Computer Architecture Lab
Assignment 0 - Report

Chidaksh Ravuru (200010046), Pavan Kumar V Patil (200030041)

August 9, 2022

1 Problem Statement

Consider the scenario where one country, called the defending country (DC), wishes to defend its border against another country, called the attacking country (AC), whose aim is to send an infiltrator to cross the border and enter DC's land. DC decides to deploy a wireless sensor network along the border. If a sensor detects an infiltration attempt, DC can then send its troops to counter the infiltration. Task is to simulate the described border crossing scenario in software. Define classes for each element: Border (you can consider the length of the border to be some large value, say 1000), Sensor and Infiltrator. Also, define a class Clock that captures the time in our simulated world.

2 Assumptions

2.1 Assumption on Length:

In accordance with the Problem Statement, the length has to be infinitely long. So, we have chosen Length (L) as 1000 as given in the Problem Statement.

$$L = 1000$$

2.2 Assumption on Initial position of Infiltrator:

We assumed that the Infiltrator starts at point $x = -1$ and $y = Length/2$. This says that he is outside the border but at the center of the Length of the border.

2.3 Assumption on movement of Infiltrator:

According to the Problem Statement infiltrator can move in any one of the neighbouring nine directions. But we assumed that the infiltrator only moves ahead from AC to DC, without taking any step backwards. Because the main goal of the infiltrator is crossing the boundary and entering DC so it is just waste of time taking a step backwards.

3 Parameters

3.1 Probability p:

We varied probability between 0.001 to 0.9. We didn't take probability as 0 or 1 (the extreme cases) because when probability is 0 infiltrator will reach the DC without any obstruction and when p is 1 it is impossible to reach the DC. So we didn't take the extreme cases while generating the plots. We took 20 values equally spaced between 0.001 to 0.9 to generate the plot.

3.2 Width w:

We varied width between 1 to 100. We took 100 values equally spaced between 1 to 100 for the width.

4 Approach

Given the above Parameters p and w. We made a mesh gird of the values of p and w, where we took every value of p with every value of w and calculated average rum time over 5 iterations. We took the results into a file and then plotted a 3D line plot between average time taken, probability of sensor being on and width of the boundary.

5 Code Workspace

We created a clean code workspace for the project. We have created 5 main files one for each task of the Software simulator. The 5 files are:

- sensor.java
- border.java
- infiltrator.java
- clock.java
- Main.java

In the folder we have **input.txt** which contains all the mesh grid values of p and w. **output.txt** contains the time taken to run for each combination of p and w. **generate_input.py** generates all the inputs required to draw the graph. **graph.py** does all the plotting work and displays the plot. We have a bash file **run.sh** to help client to run the software easily.

6 How to run the code

To reduce the work of the client using this software we made a bash file run.sh which when ran produces output. The bash command is:

>>> bash run.sh

7 Results

7.1 Effect of width on time taken to come out

As the parameter 'w' (width) was increased, the time taken by the infiltrator to enter DC also increases but time taken increases linearly with width as

$$E[time\_taken] \propto w$$

7.2 Effect of probability on time taken to come out

As the parameter 'p' (probability) was increased, the time taken by the infiltrator to enter DC also increases but increases exponentially with the probability.

7.3 Graphs

- X-axis = width of the border.
- Y-axis = Probability of sensor being on.
- Z-axis = Time taken to cross the border.

Figure 1: Variation of Probability, Width and Average Time