Computer Architecture Lab Report

Shubh Agarwal(210020047) && Saksham Chhimwal(210010046)
Github Repo Link:ShubhAgarwal-dev/ComputerArchitecture (private)

Documentation

We have prepared five classes by the name:

- **Border**: Represents a border with a grid of sensors. Each sensor can have a state of either "true" or "false," representing its activated or deactivated state. This class aims to create a border with sensors and manage their states.
- **Clock**: This represents a basic timekeeping mechanism. The class encapsulates the concept of time and provides methods to manipulate and retrieve the time value.
- **Infiltrator**: Represents an entity capable of moving through a simulated environment with a priority-based movement policy. A set of rules determines the infiltrator's movement based on its current position and the state of the environment's sensors.
- **Main**: Conducts simulations of an infiltrator navigating through a simulated border environment with sensors. It generates and writes data about the infiltrator's movements and the time it takes to reach the border or get caught by the sensors.
- **PqCompares**: Implements the Comparator interface. The purpose of this **comparator** is to establish a priority order for elements of type **Integer[]** based on their first element values. This custom comparator can be helpful when working with priority queues or sorting collections.
- **Sensor**: This represents a simple sensor entity with different constructors to set its state based on different conditions. The class provides methods to query and modify the state of the sensor.

`Input (Command Line Parameters)`

`Supply the Java class file with these parameters while executing using the command line`

```
javac main <borderLength> <borderWidth> <probability>
```

The default value of these parameters are

- **borderLength**: 1000 block
- **borderWidth**: 5 blocks
- **probability(<1)**: 0.5

```
Output File(`borderLength_borderWidth_probability.txt`)
```

The first three lines contain these parameters, and then lines 4 to 503 will have the time taken by the agent to reach the border.

Our agent is following the policy, which enables the agent to cross the border with certainty.

Agent Movement Policy

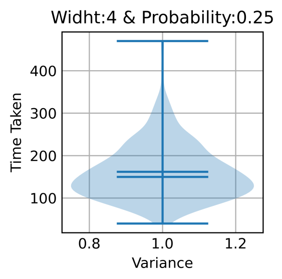
The agent has complete knowledge of all the surrounding sensors:

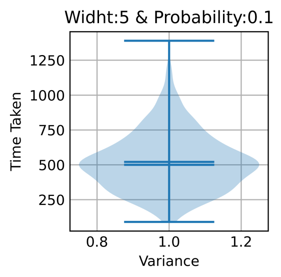
- If the sensor on which agent is ON, then it will be stationary for about 10 secs.
- Else, the agent will look for another cell to move, prioritizing the OFF cells in front and never taking any cell behind it.

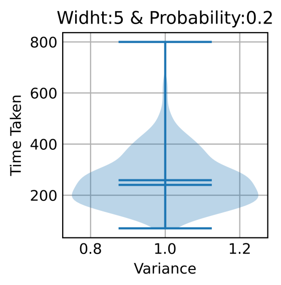
This way, the time taken by the might be long,, but it always crosses the border.

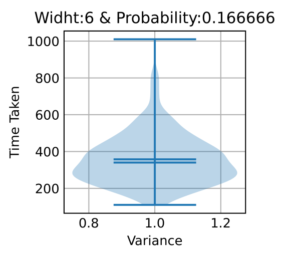
Observation

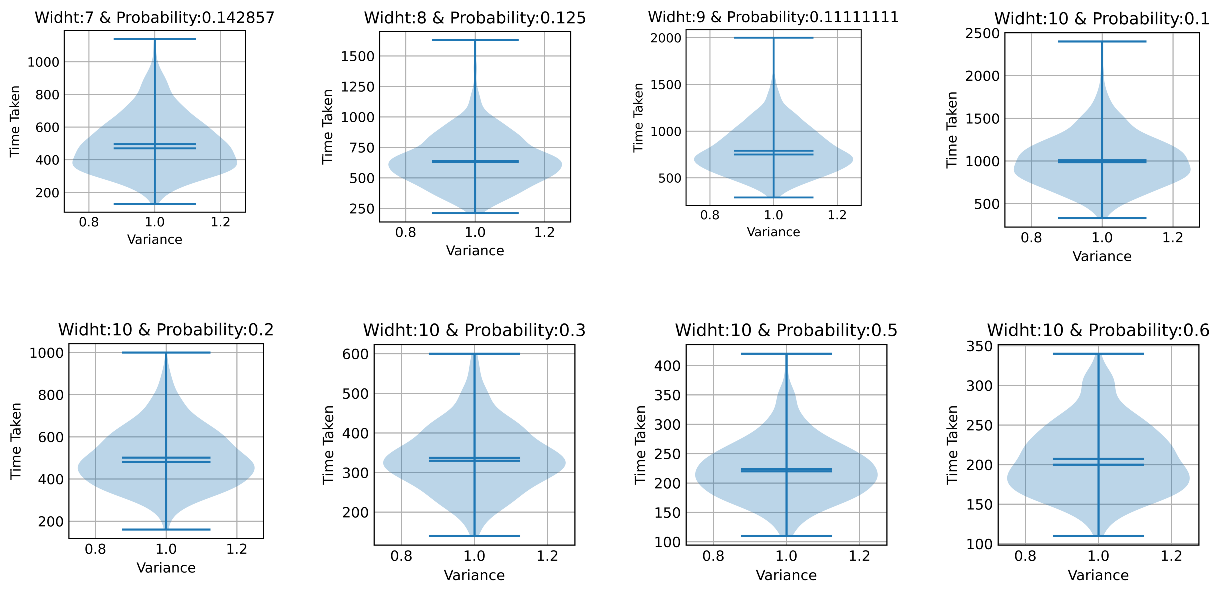
- We ran tests with different widths and probabilities and recorded observations for many runs, shown below.







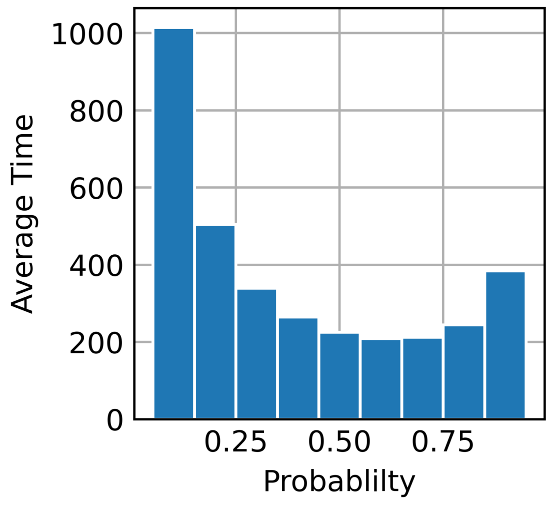




These are the violin plots, and these clearly reflect as the width increases, the time taken to cross the border also increases. The lines inside the plots reflect the mean of the time taken to cross the border over all the runs we did.

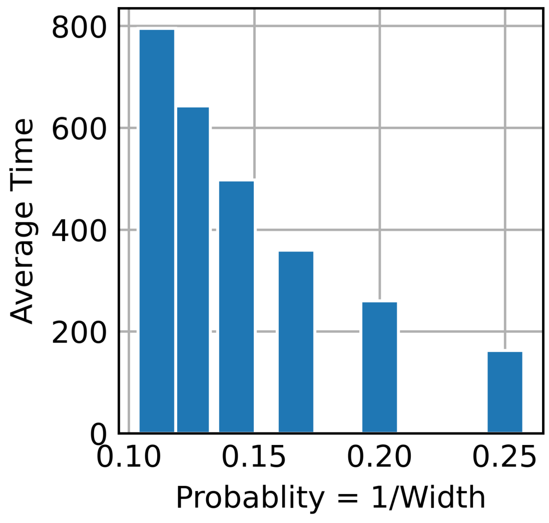
- Next, we kept the border length constant, varied the probability, and calculated the mean of the timings of each run.

  The width in these runs is 10, and the probability varies from 0.1 to 0.9.



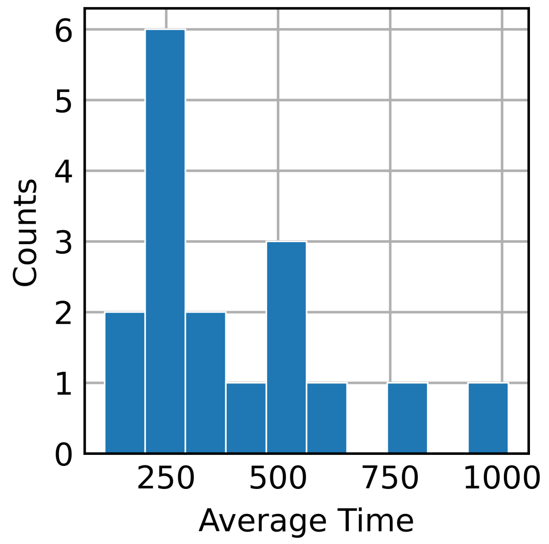
The average time to cross the border reduces as the probability increases. But sometimes, even at high probability, the infiltrator takes more time in some test cases, because of which the average time shoots up.

- After this, we had a plot in which we kept a relationship between the width and probability, where probability varies as the inverse of the width.



Here we can see that the average time decreases as the probability increases, and the plot looks like some exponential decay.

- Finally, we plotted the histogram of average time and how many runs touch that average time over all the variations of width and probability. The plot is shown below.



Here we can see that the most frequent time over all the variations is somewhere between 250 seconds and 500 seconds.