Efficient Knowledge Transfer in Shaping Reinforcement Learning

Sholeh Norouzzadeh Lucian Buşoniu Robert Babuška

Delft Center for Systems and Control, Delft University of Technology, The Netherlands
sholeh.norouzzadeh@gmail.com, {i.l.busoniu, r.babuska}@tudelft.nl

Abstract: Reinforcement learning is an attractive solution for deriving an optimal control policy by on-line exploration of the control task. Shaping aims to accelerate reinforcement learning by starting from easy tasks and gradually increasing the complexity, until the original task is solved. In this paper, we consider the essential decision on when to transfer learning from an easier task to a more difficult one, so that the total learning time is reduced. We propose two *transfer criteria* for making this decision, based on the agent's performance. The first criterion measures the agent's performance by the distance between its current solution and the optimal one, and the second by the empirical return obtained. We investigate the learning time gains achieved by using these criteria in a classical gridworld navigation benchmark. This numerical study also serves to compare several major shaping techniques.

Keywords: reinforcement learning; shaping; transfer learning.

1. INTRODUCTION

Reinforcement learning (RL) (Sutton and Barto, 1998) is a highly active research area in artificial intelligence and control. At each step, a RL agent receives the current state of the environment and chooses an action. The action changes the state of the environment, and the value of this transition is communicated to the agent through a scalar reward. The goal is to maximize the return, i.e., the cumulative long-term reward received while interacting with the environment.

In practice, as the task becomes more complex, conventional RL methods may take too long or even fail to find a good solution. *Shaping* is a class of techniques that aim to tackle this problem by starting from an easier version of the task and gradually increasing the complexity, until the original task is solved. The idea of shaping (Skinner, 1938) is inspired from human learning, which often exploits the same principle to solve complex tasks. Unfortunately, data about how – and to what extent – the various types of shaping improve the learning speed and performance are scarce in the literature, while the manner in which results are reported is not always satisfactory. In particular, the time spent on solving the easier task(s) is often disregarded, and acceleration is claimed after comparing the learning time in the original task after shaping, with solving the original task from scratch – as also noticed by Taylor and Stone (2009).

In this paper, we take the stance that shaping techniques must reduce the *total* learning time, including the time spent in the easy task(s). From this standpoint, we provide a twofold contribution. Firstly, we propose two performance-based *transfer criteria* for making the essential decision on when to transfer knowledge from an easier task to a more difficult one (or when to stop learning in final tasks). If the parameters in these criteria are properly set, transfer happens after a suitable amount of training has been performed in the easy task, without spending too much time on fine-tuning the solution to an accuracy that is not needed to accelerate learning in the original task. The first criterion evaluates the performance by the distance between the current solution and the optimal one. While this criterion is useful to analyze what shaping can achieve, it is restrictive as it requires knowledge of the optimal solution. The second criterion is much more general, and relies on the empirical return from a set of representative states.

Our second contribution is an introduction and numerical comparison, in terms of total learning time, of four representative shaping techniques. These techniques simplify the task by modifying, respectively: (i) the dynamics, (ii) the reward function, (iii) the action space, and (iv) the initial state and the state space size (Erez and Smart, 2008). The example used in the numerical study is a classical gridworld (GW) navigation benchmark, and our two novel criteria are employed to decide when to transfer or stop learning. Thus, this study doubles as an experimental evaluation of the new criteria.

After describing the necessary background on RL and the GW problem in Section 2, we discuss shaping methods in Section 3. In Section 4, we propose the novel transfer criteria, while Section 5 presents the results of our simulations. Finally, Section 6 concludes the paper.

2. REINFORCEMENT LEARNING PRELIMINARIES

Consider a Markov decision process defined by the tuple $M = (S, A, P, R, \gamma)$, where S is the state space, A is the action space, P is the transition probability function, R is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

At each time step t, the agent receives the current state s_t and takes an action a_t, which causes a transition to the next state s_{t+1}; the probability of reaching a given next state s' is $P(s_t, a_t, s')$. A reward $r_{t+1} = R(s_t, a_t, s_{t+1})$ is received that assesses the quality of the transition. The agent then receives the new state and the whole cycle repeats. The goal is to find an optimal policy $\pi^* : S \to A$ that maximizes the expected discounted return for any initial state s_0. This type of return is the sum of the infinite-horizon reward sequence, exponentially weighted using the discount factor:

$$\mathcal{R}(s_0) = \sum_{k=0}^{\infty} \gamma^k r_{k+1} \tag{1}$$

In online RL, the optimal policy must be obtained without using knowledge of the environment (i.e., of P and R). Instead, the agent must learn by interacting online with its environment. A well-known online RL algorithm is Q-learning (Watkins and Dayan, 1992; Sutton and Barto, 1998), which in its simplest form is given by:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a)] \tag{2}$$

where the Q-function approximates returns obtained starting from each state-action pair. Under appropriate conditions, Q-learning asymptotically converges to the optimal Q-function Q^*, which can be used to act optimally with:

$$\pi^*(s) = \arg\max_a Q^*(s, a)$$

A crucial convergence condition is that the algorithm keeps selecting every action with non-zero probability. This is called *exploration*, and in this paper, we use the classical ε-greedy exploration, which at each step selects actions according to:

$$a_t = \begin{cases} \text{a random action} & \text{with probability } \varepsilon \\ \arg\max_a Q(s_t, a) & \text{with probability } 1 - \varepsilon \end{cases} \tag{3}$$

Example: Gridworld Navigation. Gridworld (GW) examples are commonly used in RL. Despite their simplicity, GWs are useful abstractions of a variety of real-world tasks, such as robot navigation. They can in principle also be used to approximate general nonlinear systems, using discretization.

We consider here the 5 × 5 GW shown in Figure 1. The agent is located in one of the cells, and its position constitutes the discrete state signal. At each step, it can move one cell into any of the four cardinal directions, leading to four possible actions. If a move leads into a wall, it fails and the agent stays put. The aim is to find the goal state (s_{goal}) in a minimum number of steps. This state is terminal, so after reaching it the trial finishes and the agent is reset to a random initial state.

The reward function:

$$R(s_t, a_t, s_{t+1}) = \begin{cases} 10 & \text{if } s_{t+1} = s_{\text{goal}} \\ -5 & \text{if } s_{t+1} = s_t \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

outputs a large positive reward upon reaching the goal state, a negative reward (penalty) upon hitting a wall, and a smaller negative reward for any other step; this latter term enforces the RL algorithm to search for a minimum-time / minimum-distance solution. □

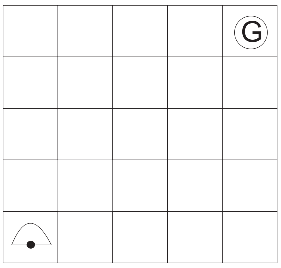

Fig. 1. GW schematic showing the agent (bottom-left) and the goal state (top-right).

3. SHAPING

In general, the shaping process starts by giving the agent a series of relatively easy problems building up to the harder problem of ultimate interest. However, shaping does not always need to return to solving the original task after solving the easy version. Two ways of performing shaping can therefore be identified:

- Temporarily simplifying the task, solving it, and then transferring the knowledge to the original task. There may be a single simplified task, or a sequence of tasks gradually increasing in complexity.
- Permanently simplifying the task. If the complexities removed are redundant, the optimal policy in the easy task is the same as in the original task, and thus the original task need not be explicitly solved. Instead, the policy found in the easy task can be used directly.

In this paper we focus on 4 major shaping methods: (i) *dynamics shaping*, (ii) *reward shaping*, (iii) *action shaping*, and (iv) *state shaping*. As their names imply, these methods modify, respectively, (i) the dynamics, (ii) the reward function, (iii) the action space, and (iv) the initial state, possibly together with the state space size (Erez and Smart, 2008). Below, we briefly describe each of the four methods, and exemplify how they can be used in the GW example introduced in Section 2. Under certain conditions, methods (ii) and (iii) can be used to modify the task permanently, as we will detail below.

Dynamics Shaping. In some problems, the physical properties of the system can be changed to simplify the task (Randløv, 2000). Changing the physical properties corresponds to changes in the transition probabilities. Of course, such a change will not always be possible.

Example: Dynamics Shaping for the GW. Consider the 5 × 5 GW, which – for the purposes of this example – now includes several obstacles in the way of the agent, see Figure 2 (left). For the purposes of shaping, the obstacle near the goal can be removed (Figure 2, right), and the resulting easier problem can be solved. Then, the knowledge gained is transferred to the harder, original problem with all the obstacles included. □

Reward Shaping. One of the most popular shaping techniques replaces the original reward function R by a shaping reward function R' , which guides the agent toward learning a good policy faster (Singh, 1992; Dorigo and Colombetti, 1998; Ng et al., 1999). Usually, R' is derived from R, e.g., by adding shaping rewards in certain situations. For example, the so-called progress indicators

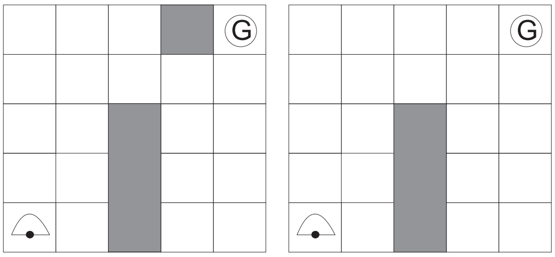

Fig. 2. Dynamics shaping for the GW. The original problem (left) is simplified by removing the obstacle near the goal (right).

(Matarić, 1997) are associated with specific goals and provide a metric of improvement relative to those goals. Under appropriate conditions, the optimal policy remains the same when using R', so the change can be permanent (Ng et al., 1999).

The weak points associated with reward shaping are that it requires significant design effort, it results in less autonomous agents, and if done improperly it may lead to unexpected (and undesirable) behavior.

Example: Reward Shaping for the GW: One progress indicator that can be used in the GW of Figure 1 is a direction-based reward. If the agent moves towards (away from) the goal state, it receives a positive (negative) additional reward:

$$R'(s_t, a_t, s_{t+1}) = R(s_t, a_t, s_{t+1}) + \begin{cases} 6 & \text{if } a_t = \text{up or right, and } s_{t+1} \neq s_t \\ -15 & \text{if } s_{t+1} = s_t \\ -9 & \text{if } a_t = \text{down or left} \end{cases} \quad (5)$$

Note that collisions are also penalized more (second branch), to prevent the agent from learning to prefer collisions instead of going down or left. The reward change can be permanent, since the optimal solution with R' remains the same as with R (go right along the upper edges of GW; go up along the right edge; otherwise go either right or up). □

Action Shaping. Agents learning to reach a certain goal by trial-and-error normally require a slow and laborious search through the space of possible actions. Thus, constraining the initial size of the action space can significantly reduce the learning time and effort (Schlesinger et al., 2000; Vereijken et al., 1992). Additionally, if the actions removed do not appear at all in the optimal policy, the simplification can be permanent. Note that removing actions is not possible when all the actions are needed to obtain a good policy.

Example: Action Shaping for the GW. Since the goal state is in the upper-right corner of the GW (see Figure 1), removing the actions 'left' and 'down' has no negative effect, as we are sure that these actions do not occur along the optimal paths (they are not in the optimal policy). For the same reason, the simplification can be permanent. Removing these two actions halves the action space. □

State Shaping. In many goal-based tasks (i.e., tasks having a goal state or region that must be reached), learning is simpler closer to the goal, and becomes progressively harder farther away from the goal (Boyan and Moore, 1995). This happens because the shorter paths to the goal reduce the risk of prolonged fruitless exploration. To accelerate learning in such tasks, learning can initially start from a close neighborhood of the goal state, which is then increased in size. The optimal solution is learned by 'growing' it out from the goal state toward the rest of state space.

Example: State Shaping for the GW. The initial position of the agent can be moved closer to the goal, and the GW size (the state space region that the agent can move in) can be reduced at the same time. Without reducing the GW size, learning may not be accelerated as much, because for any initial state, exploration may drive the agent throughout a large portion of the state space.

Meaningful GW sizes range from 2 × 2 to the original 5 × 5. The smaller GWs always contain the goal state at their upper-right corner, see Figure 3. Two scenarios can be considered: one-step transfer, and diagonally extending the GW. For one-step transfer, a single smaller GW is picked, and then the transfer to the original task is made. For diagonally extending the GW, the size is gradually increased, starting from the 2 × 2 GW until the original 5 × 5 GW. □

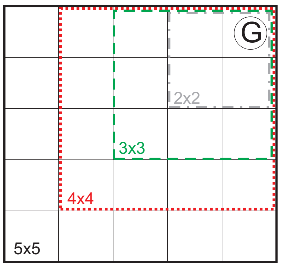

Fig. 3. State shaping for the GW. Each sub-GW is rendered in a different style and color.

4. TRANSFER CRITERIA

If the task cannot be permanently modified, an essential decision in shaping is to decide *when to transfer* learning from an easier task to a more difficult one. In this section, we propose two transfer criteria based on the agent's performance. The first criterion is based on the distance between the current solution and the optimal one, while the second criterion employs the empirical return from a set of representative states. Learning is transferred when the performance of the agent reaches an acceptable level and no longer changes significantly.

Even when shaping is permanent and no transfer is needed, the criteria developed are useful to decide when to stop learning, and then judge whether shaping has accelerated learning in comparison to starting from scratch.

4.1 Distance to the Optimum Criterion

The first criterion that we use relies on the distance between the current and the optimal solution. The distance is measured using the number of time steps needed to reach the goal, making this criterion specific to minimum-time goal-based tasks common in RL, such as GW navigation.

At the end of each trial k, the agent is initialized in simulation in every feasible state and left to follow the policy found at the end of the trial, keeping learning and exploration turned off. Then, a (positive) trial score is computed using:

$$J_k^{\text{dist}} = \frac{1}{\text{card}(S)} \sum_{s_0 \in S} (T_k(s_0) - T^*(s_0)) \tag{6}$$

where card denotes set cardinality, $T_k(s_0)$ is the number of steps taken by the policy to reach the goal state starting from s_0, and $T^*(s_0)$ is the optimal number of steps. Learning is transferred when the trial score drops below a threshold δ^{dist} for d consecutive trials:

$$J_{k'}^{\text{dist}} < \delta^{\text{dist}}, \text{ for } k' = k - d + 1, \dots, k - 1, k \tag{7}$$

Note the criterion is only used after $k \geq d$ trials have elapsed. Testing for d successive trials ensures that the decrease is not accidental, but remains steady.

If convergence to the exact optimal solution is required, then $\delta^{\text{dist}} = 0$. However, this is not always desirable in shaping, since a rough, suboptimal solution may still speed up the learning in the original problem, without requiring fine-tuning until it becomes optimal. In this case, $\delta^{\text{dist}} > 0$, and a good value will depend on the problem at hand.

Since this criterion guarantees the solution found is as close to optimal as desired, it is very useful in the analysis of shaping for minimum-time, goal-based tasks. However, it is of limited use in general, also because it requires knowledge of the optimal solution. Next, we provide a more general criterion that works in any type of problem and does not require prior knowledge.

4.2 Empirical Return Criterion

Similarly to the distance to the optimum criterion, for the empirical return criterion the agent follows, after each trial, its current policy from a set of initial states, keeping learning and exploration turned off. However, this time the number of steps or the optimal solution is not used; instead, only the empirical return accumulated along the trajectory is computed, leading to the trial score:

$$J_k^{\text{ret}} = \frac{1}{\text{card}(S_0)} \sum_{s_0 \in S_0} \mathcal{R}(s_0) \tag{8}$$

Rather than requiring good performance across the entire state space, a smaller set S_0 of representative initial states can be employed. This is useful to focus the criterion on interesting parts of the state space. Note that $\mathcal{R}(s_0)$ can be estimated with arbitrary accuracy even if there are no terminal states, by using the fact that the return is bounded by $\frac{\|R\|_\infty}{(1-\gamma)}$, where $\|R\|_\infty$ is the maximum absolute reward. To ensure an accuracy $\varepsilon_{\mathcal{R}} > 0$, a T-steps long trajectory is sufficient, where $T = \lceil \log_\gamma \frac{\varepsilon_{\mathcal{R}}(1-\gamma)}{\|R\|_\infty} \rceil$.

Learning is transferred when the trial score no longer changes significantly:

$$|J_{k'}^{\text{ret}} - J_{k'-1}^{\text{ret}}| < \delta^{\text{ret}}, \text{ for } k' = k - d + 1, \dots, k - 1, k \tag{9}$$

The criterion is only used after $k > d$ trials have elapsed.

Like for the distance to the optimum, the threshold δ^{ret} is very important and a good value for it will depend on the problem. A general guideline can be given, however, by using the already mentioned bound on the return. In particular, the difference between the scores of two consecutive trials is – conservatively – at most $2\frac{\|R\|_\infty}{(1-\gamma)}$, so δ^{ret} can be chosen as:

$$\delta^{\text{ret}} = \beta \cdot 2 \frac{\|R\|_\infty}{(1-\gamma)} \tag{10}$$

where β is a small positive constant that (a) accounts for the conservativeness of the bound, and (b) imposes how stable the solution should be. Note that this criterion easily generalizes to continuous-variable tasks.

5. SIMULATION RESULTS

In this section we investigate the effects of using the two transfer criteria in shaping RL, for the GW problem and the shaping techniques considered. In particular, we compare the number of trials required to learn with and without shaping, exploiting the proposed criteria to decide when to transfer learning (as well as when to stop learning in the original task, and in the shaped task when shaping is permanent).

Throughout all the experiments, the transferred knowledge is represented by the Q-table. This is a choice, and in general (parts of) other RL elements could be transferred, such as the policy or a learned model of the environment. To more easily identify the various variants of tasks in these experiments, the following naming convention will be used: an *easy task* is a simplified version of the original task, used during shaping; the *shaped task* is the original task, solved at the end of the shaping process; and the *unshaped task* is the original task, but solved from scratch, without using shaping. When shaping is permanent, the easy task is the same as the shaped task, and we use the latter name.

Q-learning (2) with ε-greedy exploration (3) is used throughout, with a constant learning rate $\alpha = 0.9$ and a constant exploration rate $\varepsilon = 0.1$. The discount factor γ is 0.95. In each task (easy, shaped, or unshaped) the maximum number of trials for which learning is allowed to run is 200, and the maximum number of steps in each trial is 300.

All four shaping methods are applied, using the GW examples described in Section 3. Due to space limitations, we will only discuss in detail two of the experiments: modifying the action space (permanently simplifying the task), and modifying the initial state, the second scenario, i.e., gradually extending the GW (temporarily simplifying the task). Table 1, given at the end of this section, summarizes the results of *all* the experiments.

5.1 Results for the Distance to the Optimum Criterion

The parameters of the distance to the optimum criterion are $d = 10$, $\delta^{\text{dist}} = 0$, which means the solution must be optimal for 10 consecutive trials. The 0 value for δ^{dist} was chosen because it gives the strictest criterion, and in this sense provides a lower bound on the improvement in performance that shaping can achieve.

First, *action shaping* is discussed. Figure 4 compares the results when the problem is solved from scratch (left), to those obtained by solving the problem with shaping, after permanently removing the actions 'down' and 'left' (right). In each graph, the line represents the mean performance across 30 independent runs of the experiment, while the shaded region represents the 95% confidence intervals on this mean.[1] Removing the two actions significantly increases the learning speed: the performance index of the original task converges at trial 148 while for the shaped task it converges at trial 48.[2]

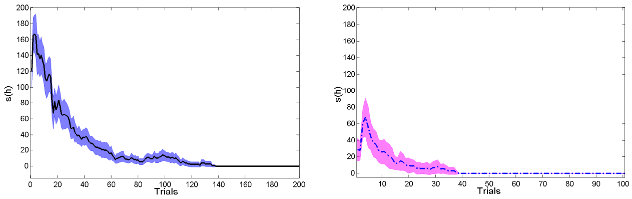

Fig. 4. Distance to the optimum: action shaping. Unshaped task (left), shaped task (right).

For *state shaping*, we consider the second scenario of the example in Section 3, i.e., gradually increasing the size of the GW from 2×2 to 5×5. The results are shown in Figure 5. For the 2×2 GW, learning converges at trial 25, for the 3×3 GW at trial 43, for the 4×4 at trial 65, and for the 5×5 GW converge at trial 41. Adding all these numbers together, shaping RL requires 174 trials, which is *larger* than starting form scratch (148, see Figure 4, left). Note that the experiments are left to run for a longer number of trials just to illustrate that they have indeed converged, but learning is in fact transferred as soon as the transfer criterion is satisfied.

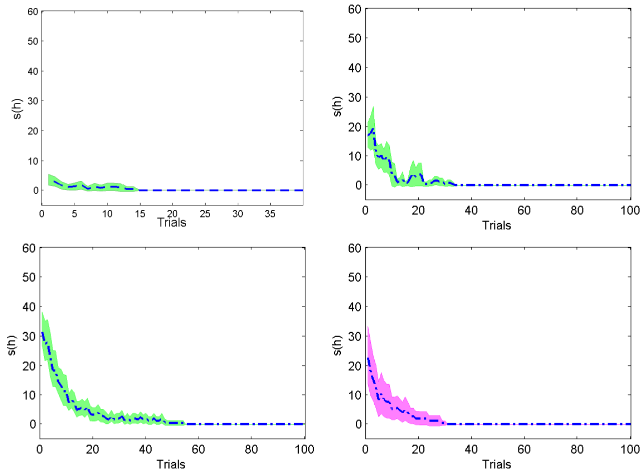

Fig. 5. Distance to the optimum: state shaping. The easy tasks are 2×2 GW (top left), 3×3 GW (top right), 4×4 GW (bottom left), and the shaped, final task is the 5×5 GW (bottom right).

5.2 *The Empirical Return Criterion*

For the empirical return criterion, we only consider one representative initial state (S_0 is a singleton consisting of this state), making this criterion less strict than the distance to the optimum. This initial state is the lower-left corner of the GW, and is representative because it is the furthest from the goal. As for the distance to the optimum, we use $d = 10$ and $\delta^{\text{ret}} = 0$, which means the score obtained must be constant for 10 consecutive trials.

Like before, we first consider *action shaping*. Figure 6 shows the results for the unshaped task (left) and the shaped task (right). The empirical return of the shaped task becomes constant at trial 32, while for the unshaped task it becomes constant at trial 89, so action shaping has significantly improved the learning speed.

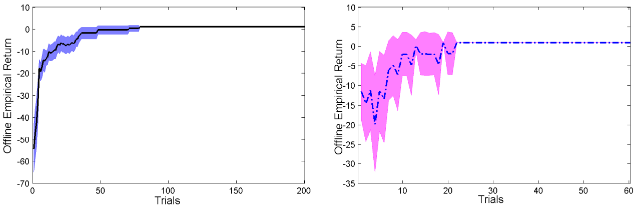

Fig. 6. Empirical return: action shaping. Unshaped task (left), shaped task (right).

Figure 7 shows the results for *state shaping*, the second scenario (gradually increasing the GW). As the figure shows, the empirical return for the 2×2 GW becomes constant at trial 17. The empirical return becomes constant at trials 17, 25, and 31, respectively for the 3×3, 4×4, and 5×5 GW. The total number of trials is 90, 1 trial more than starting from scratch.

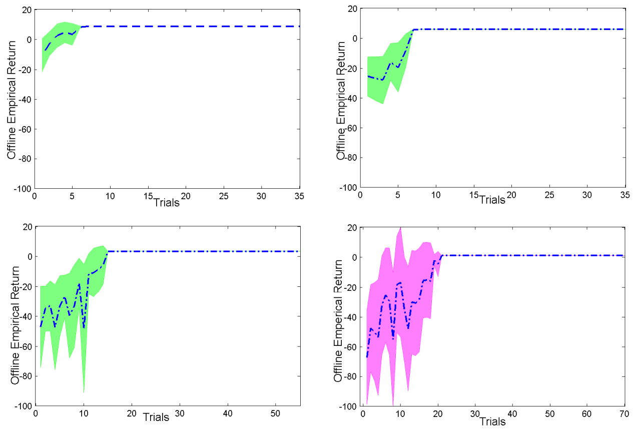

Fig. 7. Empirical return: state shaping. The easy tasks are 2×2 GW (top left), 3×3 GW (top right), 4×4 GW (bottom left), and the shaped, final task is the 5×5 GW (bottom right).

5.3 *Discussion and Conclusions*

Table 1 summarizes the results for all the four shaping methods and both transfer criteria. The dynamics and reward shaping experiments were constructed as in the

[1] In all the figures, solid line and blue shade are used for the *unshaped* task, dashed line and green shade are used for the *easy* task (not appearing in this particular figure), and dashed-dotted line and pink shade are used for the *shaped* task. The linestyle can be used to identify the graphs in grayscale print.

[2] A mean value for the number of trials to convergence is reported, by considering that the algorithm has converged d trials after the mean distance becomes equal or less than the threshold δ^{dist}.

Table 1. Summary of the results.
'State shaping I' is the one-step scenario for state shaping, while 'state shaping II' is the gradual-increase scenario. For one-step state shaping, the size of the easy task is mentioned, e.g., 3×3.

Criterion:	Distance to the optimum			Empirical return		
Learning speed ⇒ ⇓ Type of shaping	Unshaped [no. of trials]	Easy+shaped [no. of trials]	Improvement [percent]	Unshaped [no. of trials]	Easy+shaped [no. of trials]	Improvement [percent]
Dynamics shaping	153	148	3.26	96	95	1.04
Reward shaping	148	93	37.16	89	55	38.2
Action shaping	148	48	67.56	89	32	64.04
State shaping I 2×2	148	171	-15.54	89	88	1.12
State shaping I 3×3	148	164	-10.81	89	72	19.10
State shaping I 4×4	148	153	-3.37	89	68	23.59
State shaping II	148	174	-17.56	89	90	-1.12

examples of Section 3 (their results are not reported in figures).

The first important conclusion is that with both performance indices, the best shaping method is modifying the action space, followed by reward shaping. Removing two actions gives the largest gain because it effectively halves the space of solutions the agent must explore (and many of the remaining solutions are optimal, as the difference between going up and going right is only relevant at the edges of the domain). Reward shaping works well because it adds significant information about the desired solution to the reward function, see (5). Dynamics shaping provides minimal gains, and state shaping can even lead to an increase in total learning time.

In particular, state shaping is detrimental when the distance to the optimum criterion is used, while it can offer benefits with the empirical return criterion. This latter outcome illustrates that with the zero threshold we chose, the distance to the optimum criterion provides a lower bound on the benefits of shaping, being for instance much stricter than the empirical return criterion.

A critical point to be considered when analyzing the numbers in Table 1 is that they represent the largest number of trials to solve a certain task over 30 experiments. Thus, these numbers show the worse case number of trials. The results show that even in the most conservative way shaping can improve the speed of learning. It is also important to note that shaping is not guaranteed to always increase the learning speed – indeed, it can sometimes decrease it. Finally, we mention that the characteristics of the problem determine which shaping methods are applicable to it.

6. SUMMARY AND FUTURE WORK

In this paper, we proposed two novel criteria to decide when to transfer learning from easier to more difficult tasks, in the context of shaping RL. We used these criteria to study how several types of shaping influence the learning speed, for a GW navigation example. Unlike in most of the results reported in the literature, we considered the *total* learning time, including the time spent in the easy task(s), when assessing the benefits of shaping. In additional studies, not reported here, we confirmed the efficiency of the empirical return criterion for a more complex, elevator control problem.

We measured the learning speed by the number of trials needed to achieve a certain level of performance. Other measures of the learning speed are possible, such as the total number of *steps* that the agent takes until reaching a given performance level. The overall effects of shaping may change when different measures of learning speed are used, although the transfer criterion itself remains unaffected. Another interesting question is whether it could sometimes be useful to *return* to easier tasks if performance in the more complex tasks does not sufficiently improve. Moreover, there is considerable room – and need – for more theoretical work in shaping RL. A possible theoretical direction is to formally characterize which 'easy' tasks can improve the learning speed in a given original task.

REFERENCES

J.A Boyan and A.W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in Neural Information Processing Systems 7*, pages 369–376. MIT Press, 1995.

M. Dorigo and M. Colombetti. *Robot Shaping: An Experiment in Behavior Engineering*. MIT Press, 1998.

T. Erez and W.D. Smart. What does shaping mean for computational reinforcement learning. In *Proceedings 7th IEEE International Conference on Development and Learning (ICDL 2008)*, pages 215–219, 2008.

M.J. Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, 1997.

A.Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings 16th International Conference on Machine Learning (ICML-99)*, pages 278–287, Bled, Slovenia, 27–30 June 1999.

J. Randløv. Shaping in reinforcement learning by changing the physics of the problem. In *Proceedings 17th International Conference on Machine Learning (ICML 2000)*, pages 767–774, 2000.

M. Schlesinger, D. Parisi, and J. Langer. Learning to reach by constraining the movement search space. *Developmental Science*, 3:67–80, 2000.

S.P. Singh. Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning*, 8(3–4):323–339, 1992.

B.F. Skinner. *The Behavior of Organisms: An Experimental Analysis*. Copley Publishing Group, 1938.

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

M.E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10 (1):1633–1685, 2009.

B. Vereijken, R. V. Emmerik, H. Whiting, and K. Newell. Free(z)ing degrees of freedom in skill acquisition. *Journal of Motor Behavior*, 24:133–142, 1992.

C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8: 279–292, 1992.