Can learning robot solve a 2-D jeep problem?

Akira Imada

Brest State Technical University
Moskowskaja 267, Brest 224017 Republic of Belarus
akira@bstu.by

Summary. Although the topic of the *"robot navigation by learning"* has quite a long history, we still have many problems which are currently unsolvable. In this paper, we propose, as such a so-far-unsolvable problem, a benchmark for robot navigation in a two dimensional grid-world. A robot starting from somewhere in the grid should look for the goal of which the robot has no information about where. The grid-world is totally simple – no corridor, wall or obstacle. The constraint that makes it unsolvable is an energy the robot needs to move the grid with. The goal is put far away from the starting point such that the robot must refill fuels to reach the goal. The task is so-called a *jeep-problem*. We modified the original one-dimensional version of the problem for a robot navigation. A robot navigates in a desert with a jeep which can carry a limited amount of fuel, starting from its base where the jeep can return later to refill the fuel. The jeep has a container to put some of its fuels somewhere in the desert to use in future. The task is to find the goal by repeating the procedure: (i) start the base; (ii) navigate the desert; (iii) put fuels somewhere or find the fuels to get; and (iv) return to the base. This is extremely difficult and most robots less likely to survive in the desert.

Keywords: learning navigation robot, jeep problem, a needle in a haystack.

1 Introduction

Sometimes, we observe a random behavior of a computer system results in a similar to, or even better than, the one by a learned or intelligently designed system. Can we say, "No," when we talk about this issue concerning a robot navigation? That is to say, "Can a learning scheme eventually give a robot better skills than a random one?" This is the topic of this paper. We propose, for the purpose, a benchmark for a navigation robot which is supposed to elaborate its behavior by learning through a number of trials.

The benchmark we are proposing here is simple enough. It would be challenged even by random trials-and-errors, on condition that the world is small enough.

Then questions are, (i) "What if the world becomes realistically large?" and (ii) "Can a learning scheme make the performance more efficient than those by random behavior via a series of experiences of multiple trials later?"

These problems are studied in simulation. A navigation robot explores a fictitious $N \times N$ grid-world.

As a preliminary experiment let's try a kind of two-dimensional version of *a-needle-in-a-haystack problem.* The task is to look for a needle, or equivalently, uniquely pre-determined location in the grid. The robot has no information of where the needle is. The navigation is, at the first trial, by random walk. The robot is expected to eventually find the location, unless N is very large. Question is whether the robot can minimize the path length by a learning algorithm later?

Then the benchmark we want to challenge in this paper. The task is similar to the above mentioned preliminary one, but the robot needs a fuel to move and the robot cannot carry enough fuels to move the grid forever? It is easy to guess the task is much more demanding. This task is an extension of a mathematical puzzle called a *jeep-problem* in which a jeep explores one-dimensional desert under a constraint. A robot here navigates a two-dimensional grid-world instead of one-dimensional desert. While in the preliminary task we do not assume energy consumption of the robot, the jeep here needs fuels to move from one location to the next.

We now take a look at the problems more in detail.

2 To Challenge

2.1 A preliminary experiment

We assume here $N \times N$ grid as our world. Location in the lattice is given by an integer coordinate (i, j) where $i, j = 1, 2, \cdots, N$. The entrance of the world is (1,1) and the exit is (N, N), for example. Or, more generally, two points as start and goal somewhere deep inside. The task of the robot is to look for the exit starting at the entrance.

Let a robot try this task by random walk. The expectation of number of steps of a robot to reach the exit is $O(N^2)$. The result of our experiment to confirm this is shown in Fig. 1 by the average number of steps in 100 runs plotted as a function of N^2.

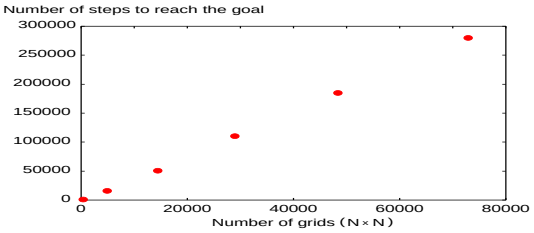

Fig. 1. Average number of steps during 100 runs until a robot who explores the $N \times N$ grid starting from $(1, 1)$ by a random walk eventually reaches (N, N) as a function of number of cells in the grid N^2.

This might be called *a two-dimensional version of a-needle-in-a-hay-stack* problem.[1]

We try this experiment by increasing the grid size. As size grows the task becomes difficult. The experiment is still on going, but according to our so-far observation of 1000 different runs, when grid size is 17000 $\times$ 17000, the minimum steps required was 25,987,691, and the robot reached the needle only in 119 out of 1,000 runs. The number of steps is starting to explode.

The question then will be, "A learning can enhance the efficiency?" In other words, "If a robot try it multiple times under a learning scheme, then the number of steps of the robot to reach the exit becomes shorter than the previous trials, or hopefully minimized?"

In Fig. 2 we show the experiment when the grid size is 96 $\times$ 96. First, by a random walk. Starting from (24,24) a robot walks aiming the goal at (72,72) of which the robot had no *a-priori* information.[2] We observed a 100 such runs and in the left of the Fig. 2 we show the minimum path out of those 100 different walks.

Then we tried a evolutionary learning, just as an example among many others, in which a possible trace of robot is expressed by a chromosomes whose gene is either 0, 1, 2, or 3, meaning to move one cell to the north, south, east or west, respectively. The length of one chromosome is N^2. Following this chromosome from one gene to the next, the robot moves from one cell to the next. After each movement it is checked if the current location is the goal or

[1] The problem in general from a computational context was firstly described in 1987 by Hinton & Nowlan [1] as *a needle being a unique configuration of 20-bit binary string while all other configurations being a haystack.*

[2] The reason for the start and goal are far inside the grid is, otherwise robot found a more clever warp utilizing our toroidal character of the grid. Namely, robot could reach from (1,1) to (N, N) just with 1 step at the minimum.

not. If it is the goal the walk is completed, otherwise to the next cell. The longest possible path length would be thus N^2 but most likely much less than that. The fitness is the number of steps to the goal. If the robot did not reach the goal after following all the N^2 genes, the walk failed. The selection is by *fitness proportionate*, reproduction is by *one-point crossover* and mutation is by replacement with other gene chosen at random with the probability of $1/N$. A result of trace after the evolution is shown in the right of the Fig. 2, and the fitness evolution is shown in Fig. 3.

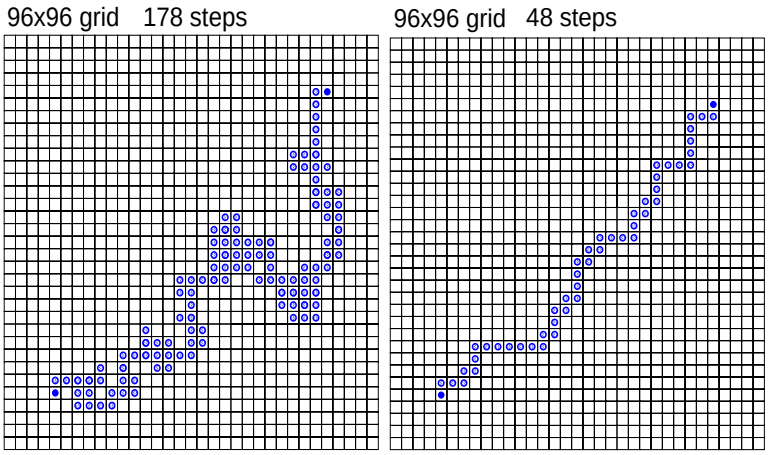

Fig. 2. In the grid-world of 96 starting from (24,24) a robot walks aiming the goal at (72,72) of which the robot had no *a-priori* information. Left: The path of minimum length among 100 trials by random walk. Right: Minimal path the robot found after an evolutionary learning as shown in Fig. 3. (Marginal area is omitted.)

2.2 Can a learning robot survive in a desert with a jeep?

Next, let us imagine that we are in a base located at the center of a desert. Again our world is a two-dimensional grid. This time, for some reasons which later become clear, size is 77×77, the coordinate of the bottom-left corner is $(-38, -38)$, and the top-right corner is $(38, 38)$ which is the only exit of the desert. The base is located at the origin $(0, 0)$. The grid is toroidal, that is, if the coordinate becomes $(N + 1)$ and $-(N + 1)$ then it is replaced with $-N$ and N, respectively.

A robot leaves the base with a jeep. The jeep moves the desert of grid from one cell to the next, each time by consuming one unit of fuel. The jeep has a tank for fuel whose capacity is 30 units. The jeep also has a container with which

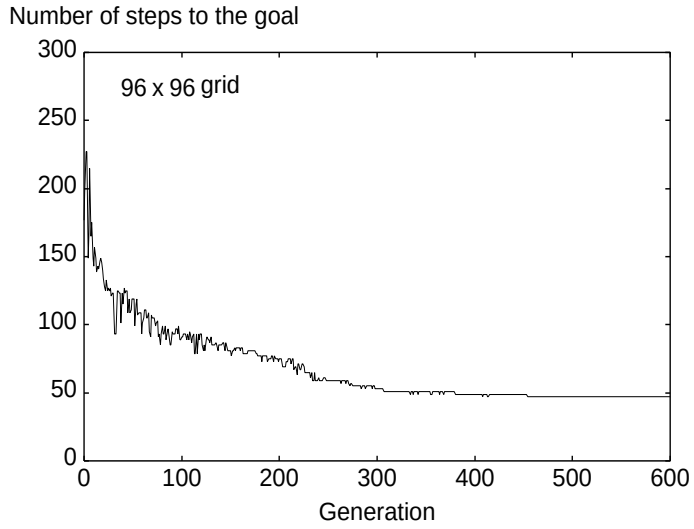

Fig. 3. An evolution of the number of steps to the goal starting with a population of random walks. We can see the convergence to the global minimum of 48 Manhattan distance.

the robot can store some amount of fuel in the tank to put at any location in the desert for the next time usage. Since the exit is 76 *Manhattan-distance* apart from the base, the tank full of 30 units are not enough to reach the exit. The robot is allowed to go back to the base twice to refill the tank.

This is an extension of so-called a *jeep problem* where a jeep should maximize its penetration to one-dimensional desert under a constraint. See, for example, the WWW page of *Wolfram MathWorld*.[3] It reads:

> *"Maximize the distance a Jeep can penetrate into the (one-dimensional) desert using a given quantity of fuel. The Jeep is allowed to go forward, unload some fuel, and then return to its base using the fuel remaining in its tank. At its base, it may refuel and set out again. When it reaches fuel it has previously stored, it may then use it to partially fill its tank. This problem is also called the exploration problem (Ball and Coxeter 1987)."*

As far as we know, this has never extended to a two-dimensional world. We now paraphrase the already known solution of the original version found in the page.

1) Start with 30 units of fuel.

[3] http://mathworld.wolfram.com/JeepProblem.html.

2) Go forward 10 distances, put 10 units, and then go back to the base with the remaining 10 units of fuels, and refill 30 units again.
3) Go forward 10 with 30 units refilled, spending 10 units, and get 10 units there.
4) Go forward 6 further, spending 6 units and put there 8 units.
5) Go back to the base spending remaining 16 units
6) With 30 units again, go forward 16 , spending 16 units, and get 8.
7) Go forward further until spending all the remaining fuel, and eventually reach the point which is 38 apart from the base.

This is how the jeep can penetrate to the desert with the maximum distance when allowed to go back to the base twice. You now notice the reason why those parameters in our two-dimensional version described here are highly artificially devised. It is to fit the problem.

We now summarize the problem.

Challenge (Jeep's survival in a desert) *Assume* 77×77 *toroidal lattice each of whose cells is expressed by (i,j) where* $i, j = -38, -37 \cdots, 0, 1, 2, \cdots, 37, 38$. *We call this grid a desert. The desert has only one exit at (38,38). Starting from (0,0) a robot navigates a jeep from one cell to the next. In order for the jeep to move one cell, it needs to spend one unit of fuel, and the jeep has the tank whose capacity is 30 units. The jeep also has a container with which the robot put some amount of fuel to any location of the desert for the next time usage. Allowing to go back to the base twice, can the robot learn how to reach the exit through a multiple times of experiences of failure?*

Or, we might be modified it like the original one, as follows. *The size is more generally* $N \times N$ *for a large enough* N. *Starting from also the base at (0,0) and being allowed to go back to the base* R *times to refill the fuels, the robot should penetrate the maximum distance from the base instead of aiming the exit of the desert.*

3 Possible Approaches

Though, at this moment of writing this article, none of our experiments below or other attempts has not given us a satisfactory result, we show the followings just as strategies to start with, if any.

3.1 Evolutionary learning

Probably the simplest approach will be an evolutionary learning. Chromosomes each of whose gene is a pair of integer (i, x). The first integer i is either 1, 2, 3, 4, 5, or 6 which indicates robot to move north, south, east, west, put

fuels, or get fuels, respectively. If i is either 1, 2, 3, or 4, then x means how many cells should robot move consecutively, and if i is 5 or 6, then x, means how much fuels should robot put or get. In reality, however, evolution of a population of such random chromosomes toward a better performance is much more difficult than a search for a needle in a haystack.

3.2 Learning Navigation with Memory

In this situation, it would be natural to employ a *memory* for an effective action of robot. We have had lots of such reports in various types of environment. See, for example, Remazeilles & Chaumette (2007) [2]. Most closely related, among others, is the work by Srinivasan (2006) [3] in which the author wrote, *"The agent will thus be able to remember a history of previously encountered states, and instead of taking decisions based on single states, it will consider a history of states as a whole. A history can hence be defined as a sequence of the last n observations and the agent's memory becomes a collection of all such histories that have been encountered in the past."*

3.3 Enforcement Learning

Yet another, and probably the most possible one, is by exploiting standard reinforcement learning, like SARSA [4] and Q-learning [5]. Unfortunately, however, we still don't know how we design Q value or reinforcement function, whichever with memoryless or memory-based policy. It would not be difficult to minimize the total steps that the jeep takes to reach the goal, or maximize the penetration into the desert, but when, where and how much fuel should be put would not be rewarded/penalized in a simple way.

3.4 Quantum Robot approach

Since Grover's [6] assertion in 1997 that quantum mechanics helps in searching for a needle in a haystack with $O(\sqrt{N})$ steps while classical computer requires $O(N)$ steps[4], a fair amount of approaches exploiting a quantum random search has been proposed. See, for example, Shenvi (2003) et al. [7]. As for searching a space by a mobile robot, Beninof (2002) [8] proposed a *quantum robot*. It might be interesting to see what Beninof wrote: *"For this initial memory state all 2^N searches are carried out coherently. Since the path lengths range from 0 to 2^N, the quantum robot can search all sites of R and return to the origin in $O(N \log N)$ steps. Since this is less than the number of steps, $O(N^2 \log N)$, required by a classical robot, the question arises if Grover's algorithm can be used to process the final memory state to determine the location of S. If this is possible, the overall search and processing should require $O(N \log N)$ steps which is less than that required by a classical robot."*

[4] As N is the number of points in search space in his equation, it is $O(N^2)$ in our context.

4 Summary

Many problems are still unsolvable not due to our current computer resources such as speed and memory but due to difficulty concerning how we find the algorithm to solve it. We believe the problem we proposed in this paper is one of such problems.

Knaden et al. wrote, *"Desert ants use path integration as their predominant system of long-distance navigation"*, one of what we have in mind is to apply a nature's wisdom to this problem. The authors went on to write according to their observation of real ants in desert in Tunisia, *"Ants had reset their home vector to zero state, and had therefore been able to reload their learned feeder vector, and consequently departed from the nest in the feeder direction."*

A search by quantum random walk has already been mathematically proofed to be more efficient than the one with our currently available computers. Most attention so far has been on general search on hypercube. We think our benchmark proposed in this paper is a good one also as a test for the quantum computation, because we could design a *wave-function* quite easily which includes an information on both geometry and the location where reserved fuels are. This is our motivation of this study too.

References

1. Hinton, G. E., and S. J. Nowlan (1987) "How Learning can guide evolution?" Complex Systems, Vol. 1, pp. 495–502.
2. Remazeilles, A., and F. Chaumette (2007) "Image-based robot navigation from an image memory." Robotics and Autonomous Systems archive, Vol. 55(4), pp. 345-356.
3. Srinivasan, B. (2006) "Analysis of memory-based learning schemes for robot navigation in discrete grid-worlds with partial observability." Available at author's WWW page but the author says this is just "Other paper." `(http://www.stanford.edu/~bharsrin/docs/papers/)`
4. Sutton, R. S. (1996) "Generalization in reinforcement learning: Successful examples using sparse coarse coding." Advances in Neural Information Processing Systems 8, MIT Press. pp. 1038-1044.
5. Watkins, C. J. C. H. (1989) "Learning from delayed rewards." PhD thesis, University of Cambridge.
6. Grover, L. (1997) "Quantum mechanics helps in searching for a needle in a haystack." Physical Review Letter, Vol. 79, pp. 325–328.
7. Shenvi, N., J. Kempe, and B. Whaley (2003) "A Quantum random walk search algorithm." Physical Review A, Vol. 67, pp. 052307–052318.
8. Benioff, P. (2002) "Space searches with a quantum robot" Contemporary Mathematics, Vol. 305, pp. 1–12.
9. Knaden, M., and R. Wehner (2006) "Ant navigation: resetting the path integrator." Journal of Experimental Biology Vol. 209, pp. 26–31.