USTH - Project MI 2.05
Subject 2: Impacts of economic migration

Cao Anh Quan

12 June 2018

1 Introduction

1.1 Context

The drop in revenues in certain categories of farmers may force them to migrate and find other activities elsewhere, leaving their parcels unoccupied. I make the assumption that empty parcels become available for the other farmers, who can buy and use them.

1.2 Goal

Adapt the behavior of the farmers so that they have (1) a probability to migrate with respect to their revenue; (2) a probability to buy empty parcels in their neighborhood if their capital is sufficient. Explore the impact of these changes on the global land-use.

1.3 Project Archive

Project URL: https://github.com/caoquan95/GAMA-project. There are 3 folders in the project:

1. GAMA - Contains GAMA code
2. Java - Contain Java code for the Calibration
3. R - Contain R code for the Sensitivity Analysis
4. Report.pdf - The report of the project

2 Plan

In order to simulate the economic migration, there are 3 aspects I need to implement:

1. Economic activities of the farmer: How the farmers spend their money, gain profit.
2. Lands selling activities: How the farmers sell their lands and migrate.
3. Land buying activities: How the farmers buy empty lands.

3 Modification

3.1 Economic activities of the farmer

I load the **cost** and **price** for each crop. Every year, the farmers will have to spend the money to buy crop. The cost that the farmers have to pay is the product of the crop and the parcel area which is owned by the farmers.

```
expense <- parcel_area * crop_cost
```

Every year, the farmers will gain the amount of money equal to the product of the price and the parcel area owned by the farmers. I add another parameter *risk_control* multiply to the original risk of the crop in order to control the profit of the farmers. if the risk is high, the farmers will be more likely to lose the money.

```
revenue <- parcel_area
    * crop_price * (1 - lu.risk * risk_control)
```

Therefore, the amount of money the farmers gain will be:

```
profit <- revenue - expense
```

I add a field money to the farmer and every year, I will add the profit to the money of the farmer.

```
money <- money + profit
```

The size of farmer represents his wealth. The richer he is, the bigger he is.

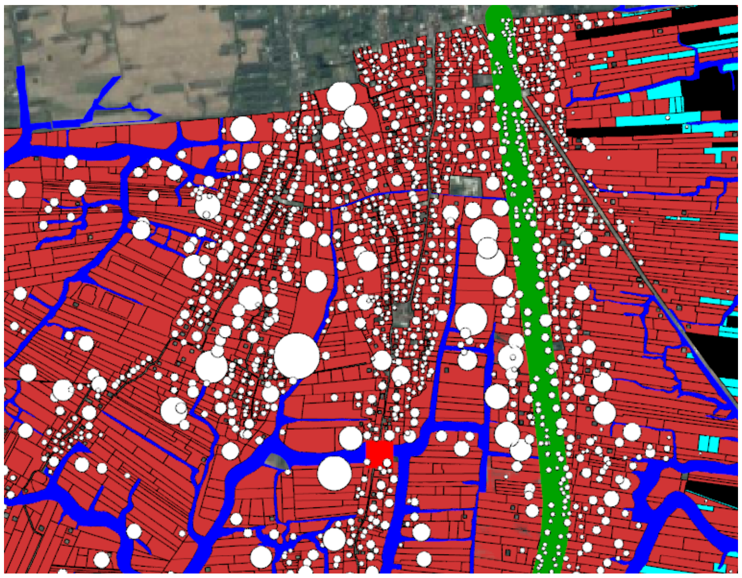

Figure 1: The richer he is, the bigger he is.

3.2 Lands selling activities

I change from **one farmer owns one parcel** to **one farmer owns many parcels**. I add a field named *sell_prob* to the farmer. The higher the *sell_prob*, the higher chance that the farmer will sell the parcel.

I add a parameter *sell_prob_change* which represents the amount of sell probability decrease after each successful year (profit > 0) or increase after each unsuccessful year (profit < 0).

Every year, if the farmer has positive profit, he will less likely to sell his lands and will be more likely to buy lands.

```
if (profit < 0) {
    sell_prob <- sell_prob + sell_prob_change;
} else {
    sell_prob <- sell_prob - sell_prob_change;
}
```

If the farmer has no money left, the sell probability will be 1.0. He will sell his parcels until his money if positive. If the farmer has no parcel left, He will migrate.

```
loop while: (flip(sell_prob) and length(my_parcels) > 0)
{
    // if the farmer have no money left,
    // he/she sells his/her parcel
    parcel p <- self.my_parcels[0];
    self.money <- self.money + p.price; //
    p.owner <- nil;
    remove p from: my_parcels;
    if (money > 0) {
        sell_prob <- 0.0;
    }
}

if (length(my_parcels) = 0)
{
    // farmer migrates when he/she owns no land
    self.migrated <- true;
}
```

The parcels that have no owner will have the color **black**.

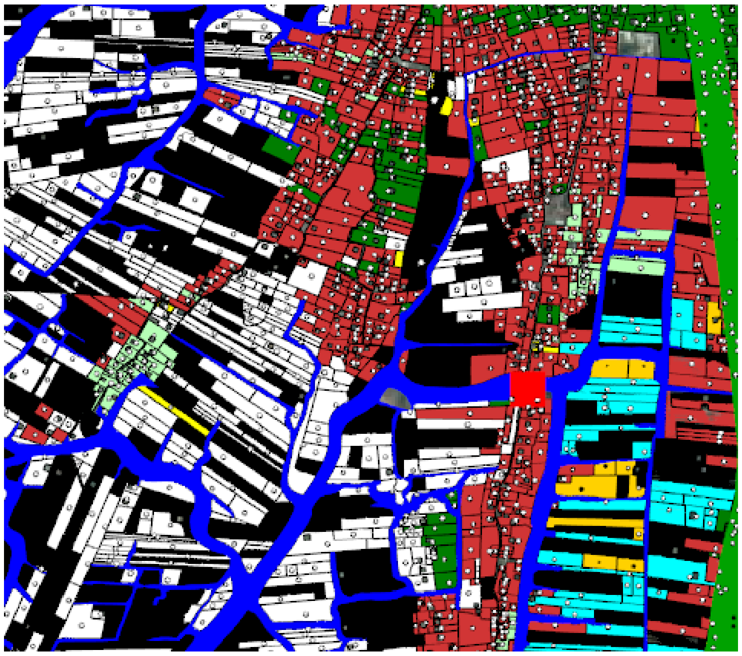

Figure 2: The parcels that have no owner will have the color **black**.

3.3 Land buying activities

The farmer looks at his vicinity to find the parcels without an owner to buy. If he has enough money, he will buy the land with the probability of

$$buy\_prob = 1 - sell\_prob$$

The distance that the farmers can look at to buy the parcels will be proportional to the **number of parcels** and the **size of parcels**, which is also proportional to the **wealth** he has.

```
can_purchase <- (parcel at_distance
    (50 * (length(my_parcels)) + size)
    where (each.owner = nil))
```

The parcels that have the same owner will be planted the same crop.

4 Result

4.1 Model Exploration

At the beginning, each farmer owns one parcel and I random the money that each farmer own.

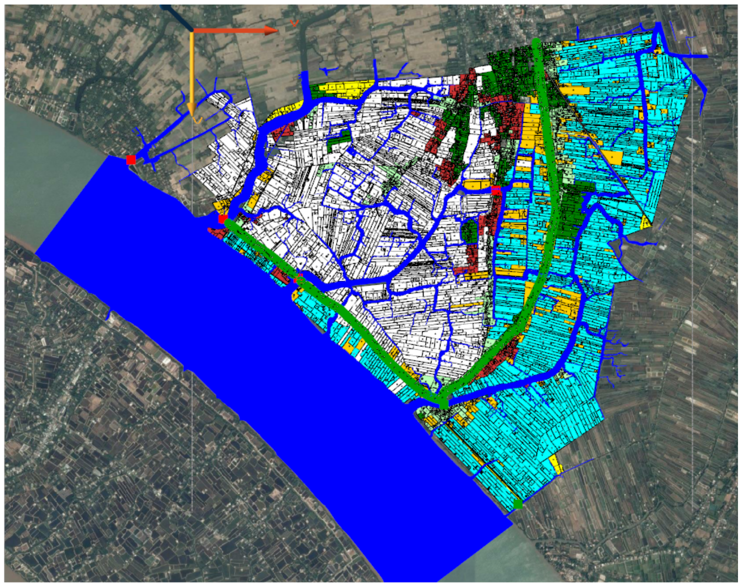

Figure 3: Map

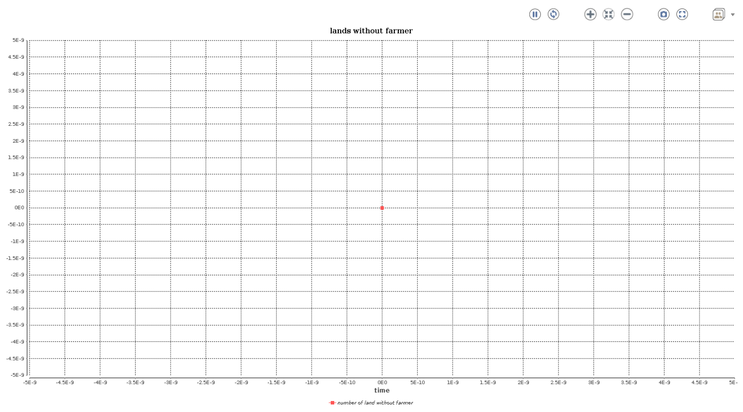

Figure 4: Lands without farmer

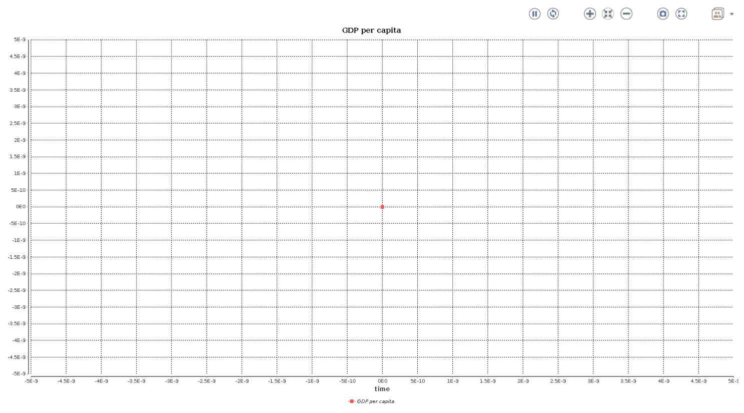

Figure 5: GDP per capita

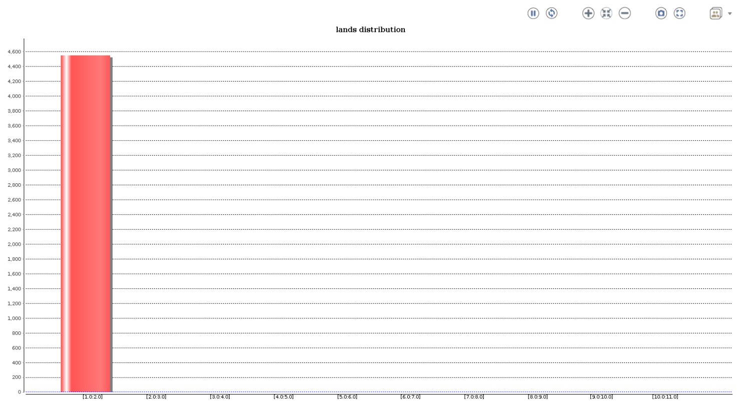

Figure 6: Land distribution - each farmer owns 1 land at the beginning

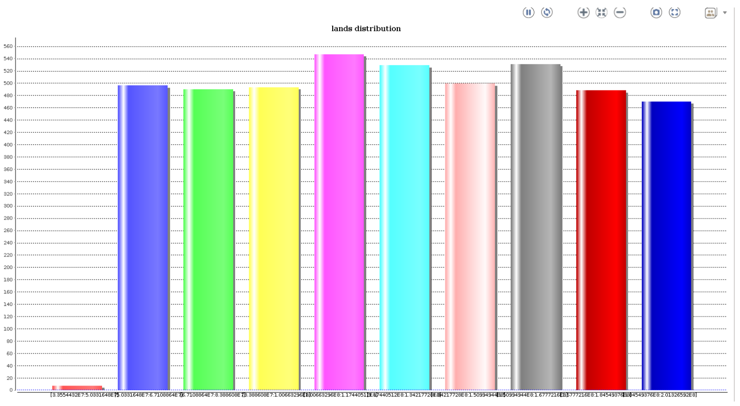

Figure 7: Wealth distribution - each farmer has random amount of money

After several cycles, some farmers cannot make profit and they migrate. It leads to the drop of GDP per capita and the increase of the number of parcels without farmer. Many farmer now own zero parcel while some other farmers, who made profit now have enough money to own up to 6 or even 7 parcels. We can see a lot of black parcels on the map which is the parcels without owner. According to the wealth distribution chart, some farmers are much richer than the others and some are in debt.

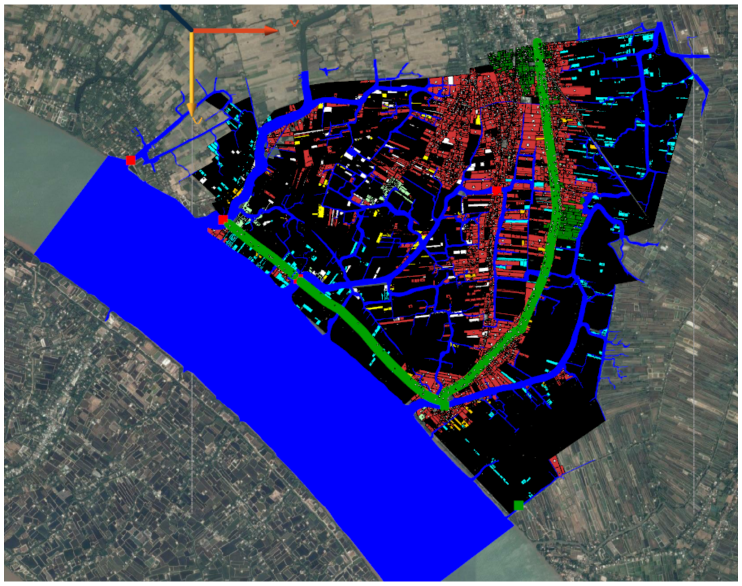

Figure 8: Map

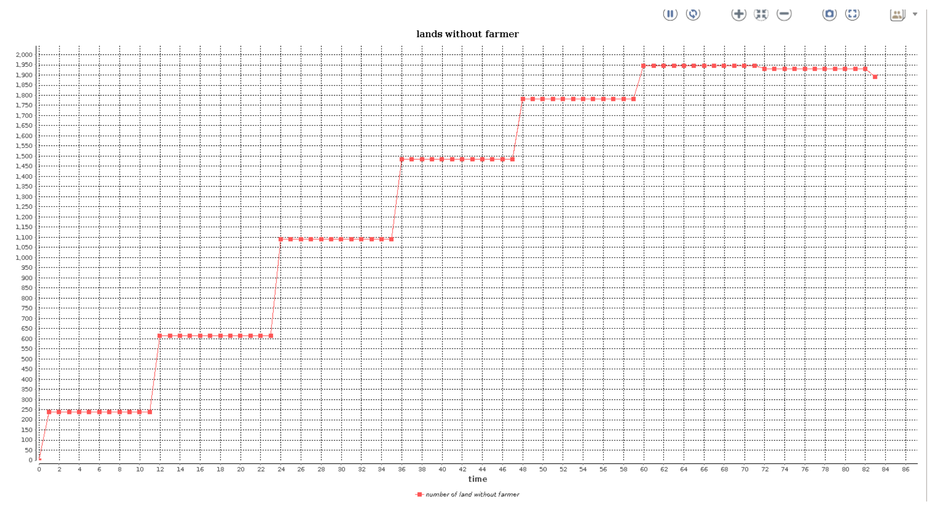

Figure 9: Lands without farmer

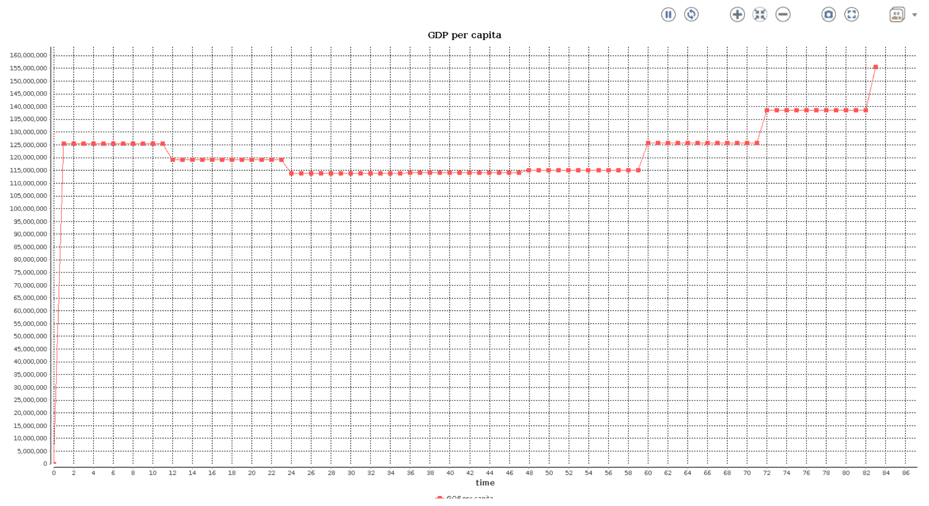

Figure 10: GDP per capita

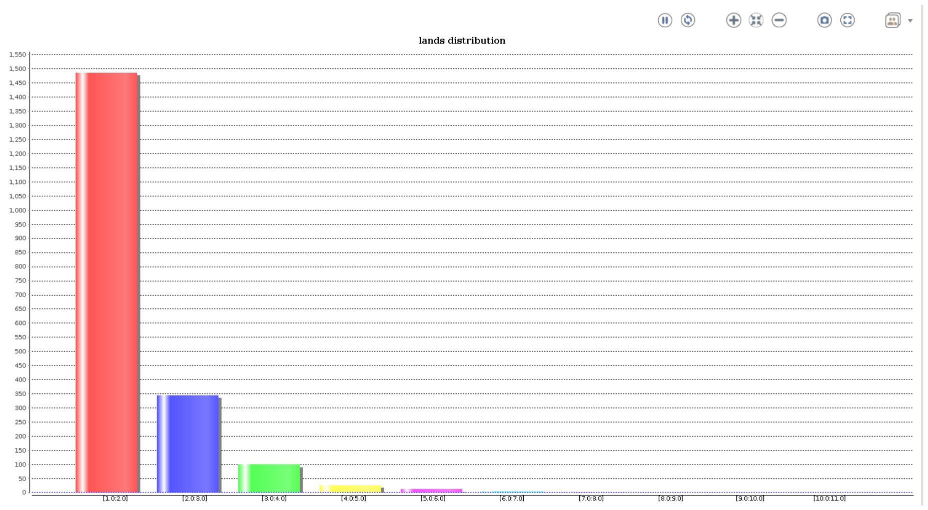

Figure 11: Land distribution

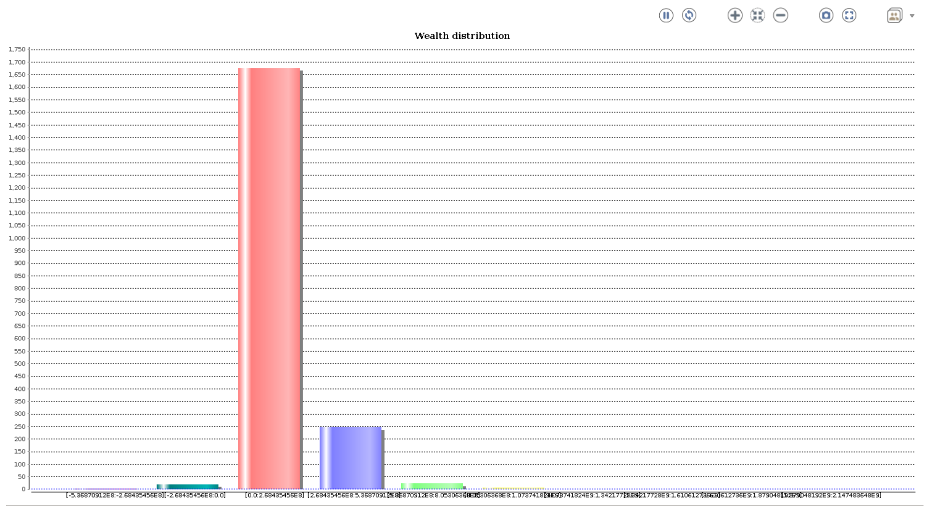

Figure 12: Wealth distribution

Finally, We can see that, all lands are purchased by the rich farmers. The number of parcels without farmer is decreased to zero and the GDP increases rapidly. Some farmers are very rich. Most of farmers have 0 to 16 parcels and some of them have up to 64 parcels. All the farmers, who are in debt and do not make a profit, had migrated.

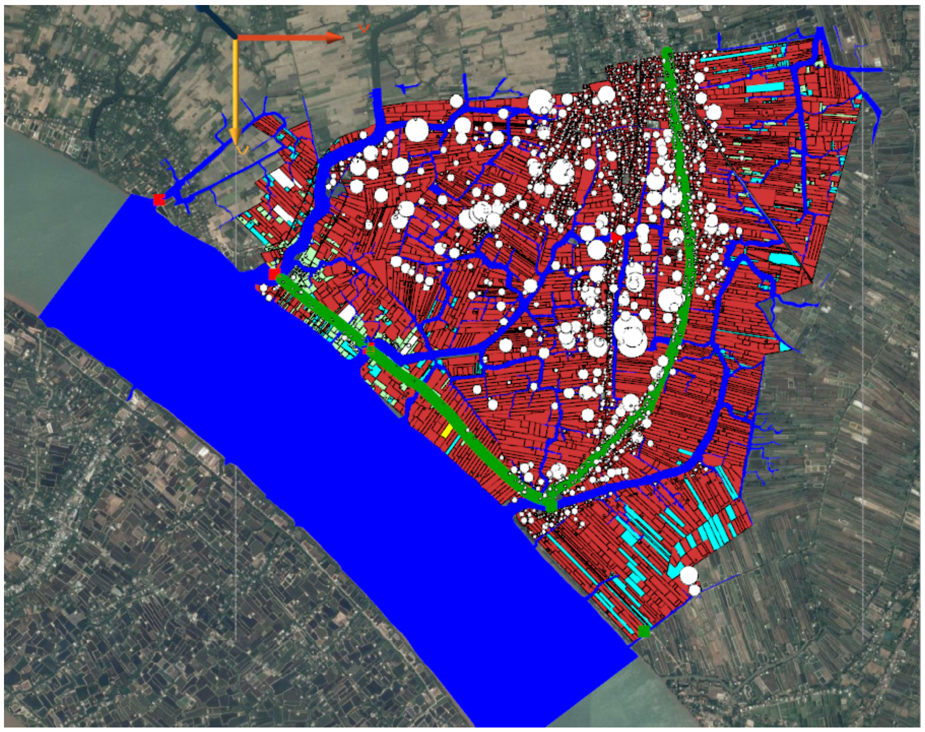

Figure 13: Map

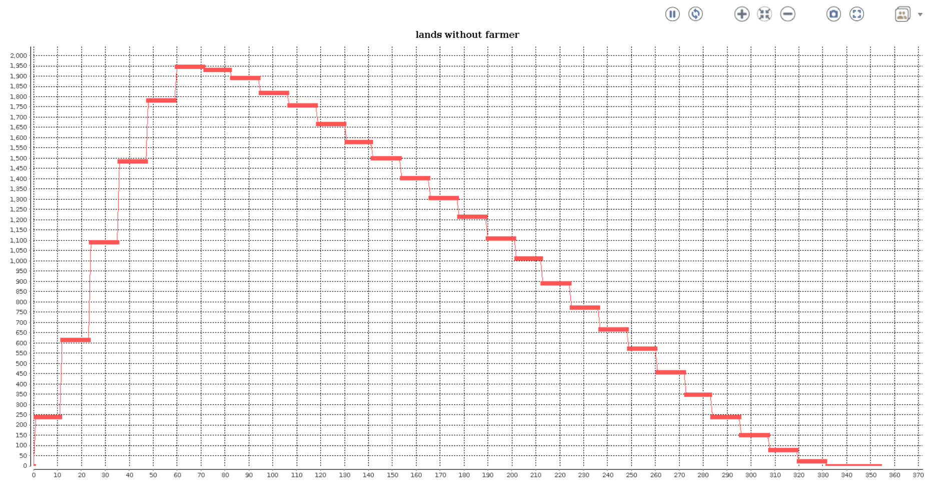

Figure 14: Lands without farmer

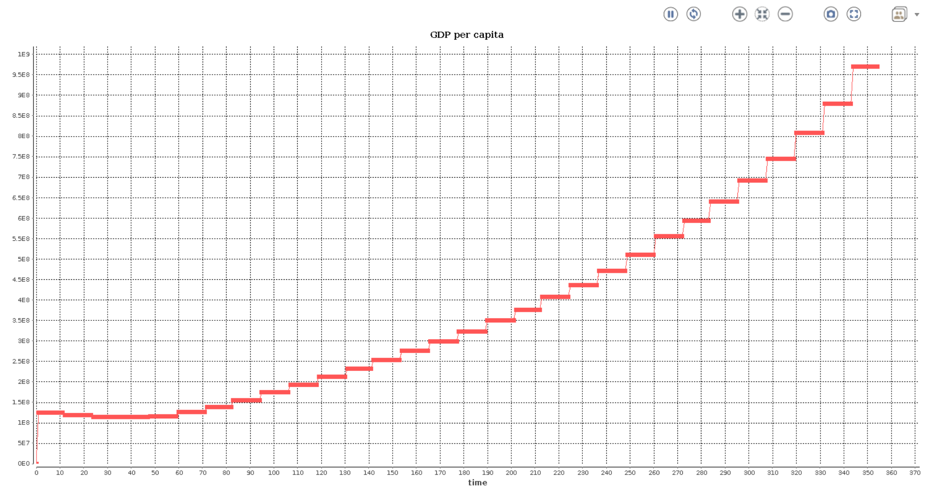

Figure 15: GDP per capita

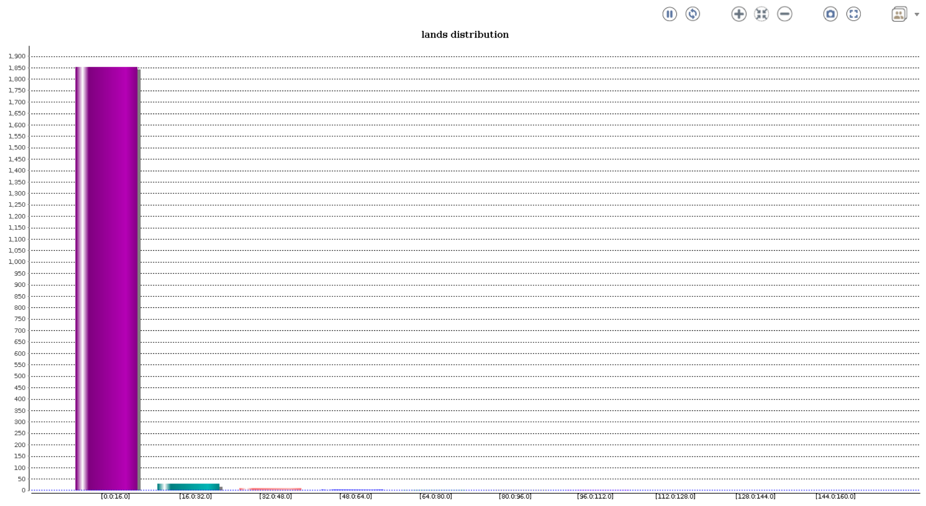

Figure 16: Land distribution

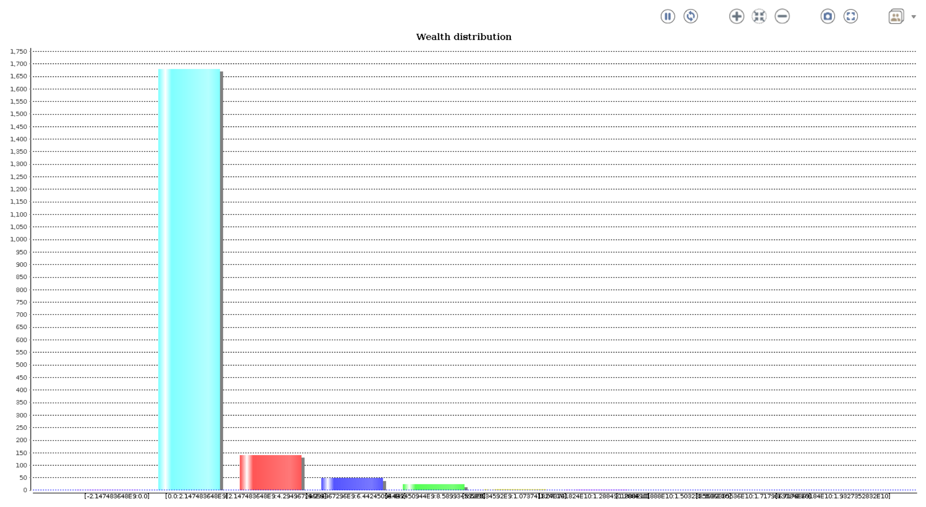

Figure 17: Wealth distribution

4.2 Sensitivity Analysis

4.2.1 CSV result

```
'Risk control','Land price','Prob change','Error','Farmer'
0.5,50000.0,0.1,2949.0,2996.5
0.5,50000.0,0.30000000000000004,3042.5,2182.5
0.5,50000.0,0.5,3083.0,1862.0
0.5,100000.0,0.1,2995.0,2898.5
0.5,100000.0,0.30000000000000004,3046.0,2104.5
0.5,100000.0,0.5,3066.0,1804.5
0.5,150000.0,0.1,3030.5,2862.5
0.5,150000.0,0.30000000000000004,3038.0,2061.0
0.5,150000.0,0.5,3062.5,1771.0
0.5,200000.0,0.1,3027.0,2870.0
0.5,200000.0,0.30000000000000004,3044.5,2012.0
0.5,200000.0,0.5,3042.0,1753.0
1.0,50000.0,0.1,2934.5,2870.5
1.0,50000.0,0.30000000000000004,2999.0,2062.0
1.0,50000.0,0.5,3052.0,1754.5
1.0,100000.0,0.1,2963.5,2769.0
1.0,100000.0,0.30000000000000004,2998.5,1944.0
1.0,100000.0,0.5,2999.5,1637.0
1.0,150000.0,0.1,3004.5,2710.0
1.0,150000.0,0.30000000000000004,3014.5,1896.5
1.0,150000.0,0.5,2987.0,1581.5
1.0,200000.0,0.1,3008.5,2672.0
1.0,200000.0,0.30000000000000004,3027.0,1878.5
```

4.2.2 R code for Variance Analysis

```
file <- "/home/quan/gama_workspace/ProjectUSTH-LUP/GAMA/models/
    resVar.csv"
print(file)
dataexf <- read.table(file, header=T, sep=",", dec=".")
summary(dataexf)
aovexf <- aov(Error ~ Land.price * Prob.change * Risk.control, data
     = dataexf)
summary(aovexf)
round(summary(aovexf)[[1]][2]/sum(summary(aovexf)[[1]][2])*100,2)
```

4.2.3 Variance Analysis Result

```
                                      Sum Sq
Land.price                              0.71
Prob.change                            12.63
Risk.control                           14.03
Land.price:Prob.change                 11.32
Land.price:Risk.control                10.67
Prob.change:Risk.control               11.76
Land.price:Prob.change:Risk.control     3.88
Residuals                              35.01
```

I can see that more 20% of the variance is due to the **Risk.control** and **Prob.change** parameter. Therefore, those are the most important parameters. The **residual** accounts for 35.01% - pretty unstable model. The Land.price almost has no impact on the model.

4.3 Calibration

From the previous subsection, I knew that the Land.price does not have much impact on the model. Therefore, I will only calibrating the **Prob.change** and **Risk.control** to minimize the **Error** to the correct land use 2010.

4.3.1 GAMA headless experiment

```
experiment Headless type: gui {
  parameter "Risk control: " var: risk_control min: 0.5 max: 2.5
    step: 0.5;
  parameter "Sell prob change: " var: sell_prob_change min: 0.1 max
    : 0.5 step: 0.2;

  output {
    monitor "error" value: parcel count (each.my_land_use.lu_code
    != each.lu_years[2010]);
  }
}
```

4.3.2 Result

After the calibration, I got the minimum error is 2969.0 with the parameter set:

```
risk_control = 2.0
sell_prob_change = 0.30000000000000004
```

Some console output from Java program

```
-----------1--------------
risk_control = 0.5
sell_prob_change = 0.1
FIN du process !!
Error: 2989.0
----------0--------------
risk_control = 1.0
sell_prob_change = 0.1
FIN du process !!
Error: 3009.0
----------1--------------
risk_control = 1.5
sell_prob_change = 0.1
FIN du process !!
Error: 3006.0
----------2--------------
risk_control = 2.0
sell_prob_change = 0.1
FIN du process !!
Error: 3048.0
----------3--------------
risk_control = 0.5
sell_prob_change = 0.30000000000000004
FIN du process !!
Error: 3026.0
----------4--------------
risk_control = 1.0
sell_prob_change = 0.30000000000000004
FIN du process !!
Error: 3007.0
----------5--------------
risk_control = 1.5
sell_prob_change = 0.30000000000000004
FIN du process !!
Error: 2969.0
----------6--------------
risk_control = 2.0
sell_prob_change = 0.30000000000000004
FIN du process !!
Error: 3046.0
Min Error: 2969.0
min parameter set:
risk_control = 2.0
sell_prob_change = 0.30000000000000004
```

Java code for calibration. I place the full file in the Java folder in the root of the project. Here is the short one.

```java
ExhaustivePlan exp = new ExhaustivePlan();
exp.setEperimentName("Headless");
exp.setFinalStep(200);
exp.setPath("/home/quan/gama_workspace/ProjectUSTH-LUP/GAMA/models/Model_Complete.gaml");
exp.setStopCondition("cycle > 200 or current_date.year = 2010");

exp.addParameter(new Parameter("risk_control", "FLOAT", 0.5, 2.5, 0.5));
exp.addParameter(new Parameter("sell_prob_change", "FLOAT", 0.1 ,
    0.5, 0.2));

exp.addOutput(new Output("error", 1, "1"));

double minError = Double.POSITIVE_INFINITY;
List<Parameter> minPs = null;


while (exp.hasNextParametersSet()) {
  System.out.println("---------" + exp.getExperimentNumber() + "
    ------------");
  List<Parameter> ps = exp.nextParametersSet();
  ps.stream().forEach(p -> System.out.println(p));

            // absolute Path mandatory !!
   String XMLFilepath = "/home/quan/Downloads/USTH/GAML/farmer" +
    exp.getExperimentNumber();
  exp.writeXMLFile(XMLFilepath + ".xml");

  GAMACaller gama = new GAMACaller(XMLFilepath + ".xml",
    XMLFilepath);
  gama.runGAMA();

  XMLReader read = new XMLReader(XMLFilepath + "/simulation-outputs
    .xml");
  read.parseXmlFile();
  double error = Double.parseDouble(read.getFinalValueOf("error"));
  System.out.println("Error: " + error);
  if (minError > error) {
    minError = error;
    minPs = ps;
  }
}
System.out.println("Min Error: " + minError);
System.out.println("min parameter set:");
minPs.stream().forEach(p -> System.out.println(p));
```