Can learning robot solve 2-D jeep problem?

Akira Imada

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

Summary. Although *robot navigation through learning* has quite a long history, we still have many problems which are very difficult to be solved, if not at all. In this paper, we propose, as a challenge, two such benchmarks in a simulated world of $N \times N$ two-dimensional lattice inside the computer. The first task is looking for a specific only one location which the robot has no *a priori* information of where. Question is a learning mechanism reduce the number of steps until it reaches the location, while random walk requires $O(N^2)$ steps. The other task is a 2-D extention of a so-called *Jeep Problem.* A robot navigate a jeep which has a limited size of fuel tank trying to put some of the fuel in some location for the future usage, then go back to its base to refill the tank, and again start to explore the lattice. The robot repeat the procedure with the goal being explore the world as long as possible, or here specifically reaching the exit with a restricted amount of fuel. The task is originally proposed and extremely difficult to eventually use a know optimal solution.

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

1 Introduction

Sometimes, a random behavior of a computer system is observed to be better than a learned or intelligently designed system. This is the topic of this paper. We propose, as benchmarks, two tasks for navigation robot which is suposed to eraborate its behavior by learning through a number of trials.

Both tasks can be acomplished more or less by random behavior. Question, however, is if a learning scheme can make the performance more efficiently than those by random behavior later through experiences of multiple trials.

Both tasks are not in a real phisical world but simulation inside computers. A navigation robot explores a fictitious world of $N \times N$ grid.

The first task is, to look for an exit which the robot has no information of where. The navigation is, at the first trial, by random walk. Unless N is very large, the robot can eventually find the exit after an average of $O(N^2)$ steps. Question is the robot can minimize the path length requied by random walk by a learning algorithm later?

The other task is more demanding. A robot navigate a 2-D grid again, but this time troidal grid for some reason. While the first task we do not assume energy consumption of the robot, but this time

2 Two Challanges

2.1 Can learning robot search it more efficiently than before?

We assume here $N \times N$ latice grid as our world. Location in the lattice is given 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). Starting at the entrance, most can navigate itself eventually found the exit only by *random walk*, unless the grid is a very large one. The expectation of number of steps of a robot to reach the exit is $O(N)$. The result of our experiment of seeking the average number of steps during 100 trials are plotted in Fig. 1 as a function of N.

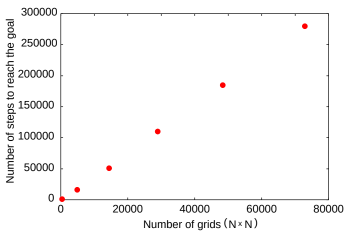

Fig. 1. Average number of steps during 100 runs until a robot who explores the $N \times N$ grid starting from $(0, 0)$ by a random walk eventually reaches (N, N) as a function of Number of number of grids N^2.

This might be called *a 2-dimensional version of a-needle-in-a-hay-stack* problem. [1]

The question then will be, “A learning can enhance the efficiency?” That is to say, “If a robot try it multiple time by a learning the number of steps of

[1] The problem in general from computational context was described in 1987 by Hinton & Nowlan [4] as *a needle being a unique configuration of 20-bit binary string while all other configurations being a haystack.*

the robot to reach the exit becomes shorter than the previous trials, and/or hopefully minimize the path eventually?"

Now let us surmmarize the problem with a little redundancy though.

Challange 1 (2-D Needle in a Haystack Problem) *Assumming $N \times N$ troidal lattice each of which cell is expressed by (i, j) $(i, j = -N, \cdots, 0, 1, 2, \cdots N)$, and the lattice has only one exit at (N, N). Starting from $(0, 0)$ a robot navigate from one grid to the next. Through a multipul traial, can the robot learn how to take a path shorter than before?*

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

Let us now imagin that we are in a base located at the center of a dessert. Yes again our world is 2-D 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 dessert. The base is located at the origin $(0, 0)$. The grid is troidal, that is if the coordinate becomes $(N+1)$ and $-(N+1)$ then it is replaced with $-(N+1)$ and $(N+1)$, respectively.

A robot leave the base with a jeep. The jeep move the grid horizontally or vertically, one grid to the next each time with a consumption of one unit of fuel. The jeep has a tank for fuel whose capacity is 30 units. The jeep also has a container whith which 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 77 Manhattan distance apart from the base and as such the full tank of 30 units is not enough to reach to the exit. The robot allowed to go back to the base twice and full the tank again.

This is an extention of so-called *Jeep Problem* where a jeep should maximize its penetration to 1-D dessert under a condition. See, for example, the web page of *Wolfram MathWorld* (`http://mathworld.wolfram.com/JeepProblem.html`). It reads:

> *"Maximize the distance a Jeep can penetrate into the 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 2-D world, though. As you might notice already, the solution in our context is as follows and you might notice now the reason why those parameters here are highly artificially devised. It is to fit the problem.

1) Start with 30 units of fuel.
2) Go forward 10 distances, put 10 units, and then go back with the remaining 10 units of fuels.
3) With 30 units again, go forward 10 , spending 10 units, and get 10 units there.
4) Go forward 6 further spending 6 units and put 8.
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.

Challange 2 (Jeep's suvival in a desert) *Assume $N \times N$ troidal lattice each of whose cells is expressed by (i, j) $(i, j = -38, -37 \cdots, 0, 1, 2, \cdots, 37, 38)$. We call it a dessert. The dessert has only one exit at $(-38, 38)$. Starting from $(0, 0)$ a robot navigate a jeep from one grid to the next. In order for the jeep to move one cell it needs to spend one unit of fuel, and the jeep has 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 two times to the base, can the robot learn how to reach the exit throug a multiple times of failure experiences?*

3 Possible Approches

Though, at this moment of writing this article none of our experiments below or other attempts of us has not given us a satisfactry result, we show them just as strategeis to start, if any.

3.1 Messy Genetic Algorithm

One of the simplest approaches would be using a genetic algorithm (GA) representing the path of the robot by a what is called a *chromosome*. The problem, however, would be a length of the chromosome. Usually GA evolves a population of fixed length of chromosomes. The path of the robot is different from navigation to navigation. Then one idea is to use a so-called *messy genetic algorithm (mGA)* in which flexible length of chromosomes can be evolved [1].

In mGA each gene is tagged with an additional number which expresses the location of the gene in its chromosome when created at random in the first generation. With the selection operation being similar with the usual GA, *crossover* is done with two operations called a *splice* and *cut*. The *cut* splits a stirng at a position chosen at random. Each of selected two parents are spliced into two parts. The first part of the first parent and the last part of the second parents, or vice versa, are concatenated and thus construct one offspring. So this reproduction works without taking care of parents' and offspring's length.

We now take a brief look at how we can apply mGA to robot navigation. Assuming we have for different gene values, 1, 2, 3, 4 which means move right, left, up and down, respectively. As an example, two parents are

$$((1\ 2)(2\ 2)(3\ 1)(4\ 2)(5\ 2)(6\ 3)(7\ 3)(8\ 3)(9\ 4)$$

and

$$((1\ 3)(2\ 3)(3\ 4)(4\ 4)(5\ 2)(6\ 3)(7\ 3)(8\ 4)(9\ 3).$$

Then an extreme example from those possible offspring would be

$$((1\ 2)(2\ 2)(3\ 1)(4\ 2)(5\ 2)(6\ 3)(7\ 3)(8\ 3)(2\ 3)(3\ 4)(4\ 4)(5\ 2)(6\ 3)(7\ 3)(8\ 4)(9\ 3)$$

and

$$((1\ 3)(9\ 4)).$$

See Fig. 2 below. fitness might be hamming distance of the closest point the path approched to (N, N).

Though reader might think this would be for evolving the length of chromosoms from short ones to long ones, it is opposit actually.

Thus, starting with a population of those chromosomes mGA in the 2nd generation creates chromosomes whose length ragnes from 2 to 98

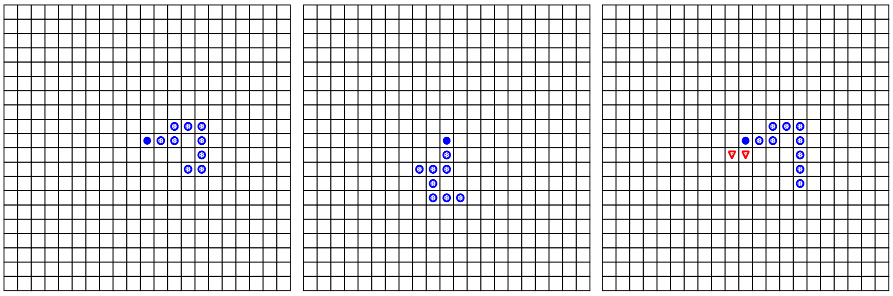

Fig. 2. Please write your figure caption here

3.2 Learning Navigation with Memory

It is natural to employ a memory for an effective navigation of robot, which allow the robot learn from previous experience. Hence we have had lots of such report. See, for example, Remazeilles & Chaumette (2007) [2]. Most closly related is the work by Srinivasan (2006) the title reads ... *Memory-Based Learning Schemes for Robot Navigation in Discrete Grid-Worlds ...*" in which author wrote, *"we attempt to find a solution to the problem of robot* *navigation using different memory-based learning schemes, and compare these schemes in terms of learning speed and efficiency.*

Our grid world, however, has no wall, no obstacles, or no corridor to be followed. It is really simple world. What makes a navigation difficult is no information about the destination. Under this simple situation our concern is to minimize a distance – Manhattan distance. Hence robots should have a memory to store previous paths experienced.

3.3 Quantum Robot approach

Since Grover's [5] 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, a fair amount of approches expliting a quantum random search has been proposed. See, for example, Shenvi (2003) et al. [6]. As for searching a space by a mobile robot, Beninof (2002) proposed a *quantum robot* [7]. It might be interesting to seewhat 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 Discussion

In this specific example of location, question is "Can a learning give a robot an information that to move left and to move down is not good strategies, or not.

References

1. Goldberg,D. E., B. Korb, and K. Deb (1989) "Messy genetic algorithms: motivation, analysis, and first results." Complex Systems Vol.3, pp. 493–530.
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. B. Srinivasan (2006) "Analysis of Memory-Based Learning Schemes for Robot Navigation in Discrete Grid-Worlds with Partial Observability." Available at authou's WWW page with a description just "Other paper." `(http://www.stanford.edu/~bharsrin/docs/papers/)`

4. G. E. Hinton, and S. J. Nowlan (1987) “How Learning Can Guide Evolution.” Complex Systems, Vol. 1, pp. 495–502.
5. L. Grover (1997) “Quantum mechanics helps in searching for a needle in a haystack.” Physical Review Letter, Vol. 79, pp. 325–328, 1997.
6. N. Shenvi, J. Kempe, and B. Whaley (2003) ”A Quantum Random Walk Search Algorithm.” Physical Review A, Vol 67
7. P. Benioff (2002) ”Space searches with a quantum robot” In Quantum Computation and Information
8. Kipphan H (2000) Handbook of printmedia. Springer, Berlin Heidelberg New York
9. Brandt J, Hein W (2001) Polymer materials in joint surgery. In: Grellmann W, Seidler S (eds) Deformation and fracture behavior of polymers. Engineering materials. Springer, Berlin Heidelberg New York
10. Che M, Grellmann W, Seidler S (1997) Appl Polym Sci 64:1079–1090
11. Ross DW (1977) Lysosomes and storage diseases. MA Thesis, Columbia University, New York