Technical University of Denmark

Google Loon Project

B.Sc.Thesis, June 10^{th} 2016

Kári Þrastarson, s131896

Supervisor:

Jan Madsen
Professor in the Department Mathematics and Computer Science
DTU

List of Tables

List of Figures

1 Introduction

2 The loon project

3 The Model

3.1 Introduction

The loon project aims to launch thousands of balloons, each carrying expensive equipment and weighing little over 200 kilograms[1]. It would be foolish to attempt such a thing before making sure that the end goal is plausible, if not definitely possible. This can be done relatively cheaply using computer simulations, which is exactly the aim of this project. Even though the world has been simplified significantly and many laws of physics overlooked, the simulation should give a good idea as to whether this idea is good or poor. Arguments are provided for all assumptions made during the modelling and their implications.

3.2 Description

The model focuses on the balloons and their movement around the stratosphere. The key components in the model are the following:

The balloons: The balloon is travels around the stratosphere and supplies Internet connection to the point on the grid it corresponds to.

The "grid": The earth in this model is represented by a two dimensional array, or a grid, of integers. The integers represent the number of balloons

[1] FACT CHECK

currently serving this particular space on earth. If the number is zero this is point is construed as not connected. The goal is to make all value of the grid larger than zero.

The stratosphere: The stratosphere is a collection of stratospheric wind layers which also are represented by a grid. The size of each wind layer equals the size of the earth grid. The layer object holds a collection of vectors which carry the wind strength in directions x and y. The stratosphere is organized so each wind layer occupies a certain altitude. That way each balloon is affected by only one wind layer at a time, determined by its current altitude.

The model has some modifiable key parameters which are explained in chapter 3.4. The model creates and initializes the ecosystem, which consists of the wind layers, balloons and the grid representing the earth. The simulation then moves each balloon according to its corresponding wind layer. The decision a balloon has to make is whether it should start moving upwards, downwards or stay put. This decision is the core of the model and will be developed in different ways and compared.

3.3 Data

The only data necessary for this simulation is stratospheric weather data. The blabla

3.4 Simulation and Parameters

The model is the structure and relationship between all objects. It is modifiable in some ways to be able to measure and compare performance of different algorithms. The parameters that should be adjusted before simulation are the following:

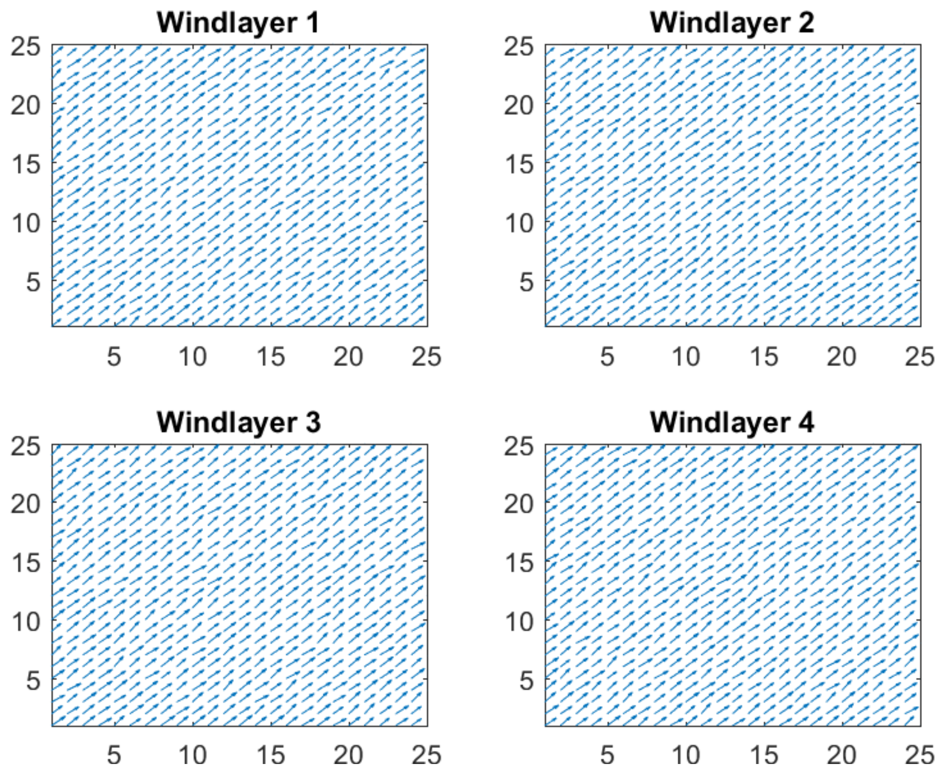

Figure 1: Wind Layers

WORLD_SIZE: The integer dimension of the grid that represents the earth. This number greatly affects the run time of the model as well as accuracy.

NUMBER_OF_BALLOONS: The number of balloons in the model. This variable is usually set to the number of cells in the earth grid, i.e WORLD_SIZE^2.This means that best case scenario the coverage is 100% and all the balloons are perfectly spread.

VERTICAL_SPEED: The speed of the balloon in directions up and down. This number will be added/subtracted from the balloon's altitude each step, until its motion in that direction is stopped.

NUMBER_OF_STEPS: How many steps the simulation should run. This number should be as big as possible to properly model the behaviour of the ecosystem.

NUMBER_OF_CURRENTS: The number of wind layers, or wind currents, in the system. The more layers there is to choose from the more accurate direction the balloon should be able to choose from.

MIN/MAX_ALTITUDE: Numbers used to divide the layers. The total distance between MIN and MAX is divided equally between all the layers in the model. This is used to determine whether a balloon has reached a new wind layer. The units are abstract but should be taken into consideration when choosing vertical speed. The vertical speed corresponds to the units in this MIN/MAX scale.

The model and simulation go hand in hand. When all objects have been created and configured, the simulation is run.

The simulation moves all balloons according to the established stratosphere. Each step of the simulation the following process is as follows:

1. Apply decisions: All balloons decide, using the control algorithm, whether they should start moving up or down or stay in the current altitude.

2. Apply currents: When the decision has been made, all the balloons are moved according to the wind layer they are currently located in. Furthermore they are moved up or down according to the model parameter for vertical speed.

3. Update statistics: The statistics of the simulation are gathered on the fly so after each iteration the data is re-evaluated and stored for further analysis.

3.5 Assumptions

3.6 Measurements

3.7 Design

3.8 Implementation

3.9 Control Algorithms

3.9.1 Description

3.9.2 Algorithm 1

The first algorithm is straightforward and rudimentary. After all the balloons are deployed on the starting point the following logic helps the balloons determine whether it should move or not:

Algorithm .1 Control Algorithm 1

```
if More than 1 balloon at current space AND balloon is not moving then
    if At bottom layer then
        Go up
    end
    else if At top layer then
        Go Down
    end
    else
        Choose direction at random
    end
end
else
    Stay in current layer
end
```

This algorithm has an obvious downside since as it contains an element of random, which is usually not helpful in a control algorithm. Nevertheless, it spreads the balloons pretty thoroughly around the grid relatively fast. The control over the grid is however limited, and this would not be a suitable solution if the end goal is to provide stable coverage.

This algorithm might however be utilized as a part of a larger more complex algorithm. At the beginner stages of the balloon's life cycle it could prove helpful to get the balloon out of that first wind layer and join the huddle.

3.9.3 Algorithm 2

This algorithm is a step towards a more complex control algorithm. Some minor logic replaces the random generator in the previous solution.

Algorithm .2 Control Algorithm 2

```
if More than 1 balloon at current space then
|  Calculate projections
|  if optionUp < optionDown then
|  |  Go up
|  end
|  else
|  |  Go down
|  end
end
else
|  Stay in current layer
end
```

The balloon fetches the wind vectors from neighbouring wind layers and uses them to weigh its options. It computes the number of balloons occupying three different cells:

1. The cell to which the balloon's current wind layer will move it.

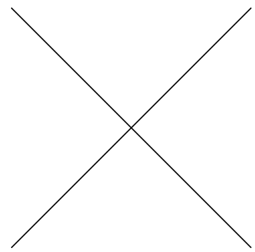

Figure 2: Calculate projections

2. The cell to which the the wind layer above the balloon's current wind layer would move it.
3. The cell to which the the wind layer below the balloon's current wind layer would move it.

The algorithm then determines which option has the cell with the lowest number of balloons and chooses that direction.

It is interesting to see that Algorithm 1 outperforms Algorithm 2 when it comes to coverage. Algorithm 1 reaches higher coverage like Figure 3 clearly shows.

3.9.4 Algorithm 3

3.10 Comparison of performance and complexity

3.11 Data analysis of the simulations

3.12 Conclusion of simulations

4 Conclusion

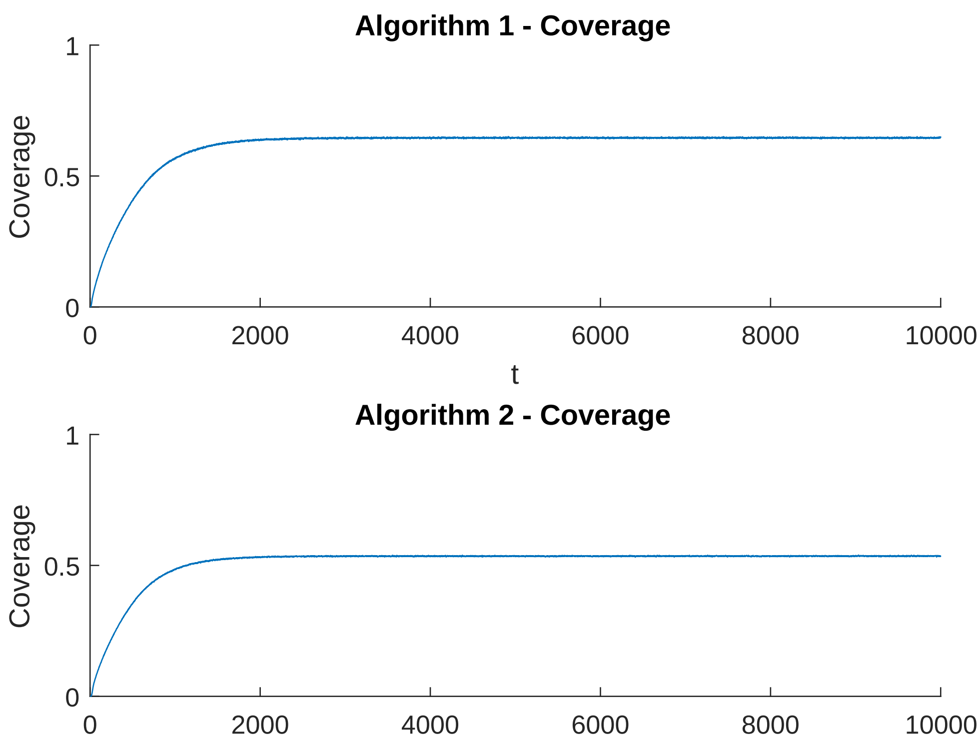

Figure 3: Algorithm 1 outperforms Algorithm 2