Grid World

Olof Harrysson, harryolo@fit.cvut.cz 18.5.2017

Problem formalization

An agent finds itself in grid world, see Figure 1. The agent can either walk north, south, west or east. The agent should get to one of the boxes marked as +1 or -1 to stop the simulation. If the agent ends up in the +1 it's rewarded with a positive reward equal to 1. Similarly it gets a negative reward, or is punished, if it ends up in the box marked with -1. The grayed out box at positions (2,2) is blocked by an object and it's therefore not possible for the agent to walk there. If the agent tries to walk towards the object, the agent simply bumps against it and remains at its previous position. Walking towards the outer edges of the world has the same consequence.

The actions in grid world is probabilistic. This means that if the agent tries to go towards a direction, it only succeeds with a probability of 0.8. The agent will have a 10% chance of going right angled from the specified direction. For example, if the agent tries to go north, it has a 80% chance of going north, a 10% chance of going east and a 10% chance of going west.

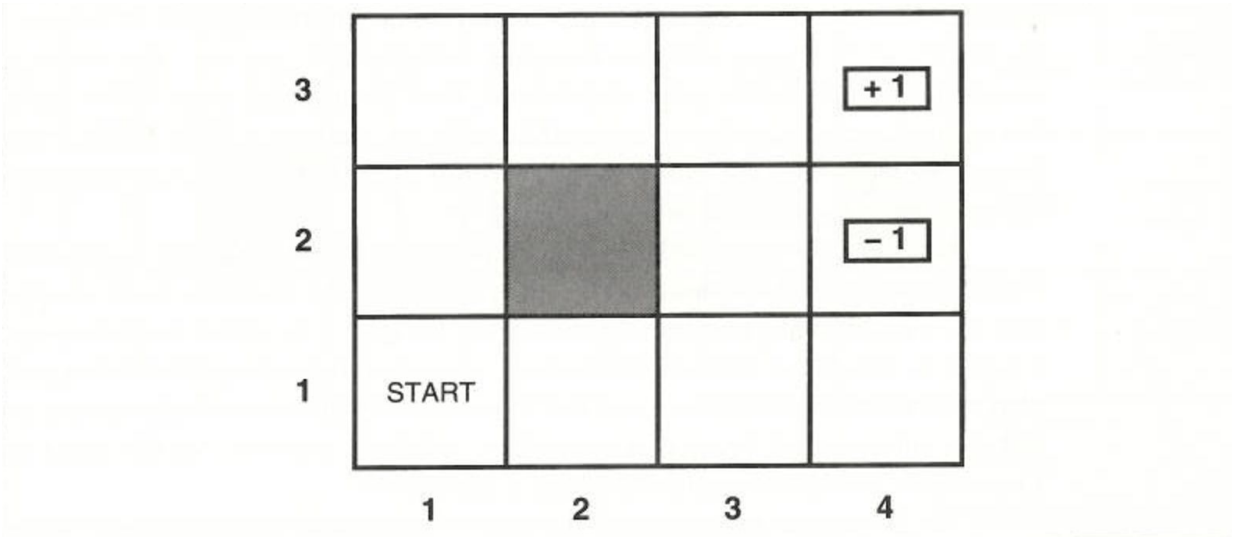


Figure 1: Grid world

To encourage the agent to device a plan, and not just wander aimlessly, a small negative reward is given each time the agent moves. This is referred to as an action cost.

One can think of this simulation as a beach where the agent's goal is to get to the water. The beach is hot so it burns the agent's feet slightly and encourages it to find it's way to the water. On the beach there is also some broken glass it wants to avoid stepping on.

The water is the +1, the glass is the -1 and the hotness of the beach is the action cost.

What is the best action the agent can take for each of the positions?

Policy Iteration

The goal with policy iteration is to find the optimal policy.

Firstly an arbitrarily policy is selected, for example do the action walk north in every position. From now we will refer to position as state. Setting the policy to down/left will not work in this implementation as for some starting states the agent will never reach a terminal state and simply keep on walking. This could be fixed with an upper bound on how many steps an agent can take.

Value iteration

The non terminal states gets an initial value of 0. The states' values are then evaluated used the policy. For example with the policy of walk north, the state next to the +1 has a chance of going into the +1 whilst the state under -1 has a smaller chance of walking into the +1 and a larger change of walking into -1. The state next to +1 will thereby get a larger value.

The states' values are determined iteratively. One can think of the values propagating outwards from the +1.

Once the values stabilize the agent chooses its action that gives the best outcome for each state. This is also affected by the probability of an action not working as intended.

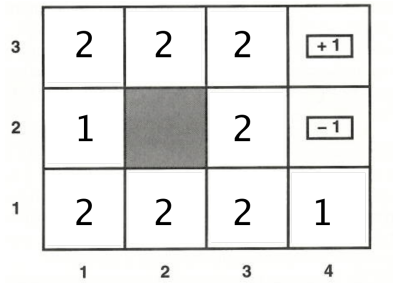
The policy is now updated and and compared to the old policy. If they are the same, that means that the policy is optimal. If they don't, the value iteration is performed again but with the new policy.

Threshold Values

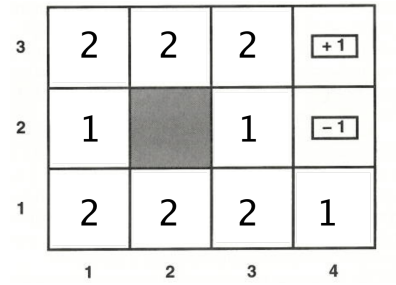
A loop was constructed that ran the pre-working solution from russel and the project implementation. They both took an action cost as a parameter and returned the optimal policy.

If they weren't equal, the project implementation was probably faulty and the program terminated. Otherwise the program printed the policy and the action cost. The following figure shows all the different policies found when running the loop 100 times with the action cost starting at -3 and ending at 0.

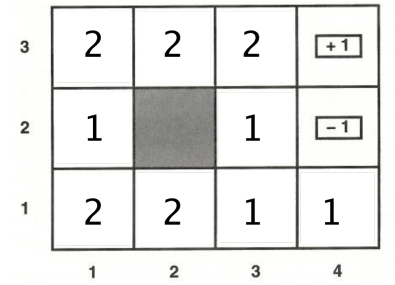
Action cost: -3



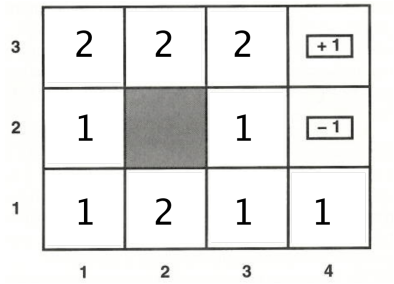
Action cost: -1.6364



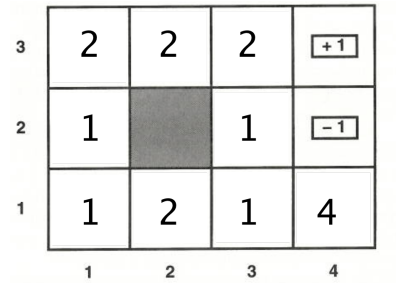
Action cost: -1.5455



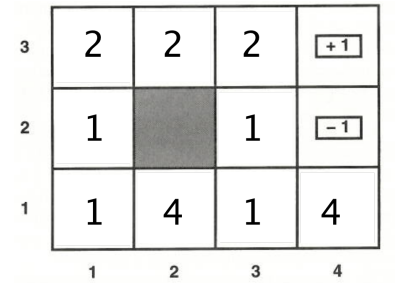
Action cost: -0.7273



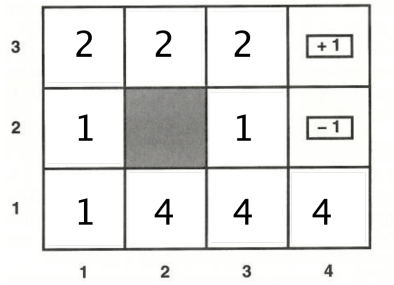
Action cost: -0.4242



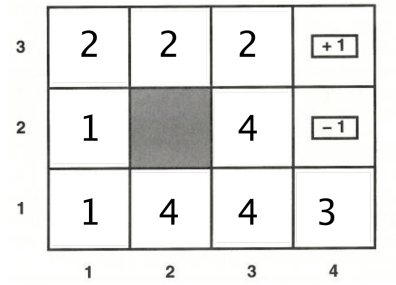
Action cost: -0.0606



Action cost: -0.0303



Action cost: 0



When the action cost is large, the agent tries to end the simulation quickly. It barely even tries to get to the +1. On the other side of the spectrum, when the action cost is 0, the agent doesn't take any chances of falling into the -1. It does this by walking directly away from it as this can still transport it to angled directions.

If the action cost is positive, the simulation never ends. The optimal strategy is to never risk falling into the terminating states. This is how the that policy would have looked.

Action cost: {any positive number}