An Exploration Strategy for RL with Considerations of Budget and Risk

Jonathan Serrano Cuevas(✉) and Eduardo Morales Manzanares(✉)

Department of Computer Science, Instituto Nacional de Astrofísica, Óptica y Electrónica, 72840 Puebla, Mexico
{jonathan.serrano,emorales}@inaoep.mx

Abstract. Reinforcement Learning (RL) algorithms create a mapping from states to actions, in order to maximize an expected reward and derive an optimal policy. However, traditional learning algorithms rarely consider that learning has an associated cost and that the available resources to learn may be limited. Therefore, we can think of learning over a limited budget. If we are developing a learning algorithm for an agent i.e. a robot, we should consider that it may have a limited amount of battery; if we do the same for a finance broker, it will have a limited amount of money. Both examples require planning according to a limited budget. Another important concept, related to budget-aware reinforcement learning, is called risk profile, and it relates to how risk-averse the agent is. The risk profile can be used as an input to the learning algorithm so that different policies can be learned according to how much risk the agent is willing to expose itself to. This paper describes a new strategy to incorporate the agent's risk profile as an input to the learning framework by using reward shaping. The paper also studies the effect of a constrained budget on RL and shows that, under such restrictions, RL algorithms can be forced to make a more efficient use of the available resources. The experiments show that as the even if it is possible to learn on a constrained budget with low budgets the learning process becomes slow. They also show that the reward shaping process is able to guide the agent to learn a less risky policy.

Keywords: Reinforcement learning · Risk · Budget · Reward shaping

1 Introduction

The purpose of executing a reinforcement learning algorithm is to generate a mapping from situations to actions so as to maximize a function reward or reinforcement signal. The agent must discover by itself which actions yield the best reward by executing an experimentation process, considering as well the impact of the current decision over future rewards. These two characteristics, trial-and-error and delayed reward, are the two most important features of reinforcement learning [1]. The agent task is to develop a knowledge of its environment by using an experimentation process. This knowledge is to be exploited afterwards by the

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2017, LNCS 10267, pp. 105–116, 2017.
DOI: 10.1007/978-3-319-59226-8_11

agent to obtain a reward. However, most RL algorithms learn an (near) optimal policy without considering a learning cost. The process of learning has a cost because of the exploration process [2], since deciding to explore an unknown area implies an expense of some sort, and visiting certain areas in the environment might lead to large costs. If the agent has a limited amount of resources, or a budget, to pay for these costs, the learning process has to be optimized accounting for it. This optimization becomes critical within certain applications.

An example of such applications might appear in robotics [3], where a robot has to learn a task before running out of batteries but, at the same time, avoid certain actions that might yield a catastrophic outcome, such as the destruction of the robot. Another example might occur in a finance application [4], where a policy tries to maximize the utility at a certain time horizon, but at the same time, avoiding any chance of running out of money. The concept of risk arises naturally on the latter example, and also the concept of risk aversion, if the problem is stated as to learn a policy which yields the largest reward but at the same time minimizing the risk of running out of money. In general terms our problem is to optimize a certain parameter observing, at the same time, a safety margin over another parameter. To deal with these type of problems some techniques and algorithms have been developed under the concept of Safe Reinforcement Learning or SRL [5]. SRL can be defined as the process of learning policies that maximize the expectation of the reward in problems where it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes [6]. Therefore one could say that SRL studies the process of reinforcement learning accounting for the safety of the agent.

So far we have the problem of learning a policy to optimize a resource, while reducing the probability (or risk) of running out of such a resource during the learning process. Since the traditional learning algorithms only aim to optimize a reward, this work proposes the use of reward shaping to model the concept of risk profile β_p and learn policies accounting for it.

The remainder of this paper is organized as follows. Section 2 describes the most closely related work. In Sect. 3 the proposed approach is described in detail. Section 4 describes the learning environment and in Sect. 5 the experimental results are given. Finally conclusions and future research work are given in Sect. 6.

2 Related Work

SRL is a requirement in many scenarios where the safety of the agent is particularly important and, for this reason, researchers are paying increasing attention not only to maximize the long-term reward, but also to damage and risk avoidance [7,8]. SRL is a relatively new topic, therefore there is still some debate on how to classify the different techniques used to accomplish it. However, García and Fernández [6] proposed a SRL taxonomy which classifies the SRL techniques in two broad groups. The first group includes techniques which modify

the optimality criterion, the second group includes techniques which modify the exploration process through the incorporation of external knowledge or the guidance of a risk metric. This work fits into the first group, since it proposes and modifies the reward function in order to consider the agent's risk profile.

Risk metrics are considered in several forms in the RL literature, but in most of them the risk is related to the stochasticity of the environment, therefore it is related to the inherent uncertainty of the environment [9]. Dealing with environment uncertainty is not easy because in those environments, even an optimal policy (with respect to the return) may perform poorly in some cases. For this reason, and in order to be able to test our exploration strategy without the randomness which the environment's inherent uncertainty might generate, our work will deal only with stationary rewards, and our risk metric will be tightly related to the amount of resources the agent has at any given time and to the probability of running out of these resources.

In RL, techniques for selecting actions during the learning phase are called exploration/exploitation strategies. Most exploration methods are based on heuristics, rely on statistics collected from sampling the environment, or have a random exploratory component, i.e. $\epsilon - greedy$, which aim to explore the state space efficiently. To avoid risky situations, the exploration process is often modified by including prior knowledge of the task. This prior knowledge can be used to provide initial information to the RL algorithm biasing the subsequent exploratory process [10,11], to provide a finite set of demonstrations on the task [12], or to provide guidance [13]. It is important to mention that most of these exploration methods are blind to the risk of actions, and all of them are blind to the notion of a budget. It is left as future work to make the exploration dependent on the budget.

Finally some background on reward shaping will be given now. The practice of reward shaping in reinforcement learning consists of supplying additional rewards to a learning agent to guide its learning process, beyond those supplied by the underlying MDP [14], thus shaping its behavior. This shaping process results on a faster convergence time to an optimal policy because the additional rewards provided to the agent makes the exploration process more efficient. Therefore, reward shaping has the potential to be a very powerful technique for scaling up reinforcement learning methods to handle complex problems [15], and it can be used with any reinforcement learning algorithm such as Q-Learning [16].

3 Learning Framework Description

Our learning framework was inspired by the real-life perception of danger and the concept of risk averseness. As shown in Fig. 1, the same danger can be perceived as larger or smaller according to the agent who observes it. For a risk-seeking agent, the danger perception is diminished, but for a risk-averse agent, the danger is amplified. The same idea applies for budget; any investment is seen as less risky as the amount of budget increases. Now we will explain our learning framework.

Fig. 1. Risk-averse agent experiment.

Formally the agent starts with a given amount of resources B_0, called budget, then at each time $t = i$ it receives a reward r_i, therefore the accumulated reward at time, or step, t is:

$$R_t = \sum_{i=1}^{t} r_i \tag{1}$$

And the accumulated budget at time t is shown in Eq. 2. In this research, each step represents a cost so the budget is reduced accordingly to the length of the path.

$$B_t = B_0 + R_t \tag{2}$$

In order to guarantee that a policy can be found with a given budget we must ensure that $B_0 > E(R_t)$, where $E(Rt)$ corresponds to the expected value of the reward's sum assuming no discount rate. For this simple gridworld scenario with only two possible paths this value is simple to calculate as it will be shown later.

3.1 Shaping Rewards

Intuitively we are trying to learn a policy for some Markov Decision Process (MDP) $M = (S, A, T, \gamma, R)$[1], and we wish to help our learning algorithm by giving it additional shaping rewards which will hopefully guide it towards learning a policy which accounts for β_p. To formalize this, we assume that, rather than running our reinforcement learning algorithm on $M = (S, A, T, \gamma, R)$, we will run it on some transformed MDP $M' = (S, A, T, \gamma, R')$, where $R' = f(R)$ is the reward function in the transformed MDP, and and f can take several forms.

[1] Where S is a set of states, A is a set of actions, T is a transition function, γ is a discount factor and R is a reward function.

In traditional reward shaping research, it has been an additive function, in this research we use a simple, yet effective form to represent the reward provided to the learning algorithm. So, if in the original MDP M we would have received a reward $R(s, a, s')$ for moving from s to s' by executing action a, then in the new MDP M' we would receive reward $\beta_p R(s, a, s')$ on the same event. Now our job is to select the value of β_p to properly shape the reward and derive a risk-aware policy.

3.2 Mapping the Risk Profile Using Reward Shaping

We used a simple approach to map risk profile by using reward shaping: by carefully selecting the value of β_p. If we pick $\beta_p > 1$ the agent will have a risk averse profile, and if $0 < \beta_p \leq <1$, the agent will have a risk seeking profile. Effectively, with this simple mechanism, a risk averse agent will consider the rewards (costs) in a pessimistic way and will take lower risks, while a risk seeking agent will consider the rewards in an optimistic way and will take higher risks.

In this work we only change the original value of R when the rewards are negative, thus changing the agent's perception of the investment or effort required to earn any given final reward B_t. As a sidenote, in an analogous manner the rewards could be shaped as well by affecting only the positive rewards, however, in order to be consistent with the explanation given on Sect. 3, only the negative rewards were shaped.

4 Scenario Description

In order to test the ideas about reinforcement learning considering budget and risk we used Q-Learning as the reinforcement learning algorithm, and a grid world with some considerations which will be described on this section.

The strategy was tested in a 10×5 grid world, as shown in Fig. 2. The bottom left square has the coordinate (0, 0), while the upper right square has the coordinate (10, 5). The grid world includes a wall which the agent cannot cross (marked in black) and two special squares marked with an E and a \$ sign. The E shows the position of an exit, while the \$ square provides the agent with a special reward. As a convention the reward provided by any given square where $x = i$ and $y = j$ will be named as $r_{i,j}$, and the reward provided by the \$ square will be named as $r_{\$}$ regardless of its position. The reward $r_{\$} \geq 1$, while the rest of the squares provide a reward equal to -1.

The task that the agent has to perform is to find its way to the E square in order to maximize its final reward R. The agent can as well decide to get the coin first and then head to the exit. Since the reward of every grid world square equals -1, the reward the agent will receive after reaching the exit will depend on the Manhattan distance [17] of the route it chooses and on wether it decides to pick up the coin or not. Let's call the shortest route which picks up the coin as $Route_1$ and the shortest route which does not pick up the coin as $Route_2$. The Manhattan distance of $Route_1$ is $MD(Route_1) = 17$, while

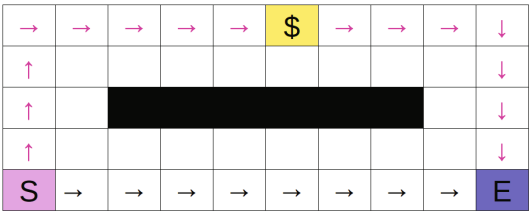

Fig. 2. Simple grid world used to show the effect on the learned policy of using reward shaping to modelate the user's risk profile.

$MD(Route_2) = 9$, therefore the reward for $Route_1$ is $R(Route_1) = -17 + r_\$$ and $R(Route_2) = -9$. Note that $R(Route_1)$ considers the reward provided by the \$ square. The accumulated reward and the accumulated budget are calculated as stated by Eqs. 1 and 2. Considering that the agent requires one time step to move from one square to another one, the time required to complete the task following $Route_1$ is $T(Route_1) = 17$ and $T(Route_2) = 9$. One could anticipate that the agents decision to choose $Route_1$ will depend on $r_\$$, however it will be shown that it depends as well on the agents risk profile β_p.

As mentioned before the initial budget B_0 is the amount of resources that the agent has to complete its task. If we think in terms of a robot and fuel then the budget is the amount of fuel the agent has, while $r_\$$ corresponds to a fuel tank that the robot can find and use. The variable $r_{i,j}$ is the reward received at any given (i, j) square, so following on with the robot example, it represents the fuel the agent has to spend in order to move. The agent aims to find the exit with as much fuel remaining as possible. If the agent runs out of fuel before finding the exit then the task is considered as failed and the game is over. With this in mind a game start when the agent receives its budget and ends if any of the following conditions occur:

- Condition 1 (Tn_1). The agent completed the task and found the exit.
- Condition 2 (Tn_2). The agent ran out of budget.

Note that finding the \$ square is not part of the task, therefore the agent has to decide wether it is convenient to visit it to receive $r_\$$ or not.

5 Experiments

To test our experiments we used the simulation software Burlap [18] and the RL algorithm Q-Learning shown in Eq. 3 with $\gamma = 0.90$ and the values for α shown in Table 1. The experiments aim to prove that (i) it is possible to learn a policy which completes a task with a constrained budget and (ii) that reward shaping is a good alternative to learn policies which account for the agent's risk profile.

$$Q(s_t, a_t) \leftarrow (1 - \alpha_t)Q(s_t, a_t) + \alpha_t[R(s) + \gamma \max_a Q(s_{t+1}, a)] \tag{3}$$

Table 1. Experiment values.

Variable	Value(s)	Comments
α	$\{0.3, 0.9\}$	The learning rate
B_0	$\{20, 30, 40, 50\}$	The budget at the beginning of each game
$r_\$$	$\{9\}$	The reward for visiting the square \$
β_p	$\{0.5, 2\}$	The agent's risk profile

In order to prove these ideas we executed experiments changing the values of β_p and B_0. The used values are shown in Table 1.

To refer to any particular combination of these variables a shorthand is used. For instance the shorthand *COIN9-0.9-2-20* is used to label an experiment with $r_\$ = 9$, $\alpha = 0.9$, $B_0 = 25$ of a risk-averse agent (the number 2 represents a risk-averse agent while the number 1 represents a risk-seeking agent).

5.1 Experiment Settings

The series of experiments is divided in learning episodes (LE), each one of these started when the agent received its initial budget and ended when either Tn_1 or Tn_2 was accomplished. One experiment consisted of 500 learning episodes and each experiment was repeated for $n = 25$ times. The metrics used to evaluate the results are described in Table 1.

Table 2. Experiment metrics.

Variable	Description
EXIT	For any given LE it represents the percentage of n where the agent reached the exit
C1	For any given LE it represents the percentage of n where the agent visited square \$
FR	Average final reward that the agent received at the end of any given learning episode

Table 3 shows that the agent can only receive three different rewards: any movement on any direction gives the agent a reward equal to -1, if it reaches square \$ it receives a reward of 9, and if it runs out of budget it receives a reward of -10. Considering this, and assuming that the agent does not run out of budget, then $R(Route_1) = -17 + 9 = -8$ and $R(Route_2) = -9$; therefore, if the agent is willing to tolerate the risk of running out of budget it should try to learn a path similar to $Route_1$, otherwise its safer choice is a path like $Route_2$.

To determine the value of $E(R_t)$ we assumed that there is an equal probability for the agent to pick either $Route_1$ or $Route_2$. Therefore $E(R_t) = 0.5 \cdot -8 + 0.5 \cdot$

Table 3. The possible rewards the agent can receive.

State change	Reward
Transition to any square	-1
Transition to square \$	9
Transition to $B_0 = 0$	-10

$-9 = -8.5$. From this simple calculation we can tell that B_0 has to be larger than $|-8.5|$ so that it is sufficient to complete the task.

5.2 Experimental Results

Each plot shown in this section includes three subplots, and each one with one of the metrics described on Table 2. At any given subplot four series are plotted, and these are labeled as described in Sect. 4. The subplots share the horizontal axis which represents the learning episode LE.

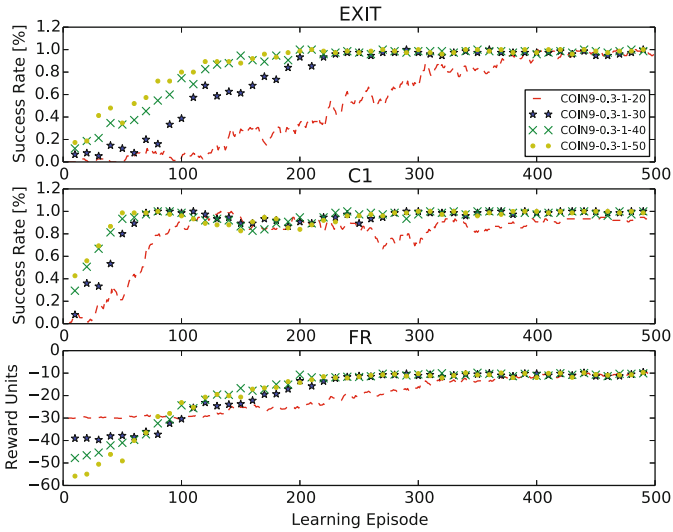

Fig. 3. Risk-seeking agent experiment.

Figure 3 shows the learning process of a risk-seeking agent ($\beta_p = 0.5$), and Fig. 4 shows the learning process of a risk-averse agent ($\beta_p = 2$). Both risk profiles were modeled by using reward shaping and the difference between them is clear: the risk-seeking agent learns to follow $Route_1$ while the risk-averse agent learns that it is better to follow $Route_2$ while both learn to find the exit located at the bottom right square of the gridworld. This can be told by observing the

subplots labeled as $EXIT$ and $C1$ of each plot. On the $EXIT$ subplot after learning episode 400 all the plotted series have a success rate larger than 75%. On the $C1$ subplot is where the differences between the risk-seeking and risk-averse agent are: after learning episode 400 the risk-seeking success rate for this subplot is more than 80%, while for the risk-averse agent is less than 40%, therefore the latter is developing a policy which ignores the square $ and instead heads directly towards the exit square.

Each experiment was tested with 4 different values of B_0: 20, 30, 40 and 50. Regardless the value of α and the agent's risk profile, the more budget the faster the agent learns its final policy, however it also requires more money to learn the same policy.

Now lets analyze the impact of modifying the budget by observing Fig. 3. As mentioned before 4 different values of B_0. Also if the agent runs out of budget (condition Tn_2), it receives a reward of -10. These two facts support the agents behavior near learning episode 0 on subplot FR, which plots the variable B_t at the moment t when either condition Tn_1 or Tn_2 is reached: the minimum value B_t that the agent can reach corresponds to $-B_0$ minus the penalization for running out of budget. For this reason the minimum value for B_t of series *COIN9-0.3-1-20* is -30. Another important observation on FR subplots is that as B_0 increases, it allows the agent to learn its policy faster, however the tradeoff is that the learning process becomes more expensive since the area below the x-axis and the FR curve increases proportionally to B_0. Table 4 shows this area from $t = 0$ to time $t = 100$ and from time $t = 0$ to $t = 200$ for the FR subplot shown on Fig. 3.

Our final analysis is related to the impact of learning rate α. A high learning rate value was used on the experiments reported previously, therefore, just to prove that this decision has no relevant impact on the learned policy we will report an experiment with a low learning rate value. Figure 5 shows the results of the same experiment plotted on Fig. 3, which corresponds to a risk-seeking agent. The only difference is the value of *alpha*: the first one uses Q-Learning with $\alpha = 0.9$ and the latter uses $\alpha = 0.3$. Both agents learn a policy which follows $Route_1$, however the agent which uses $\alpha = 0.9$ develops its policy faster. Since the rewards are stationary we expected this behavior. Therefore we can state that the value of α makes no impact on our process of modeling risk profile by using reward shaping.

Table 4. Learning cost for different values of B_0

B_0	FR @ 100	FR @ 200
20	44,896	84,799
30	55,937	88,879
40	59,931	86,766
50	63,647	91,013

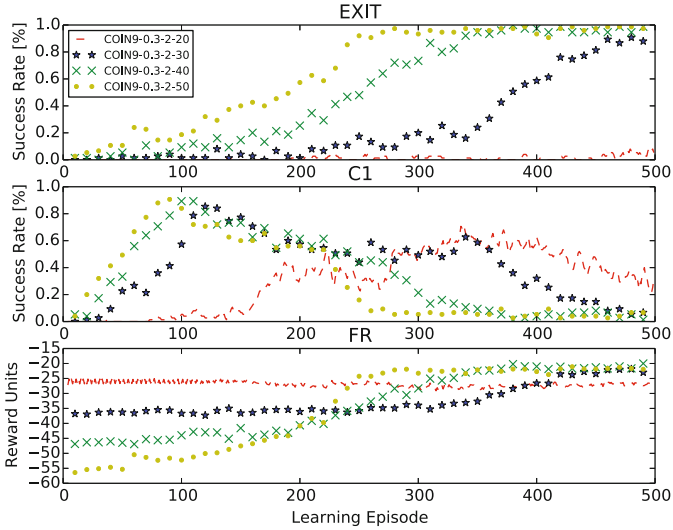

Fig. 4. Risk-averse agent experiment.

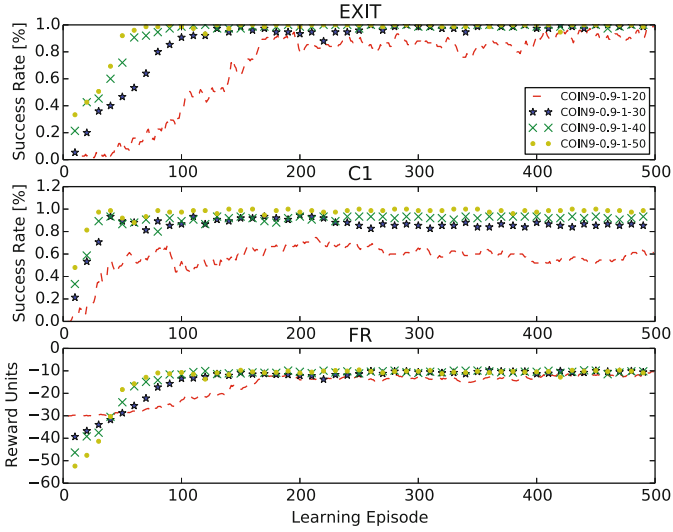

Fig. 5. Risk-seeking experiment with low learning rate value.

So far our experiments showed that: it is possible to learn an optimal policy with a constrained budget, given that this budget is large enough to cover a minimum cost[2]; and that with a high budget the agent learns the optimal policy

[2] In this case a transportation cost set by the distance between the agent's starting point and the exit square.

faster, but since it is allowed to explore more, the learning cost increases as shown in Table 4.

6 Conclusions and Future Work

This work provided some references about the lack of consideration of budget in RL, and how relevant the concept of budget is within several RL applications. As mentioned before, RL techniques attempt to optimize a reward no matter the learning cost, and this situation is not suitable when the learning process requires resources which are limited, as occurs in many real scenarios. Furthermore, not all agents share the same risk tolerance, however RL techniques do not account for this issue either.

For these reasons we aimed to work on a SRL technique which consider budget and risk profile. We provided the agent a negative reward for running out of resources and weighted this reward by using reward shaping. The negative reward received by the agent if it runs out of budget forces it to consider the risk of trying to improve its final reward, while the shaping process allows us to quantify a qualitative agent's attribute, such as how risk tolerant it is, and use it as an input for the learning process.

Our results showed that it is possible to learn a policy even on a constrained budget. They also showed that the reward shaping process helps learning algorithm to understand the agent's risk preferences, and allows it to learn different policies according to its risk profile, i.e. sometimes the agent decided that, in order to maintain a safety level, it was not an optimal policy to visit square $.

Our future work will be focused on developing strategies to determine the minimum budget required by a given scenario in order to determine if a solution is reachable with a certain budget or at a given time horizon for any given profile. This task is not hard to do in our simple grid world, however it becomes a very important issue in non-trivial scenarios. We also aim to develop exploration strategies which consider a budget input; the idea that justifies to have this target is that an agent should be less prone to exploring new actions as it resources run low. There is also work to be done related to improving the exploration process accounting again for budget and risk profile.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
2. Thrun, S.B.: Efficient Exploration in Reinforcement Learning. Springer, New York (1992)
3. Mahadevan, S., Connell, J.: Automatic programming of behavior-based robots using reinforcement learning. Artif. Intell. **55**(2), 311–365 (1992)
4. Nevmyvaka, Y., Feng, Y., Kearns, M.: Reinforcement learning for optimized trade execution. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 673–680. ACM (2006)

5. Thomas, P.S.: Safe reinforcement learning (2015)
6. García, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res. **16**, 1437–1480 (2015)
7. Mihatsch, O., Neuneier, R.: Risk-sensitive reinforcement learning. Mach. Learn. **49**(2–3), 267–290 (2002)
8. Heger, M.: Consideration of risk in reinforcement learning. In: Proceedings of the Eleventh International Conference on Machine Learning, pp. 105–111 (1994)
9. Coraluppi, S.P., Marcus, S.I.: Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes. Automatica **35**(2), 301–309 (1999)
10. Driessens, K., Džeroski, S.: Integrating guidance into relational reinforcement learning. Mach. Learn. **57**(3), 271–304 (2004)
11. Martín H., J.A., Lope, J.: Learning autonomous helicopter flight with evolutionary reinforcement learning. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2009. LNCS, vol. 5717, pp. 75–82. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04772-5_11
12. Abbeel, P.: Apprenticeship learning and reinforcement learning with application to robotic control. In: ProQuest (2008)
13. Garcia, J., Fernández, F.: Safe exploration of state and action spaces in reinforcement learning. J. Artif. Intell. Res. **45**, 515–564 (2012)
14. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: theory and application to reward shaping. In: ICML, vol. 99, pp. 278–287 (1999)
15. Dorigo, M., Colombetti, M.: Robot shaping: developing autonomous agents through learning. Artif. Intell. **71**(2), 321–370 (1994)
16. Devlin, S., Kudenko, D.: Dynamic potential-based reward shaping. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 433–440 (2012)
17. Black, P.E.: Manhattan distance. Dict. Algorithms Data Struct. **18**, 2012 (2006)
18. MacGlashan, J.: Brown UMBC reinforcement learning and planning BURLAP. http://burlap.cs.brown.edu/. Accessed 5 Jan 2017