Reward function and initial values : Better choices for accelerated Goal-directed Reinforcement Learning

Laëtitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat

Laboratoire d'Automatique de Besançon UMR CNRS 6596,
24 rue Alain Savary, 25000 Besançon, France,
{laetitia.matignon, guillaume.laurent, nadine.piat}@ens2m.fr

Abstract. An important issue in Reinforcement Learning (RL) is to accelerate or improve the learning process. In this paper, we study the influence of some RL parameters over the learning speed. Indeed, although RL convergence properties have been widely studied, no precise rules exist to correctly choose the reward function and initial Q-values. Our method helps the choice of these RL parameters within the context of reaching a goal in a minimal time. We develop a theoretical study and also provide experimental justifications for choosing on the one hand the reward function, and on the other hand particular initial Q-values based on a goal bias function.

1 Introduction

The reinforcement learning (RL) paradigm [1] provides techniques in which an agent can optimize environmental payoff for the autonomous resolution of tasks. A RL agent tries to *learn a policy*, *i.e.* it learns by trial-and-error to select actions that maximize its expected discounted future rewards for state-action pairs, represented by the action values. Q-learning [2] is a commonly form of RL where the optimal policy is learned implicitly in the form of a Q-function.

One of the main limitations of RL is the *slowness in convergence*. Thus, several methods have been proposed to speed up RL. They involve the incorporation of prior knowledge or bias into RL. [3] proposed a methodology for *designing reward functions* that take advantage of implicit domain knowledge. It involves the use of continuous reward functions and *progress estimators*. Likewise, with *reward shaping*, the rewards from the environment are augmented with additional rewards [4]. However, reward shaping can lead the agent into learning suboptimal policies and so, traps the system. [5] completed the reward shaping study and moreover, proved certain similarities between potential-based shaping and *initial Q-values*. Indeed, the most elementary method for biasing learning is to choose the initial Q-values [6]. So [7] studied various representations of reward functions and the complexity of Q-learning methods depending on the choice of RL representation. Finally, concerning *imitative reinforcement*, [8] proposed to give the learning agent access to the Q-values of the experienced agent.

Thus, reward function and initial Q-values play an important part in RL. Nevertheless, although RL has been studied extensively and its convergence properties are well known, in practice, people often choose reward function on one's intuition and initial Q-values arbitrarily [1]. In this paper, we discuss the effects of RL parameters on the policy in order to suggest a generic analysis. We validate our analysis with Q-learning algorithm. The main issue is to shed light on *how to correctly initialize RL parameters in order to obtain the desired optimal behavior in a minimal time within the context of a goal directed task.*

2 Reinforcement Learning

The framework of most of RL algorithms is a *Markov Decision Process* (MDP), defined as a finite set of states, S, a finite set of actions, A, and a transition function $T : S \times A \times S \rightarrow [0;1]$ giving for each state and action a probability distribution over states. $R : S \times A \times S \rightarrow \mathbb{R}$ is a reward function giving the expected immediate reward or *reinforcement* received under each transition. The goal is to learn a mapping from states to actions, called a *policy*, π.

In this work, we have validated our analysis with *Q-learning* [2] algorithm. In Q-learning, an *action-value function* $Q^{\pi}(s,a)$ is estimated over the learning process and stored in a tabular representation. An *action-value* represents the expected sum of rewards [1] the agent expects to receive by executing the action a from state s and following the policy π . The optimal action-value function Q^* is known to be the unique solution to the Bellman equation,

$$Q^*(s,a) = \sum_{\forall s' \in S} T(s,a,s') \Big[R(s,a,s') + \gamma \max_{a'} Q^*(s',a') \Big] \ . \tag{1}$$

Q-learning is an *off-policy* method and its updating rule is :

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

where r is the reward received for the transition from the state s to the new state s' by executing the action a. $\alpha \in]0;1]$ is the learning-rate parameter and $\gamma \in [0;1[$ the discount factor. Under some cases [9], Q-learning algorithm is guaranteed to converge to the optimal value function. The Q-Learning algorithm chooses the action according to an exploration/exploitation criteria. We used the *ϵ-greedy method*, in which the probability of taking a random action is ϵ and, otherwise, the selected action is the one with the largest Q-value in the current state [2].

3 Choice of uniform initial Q-values with binary rewards

We make the assumption that two general trends stand out : *the global policy* and *a specific behavior at the beginning* of the learning process. In this section,

[1] The rewards are discounted by a discount factor γ that controls the balance between the significance of immediate rewards and future rewards.

[2] If several values are identical, the choice will be random among greedy actions.

we are going to set out that these both tendencies depend on the shape of the reward function and on the initialization of the action-value function. We first considered *a binary reward function* which has the advantage to include a lot of cases and it is possible to extrapolate.

3.1 Optimal policy

The binary reward function is such as the reward received is always r_∞ except if the new state is the goal state and then, the reward is r_g. It's given by :

$$\forall s \in S \ \forall a \in A, \ R(s, a, s') = \begin{cases} r_g \text{ if } s' = s_g \\ r_\infty \text{ else} \end{cases} \tag{3}$$

where s' is the state obtained by executing the action a from s, and s_g the goal state. In case of all rewards are identical ($r_g = r_\infty$), the solution of the Bellman equation (1) is a constant noted Q_∞,

$$\forall s \ \forall a \ Q^*(s, a) = Q_\infty = \frac{r_\infty}{1 - \gamma} \ . \tag{4}$$

That is to say that during the learning process, Q-values for all state-action values converge to Q_∞. Nevertheless, if $r_g \neq r_\infty$, Q_∞ is the limit of the action-value function $Q^*(s, a)$ when the distance between s and s_g tends toward infinity. So, toward the goal, states are *more and more or less and less attractive* depending on r_g and Q_∞. On the one hand, if $r_g > Q_\infty$, the Q-value of state-action pairs moving to the goal state will be more and more attractive than Q_∞. So the global optimal policy is *the shortest way* toward s_g. On the other hand, if $r_g < Q_\infty$, the optimal policy is random everywhere except *a local repulsion of* s_g.

As a matter of course, the shortest way toward the goal is the sought optimal policy concerning goal-directed tasks. *So r_g must always be superior to Q_∞.*

3.2 Behavior at the beginning of the learning process

As well as the reward function, the initial value Q_i of the action-value function has an effect on the policy, but only at the beginning of the learning process. We believe that a global trend can be underscored during the first trials of the learning process.

Let's examine what the Q-values are expected to be at the beginning. If we calculate the first updating of a state-action pair (s, a) thanks to (2), such as the next state s' is not the goal state and has not been updated, we have :

$$\begin{aligned} Q(s, a) &\leftarrow Q_i + \alpha \left[r_\infty + (\gamma - 1) Q_i \right] \\ &\leftarrow Q_i + \alpha (1 - \gamma)(Q_\infty - Q_i) \ . \end{aligned} \tag{5}$$

So the discriminating value of Q_i is also Q_∞. According to the value of Q_i compared to Q_∞, *states already visited will be more or less attractive* as long as the agent has not reached plenty of times the goal state.

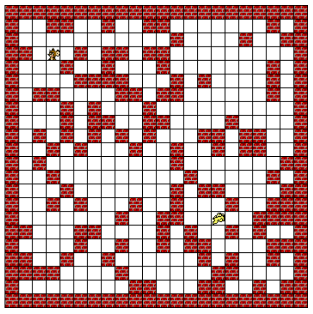
- **If** $Q_i > Q_\infty$ **:** states which have already been visited will have a value lower than the value of states which haven't yet been visited ($Q(s,a) < Q_i$). In other words, states which haven't yet been visited will be more attractive. It induces the agent *to explore* more systematically at the beginning of the learning than a random exploration. We called it *systematic exploration behavior.*
- **If** $Q_i < Q_\infty$ **:** states which have already been visited will have a value superior to the value of states which haven't yet been visited ($Q(s,a) > Q_i$). That's to say states which have already been visited will be more attractive. It leads the agent into less exploration at the beginning, which considerably slows down the learning. We named this behavior "*moving round in circles*". Of course, it's better to avoid it.
- **If** $Q_i = Q_\infty$ **:** states which have already been visited will have the same value than states which haven't yet been visited ($Q(s,a) = Q_i$). So we obtain at the beginning *a pure random behavior.*

3.3 Gridworld experiments

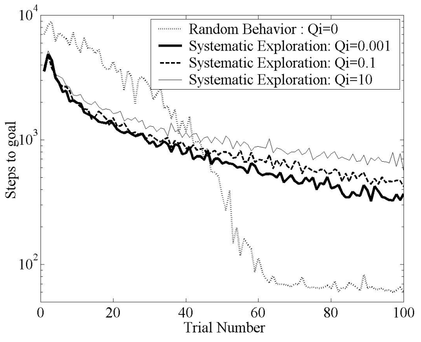
We have discussed how traditional reward functions and arbitrary initial Q-values may slow down the learning of an interesting policy. At this point, we are going *to validate this previous analysis* and we have chosen for simplicity and clarity to use at first a non-deterministic gridworld domain to demonstrate how the behavior is influenced by using binary rewards and different initial Q-values.

Benchmark. The system is represented by a mouse going around a maze (Fig. 1a). Each mouse's position is a discrete state. When the mouse reaches the goal state, the trial ends. The mouse chooses from four actions, representing an intention to move in one of the four cardinal directions (N,E,S,W). An action that would move the mouse in a wall instead leaves the mouse in its current position. Any movement moves the mouse in the intended direction with probability 0.6, and otherwise in a random state of the four neighboring states of the expected state. All trials use Q-learning with a learning-rate α of 0.1, a discount factor γ of 0.9, a tabular Q-table initialized uniformly Q_i and follow a policy where the greedy action is taken with a probability 0.9 ($\epsilon = 0.1$).

Binary reward function. Our reward function replicates the function given in (3), with $r_g = 1$ and $r_\infty = 0$. With such a reward function, the optimal policy is *the shortest way toward the goal* and the discriminating value of Q_i is 0 (Q_∞=0). Fig. 1b illustrates our previous analysis. As can be readily seen, using *the systematic exploration behavior* helped speed up learning during the first trials. The agent visited every nook and cranny of the complex maze and the goal state s_g was discovered faster than in case of a random exploration. But given that the agent was always spurred on to explore, it always took more steps to reach the goal after some trials. Besides, we used different values of Q_i for the systematic exploration and we notice the more Q_i was superior to Q_∞,



(a) Gridworld



(b) Experiments

Fig. 1. On the left, non-deterministic 20×20 gridworld with a single start state (state $[2, 2]$) and a goal state (cheese) (state $[14, 14]$). On the right, gridworld experiments with different Q_i averaged over 50 independent runs. Sreps to goal *vs.* trial number.

the more the agent explored. So, *the more the difference between Q_i and Q_∞ is important, the more the general behavior is underlined.*

Concerning the *moving round in circles behavior* ($Q_i < 0$), we do not submit any experiments because the agent took too much time to reach the goal. Anyway, *this behavior has to be avoided.*

3.4 Conclusion

This shed light on *the importance of initial Q-values.* The choice of Q_i is not trivial and must be done according to the desired behavior. When the system naturally goes away from the goal, a systematic exploration should be preferred in order to speed up learning at the beginning. Indeed, systematic exploration forces the controller to explore unvisited states, and so to approach the goal.

4 Choice of continuous reward function and heterogeneous initial Q-values

In order to broaden the scope of our study, we propose henceforth to use first two different *continuous reward functions* with uniform initial Q-values, and secondly to initialize the action value function with *a goal bias function.*

4.1 Reward function using progress estimators

First, we propose to study the use of a continuous reward function instead of binary rewards. Thus, some authors introduce reward functions by using *progress estimators* [3] or reward shaping [4]. Progress estimators provide a measure of improvement relative to an objective. They do not supply a complete information but only partial, goal-specific "advice". For instance, concerning the gridworld, the progress estimator can be an assessment of the expected number of steps needed to get to the goal from the new state s', defined as $\varphi(s', a) = d(s', s_g)$. d is the manhattan distance between the new state s' and s_g. The aim of the agent in the gridworld is to minimize this function and the parameters could be then :

$$\begin{cases} \mathrm{r}(s, a, s') = -\varphi^2 = -d^2(s', s_g) \\ Q_i = 0 \end{cases} \tag{6}$$

This way, the agent is less and less punished by approaching the goal. The global policy is the shortest way toward the goal. Given our maze, this reinforcement is spurious as there are plenty of walls between the initial state and the goal. In particular, it entails an *unlearning phenomenon* after few trials. Indeed, if the agent was taken off toward a dead end (that moves the agent closer to the goal), it would get out of the trap only thanks to exploration because states are more and more attractive toward the goal. At the beginning, Q-values are near 0 so systematic exploration is strong : going out of the trap is possible. But after few trials, turning back is tantamount to choosing a less attractive Q-value and will happen only if several exploration actions follow one another, *i.e.* seldom.

So, both reward shaping and progress estimators are risky. It's better to ban these approaches insofar as they may lead to a pernicious behavior.

4.2 Continuous reward function inspired by gaussian function

Consequently, we propose a continuous reward function such that on the one hand, *r is uniform* for some states far from the goal in order to avoid the unlearning phenomenon, and on the other hand, there is *a reward gradient* in a zone around the goal. We suggest the reward function inspired by *the gaussian function* :

$$\mathrm{r}(s, a, s') = \beta e^{-\frac{d(s', s_g)^2}{2\sigma^2}} \ . \tag{7}$$

Q_i values are uniform, β adjusts the amplitude of the function and σ, the standard deviation, specifies the *reward gradient influence area*. As a matter of course, "moving round in circles" behavior must be avoided, *i.e.* $Q_i \geq \frac{\beta}{1-\gamma}$.

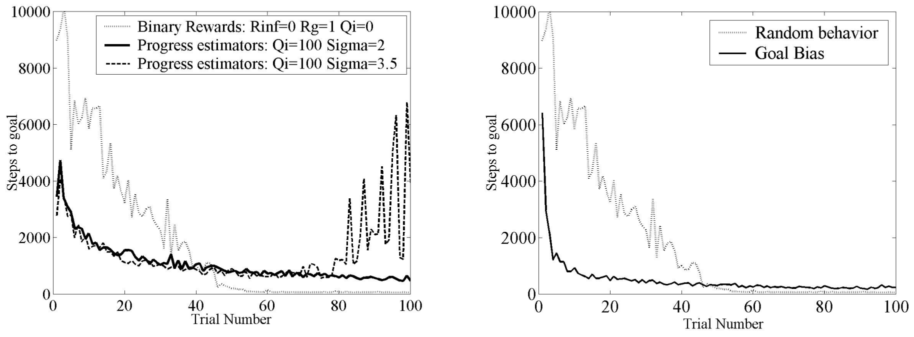
For the gridworld task, we have chosen $Q_i = 100$ and $\beta = 10$. Fig. 2a shows the unlearning phenomenon as from 80 trials with $\sigma = 3.5$ [3]. Indeed, the reward gradient influence area is too large and includes few dead ends. On the contrary, if the reward gradient influence area is only 6 steps around the goal ($\sigma = 2$), there won't be any unlearning phenomenons and the learning process is accelerated.

Such a continuous reward function is *adjustable* so as to avoid a harmful behavior. Anyway, the best approach would be to influence fleetingly the learning.

4.3 Goal Bias

In view of the importance of the action-value function initialization, we propose to be inspired by *progress estimators* in order to *initialize the action-value func-*

[3] *i.e.* states 10 steps away from the goal have uniform r.



(a) Continuous reward (b) Goal bias

Fig. 2. Gridworld experiments averaged over 20 independent runs. Steps to goal *vs.* trial number. On the left, reward function inspired by gaussian function . $Q_i = 100$; $\mathrm{r}(s, a, s') = 10e^{-\frac{d(s',s_g)^2}{2\sigma^2}}$. On the right, goal bias function with binary rewards. Random test is $Q_i = 0$. Goal bias function is $Q_i(s, a) = 0.001e^{-\frac{d(s,s_g)^2}{2\times 13^2}}$.

tion with more precise information. In this section, the reward function is the binary one given by (3) with $r_\infty = 0$ and $r_g = 1$.

We are going to settle a correct goal bias function thanks to our previous analysis. An interesting bias shall achieve *an adjustable state gradient* and in addition, must avoid the "moving round in circles" behavior. We suggest for instance *a gaussian goal bias function* :

$$Q_i(s, a) = \beta e^{-\frac{d(s,s_g)^2}{2\sigma^2}} + \delta + Q_\infty \ . \tag{8}$$

δ fixes the level of systematic exploration far away from the goal, β the amplitude of the bias and σ the bias influence area.

Concerning the gridworld, the bias is such that states near the goal are more and more interesting *a priori* than states far away from the goal. So δ and β must be chosen very small compared to one (in order to avoid too much systematic exploration). Fig. 2b presents goal bias results on the previous gridworld which are unambiguous. *The goal bias leads to a much faster learning process.* It is worth noticing that there is *no problem concerning dead ends even if the bias is wrong.* Contrary to Sect. 4.2, the effect of the goal bias function is transient. It advises the agent *only at the beginning of the learning process.*

4.4 Conclusion

Both reward shaping and progress estimators methods must be used cautiously to design a continuous reward function. Consequently, we have proposed a continuous reward function inspired by a gaussian function and whose reward gradient influence area is adjustable in order to deal with risks. *Anyway, the best solution* *is to choose a suitable goal bias function that does not lead to any problems. Our analysis helps the choice of a correct goal bias function.*

5 Experiments with the pendulum swing-up problem

Last of all, we validate our results on the continuous space control task of *a pendulum swinging upwards with limited torque* [10] (Fig. 3a). The control of this one degree of freedom system is non-trivial if the maximal output torque u^{max} is smaller than the maximal load torque mgl. The controller has to swing the pendulum several times to build up enough momentum to bring it upright and has to decelerate the pendulum early enough to prevent it from falling over.

We have chosen a two-dimensional state space $\mathbf{x} = (\theta, \omega)$. $30 \times 30 \times 9$ bases were used for the state-action space (θ, ω, u). Each trial was started from an initial state $\mathbf{x}(0) = (\pi, 0.1)$ and lasted 20 seconds. The sample time is 0.03 seconds. As a measure of the swing-up performance, we have chosen t_{up} as the time in which the pendulum stayed up ($|\theta| < \pi/4$). A trial was regarded as "successful" when t_{up} is superior to the t_{up} average of the 1000 last trials.

We tested the performance of the Q-Learning algorithm depending on the shape of the reward function and initial Q-values (Fig. 3b). In all cases, the t_{up} average after 10000 trials is around 14 seconds.

We tested first the binary reward function

$$R(\mathbf{x}, u, \mathbf{x}') = \begin{cases} 1 \text{ if } \ |\theta'| < \pi/4 \\ 0 \text{ otherwise} \end{cases} \tag{9}$$

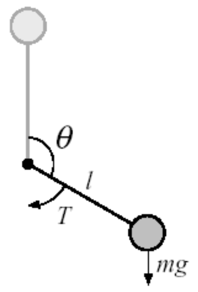
with different uniform initial Q-values. With $Q_i = 0$, the task was really difficult to learn (`bar1`) because the behavior far from the goal is random. A better performance concerning the binary reward and uniform Q_i was observed with $Q_i > 0$ (`bar2`), *i.e.* the followed behavior when $|\theta| > \pi/4$ is *systematic exploration*. The policy drives the agent to unexplored areas which are assigned higher Q-values, *i.e.* the controller is spurred on to swing the pendulum upwards. Systematic exploration is the best strategy in this specific case.

Then, we tested goal bias with binary rewards : $Q_i(\mathbf{x}) = \beta e^{-\frac{\theta^2}{2\sigma^2}} + \delta$. Given our previous results, it is obvious that the goal bias function shall favor systematic exploration when $|\theta| > \pi/4$, so we chose $\delta = 0.1$, $\beta = 1$ and $\sigma = 0.25$ (`bar3`). Goal bias does not improve obviously the learning.

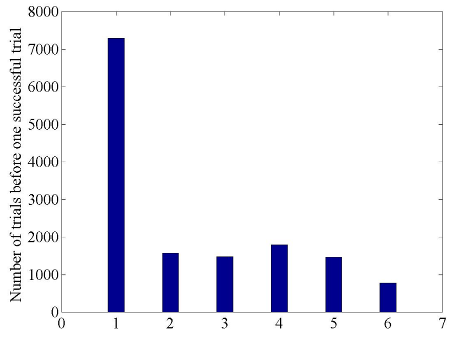
The system is a continuous space control task so the classical reward [10] is given by the height of the tip of the pendulum, *i.e.* $R(\mathbf{x}, u, \mathbf{x}') = \cos(\theta')$ and the arbitrary Q_i value is 0 (`bar4`). Thus, the pendulum moves round in circles when $|\theta| < \pi/2$ and explores systematically when $|\theta| > \pi/2$ at the beginning. The result is disappointing.

We applied our continuous reward function (7) with $Q_i = 10$ and $\beta = 1$. The distance is defined as $d(\mathbf{x'}, \mathbf{x}_{up}) = \theta'$ with $\mathbf{x'}$ the new state and $\mathbf{x}_{up}$ the goal state. So the continuous reward function is

$$R(\mathbf{x}, u, \mathbf{x}') = e^{-\frac{\theta'^2}{2\sigma^2}} \ . \tag{10}$$



(a) A pendulum



(b) Experiments

Fig. 3. On the right, comparison of number of trials before one successful trial. The simulation lasted 10000 trials averaged over 10 independent runs.
`bar1`: {binary reward ; $Q_i(\mathbf{x}) = 0$} `bar2`: {binary reward ; $Q_i(\mathbf{x}) = 0.1$}
`bar3`: {binary reward and goal bias } `bar4`: {$R(\mathbf{x}, u, \mathbf{x}') = \cos(\theta')$; $Q_i(\mathbf{x}) = 0$}
`bar5`: {continuous reward and $Q_i(\mathbf{x}) = 10$ } `bar6`: {continuous reward and goal bias}

$\sigma = 0.25$ so that the reward gradient influence area is only around $|\theta| < \pi/4$. Results (`bar5`) are near the case of a binary reward and systematic exploration.

Lastly, we tried goal bias with continuous reward function. The reward function (10) is the continuous equivalent to the binary one so we have kept this one. The reward function is continuous, it is the same for Q_i which must be higher than $\frac{R(\mathbf{x})}{1-\gamma}$. So goal bias is $Q_i(\mathbf{x}) = \beta(1 + \frac{1}{1-\gamma})e^{-\frac{d^2}{2\sigma^2}} + \delta$. We have kept previous choices: $\delta = 0.1$, $\beta = 1$ and $\sigma = 0.25$. This last simulation (`bar6`) is the better performance concerning the pendulum.

6 Conclusion

In this paper, we have called into question the arbitrary choice of the reward function and initial Q-values within the context of *goal directed tasks*. Our analysis has resulted in *rules to correctly evaluate rewards and initial Q-values* according to the desired behavior. Notably, some values of Q_i lead to a detrimental behavior that must be avoided. Thanks to our experiments, we have confirmed the presence of bounds which mark out diverse behaviors. It is worth noticing that the farther Q_i is from the bounds, the more the characteristic behaviors are distinguished.

Moreover, we advise to be wary of reward shaping or progressive estimators that may entail pernicious behavior. A *safer adjustable continuous reward function* is also suggested. At last, thanks to our conditions on the initial Q-values, we developed a *generic goal bias function*, whose main feature is to be transient. This method turns out to be *an effective way to improve the learning performance of goal-directed tasks.* Table 1 recapitulates the better choices.

Table 1. Better choices of reward function and initial Q-values for goal-directed RL.

Binary reward function for discrete state space	Continuous reward function for continuous state space
$r(s,a,s') = \begin{cases} r_g \text{ if } s' = s_g \\ r_\infty \text{ else} \end{cases}$ Choice of r_g and r_∞ : $r_g \geq \frac{r_\infty}{1-\gamma}$	$r(s,a,s') = \beta e^{-\frac{d(s',s_g)^2}{2\sigma^2}}$
Choice of uniform initial Q-values : $Q_i = \frac{r_\infty}{1-\gamma} + \delta$	Choice of uniform initial Q-values : $Q_i = \frac{\beta}{1-\gamma}$
Choice of goal biased initial Q-values : $Q_i(s) = \beta e^{-\frac{d(s,s_g)^2}{2\sigma^2}} + \delta + \frac{r_\infty}{1-\gamma}$	Choice of goal biased initial Q-values : $Q_i(s) = \beta(1 + \frac{1}{1-\gamma})e^{-\frac{d(s,s_g)^2}{2\sigma^2}} + \delta$
$\delta \geq 0$ fixes the level of systematic exploration far away from the goal, $\beta > 0$ adjusts the amplitude of the gradient, σ specifies the gradient influence area and γ is the discount factor s is the previous state, s' the new state, s_g the goal state	

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)
2. Watkins, C.: Learning from Delayed Rewards. PhD thesis, Cambridge University, Cambridge, England (1989)
3. Mataric, M.J.: Reward functions for accelerated learning. In: Proc. of the 11th ICML. (1994) 181–189
4. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: theory and application to reward shaping. In: Proc. of the 16th ICML. (1999) 278–287
5. Wiewiora, E.: Potential-based shaping and Q-value initialization are equivalent. Journal of Artificial Intelligence Research **19** (2003) 205–208
6. Hailu, G., Sommer, G.: On amount and quality of bias in reinforcement learning. In: Proc. of the IEEE International Conference on Systems, Man and Cybernetics, Tokyo (1999) 1491–1495
7. Koenig, S., Simmons, R.G.: The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms. Machine Learning **22**(1-3) (1996) 227–250
8. Behnke, S., Bennewitz, M.: Learning to play soccer using imitative reinforcement. In: Proc. of the ICRA Workshop on Social Aspects of Robot Programming through Demonstration, Barcelona (2005)
9. Watkins, C., Dayan, P.: Technical note: Q-learning. Machine Learning **8** (1992) 279–292
10. Doya, K.: Reinforcement learning in continuous time and space. Neural Computation **12**(1) (2000) 219–245