial. These flag-based agents value a state's potential equal to one hundred times the number of flags it alone has collected. This again ensures that the maximum potential of any state is equal to the maximum reward of the environment.

We have also considered the combination of this flag-based heuristic with the general methods of joint-plan-based and individual-plan-based shaping. These combined agents value the potential of a state to be:

$$\begin{aligned} \Phi(s) = & \quad (CurrentStepInPlan + NumFlagsCollected) * \omega \\ \omega = & \quad MaxReward/(NumStepsInPlan \\ & \quad +NumFlagsInWorld) \end{aligned} \tag{5}$$

where *NumFlagsCollected* is the number of flags the agent has collected itself, *NumStepsInPlan* is the number of steps in its state-based plan and *NumFlagsInWorld* is the total number of flags in the world (i.e. for this domain - 6).

All agents, regardless of shaping, implemented SARSA with $\epsilon-$greedy action selection and eligibility traces. For all experiments, the agents' parameters were set such that $\alpha = 0.1$, $\gamma = 0.99$, $\epsilon = 0.1$ and $\lambda = 0.4$. For these experiments, all initial Q-values were zero.

These methods, however, do not require the use of SARSA, $\epsilon-$greedy action selection or eligibility traces. Potential-based reward shaping has previously been proven with Q-learning and RMax and any action selection method that chooses actions based on relative difference and not absolute magnitude (Asmuth et al. 2008). From our own experience, it also works with many multi-agent specific algorithms (including both temporal difference and policy iteration algorithms). Furthermore, it has been shown before without (but never before to our knowledge with) eligibility traces (Ng et al. 1999, Asmuth et al. 2008, Devlin et al. 2011).

All experiments have been repeated thirty times with the mean discounted reward per episode presented in the following graphs. All claims of significant differences are supported by two-tailed, two sample t-tests with significance $p < 0.05$ (unless stated otherwise).

4.1 Results and Conclusions

Figure 2 shows all agents, regardless of shaping, learn quickly within the first 300 episodes. In all cases, some knowledge significantly improves the final performance of the agents as shown by all shaped agents out-performing the base agent with no reward shaping.

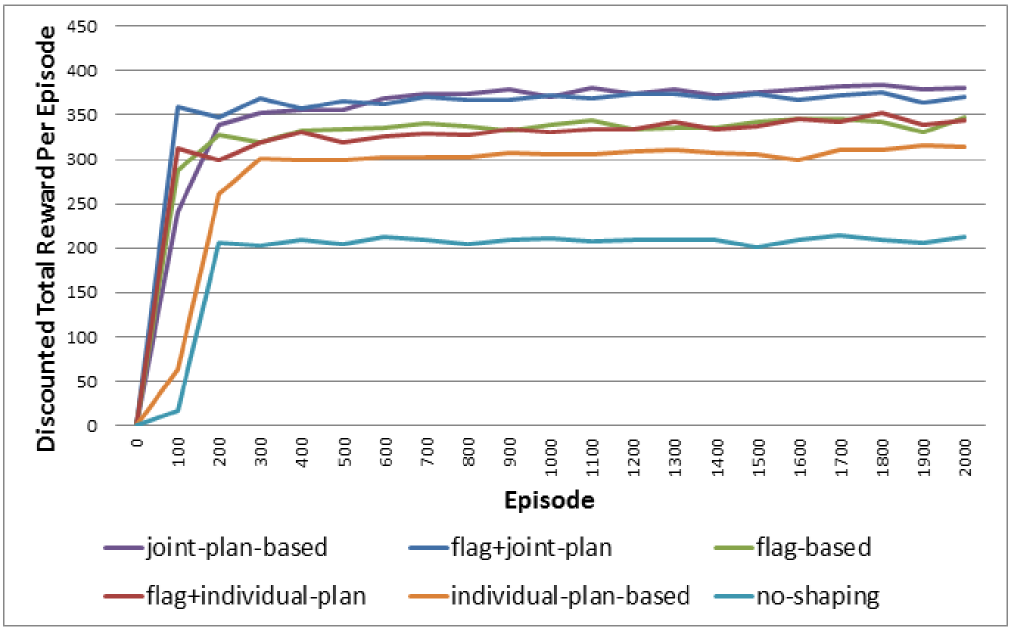
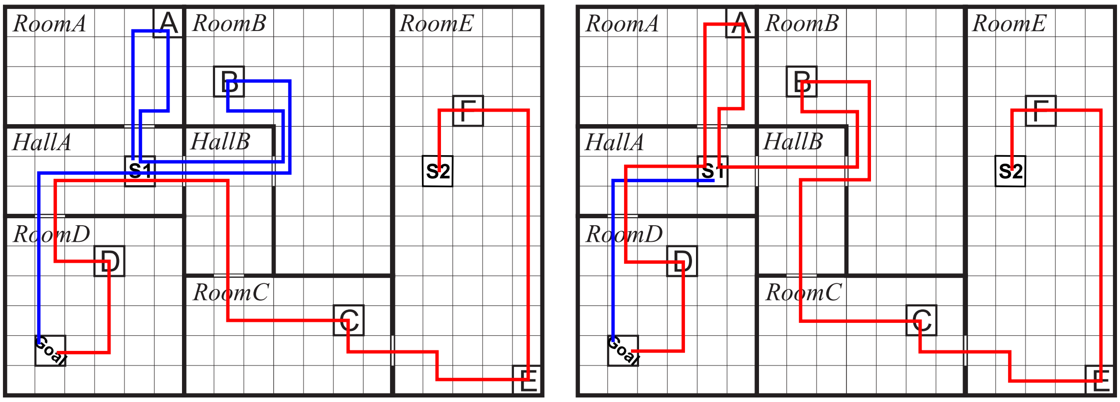


Figure 2: Initial Results

Agents shaped by knowledge of the optimal joint-plan (both alone or combined with the flag-based heuristic) significantly outperform all other agents, consistently learning to collect all six flags[2]. Figure 3a illustrates the typical behaviour learnt by these agents. Note that in these examples the agents have learnt the low level implementation of the high level plan provided.



(a) Joint-Plan-Based Agents

(b) Individual-Plan-Based Agents

Figure 3: Example Behaviour of:

The individual-plan-based agents are unable to reach the same performance as they are given no explicit knowledge of how to coordinate.

Figure 3b illustrates the typical behaviour learnt by these agents. This time we note that agent 1 has opted out of receiving its shaping reward by moving directly to the goal and not following its given plan. The resultant behaviour allows the agents to receive the maximum goal reward from collecting all flags, but at a longer time delay and, therefore, a significantly greater discount.

[2] Please note the joint-plan-based agents' illustrated performance in Figure 2 does not reach 600 as the value presented is discounted by the time it takes the agents to complete the episode.

Occasionally the agents coordinate better with agent 1 collecting flag D or, even rarer, flags D and A. Whilst this is the exception, it is interesting to note that the agent not following its plan will always choose actions that take away from the end of the other agent's plan rather than follow the first steps of their own plan.

The flag based heuristic can be seen to improve coordination slightly in the agents receiving combined shaping from both types of knowledge, but not sufficiently to overcome the conflicts in the two individual plans.

To conclude, some knowledge, regardless of the number of conflicts, is better than no knowledge but coordinated knowledge is more beneficial if available.

Given these initial results, our aim in the following experiments was to try to close the difference in final performance between individual-plan-based agents and joint-plan-based agents by overcoming the conflicted knowledge.

5 Overcoming Conflicted Knowledge

In this section we explore options for closing the difference in final performance caused by conflicted knowledge in the individual plans.

One plausible option would be to introduce communication between the agents. Another may be to combine individual-plan-based reward shaping with FCQ-learning (De Hauwere et al. 2011) to switch to a joint-action representation in states where coordination is required. However, as both multiple individual learners and individual-plan-based reward shaping were designed to avoid sharing information amongst agents, we have not explored these options.

Without sharing information, agent 1 could be encouraged not to opt out of following its plan by switching to a competitive reward function. However, as illustrate by Figure 4, although this closed the gap between individual-plan-based and joint-plan-based agents, the change was detrimental to the team performance of all agents regardless of shaping.

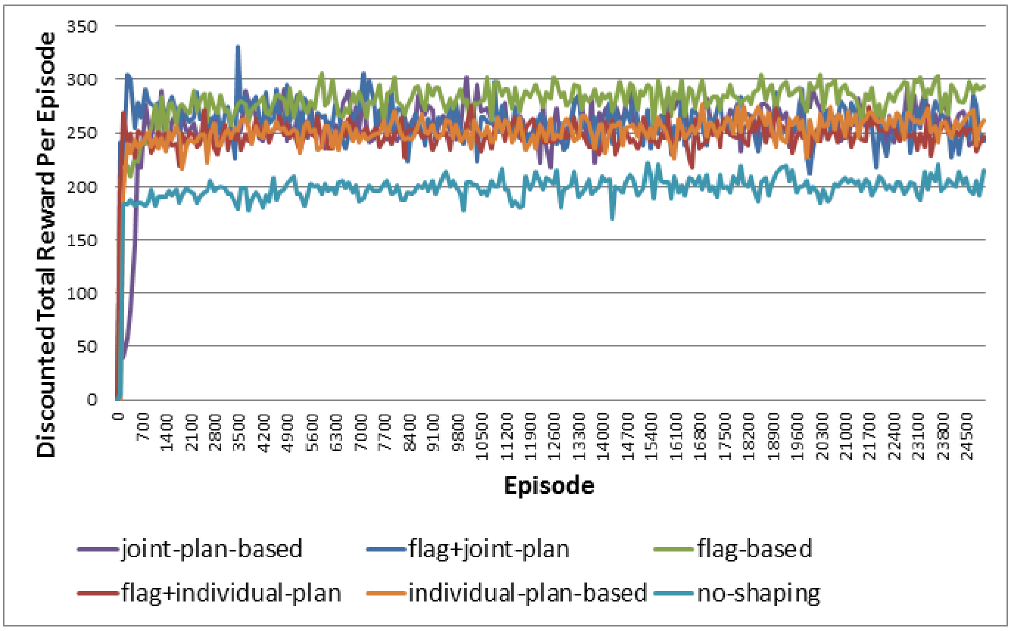


Figure 4: Competitive Reward

Specifically, individual-plan-based agent 1 did, as expected, start to participate and collect some flags but collectively they would not collect all flags. Both agents would follow their plans to the first two or three flags but then head to the goal as the next flag would not reliably be there. For similar reasons joint-plan-based agents would also no longer collect all flags. Therefore, the reduction in the gap between individual-plan-based and joint-plan-based agents was at the cost of no longer finding all flags. We considered this an undesirable compromise and so will not cover this approach further.

Instead, in the following subsections we will discuss two approaches that lessened the gap by improving the performance of the individual-plan-based agents.

The first of these approaches is increasing exploration in the hope that the agents will experience and learn from policies that coordinate better than those encouraged by their individual plans. The second approach was to improve the individual plans by reducing the number of conflicts or increasing the time until conflict.

Both methods enjoy some success and provide useful insight in to how future solutions may overcome incorrect or conflicted knowledge. Where successful, these approaches provide solutions where multiple agents can be deployed without sharing their goals, broadcasting their actions or communicating to coordinate.

5.1 Increasing Exploration

Setting all initial Q-values to zero, as was mentioned in Section 4, is a pessimistic initialisation given that no negative rewards are received in this problem domain. Agents given pessimistic initial beliefs tend to explore less as any positive reward, however small, once received specifies the greedy policy and other policies will only be followed if randomly selected by the exploration steps (Sutton & Barto 1998).

With reward shaping and pessimistic initialisation an agent becomes more sensitive to the quality of knowledge they are shaped by. If encouraged to follow the optimal policy they can quickly learn to do so, as is the case in the initial study with the joint-plan-based agents. However, if encouraged to follow incorrect knowledge, such as the conflicted plans of the individual-plan-based agent, they may converge to a sub-optimal policy.

The opposing possibility is to instead initialise optimistically by setting all Q-values to start at the maximum possible reward. In this approach agents explore more as any action gaining less than the maximum reward becomes valued less than actions yet to be tried (Sutton & Barto 1998).

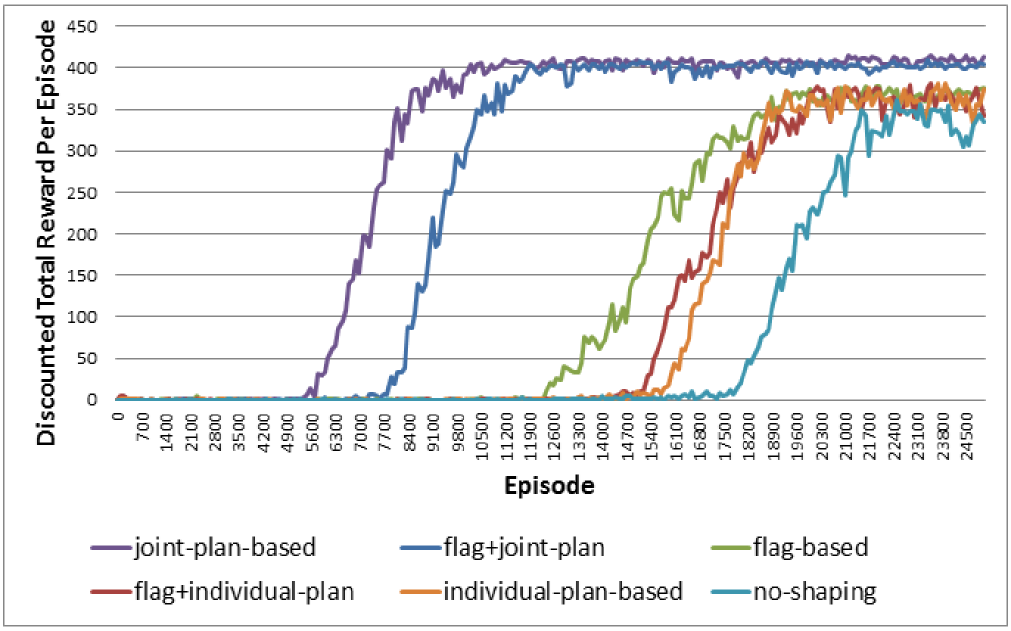


Figure 5: Optimistic Initialisation

In Figure 5 we show the outcome of optimistically initialising the agents with Q-values of 600, the maximum reward agents can receive in this problem domain.

As would be expected, increased exploration causes the agents to take longer to learn a suitable policy. However, all agents (except for those receiving flag-based or combined-flag+joint-plan shaping) learn significantly better policies than their pessimistic equivalents[3]. This reduces the gap in final performance between all agents and the joint-plan-based agents, however, the difference that remains is still significant.

Despite that, the typical behaviour learnt by optimistic individual-plan-based agents is the same as the behaviour illustrated in Figure 3a. However, it occurs less often in these agents than it occurred in the pessimistic joint-plan-based agents. This illustrates that conflicts can be overcome by optimistic initialisation but it cannot be guaranteed, by this method alone, that the optimal joint-plan will be learnt.

Furthermore, it takes time for the individual-plan-based agents to learn how to overcome the conflicts in their plans. However, this time is still less than it takes the agents with no prior knowledge to learn. Therefore, given optimistic initialisation, the benefit of reward shaping is now more important in the time to convergence instead of the final performance.

To conclude, these experiments demonstrate that some conflicted knowledge can be overcome given sufficient exploration.

5.2 Improving Knowledge

An alternative approach to overcoming conflicted knowledge would be to improve the knowledge. The individual-plan-based agents received shaping based on plans to both collect all six flags. If these plans are followed the agents will collide at their second planned flag to collect. The agent that does not pick up the flag will no longer be able to follow their plan and will therefore receive no further shaping rewards. Instead, we now consider three groups of agents that are shaped by less conflicted plans.

Specifically, plan-based-6 agents still both plan to collect all six flags, but the initial conflict is delayed until the second or third flag. The comparison of these agents to the individual-plan-based agents will show whether the timing of the conflict affects performance.

Plan-based-5 agents plan to collect just five flags each, reducing the number of conflicted flags to 4. Comparing this to both previous agents and subsequent agents will show whether the number of conflicts affects performance. These agents also experience their first conflict on the second or third flag.

Plan-based-4 agents plan to collect four flags each, reducing the number of conflicted flags to two and delaying the first conflict until the third flag. This agent will contribute to conclusions both on timing of conflicts and amount of.

As can be seen in Figure 6, both the timing of the conflict and the amount of conflict affect the agents' time to convergence. Little difference in final performance is evident in these results as the agents are still benefiting from optimistic initialisation. Alternatively, if we return to pessimistic initialisation as illustrated by Figure 7, reducing the amount of incorrect knowledge can also affect the final performance of the agents.

However, to make plans with only partial overlaps, agents require some coordination or joint-knowledge that would not typically be available to multiple individual learners. If the process of improving knowledge could be automated, for instance with an agent starting an episode shaped by its individual plan and then refining the plan as it notices conflicts (i.e. plan steps that never occur), the agent may benefit from the improved knowledge without the need for joint knowledge or coordination.

[3] For individual-plan-based agents $p = 0.064$, for all others $p < 0.05$.

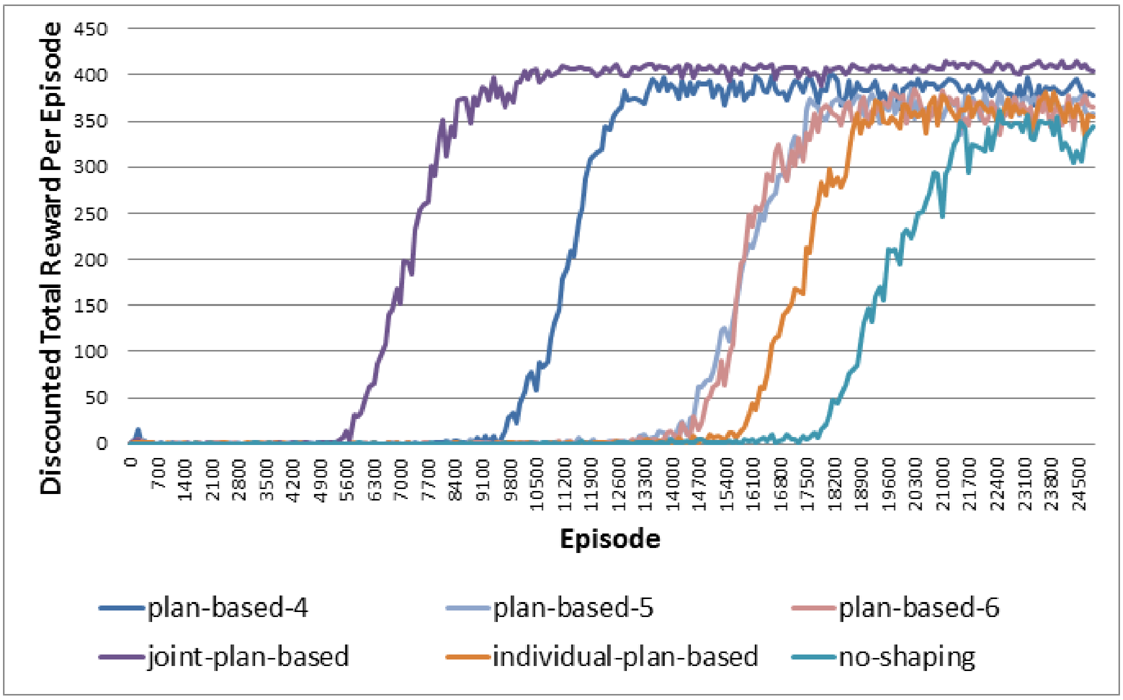


Figure 6: Optimistic Partial Plans

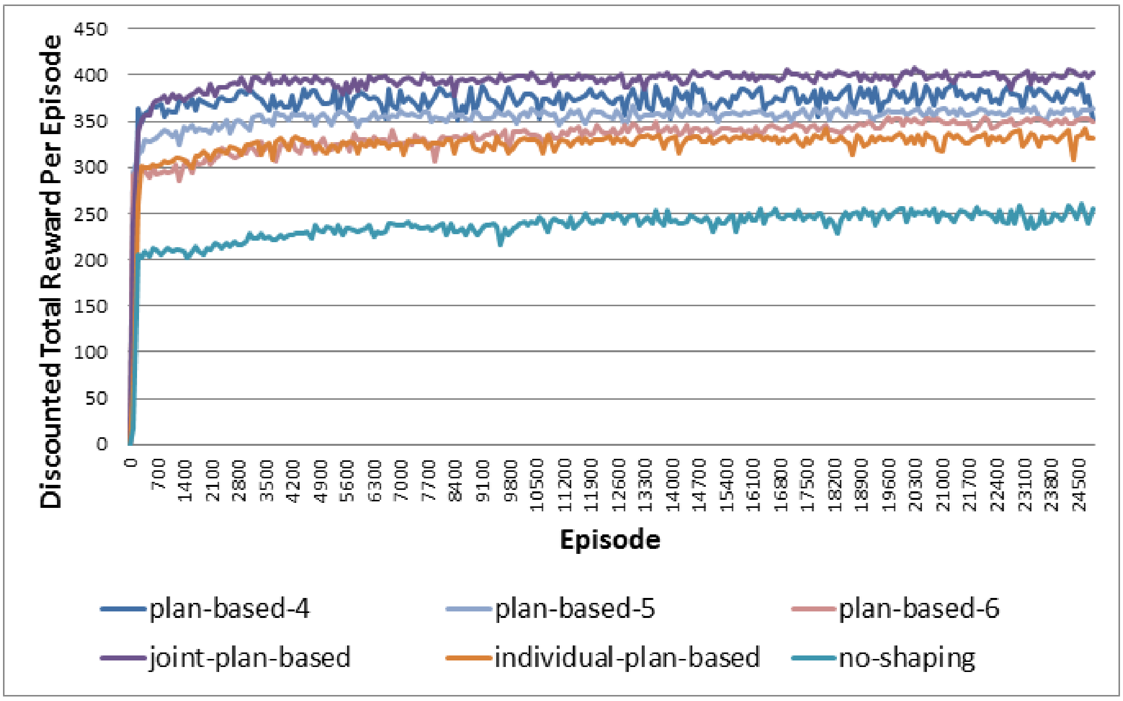
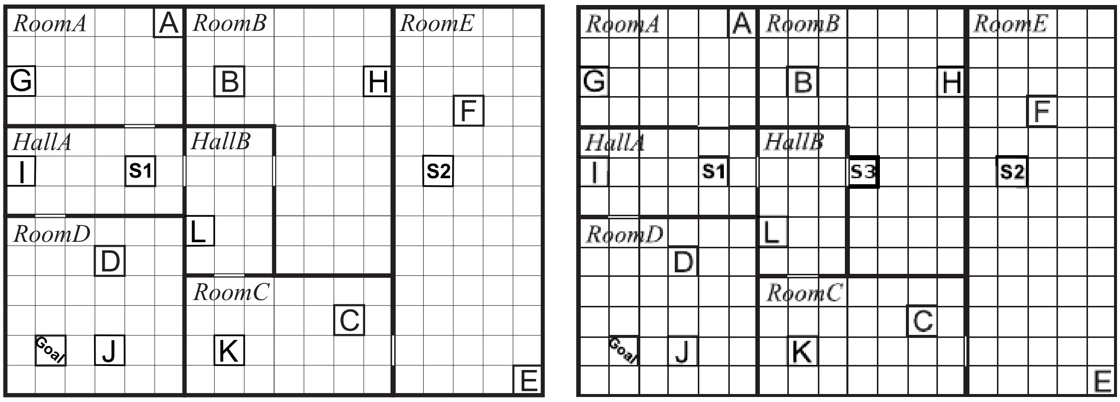


Figure 7: Pessimistic Partial Plans

6 Scaling Up

To further test multi-agent, plan-based reward shaping and our two approaches to handling incorrect knowledge we extended the problem domain by adding six extra flags[4](as illustrated in Figure 8a) and then adding a third agent (as illustrated in Figure 8b.)



(a) Extra Flags

(b) Extra Agent

Figure 8: Scaled Up Problem Domains

6.1 Extra Flags

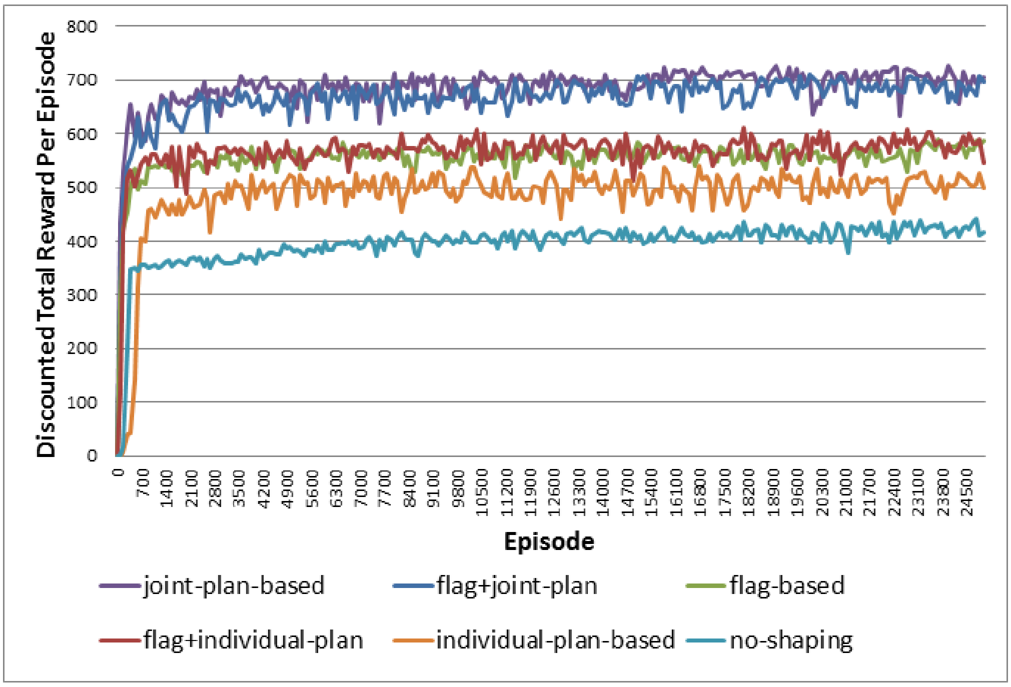


Figure 9: Pessimistic Initialisation in the Scaled Up Problem Domain

As shown in Figures 9 and 10, the results for pessimistic initialisation with 12 flags and 2 agents were effectively the same as those in the original domain except for a slightly longer time to convergence as would be expected due to the larger state space.

[4] Consequently *MaxReward* now equals 1200.

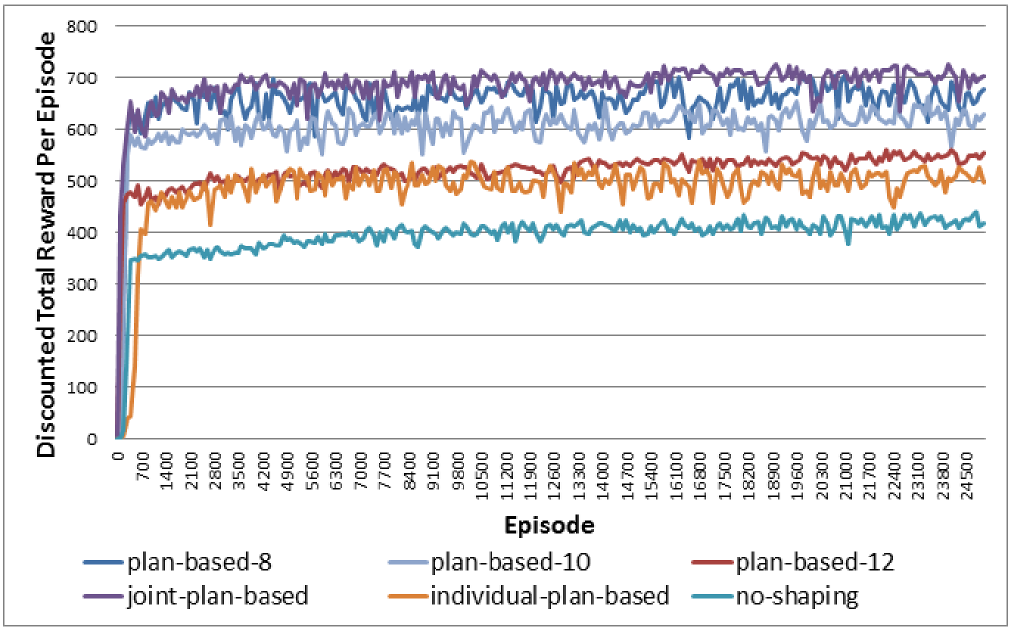


Figure 10: Pessimistic Partial Plans in the Scaled Up Problem Domain

The results for optimistic initialisation, however, took significantly longer. Figure 11 illustrates the results of just one complete run for this setting as performing any repeats would be impractical.

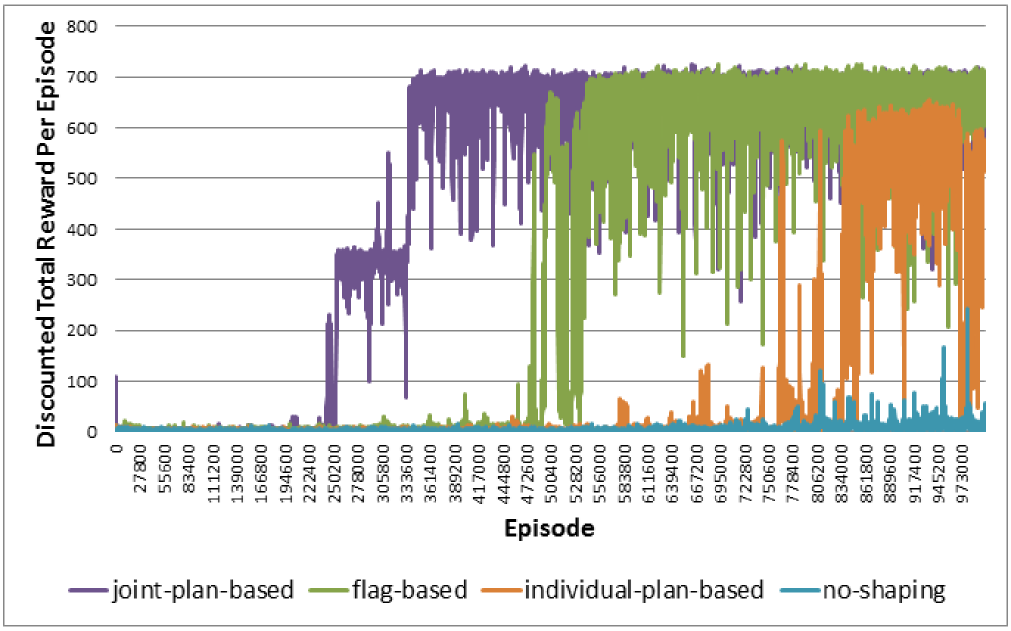


Figure 11: Optimistic Initialisation in the Scaled Up Problem Domain

Whilst these results may be obtained quicker using function approximation or existing methods of improving optimistic exploration (Grześ & Kudenko 2009), they highlight the poor ability of optimistic initialisation to scale to large domains. Therefore, these experiments further support that automating the reduction of incorrect knowledge by an explicit belief revision mechanism would be more preferable than increasing exploration by optimistic initialisation as the latter method does not direct exploration sufficiently. Instead optimistic initialisation encourages exploration to all states randomly taking considerable time to complete. A gradual refining of the plan used to shape an agent would encourage initially a conflicted joint-policy, which is still better than no prior knowledge, and then on each update exploration would be directed towards a more coordinated joint-plan.

6.2 Extra Agent

Finally, Figure 12 shows the results for pessimistic initialisation for the scaled up setting with 12 flags and 3 agents illustrated in Figure 8b. Under these settings, the performance of all agents is more variable due to the extra uncertainty the additional agent causes. This is to be expected as the underlying state-action space has grown exponentially whilst, as each agent only considers it's own location and collection of flags, the state space learnt by each agent has not grown. For similar reasons, the agents without shaping or shaped by any potential function that includes the flag heuristic perform significantly worse now than when there were only two agents acting and learning in the environment.

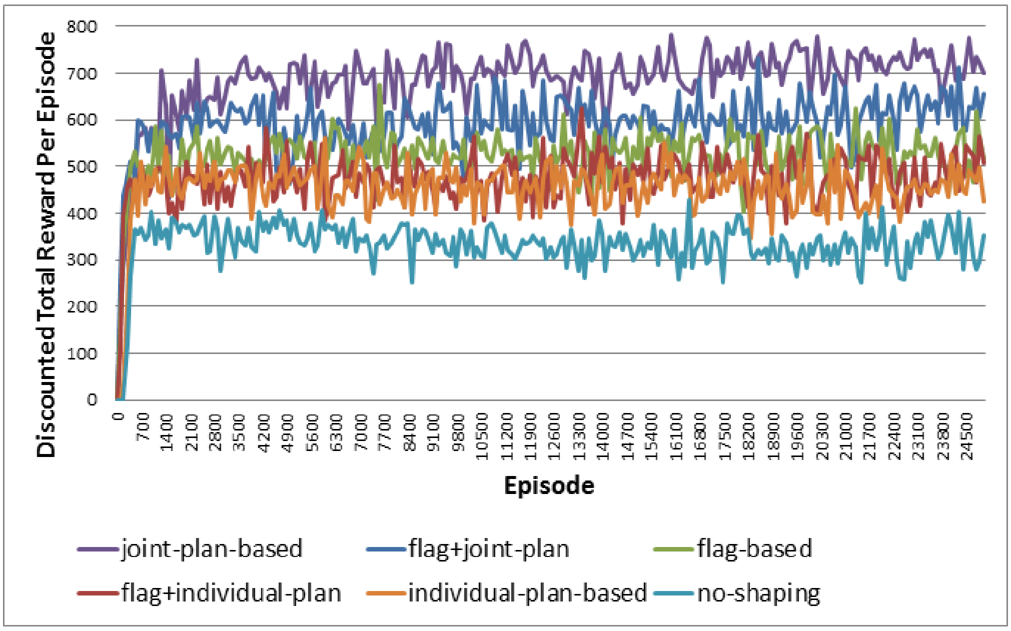


Figure 12: Pessimistic Initialisation in the Scaled Up Problem Domain with 3 Agents

Alternatively, the agents shaped by individual plans or joint plans alone have remained robust to the changes and converge on average to policies of equivalent performance to their counterparts with two agents in the environment. This was expected with the joint-plan agents as the plans received take into account the third agent and coordinate task allocation prior to learning.

However, in the case of the individual plans this is more impressive. The typical behaviour we have witnessed these agents learn is a suitable task allocation with each agent collecting some flags. This has occured because with the third agent starting in the middle less flags are contended. It is quickly learnable that flags A and G belong to agent 1, flags C,E,F and K belong to agent 2 and flags B and H belong to agent 3 with the allocation of collecting the four remaining flags having little impact on overall performance provided they are collected by someone. Whereas before, with two agents, flags B and H in particular were highly contended with both agents having similar path lengths to reach them and needing to deviate significantly from the path they would take if they were not to collect them. Coordinating in this task is exceptionally challenging and a key feature of this environment with only two agents.

7 Closing Remarks and Future Work

In conclusion, we have demonstrated two approaches to using plan-based reward shaping in multi-agent reinforcement learning. Ideally, plans are devised and coordinated centrally so each agent starts with prior knowledge of its own task allocation and the group can quickly converge to an optimal joint-policy.

Where this is not possible, due to agents unwilling to share information, plans made individually can shape the agent. Despite conflicts in the simultaneous execution of these plans, agents receiving individual-plan-based reward shaping outperformed those without any prior knowledge in all experiments.

Overcoming conflicts in the multiple individual plans by reinforcement learning can occur if shaping is combined with domain specific knowledge (i.e. flag-based reward shaping), the agent is initialised optimistically or the amount of conflicted knowledge is reduced. The first of these approaches requires a bespoke encoding of knowledge for any new problem domain and the second, optimistic initialisation, becomes impractical in larger domains.

Therefore, we are motivated to pursue in ongoing work the approach of automatically improving knowledge by an explicit belief revision mechanism. Where successful, this approach would provide a semi-automatic method of incorporating partial-knowledge in reinforcement learning agents that benefit from the correct knowledge provided and can overcome the conflicted knowledge.

References

Asmuth, J., Littman, M. & Zinkov, R. (2008), Potential-based shaping in model-based reinforcement learning, *in* 'Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence', pp. 604–609.

Babes, M., de Cote, E. & Littman, M. (2008), Social reward shaping in the prisoner's dilemma, *in* 'Proceedings of The Seventh Annual International Conference on Autonomous Agents and Multiagent Systems (AAMAS)', Vol. 3, pp. 1389–1392.

Bertsekas, D. P. (2007), *Dynamic Programming and Optimal Control*, Athena Scientific, 3rd edition.

Claus, C. & Boutilier, C. (1998), The dynamics of reinforcement learning in cooperative multiagent systems, *in* 'Proceedings of the National Conference on Artificial Intelligence', pp. 746–752.

De Hauwere, Y., Vrancx, P. & Nowé, A. (2011), Solving delayed coordination problems in mas (extended abstract), *in* 'The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)', pp. 1115–1116.

De Weerdt, M., Ter Mors, A. & Witteveen, C. (2005), 'Multi-agent planning: An introduction to planning and coordination', *Handouts of the European Agent Summer* .

Devlin, S., Grześ, M. & Kudenko, D. (2011), 'An empirical study of potential-based reward shaping and advice in complex, multi-agent systems', *Advances in Complex Systems* .

Devlin, S. & Kudenko, D. (2011), Theoretical considerations of potential-based reward shaping for multi-agent systems, *in* 'Proceedings of The Tenth Annual International Conference on Autonomous Agents and Multiagent Systems (AAMAS)'.

Devlin, S. & Kudenko, D. (2012), Dynamic potential-based reward shaping, *in* 'Proceedings of The Eleventh Annual International Conference on Autonomous Agents and Multiagent Systems (AAMAS)'.

Grześ, M. (2010), Improving exploration in reinforcement learning through domain knowledge and parameter analysis, Technical report, University of York.

Grześ, M. & Kudenko, D. (2008*a*), 'Multigrid Reinforcement Learning with Reward Shaping', *Artificial Neural Networks-ICANN 2008* pp. 357–366.

Grześ, M. & Kudenko, D. (2008*b*), Plan-based reward shaping for reinforcement learning, *in* 'Proceedings of the 4th IEEE International Conference on Intelligent Systems (IS'08)', IEEE, pp. 22–29.

Grześ, M. & Kudenko, D. (2009), 'Improving optimistic exploration in model-free reinforcement learning', *Adaptive and Natural Computing Algorithms* pp. 360–369.

Marthi, B. (2007), Automatic shaping and decomposition of reward functions, *in* 'Proceedings of the 24th International Conference on Machine learning', ACM, p. 608.

Nash, J. (1951), 'Non-cooperative games', *Annals of mathematics* **54**(2), 286–295.

Ng, A. Y., Harada, D. & Russell, S. J. (1999), Policy invariance under reward transformations: Theory and application to reward shaping, *in* 'Proceedings of the 16th International Conference on Machine Learning', pp. 278–287.

Peot, M. & Smith, D. (1992), Conditional nonlinear planning, *in* 'Artificial Intelligence Planning Systems: Proceedings of the First International Conference', Morgan Kaufmann Pub, p. 189.

Puterman, M. L. (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley and Sons, Inc., New York, NY, USA.

Randløv, J. & Alstrom, P. (1998), Learning to drive a bicycle using reinforcement learning and shaping, *in* 'Proceedings of the 15th International Conference on Machine Learning', pp. 463–471.

Rosenschein, J. (1982), Synchronization of multi-agent plans, *in* 'Proceedings of the National Conference on Artificial Intelligence', pp. 115–119.

Shoham, Y., Powers, R. & Grenager, T. (2007), 'If multi-agent learning is the answer, what is the question?', *Artificial Intelligence* **171**(7), 365–377.

Shoham, Y. & Tennenholtz, M. (1995), 'On social laws for artificial agent societies: off-line design', *Artificial Intelligence* **73**(1-2), 231–252.

Sutton, R. S. (1984), Temporal credit assignment in reinforcement learning, PhD thesis, Department of Computer Science, University of Massachusetts, Amherst.

Sutton, R. S. & Barto, A. G. (1998), *Reinforcement Learning: An Introduction*, MIT Press.

Ziparo, V. (2005), Multi-Agent Planning, Technical report, University of Rome.