Cliff walking problem

Vladimir Demin

Brest State Technical University
spas.work@gmail.com

Abstract

In this work, we test some machine learning techniques by observing how an agent explores a grid-world with a cliff. Starting with a random walk, we compare it with an agent behaviors learned with Genetic Algorithm, Reinforcement Learning and Finite State Machine.

1 INTRODUCTION

In this project, we study machine learning techniques such as Random Walk, Genetic Algorithm (GA), Reinforcement Learning (RL), and Finite State Machine (FSM), by observing an artificial agent who travels a virtual world called a grid-world (Fig. 1). See, for example (Seijen et al. 2009). The world has the point for the agent to start with, the point for the agent to aim as a goal, and dangerous cliff to which if the agent falls the agent will die.

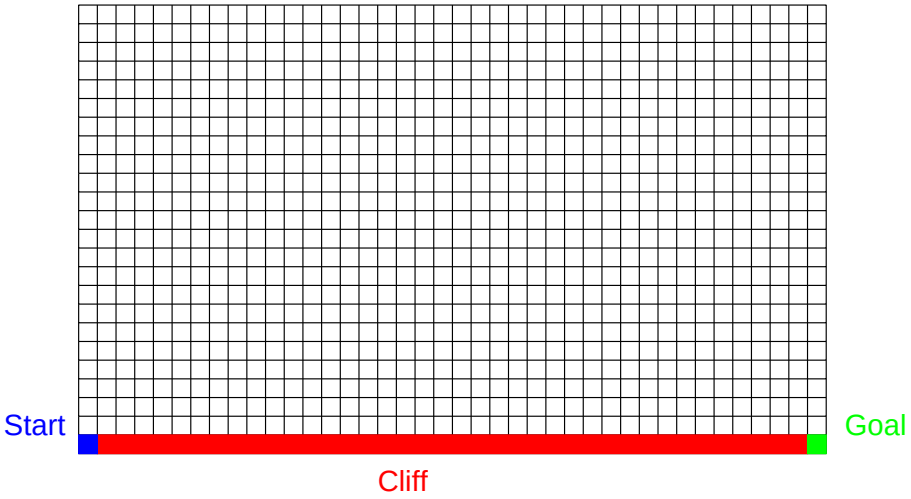

Figure 1: A grid-world with cliff.

We call "agent" who are to learn to behave intelligently, and here, agent's aim is to reach the goal. The size of the grid is $M \times N$. Agent starts at the leftmost cell in the bottom, that is, (1, N). We now denote it as (1, 1). The goal is the rightmost cell in the bottom. All the cells between (1, 2) and (1, N - 1) is the cliff. If agent enters the cliff, which means the agent falls into the cliff, then agent will die. So, the aim of the agent is to reach the goal alive. Example of the route of an agent is shown in Fig. 2.

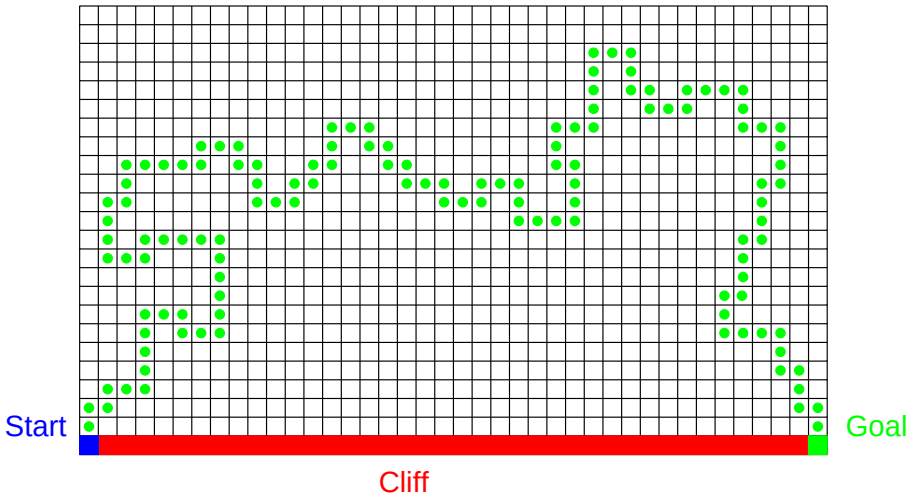

Figure 2: Example of a route that an agent succeeded in reaching the goal.

Agent can move only one cell at a time to the neighboring cell, that is, up, down, right and left, unless the agent touches the border. When the agent touches the border, the action that makes the agent cross the border is not performed but it must remain stopped at the point waiting until the next action. For example, if the agent is at (1, 3) and the action is to left, then agent remains at that point, and if the next action is to up, then it moves to (1, 4), or when the next action is right then agent moves to (2, 3).

1.1 Random Walk

Agent moves randomly one step upwards, downwards, to the right, or to the left, if the action is possible. If the movement is not possible due to the border of the grid-world, the agent does nothing and decide the next action again at random. This is repeated until the agent reaches the goal, or die by falling down into the cliff. The maximum number of steps allowed depends on the size of the grid-world. For example, agent can move 1000 steps unless it dies during its travel.

1.2 Genetic Algorithm

Borrowing the idea from the biological evolution, we can solve certain types of problems by an algorithm which is called Genetic Algorithm (GA) (Holland, 1975). In this case we express the problem we want to solve by a vector which is called chromosome. Chromosome is made up of a number of genes. At the beginning we create a population of 100 chromosomes at random. They are not good solutions at all because they are randomly created. But some are a little better than others. So we pick up two chromosomes such that better chromosomes are more likely to be chosen. Here, let's choose them at random from better half of the population. This is called truncate selection. Then, we create one child from two parents selected with an operation called uniform-crossover where we choose genes from one gene to the next by picking up gene randomly from either of the two parents.

1.3 Reinforcement Learning

In Reinforcement Learning (RL) (Sutton et al. 2005), an agent takes one state at a time chosen from predefined all possible states. In each of those states, all possible actions are given each with a probability of how likely the action will be chosen.

Agent in RL behaves following its policy. Thus, RL is defined with a set of states and a set of actions. Then, we have a table called policy in which each pair of all possible states and actions is given its probability to be occured. Starting with a random assignment of this probability at the beginning, the policy will be renewed according to the agent's experience. One of the methods to renew policy is called Q-learning.

1.4 Finite State Machine

For the exploration of our grid-world we also can exploit Finite State Machine (FSM) See (Jefferson et al. 1992). FSM is specified by some states. The number of states here should be 2^n, that is, like 2, 4, 8, 16, or 32 for the reason we will mention below. All possible inputs and outputs should be predefined, and one input in any state should result in an output and the state change.

2 EXPERIMENT

Success or failure of the agent depends on the size of the grid-world. We experiments by changing the size of the grid-world – from small one to the large one – with M and N being increased, from 4 to 100.

Also we experiment to evaluate which method is quicker and/or more efficient with the size being fixed to 100×100.

2.1 Random Walk

Running the algorithm with random number seed being changed from run to run, we count the number of agents who succeeded. Examples of the route the successful agent follows are shown as well as the route the agent failed.

2.2 Genetic Algorithm

For our problem of exploring 100×100 grid-world with cliff we create 100 chromosomes with 500 genes at random. Then we exploit truncate selection and uniform-crossover. After we evaluate a quality of chromosomes we take 50 'good' ones, which means that agent didn't die. Thus we produce 100 children. We repeat this actions until at least one of the cromosome succeeds.

2.3 Reinforcement Learning

State in our current study of agent's exploration in a grid-world is the cell where the agent locates. Hence we have $M \times N$ different states. The possible actions are up, down, right and left.

To choose the action in one state, we employ what we called ϵ-greedy strategy. That is, we choose an action at random with probability (a pre-fixed small random value such as 0.1) and the action is chosen with the highest value in the Q-table with the probability $1 - \epsilon$. We renew the policy table what is called Q-learning as follows. We now assume the state of the agent at time t is s_t and the action policy table gives is a_t.

$$Q(s_t, a_t) += \eta(r_{t+1} + \gamma \max_{\alpha} Q(s_{t+1}, a) - Q(s_t, a_t)).$$

where $\max_a Q(S_{t+1}, a)$ is the action with the highest value in the state of s_{t+1} in the Q-table, η is called learning ratio which determines how this renewal influences the overall learning, and γ is called discount rate which determines the importance of the reward received in the future. Anyway both values take a value between 0 and 1. We try this experiment by changing these two parameters. Reward at time t is expressed by r_t. In our case, r_t is -10 when the agent tried to cross the border, -500 when the agent falls into the cliff, +500 when agents reaches to the goal, and all other empty cell gives the agent reward -1.

We evaluate how many iterations algorithm needed to solve the problem on the gird-worlds from 4x4 to 100×100. We will show a good result and a bad result we obtainbed by RL with the Q-learning.

2.4 Finite State Machine

For our problem, inputs are fivefold, that are empty, cliff, goal, wall or border. Outputs are fivefold too, that is, either of go-forward, turn-left, turn-right, turn-back, or stop (when the agent reaches the goal.)

Assuming 2 state-FSM, one example of the transition table will be like:

Table 1: Example of simple FSM with two states

State	Input	output	next state
0	1	3	0
0	2	2	0
0	3	3	1
0	4	1	0
0	5	5	1
1	1	4	0
1	2	2	0
1	3	2	1
1	4	4	0
1	5	3	1

The first 2 column is automatic depending on the number of states & inputs. The 3rd and 4th columns are given at random at the beginning.

We will try to solve cliff walking problem with FSM with states of 4, 16, 32, 64 and so on. We make FSM evolve by GA creating the first generation at random.

3 RESULTS AND DISCUSSION

3.1 Random Walk

Random walk is not good method to take a goal in solving cliff walking problem. Nevertheless we obserbed successes from time to time. So, we start on this method for the sake of comparison of the results of other methods.

First, we search the maximum capability of random walk. We ovserved a fate of agents out of 1000 trials. We count the number of agents who succeeded in reaching the goal within 1000 steps with the size of grid-world increasing from 3×3 to 100×100. See Fig. 3. As might be seen in the Figure the largest case where we found at least one successful agent was 33×33.

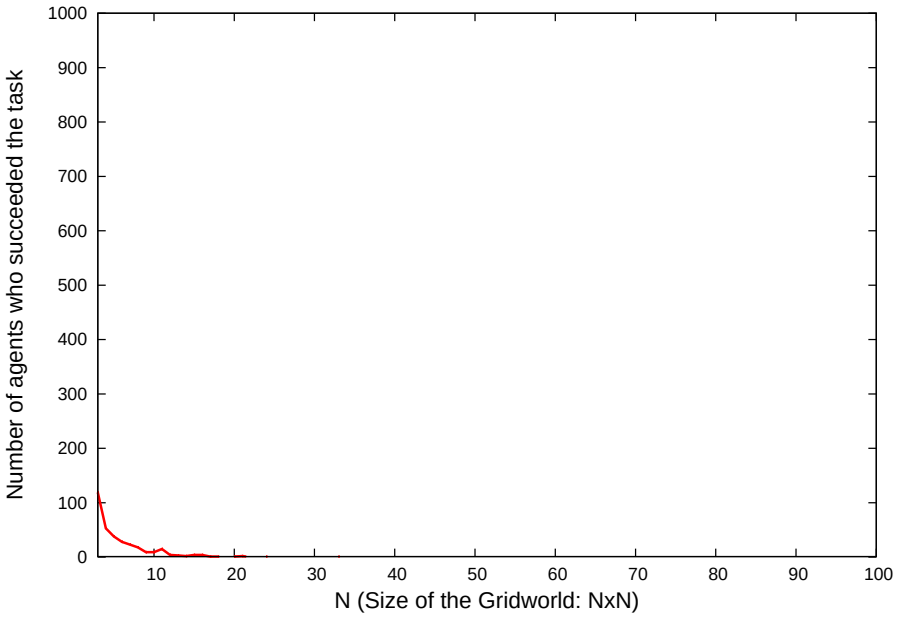

Figure 3: The number of successful agents out of 1000 runs with grid size being increased from 3×3 to 100×100.

In order to study a limit of this experiment, we allowed the agents to explore 1500 steps, instead of 1000, unless they died in the cliff. Number of trial in each size of grid-world is 100000, instead of 1000. See Fig. 4 and Fig. 5.

We can see that maximum size of grid-world in which we found a successful agent was 80×80. We need much more time to observe 1500 steps of each of 100000 agents.

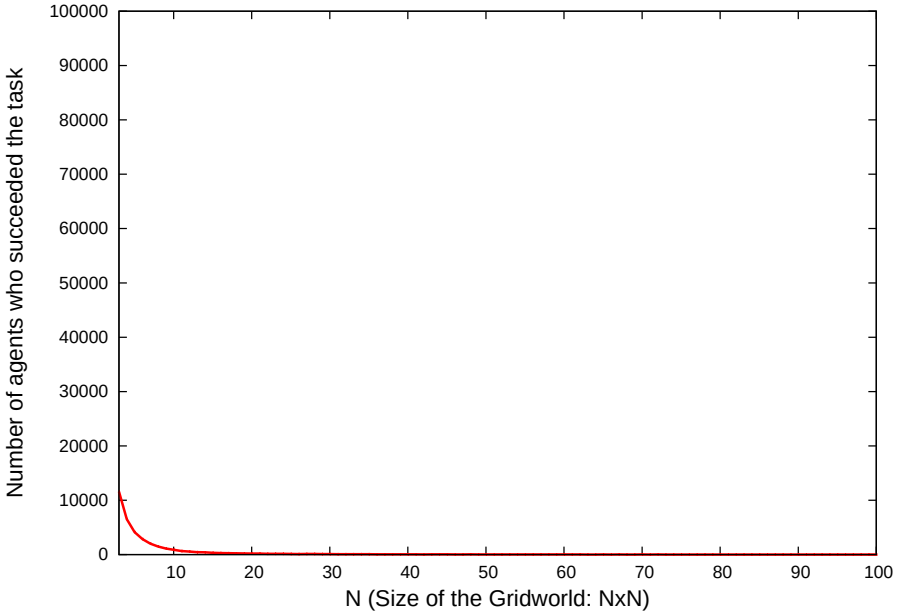

Figure 4: The number of successful agents out of 100000 agents allowed 10000 seps as a function of the size of a grid-world.

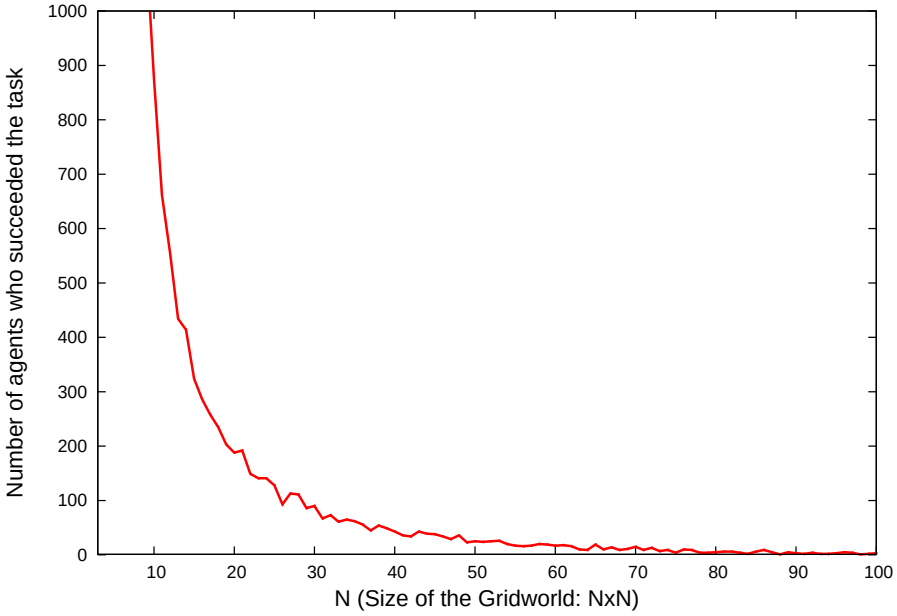

Figure 5: With an enlarged scale of Figure 4.

Even when succeeded in reaching the goal, the route is not the optimal one. In Fig. 6, we show an example of the route of such a successful agent in the grid-world of 50×50, while in Fig. 7 and Fig. 8, we show the examples of the route of unsuccessful agents. The former failed to reach the goal after 500 steps, and the latter died fallen in the cliff.

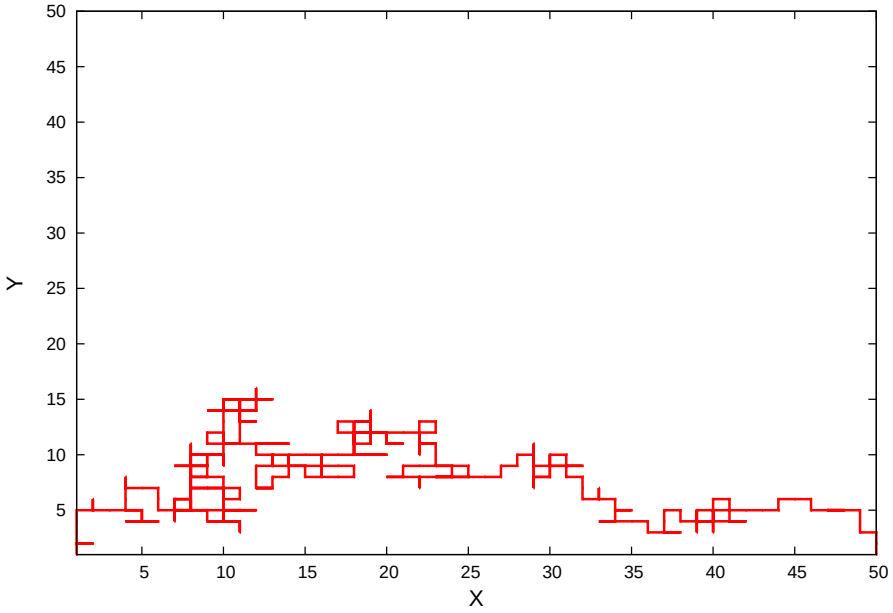

Figure 6: A route an agent reach the goal.

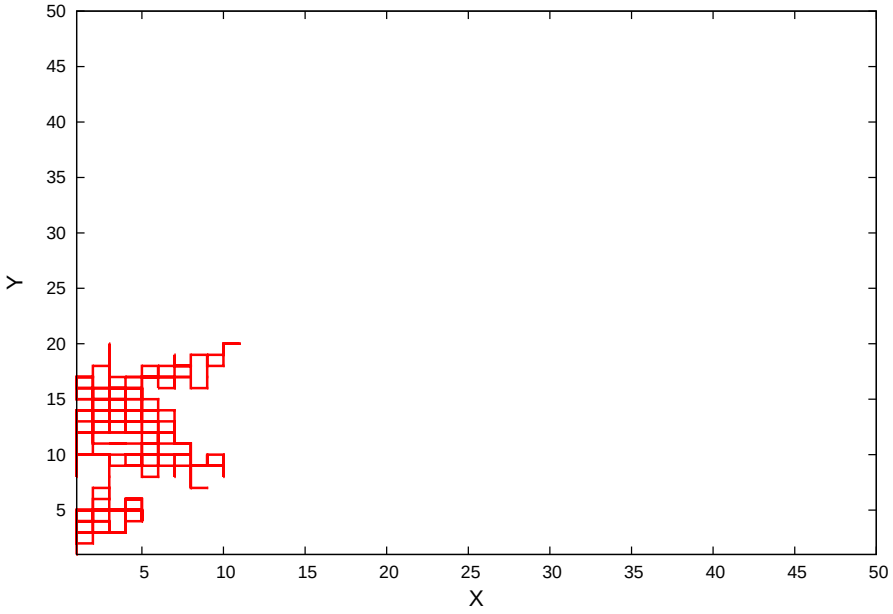

Figure 7: An example in which the agent died after number of limit steps.

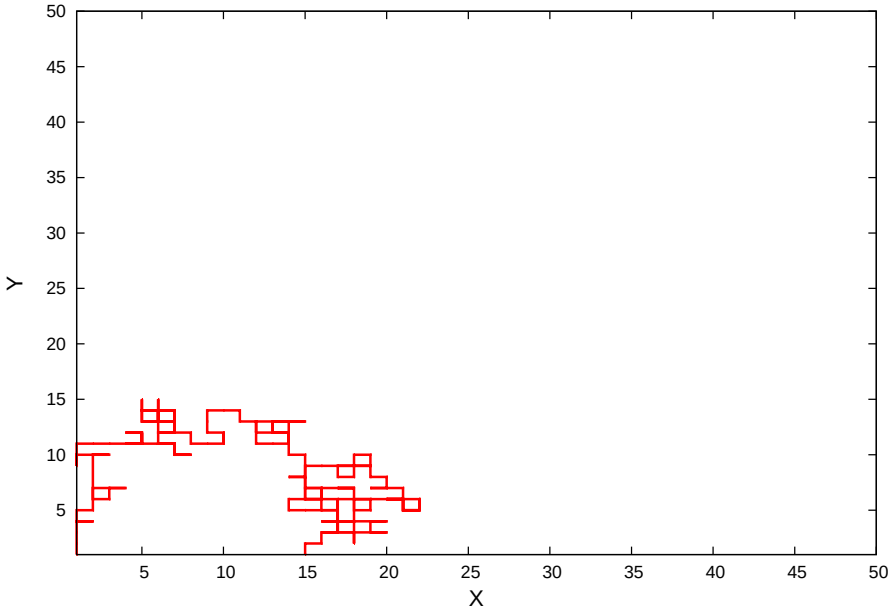

Figure 8: An example in which the agent fell into the cliff and died.

3.2 Genetic Algorithm

First, in order to study how many genes will be appropreate, we run the GA by changing the length of chromosomes from 100 to 500, estimating the minimum number of steps as a function of the length of the chromosome. See Fig 9. We can see that the length from 150 to 200 seems to be enough when our interest is the shortest route.

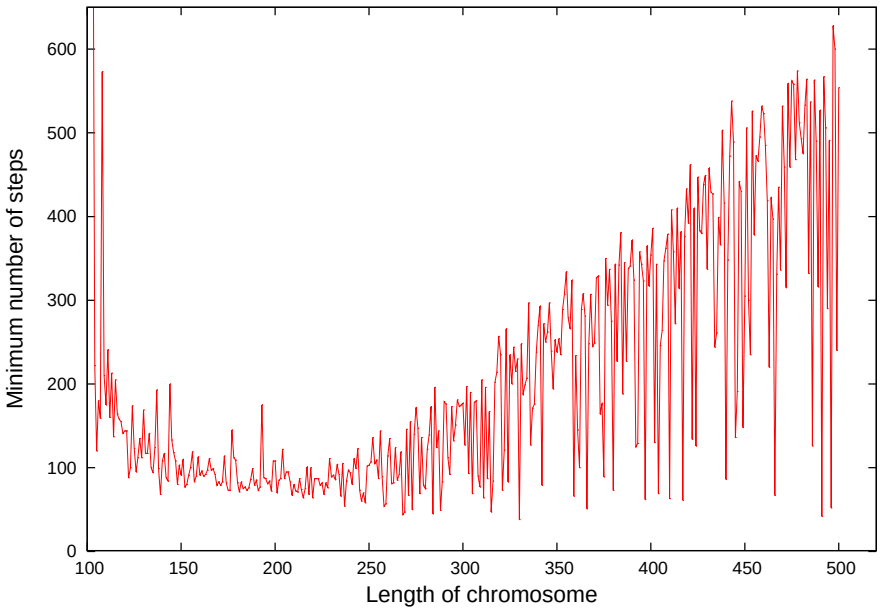

Figure 9: Minimum number of steps to the goal against the length of chromosome.

Then the question is a quality of solution. Let us now compare the obtained shortest route to the goal. The result with 500 genes and 110 genes are shown in Fig 10. and Fig. 12, respectively. The fitness value vs generation of 500 genes and 110 genes study are shown in Fig. 11 and Fig. 13, respectively

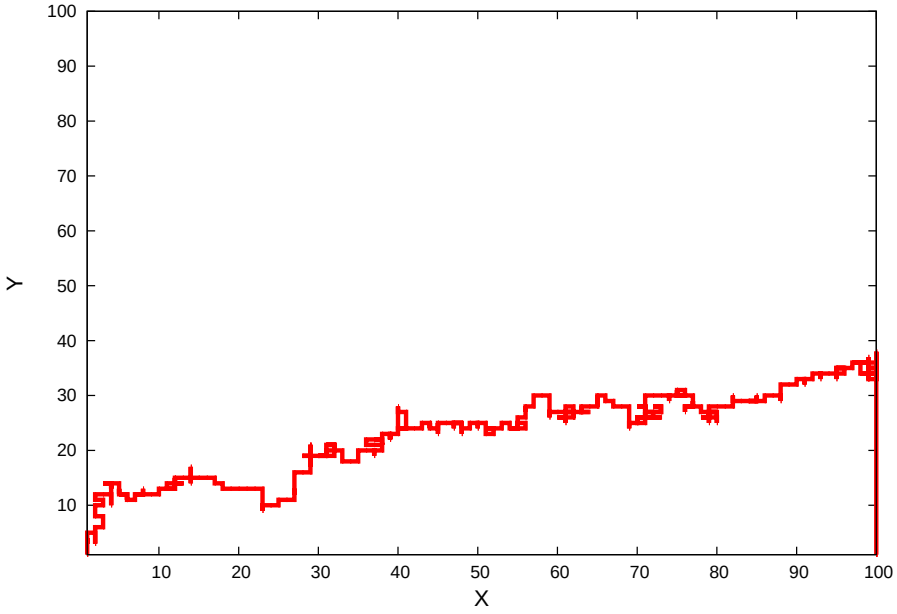

Figure 10: An example of successfull route with the length of chromosome being 500.

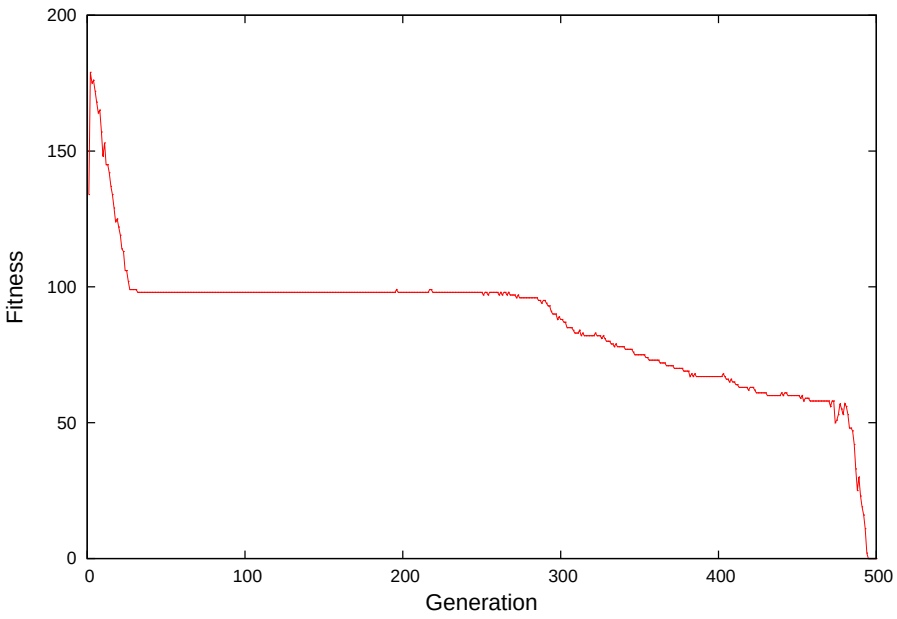

Figure 11: Fitness vs Generation of 500 steps way study.

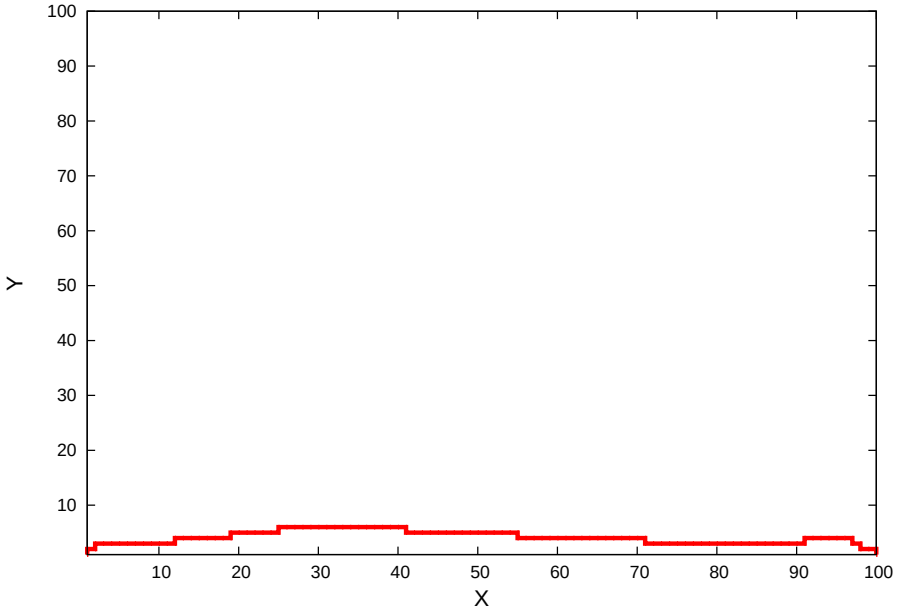

Figure 12: An example of successfull route with the length of chromosome being 110.

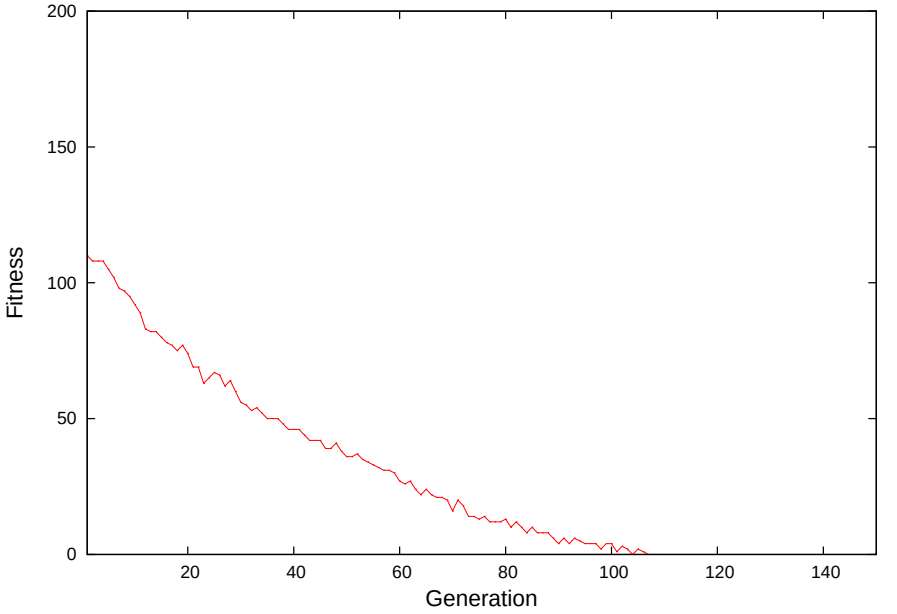

Figure 13: Fitness vs Generation of 110 steps way study.

3.3 Reinforcement Learning

We observed the route of agents who learn how to reach to the goal by RL with Q-learning.

After learning with the ϵ-greedy strategy, agent can reach the goal even in grid-world as large as 100×100. See Fig. 14.

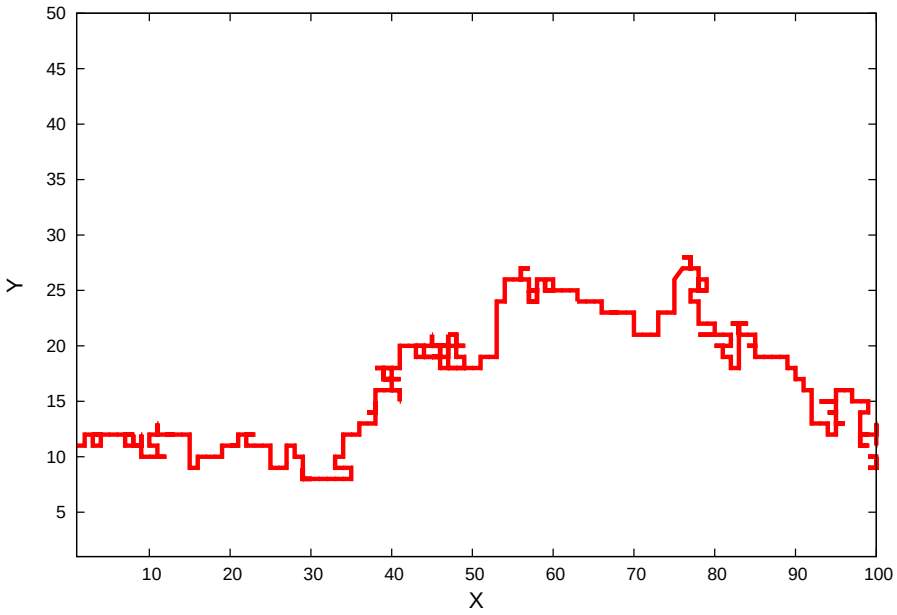

Figure 14: A route of a successful agent in 100×100 grid-world after learning.

3.4 Finite State Machine

We found that experiment using FSM is very difficult. One of the reason is a random generation of FSMs at the beginning. Those random FSMs do not work even in a little clever manner, but equally in a foolishway. So, GA cannot select seemingly a little better parents than others, which is necessary to evolve. We have not been solved yet in this work.

4 CONCLUSION

We have compared agent's performances whose behavior is determined by (i) Random Walk, (ii) Genetic Algorithm, (iii) Reinforcement Learning and (iv) Finite State Machine when the agent explores in a grid-world with cliff. Naturally, we found random walk was not good way to find the goal. GA and RL show good results. FSM has been very hard and is not ready to show a result yet.

In our experiments, RL works better than GA. Agents learned by RL were more intelligent than those by GA, in a sense that they choose routes based on its knowledge, while GA results in only a route of solution. It doesn't know anything about gridworld. On the other hand, GA converges more quickly because it needs a less iterations to learn. If we need the shortest route GA is better than RL.

5 ACKNOWLEDGEMENT

This work is a continuation of our semester project in 2009 in the Brest State Technical University (Imada 2009). I would like to thank our professor Akira Imada for giving us a stimulating topic as our semester project, indicating us how to create a creative scientific article, encouraging me to write this techinical report, and then proof-reading this manuscript.

6 REFERENCE

Holland, J. H. (1975) "Adaptation in Natural and Artificial Systems." University of Michigan Press, Ann Arbor.

Imada, A. (2009) "Exploration in a gridworld which has a cliff." (2009) Semester Project guideline, Brest State Technical University. http://neuro.bstu.by/ai/cliff-walking.pdf

Jefferson D., R. Collins, C Cooper, M Dyer, M Flowers, R Korf, C Taylor, and Alan Wang (1992) " The Genesys System: Evolution as a Theme in Artificial Life." Proceedings of Artificial Life II.

Seijen, van H, H. van Hasselt, S. Whiteson, and M. Wiering (2009) "A Theoretical and Empirical Analysis of Expected Sarsa." Proceedings of the Symposium on Adaptive Dynamic Programming and Reinforcement Learning, pp. 177-184.

Sutton, R. S and A. G. Barto (2005) "Reinforcement Learning: An Introduction." A Bradford Book. The MIT Press. pp. 185-186