A PROJECT REPORT

ON

“Agent-Based Modeling For Simulation of telecom services”

JATIN	2019UMC4508
GULSHAN SHARMA	2019UMC4530
VARUN KUMAR	2019UMC4532
SAURABH	2019UMC4563

Under The Guidance of

Prof. Surbhi Sharma

Acknowledgements

We are profoundly grateful to **Prof. Surbhi Sharma** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards **Mr. Dharmendra Singh**, whose invaluable guidance supported us in completing this project.

Jatin
Gulshan Sharma
Varun Kumar
Saurabh

ABSTRACT

An agent-based model (ABM) is a class of computational models for simulating the actions and interactions of autonomous agents (both individual or collective entities such as organizations or groups) with a view to assessing their effects on the system as a whole. Agent-based modeling is related to, but distinct from, the concept of multi-agent systems or multi-agent simulation in that the goal of ABM is to search for explanatory insight into the collective behavior of agents obeying simple rules. It combines elements of game theory, complex systems, emergence, computational sociology, multi-agent systems, and evolutionary programming. Our project aims at developing an agent based model for simulation of telecom services. Also we exploited this model for market and risk analysis.

GitHub URL

Contents

Introduction

Problem Statement of the project: To develop an ”Agent-based model for simula- tion of telecom services”.

As we know that any environment which involves various agents is going to be very complex. Also, under such scenario analysis of that environment becomes a troublesome job if done manually, because these environment carry huge number of possibilities within themselves.

A similar example of such environment is telecom market. Telecom market do have huge number of agents which includes large number of users, telecom service providers and regulator(moderator).

Telecom service providers are the agents who offer telecom services to the users. Users are meant to avail the services offered by the providers. Regulator sits at the top of the hierarchy and governs the overall market by setting up the constraints and forcing every agent to follow them effectively.

Our project aims at simulating telecom market having above mentioned agents as its constituents and thus plotting the reality of market in the computer screen. Also we used this simulation in order to analyze the market(Business analysis) and calculation of risk involved(Risk analysis) for any client(viz telecom service providers).

Technologies Used

- **Pseudo Random Number Generator (PRNG)**

Pseudo Random Number Generator(PRNG) refers to an algorithm that uses mathematical formulas to produce sequences of random numbers. PRNGs generate a sequence of numbers approximating the properties of random numbers.

PRNGs are suitable for applications where many random numbers are required and where it is useful that the same sequence can be replayed easily. Popular exam- ples of such applications are simulation and modeling applications.

Characteristics of PRNG are as follows:

- Efficient: PRNG can produce many numbers in a short time and is advantageous for applications that need many numbers
- Deterministic: A given sequence of numbers can be reproduced at a later date if the starting point in the sequence is known. Determinism is handy if you need to replay the same sequence of numbers again at a later stage.
- Periodic: PRNGs are periodic, which means that the sequence will eventually repeat itself. While periodicity is hardly ever a desirable characteristic, modern PRNGs have a period that is so long that it can be ignored for most practical purposes.

  Java Math random() method :- The java.lang.Math.random() method returns a pseudorandom double type number greater than or equal to 0.0 and less than 1.0. We have tailored Math.random() function according for carrying out our task.

Abstract Window Toolkit (AWT)

Abstract Window Toolkit (AWT) is a set of application program interfaces (API s) used by Java programmers to create graphical user interface (GUI) objects. Package java.awt Contains all of the classes for creating user interfaces and for painting graphics and images.

We have used AWT package for creating pie chart in order to represent the results effectively.

ArrayList

ArrayList is a part of collection framework and is present in java.util package. It provides us dynamic arrays in Java. Though, it may be slower than standard arrays but can be helpful in programs where lots of manipulation in the array is needed. We have used ArrayList for storage of strategies picked up by providers at a particular instance and washing them off when next instance comes in.

Syntax:- ArrayList arrli = new ArrayList(size);

- ArrayList inherits AbstractList class and implements List interface.
- ArrayList is initialized by a size, however the size can increase if collection grows or shrunk if objects are removed from the collection.
- Java ArrayList allows us to randomly access the list.

Libraries used

- jcommon-1.0.8.jar

  JCommon is a free general purpose Java class library that is used in several projects at www.jfree.org, including JFreeChart and JFreeReport.

- jfreechart-1.0.1.jar

  JFreeChart is a class library, written in Java, for generating charts. Utilizing the Java2D APIs, it currently supports bar charts, pie charts, line charts, XY-plots and time series plots.

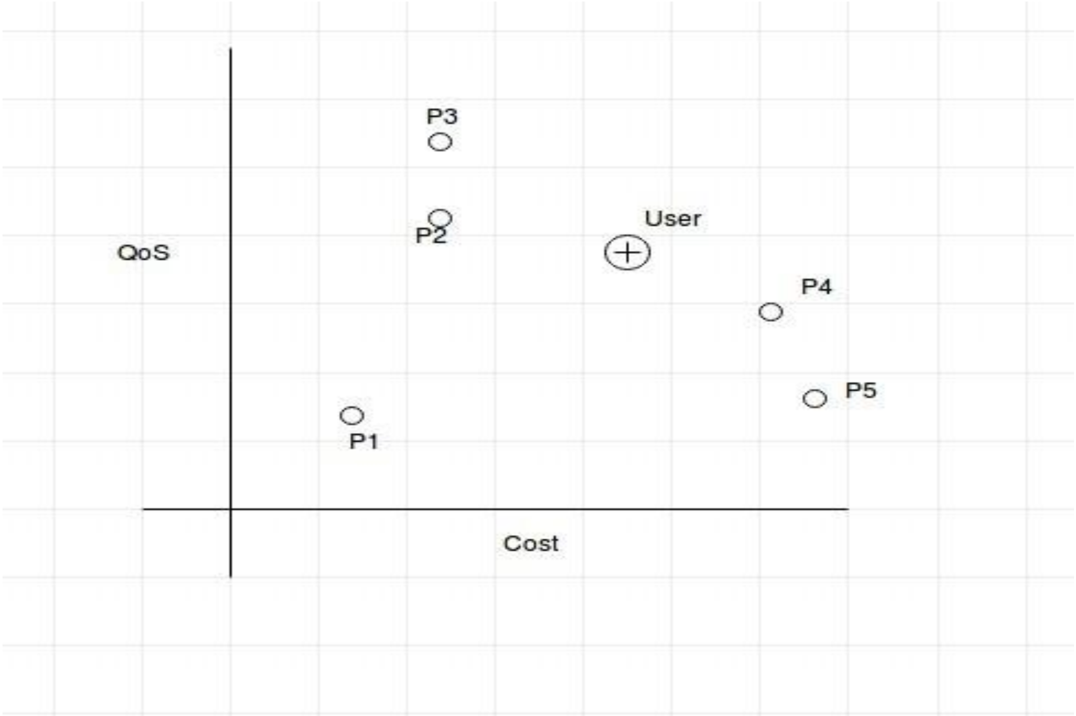
Implementation

• Process Flow

1. First of all, we have classified users in 3 categories
   a. Price conscious (20%)
   b. QoS conscious (20%)
   c. Average price and QoS conscious (60%)
2. Allocating a provider for a user which is very close to his expectations.

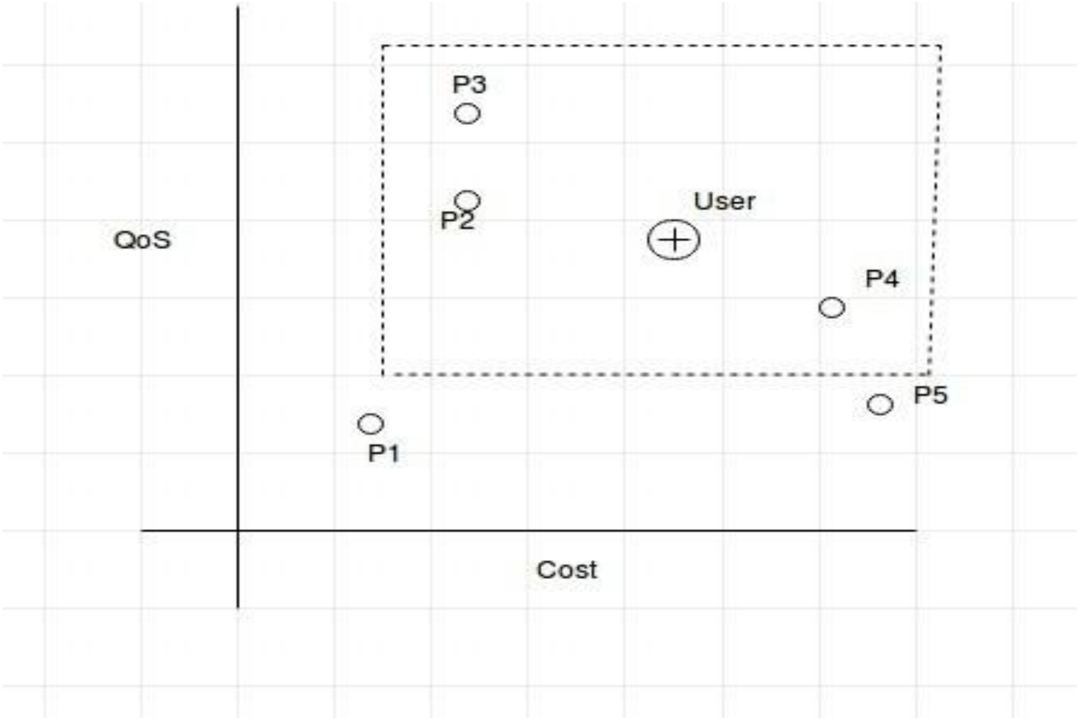
   Flow is as follows:-

   a. User will enter his expectation



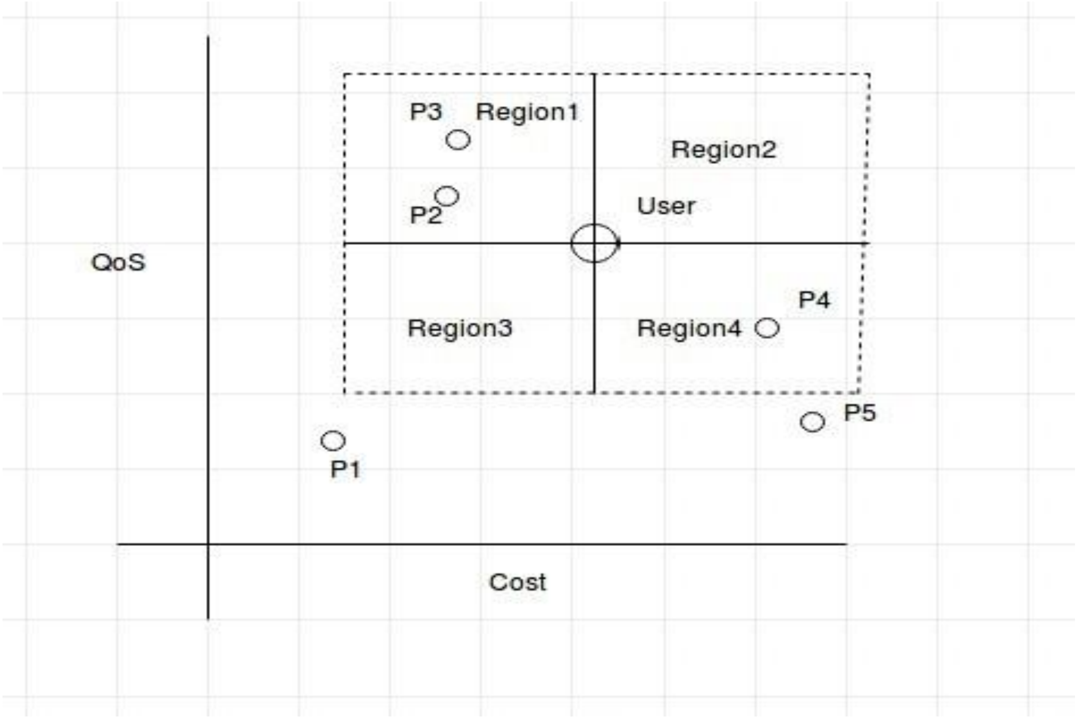
User's expectation

b. Now we will create area of inspection



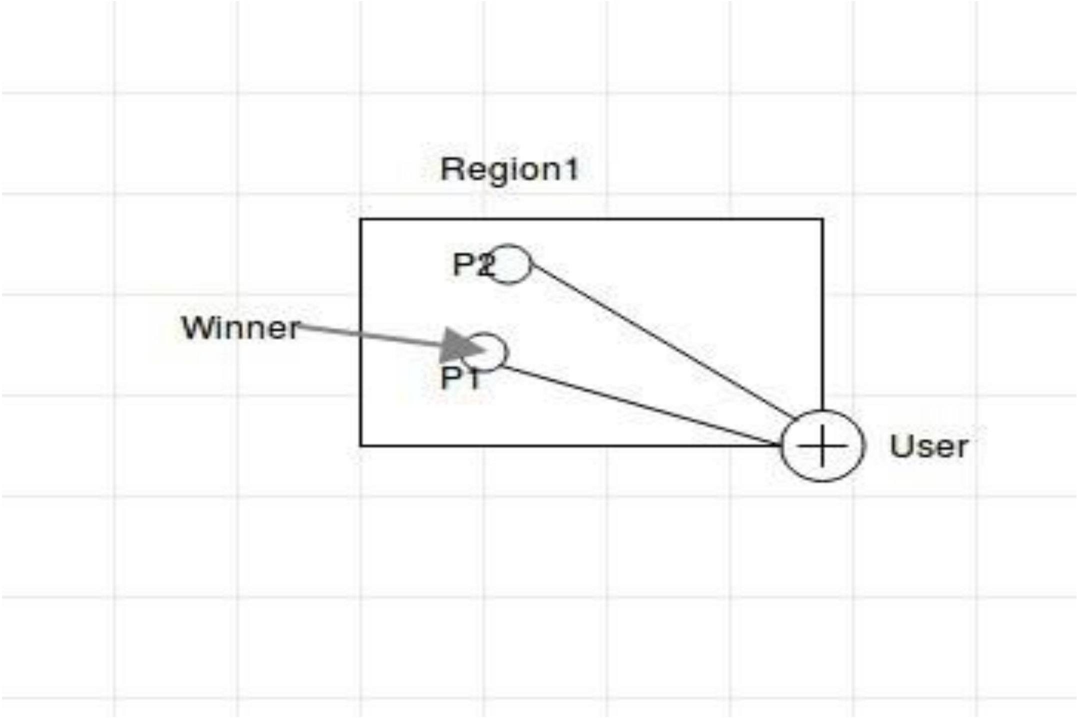
Area of inspection

c. Check for the presence of suitable provider in region 1, region 2 , region 3 and region4 (ORDER MATTERS). Also, if provider is found in one region, we will not check for remaining regions for obvious reasons.



Finding the apt provider

d. Allocate user to a provider



Winner(who grabs the user)

- ## DSIC Algorithm

We have used DSIC (Dominant Strategy Incentive Compatible) algorithm in orderto know which provider is best suited for a particular customer.

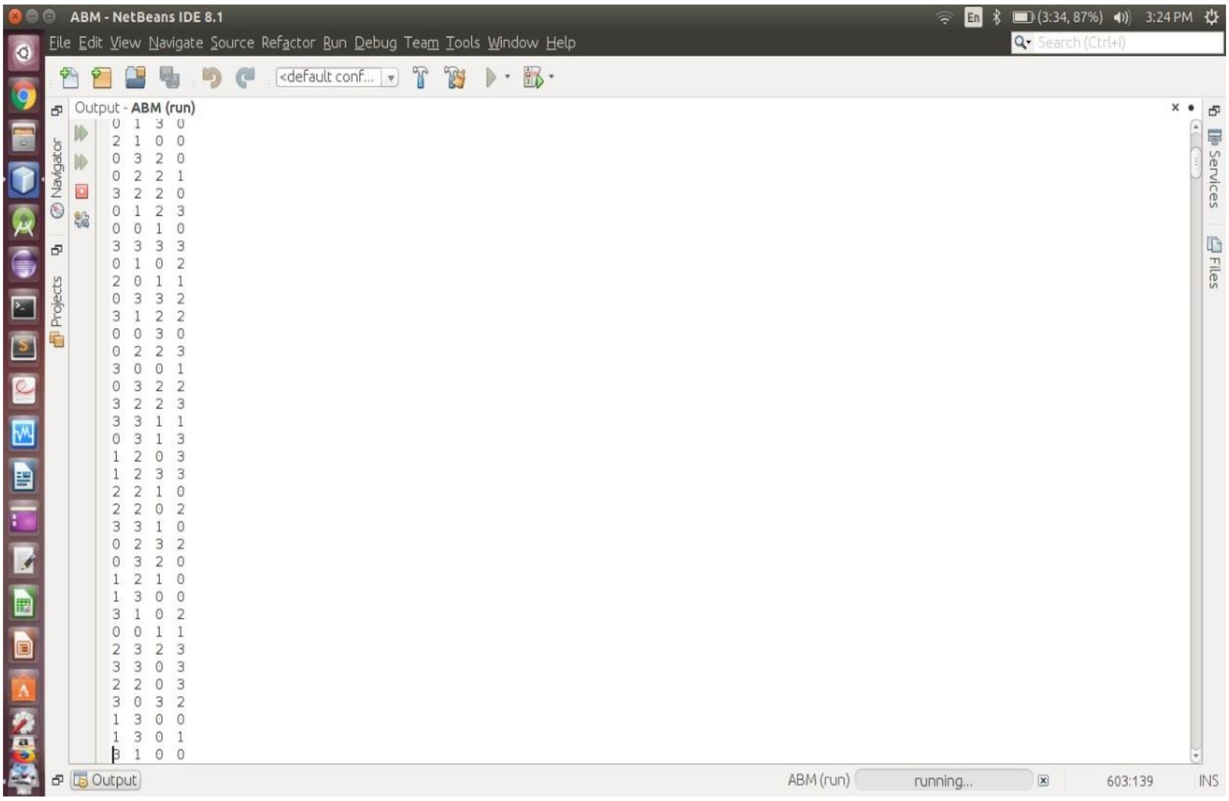
a. User will select that provider for which (cost/QoS) as expected by user is as close as possible to the (cost/QoS) provided by telecom service provider.
b. Service provider who grabs the user will get the corresponding payoff which is equal to the cost of the plan purchased by the user.

These results do follow our intuition i.e. if a client provides less QoS at a very high cost(test case T01), he is prone to risk and is in danger of being washed off fromthe market. However, if he provides good QoS at a cheaper price(test case T02), he follows a good hold in the market and hence safe percentage is on the higher side.

Screenshots of Project

- **Picked Strategy**

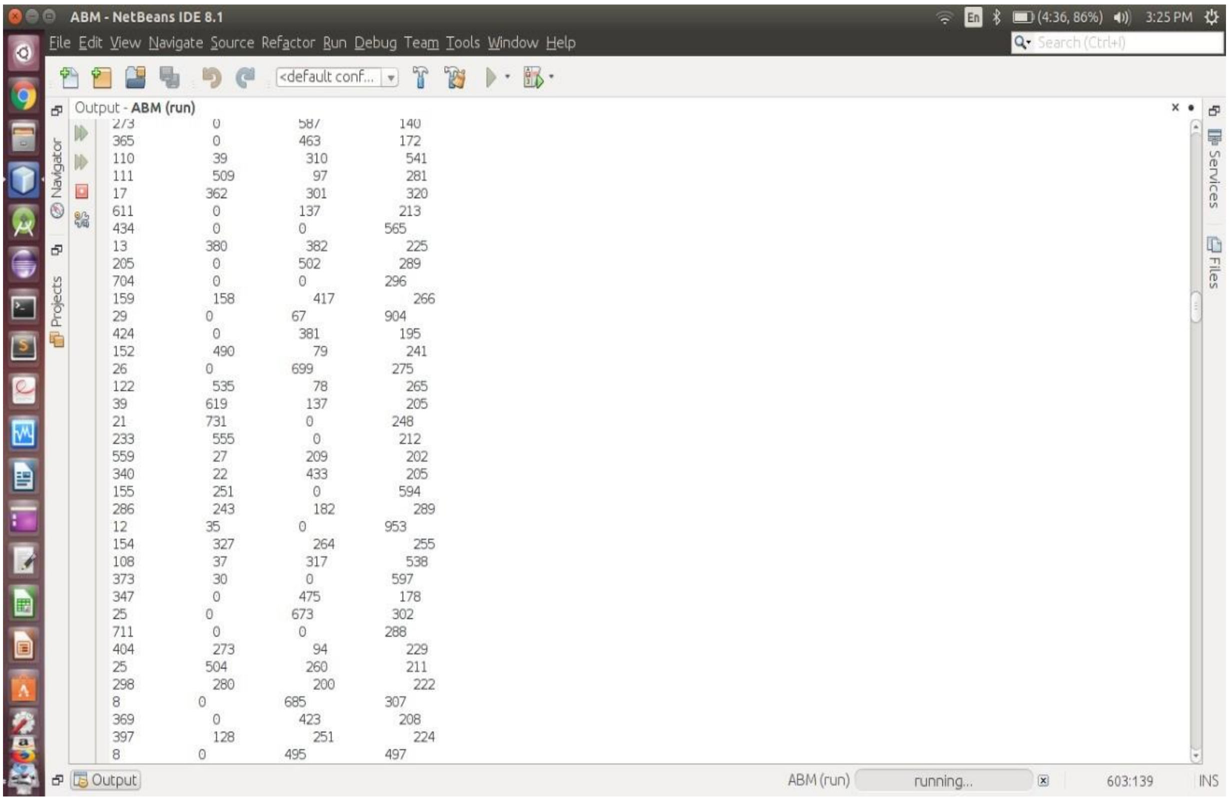
This picture depicts the set of strategies picked up by the providers at a particular instance while iterating thousand times.



```
0  1  3  0
2  1  0  0
0  3  2  0
0  2  2  1
3  2  2  0
0  1  2  3
0  0  1  0
3  3  3  3
0  1  0  2
2  0  1  1
0  3  3  2
3  1  2  2
0  0  3  0
0  2  2  3
3  0  0  1
0  3  2  2
3  2  2  3
3  3  1  1
0  3  1  3
1  2  0  3
1  2  3  3
2  2  1  0
2  2  0  2
3  3  1  0
0  2  3  2
0  3  2  0
1  2  1  0
1  3  0  0
3  1  0  2
0  0  1  1
2  3  2  3
3  3  0  3
2  2  0  3
3  0  3  2
1  3  0  0
1  3  0  1
3  1  0  0
```

- **User Distribution**

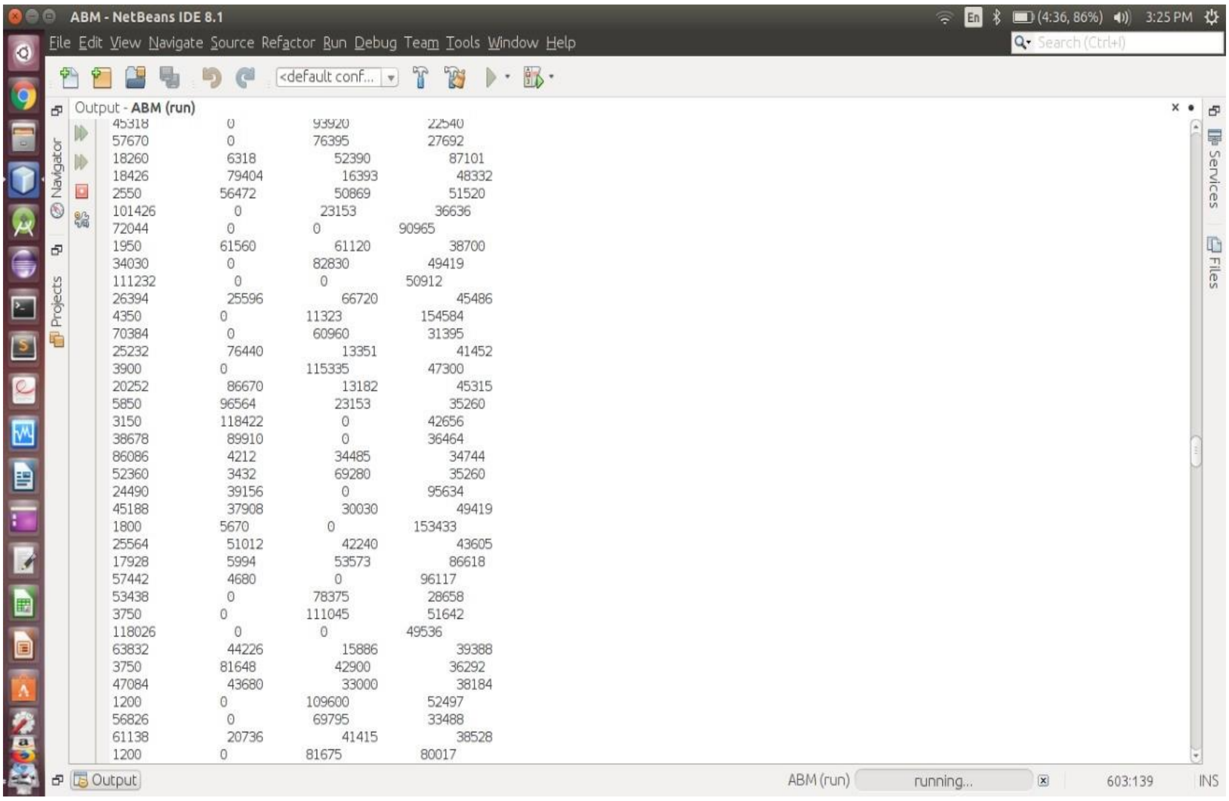
This picture depicts the distribution of users as per the DSIC(Dominant Strategy Incentive Compatibility mechanism) algorithm.



```
273          0            587          140
365          0            463          172
110           39           310          541
111           509           97          281
17           362           301          320
611          0            137          213
434          0            0           565
13           380           382          225
205          0            502          289
704          0            0           296
159           158           417          266
29          0           67           904
424          0            381          195
152           490           79          241
26          0           699          275
122           535           78          265
39           619           137          205
21           731           0           248
233           555           0           212
559           27           209          202
340           22           433          205
155           251           0           594
286           243           182          289
12           35           0           953
154           327           264          255
108           37           317          538
373           30           0           597
347          0            475          178
25          0           673          302
711          0            0           288
404           273           94          229
25           504           260          211
298           280           200          222
8          0           685          307
369          0            423          208
397           128           251          224
8          0           495          497
```

- **Service Providers Payoff**

Following picture depicts the payoff allocated to every service provider. Payoff of a provider is given by the number of users it grabbed multiplied by the cost of plan purchased by the user.



```
45318        0        93920        22540
57670        0        76395        27692
18260        6318        52390        87101
18426        79404        16393        48332
2550        56472        50869        51520
101426        0        23153        36636
72044        0        0        90965
1950        61560        61120        38700
34030        0        82830        49419
111232        0        0        50912
26394        25596        66720        45486
4350        0        11323        154584
70384        0        60960        31395
25232        76440        13351        41452
3900        0        115335        47300
20252        86670        13182        45315
5850        96564        23153        35260
3150        118422        0        42656
38678        89910        0        36464
86086        4212        34485        34744
52360        3432        69280        35260
24490        39156        0        95634
45188        37908        30030        49419
1800        5670        0        153433
25564        51012        42240        43605
17928        5994        53573        86618
57442        4680        0        96117
53438        0        78375        28658
3750        0        111045        51642
118026        0        0        49536
63832        44226        15886        39388
3750        81648        42900        36292
47084        43680        33000        38184
1200        0        109600        52497
56826        0        69795        33488
61138        20736        41415        38528
1200        0        81675        80017
```

References

[1] A Mechanism Design Approach to Resource Procurement in Cloud Computing; Prof. Shrisha Rao and Mr.Abhinandan S. Prasad

[2] Simple Additive Weighting approach to Personnel Selection problem; AlirezaAfshari, Majid Mojahed and Rosnah Mohd Yusuff

[3] https://www.coursera.org/learn/game-theory-1

[4] https://www.slideshare.net/lakshanasuresh/ahp-calculations

[5]http://iopscience.iop.org/article/10.1088/1757-899X/166/1/012020/pdf