Intelligent Systems

Seminar assignment 2 - Reinforcement Learning

Authors:
Andrejaana Andova
Marko Prelevikj

17.12.2015
Ljubljana

- **Implementation of getStateDesc()**

We used the provided base of getStateDesc() function to yield next generation state description function in order to improve the performance of the agent.
Improvements were done as follows:

- another value of the states is added to tell the agent where the fuel is, in order to make it chase the fuel
- Two extra states were added in order to perceive the diagonals of the agents, that is diagonally to the left and diagonally to the right, as shown in figure 1.

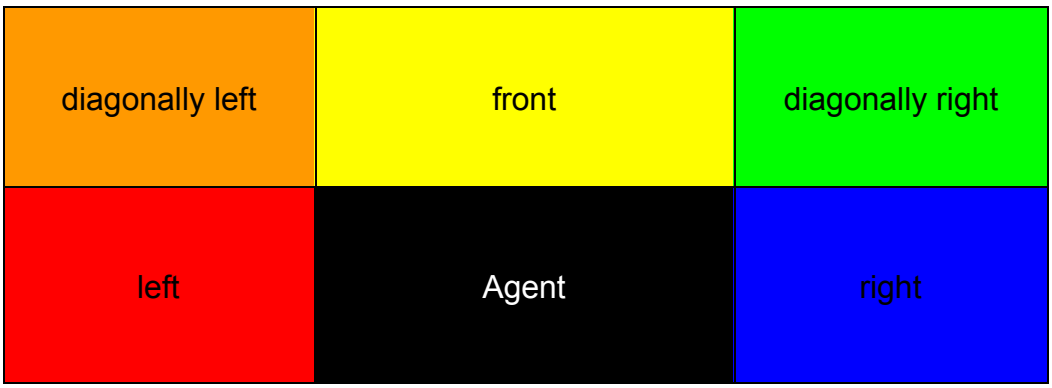

FIgure 1. Agent vision after upgrade.

To get a variety of results to analyse we used all three version of the function (provided, with fuel indicator, with fuel indicator and 2 extra states).

- **Implementation of getReward()**

The trickiest part of this assignment was writing the reward function.
In the end we managed to write 3 versions of the function which operate somewhat differently:

(method 1): The first version operates very simple, it uses only three states (left, front, right) and only 2 values of the states (1 - free, 2 - potential collision with a vehicle).

(method 2): Next, five states were used, as described in figure 1. The approach in this version is to avoid obstacles more by using 2 more states, so it can go around the rest of the vehicles and not bump into them if they are diagonally of the vehicle, which turned out to be a problem in the previous approach.

(method 3): Finally, the upgrade of the previous version is make the vehicle go around all of the other vehicles and to pick up the fuel for maximum performance.

- **Results and analysis**

The following figures are the results we got after a running the following test:

Every method was tested for the provided parameters: discount factor, number of road lanes, number of vehicles, number of maximum trials per applying of the q-learning algorithm.

Since it was not possible to analyse every permutation of the parameters, while one of the parameter was tested the rest were left at their default values with the exception of the number of maximum trials which was set to 50 due to saving processing time.

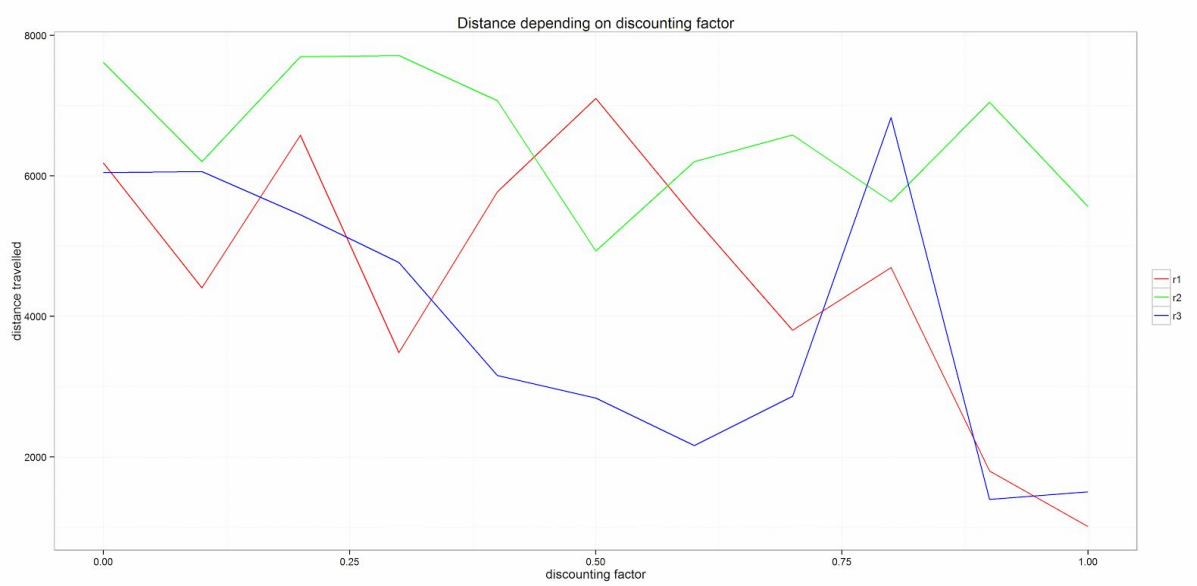

Figure 2. How the discounting factor influences the average distance travelled

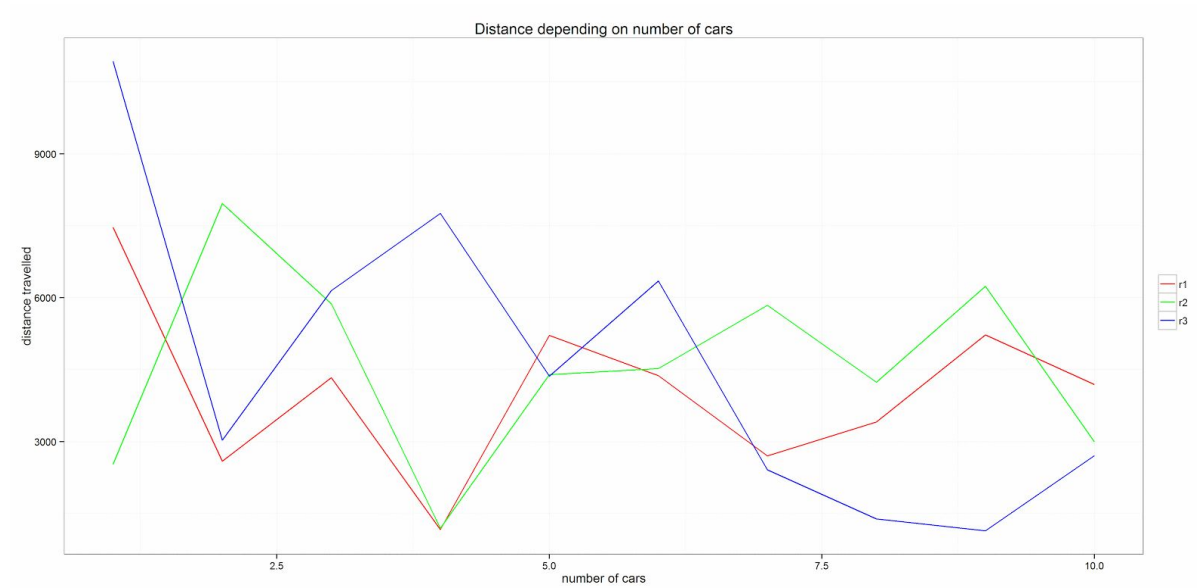

Figure 3. How the number of cars influences the average distance travelled

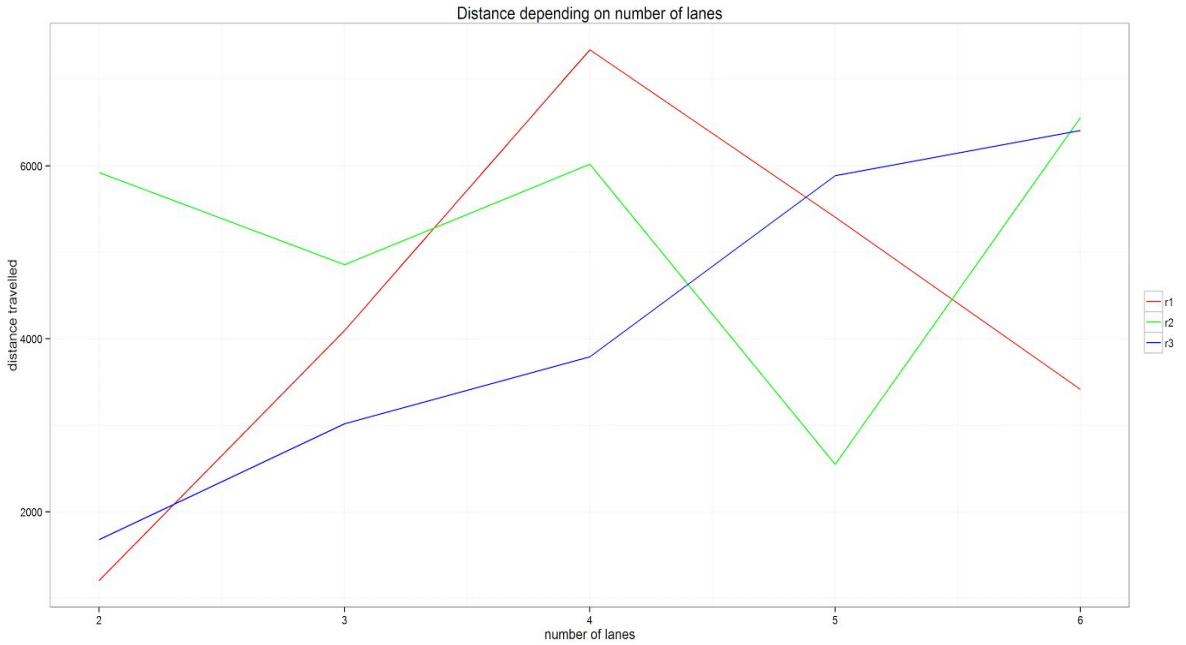

Figure 4. How the number of lanes influences the average distance travelled

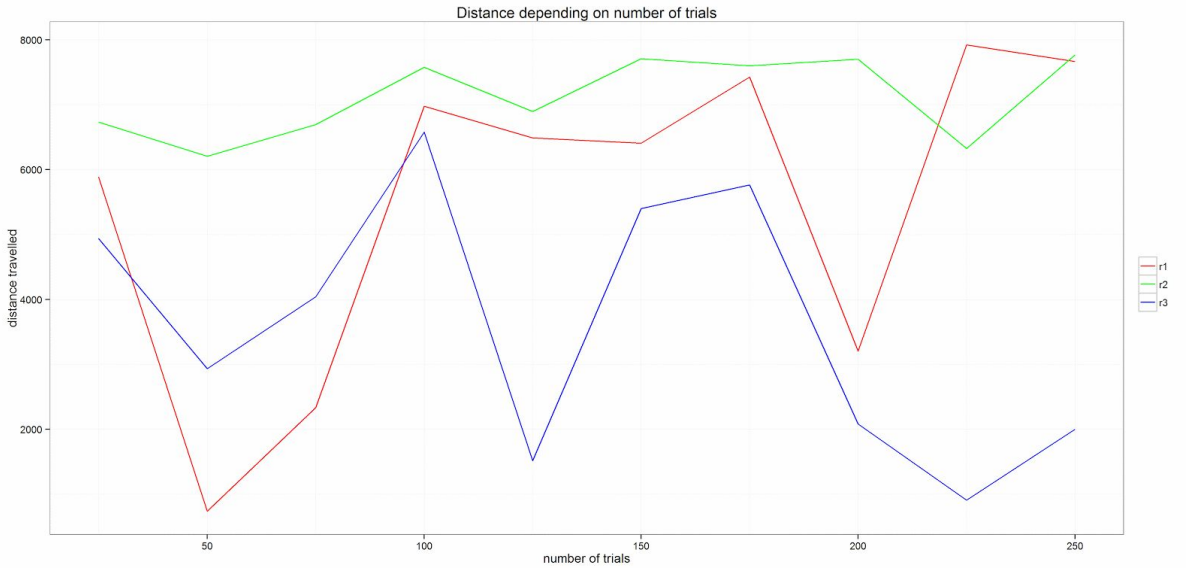

Figure 5. How the number of maximum trials influences the average distance travelled

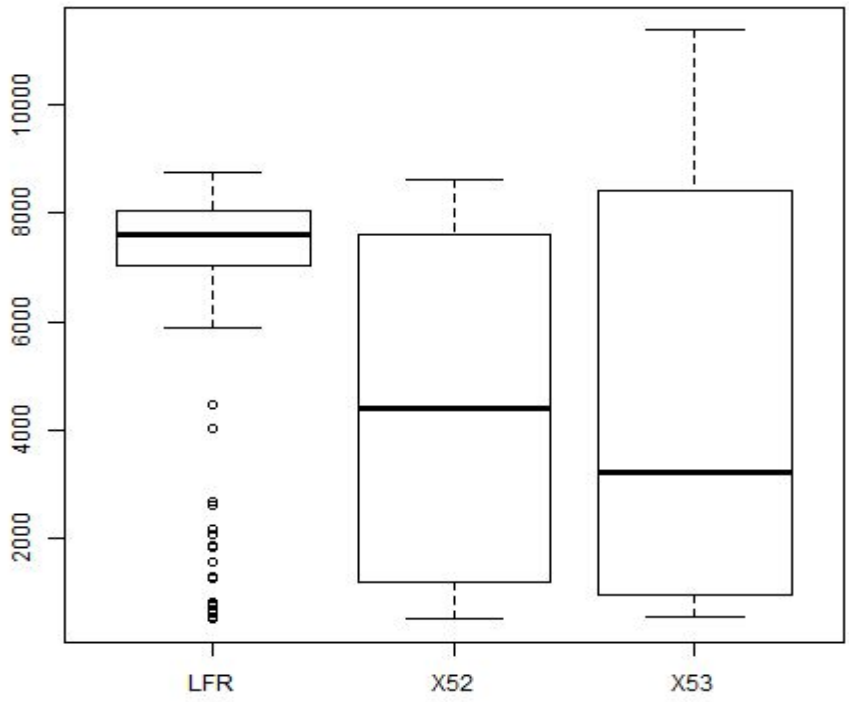

Figure 6. Different methods given 100 trials
LFR - method 1, x52 - method 2, x53 - method 3

Provided that the analysis is done with only 50 iterations it is no wonder why the best method on the graphs is the first one, the most simple one. 50 iterations are not enough for the q-learning algorithm to come out with a good estimation for the agent because there are too many states and state values that it needs to cover. But whenever it discovers one of them it surely takes full advantage of it.

On figure 2. the dependence of the discounting factor is shown. The discounting factor is the one responsible for taking into account the previous steps that were made. So, the less it is taken into account (0), the more myopic the solution is.

On figure 3. is shown that our vehicle is not the best driver driver, since it is easily bumping into them when the number of vehicles is increased. This is due to low number of iterations during the learning process and the high discounting factor (meaning that a lot of previous actions are taken into account).

On figure 4. is shown that the third method is almost linearly improving with the number of lanes, which can be explained again with the number of iterations, and as well as not many cars are in the way so it can speed up and reach further, while gathering the fuel in its proximity.

On figure 5. the dependence of the maximum number of trials is shown. The number of trials is influencing how much the agent is going to learn about the environment

and the greater the number is, the more it has learned. It should give better results. So on the figure 5. is shown that our most advanced method is not the most efficient one, but the second one is the best.

On figure 6. are shown the results of 100 simulations with the default parameters as listed previously. It is important to note that the more simple the method is, the less dispersed data you will get while learning.

Further improvements of the experiment can be done if the current experiment is repeated for all possible parameters, that way we can get a clearer picture of the knowledge of the agent and what it is sensitive to.