Assignment 2

SYSC 5104

Conceptual Model for Mobile Network

Misagh Tavanpour

Department of System and Computer Engineering

Carleton University

October 2012.

Part I

1.1. Introduction

Mobile networks or cellular networks are kind of radio networks distributed over land areas. Each of these land areas known as a cell. Each cell has at least one fixed transceiver called base station (BS) and to avoid interference it uses a different set of frequencies from neighboring cells. These cells support radio coverage over a geographic area by joining together and this enables portable transceivers such as smart phones to communicate with each other via fixed transceivers (Fig 1). This communication can be hold even if both side of this communication are moving through more than one cell during transmission. The problem in this kind of definition is about user equipments (UE) near to cell borders. Clearly this kind of users have more distance to BS at center of the cell so maybe the quality of service they receive from their serving BS won't be good enough. There is possibility here that is user can be supported by other neighboring BSs. when a user traverse near to cell border it can be supported by neighboring cell's BSs too. In this situation it can send/receive data to/from those BSs too. This could be lead to an increase in UE performance. This capability in advanced mobile networks such as LTE-Advanced is called as coordinated multiple point (CoMP).

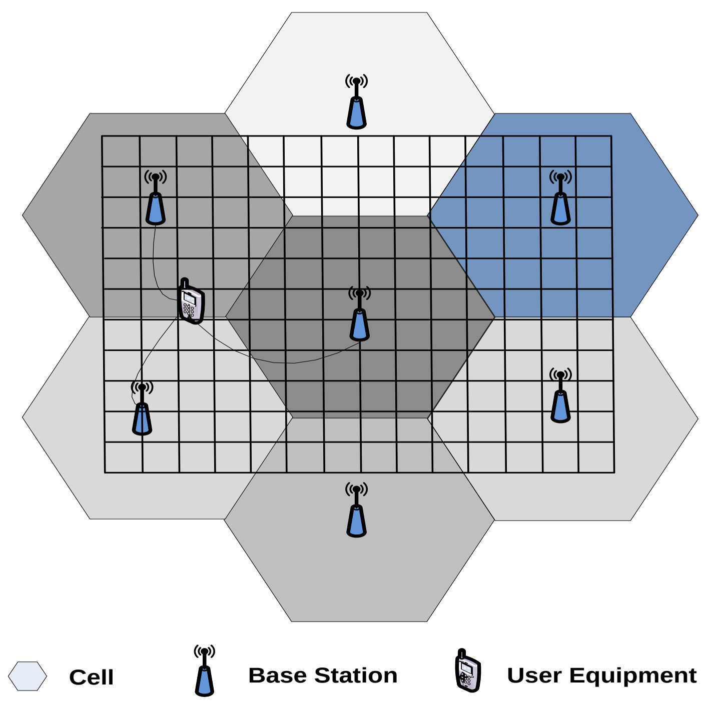


Fig 1: A sample mobile network

1.2. Model concept and basic definitions

In this Assignment, a Cell-DEVS model that tries to find out the number of BSs which cover a mobile user in different location of an area have been considered. As it can be seen in Fig 2, consider a two dimension area in which network coverage supported by number of Cells (from now on *"**C**ell"* refers to land area which has at least one BS and *"**c**ell"* refers its normal definition in Cell-DEVS). A mobile user can travel along this area and in each place the number of base stations which provide coverage for this cell can be varied from 0 (no coverage) to 3. Please note that the increase in the number of BSs that support a certain UE is not defiantly mean that that UE will experience better QoS. This increase in number of BS is base on a fact that near the Cell borders (Fig.1) UE can received the signal from neighboring Cells BSs. By using modern technique such as CoMP, more than one BS can provide service for these kinds of UEs to improve their QoS and make life easier. When a UE is close enough to its BS near the Cell center it is covered by one BS and it can receive the suitable signal and experience high QoS. If this UE starts to travel to Cell borders because of the presence of the noise and other similar factors the received signal quality starts to reduce step by step. Clearly it decreases the performance and the UE will have lower data rate and etc. By using techniques like CoMP network providers try to improve the performance of the users in Cell borders. This could be happen by coordination between different BSs in neighboring Cells to support a UE.

Therefore Mobile users can travel within the area randomly and after each movement their current position, number of UE which support them at this new position and the signal strength (according to their distance from BS) are calculated.

This model is three-dimension model and it has five planes including Existence plane, Movement plane, UE signal strength plane , Number of BSs plane and finally signal strength plane (Existence, Movement, SigStrUE, BSnum and SigStr). Fig 2 shows all this planes.

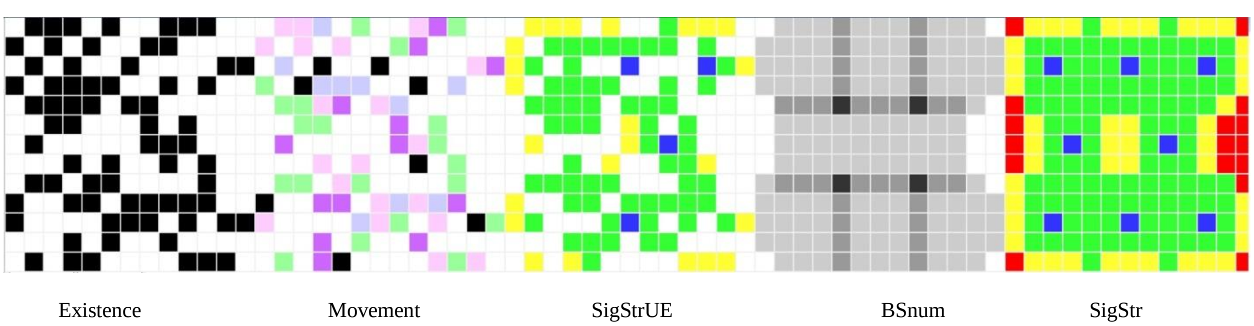


Fig 2: Different planes of model

In the first plane we keep track of UE's position during the simulation. 0 (white cells in Fig 2-Existence) means that there is no UE in a certain cell and 1 (black cells in Fig 2-Existence) means that there is one UE on that cell. Second plane contains the next movement direction of a UE. Each time the new values will be computed for the cells which will be host for UE according to the first plane so if the next value of cell in Existence plane will be 0 it means that there is no need to calculate next move in Movement plane. Movement values are random numbers between 2 to 6. Number two means UE wants to go to the east, 3 is for south, 4 is for west, 5 is for a move to north and 6 means that the UE in this cell wants to stay in same cell (Fig 3) . In case of collision between moving UEs the one with less number will move and the rest will stay in their current cell. So the descending priority for movement is east, south, west and north in order. The absence of UE in corresponding cell in plane 0 (cell which has value 0 in plane 0) is demonstrated by 8 in movement plane (white cells in movement plane in Fig 2).

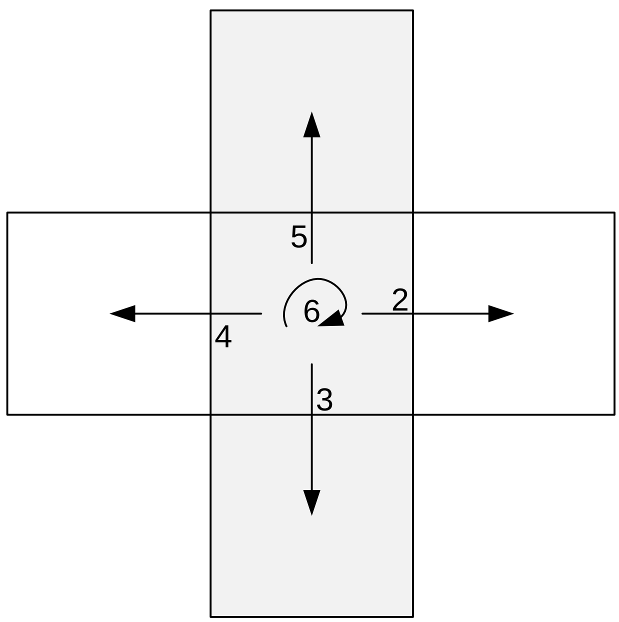


Fig 3: Movement guideline

Third plane or SigStrUE plane reveals information about signal strength of a UE in certain cell. Values in this plane are as follows: 25 (no coverage, red color in Fig 2), 26 (weak coverage, yellow color in Fig 2), 27 (good coverage, green color in Fig 2) and 28 (excellent coverage, blue color in Fig 2). For each cell this numbers is selected according to that cell distance from serving

BS at Cell center. So normally if a UE is in border of the Cell it will experience a weak coverage but if we consider that the network using CoMP technique to provide better service for such user then even UEs near Cells border can receive suitable service and they can benefit from good coverage too. In this assignment's simulations both situation were considered.

The fourth and fifth planes (BSnum and SigStr) include information about the number of BS which can support certain cell and signal strength on that particular cell. The latter plane concept is as same as third plane with tiny different between selected number for each mode (20 is for the cells without coverage, 21 is for the cells with weak coverage, 22 is for the cells with good coverage and 23 is for the cells with excellent coverage). As it was mentioned BSnum plane reveals the number of BSs which cover a certain cell. This plane values are something among 10, 11, 12 and 13 which correspond to 0 BS, 1 BS, 2 BS and 3 BS respectively. Both BSnum and SigStr planes initialized by Coverage.val file.

The rules for each plane will be discussed in more detail in next section. Also it is worth to mention that this model is not wrapped and there are no external inputs or outputs (except initialization information). Fig 4 shows the base area for modeling and simulation of this assignment. Cells and BSs distinguished and ones can understands the coverage area of each Cell and common area between Cells (in this kind of area CoMP technique can be used).

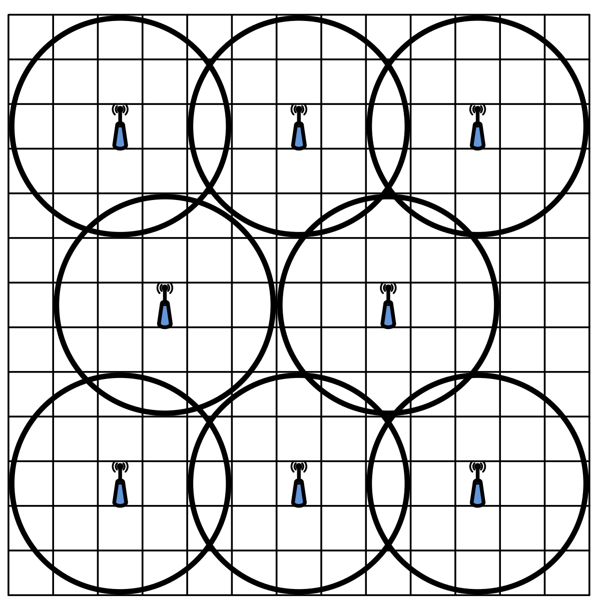


Fig 4. Sample area for simulation

Part II

2.1. Formal specification

Formal specification of the **Atomic Cell-DEVS** model and **Coupled Cell-DEVS** is as following. The more detail about the function is on the next section.

$\boldsymbol{Plane\ 0}$:

$CD = < X, Y, S, N, type, d, \tau, \delta_{int}, \delta_{ext}, \lambda, D >$

$X = \{0,1,9\}$

$Y = \{0,1,9\}$

$S = \{0,1,9\}$

$\tau(N) = S$

$type = transport$

$d = 100$

τ = as defined in Coverage.ma

$\delta_{int}, \delta_{ext}, \lambda,$ and D will be according Cell − DEVS definitions.

$\boldsymbol{Plane\ 1}$:

$CD = < X, Y, S, N, type, d, \tau, \delta_{int}, \delta_{ext}, \lambda, D >$

$S = \{2,3,4,5,6\}$

$\tau(N) = S$

$type = transport$

$d = 100$

τ = as defined in Coverage.ma

$\delta_{int}, \delta_{ext}, \lambda,$ and D will be according Cell − DEVS definitions.

$\boldsymbol{Plane\ 2}$:

$CD = < X, Y, S, N, type, d, \tau, \delta_{int}, \delta_{ext}, \lambda, D >$

$S = \{25,26,27,28\}$

$\tau(N) = S$

$type = transport$

$d = 100$

τ = as defined in Coverage.ma

$\delta_{int}, \delta_{ext}, \lambda,$ and D will be according Cell $-$ DEVS definitions.

$\boldsymbol{Plane\ 3}$:

$CD = < X, Y, S, N, type, d, \tau, \delta_{int}, \delta_{ext}, \lambda, D >$

$S = \{10,11,12,13\}$

$\tau(N) = S$

$type = transport$

$d = 100$

τ = as defined in Coverage.ma

$\delta_{int}, \delta_{ext}, \lambda,$ and D will be according Cell $-$ DEVS definitiotns.

$\boldsymbol{Plane\ 4}$:

$CD = < X, Y, S, N, type, d, \tau, \delta_{int}, \delta_{ext}, \lambda, D >$

$S = \{20,21,22,23\}$

$\tau(N) = S$

$type = transport$

$d = 100$

τ = as defined in Coverage.ma

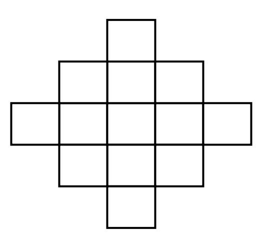
$\delta_{int}, \delta_{ext}, \lambda,$ and D will be according Cell $-$ DEVS definitions.

Formal specification of the **Coupled Cell-DEVS** model is as following:

$\boldsymbol{Plane\ 0,1}$:

$M = < X, Y, Xlist, Ylist, \eta, N, \{m, n\}, C, B, Z, select >$

$X = \{\phi\}$



$Y = \{\phi\}$

$Xlist = \{\phi\}$

$Ylist = \{\phi\}$

$\eta = 13$

$N = \{\ (-2,0,0), (-1,-1,0), (-1,0,0), (-1,1,0), (0,-2,0), (0,-1,0), (0,0,0), (0,1,0),$

$(0,2,0), (1,-1,0), (1,0,0), (1,1,0), (2,0,0) \}$

$m = 13$

$n = 13$

$C = is\ the\ cell\ space\ set, defined\ as\ C = \{C_{i,j}\ / i, j\ \in [1,13]\}$

$B = \{Not\ Wrapped\}$

$Z\ is\ the\ translation\ function$

$Select\ \{(-2,0,0), (-1,-1,0), (-1,0,0), (-1,1,0), (0,-2,0), (0,-1,0), (0,0,0), (0,1,0),$

$(0,2,0), (1,-1,0), (1,0,0), (1,1,0), (2,0,0) \}$

$\boldsymbol{Plane\ 2, 3, 4}$:

$M = < X, Y, Xlist, Ylist, \eta, N, \{m, n\}, C, B, Z, select >$

$X = \{\phi\}$

$Y = \{\phi\}$

$Xlist = \{\phi\}$

$Ylist = \{\phi\}$

$\eta = 13$

$N = \{ (0,0,0) \}$

$m = 13$

$n = 13$

$C = is\ the\ cell\ space\ set, defined\ as\ C = \{C_{i,j}\ / i, j\ \in [1,13]\}$

$B = \{Not\ Wrapped\}$

$Z\ is\ the\ translation\ function$

$Select\ \{(0,0,0)\}$

2.2. Functions and Rules

In this section rules and initialization of simulations have been discussed. There are comments beside most of rules which describe their functionality and these comment line starts with “%”. More detail about this area can be found in Coverage.ma file.

Basic definitions are as following:

```
[top]
components : coverage

[coverage]
type : cell
dim : (13,13,5)
delay : transport
defaultDelayTime : 100
border : nowrapped
neighbors :  . . .
initialvalue : 9
initialCellsValue : Coverage.val
localtransition : Existence              % {(0,0,0) .. (12,12,0) }
zone : Movement   { (0,0,1)..(12,12,1) }
zone : SigStrUE   { (0,0,2)..(12,12,2) }
zone : BSNum      { (0,0,3)..(12,12,3) }
zone : SigStr     { (0,0,4)..(12,12,4) }
```

Existence plane rules are:

```
[Existence]
rule : {randInt(1)} 100 { (0,0,0) = 9 }     %Just execute one time at first to put UEs in cells randomly.
rule : 0 100 { (0,0,0) = 1 and (0,0,3) = 10 }  %There is no coverage so UE is out of access.
rule : 1 100 { (0,0,0) = 1 and (0,0,1) = 6 }   %There is UE in this cell which wants to stay at this cell

%Go East
rule : 0 100 { (0,0,0) = 1 and (0,1,0) =  0 and (0,0,1) = 2 }

% Go South but also check that nobody from west of the destination cell does not want to move to
% destination cell. Because it will be that cell right to move first (according to priority).
rule : 0 100 { (0,0,0) = 1 and (1,0,0) =  0 and (0,0,1) = 3  and ((1,-1,0) = 0 or (1,-1,1) != 2) }

%Go West and check west and north side of destination cell.
rule : 0 100 { (0,0,0) = 1 and (0,-1,0) = 0 and (0,0,1) = 4  and ((0,-2,0) = 0 or (0,-2,1) != 2)   and ((-1,-1,0) = 0 or
(-1,-1,1) != 3) }

 %Go North and …..
rule : 0 100 { (0,0,0) = 1 and (-1,0,0) = 0 and (0,0,1) = 5  and ((-1,-1,0) = 0 or (-1,-1,1) != 2) and ((-2,0,0) = 0 or
(-2,0,1) != 3) and ((-1,1,0) = 0 or (-1,1,1) != 4) }

% If somebody wants to come and …
%Check cell has coverage. Check is there request from West, North, East or South to come to this cell.
rule : 1 100 { (0,0,0) = 0 and (0,0,3) != 10 and (((0,-1,0) = 1 and (0,-1,1) = 2) or ((-1,0,0) = 1 and (-1,0,1) = 3) or
((0,1,0) = 1 and (0,1,1) = 4) or ((1,0,0) = 1 and (1,0,1) = 5)) }

rule : {1} 100 {(0,0,0) = 1}  %When the rules reach here it means that Cell wants to move but it couldn’t

rule : {0} 100 {(0,0,0) = 0}  %Nothing happen for this Cell.
```

When a UE goes to cell without network coverage it will be discarded from mobile network and so in simulation. So if a UE travels to such cell we consider that it wants to leave the area and consequently omit it from rest of simulation.

Movement plane rules are:

```
[Existence]
rule : 9 100 { (0,0,2) = 10 }     %No need to calculate movement because no BS cover this cell.
rule : {randInt(4) + 2} 100 { (0,0,0) = 9 }  %For the first time after initialization
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 6 }   %UE wants to stay here so calculate next move

%UE wants to come to this cell from West, North, East or South. Calculate its next move.
rule : {randInt(4) + 2} 100 { (0,0,-1) = 0 and (0,-1,-1) = 1 and (0,-1,0) = 2 }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 0 and (-1,0,-1) = 1 and (-1,0,0) = 3 }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 0 and (0,1,-1) = 1  and (0,1,0) = 4  }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 0 and (1,0,-1) = 1  and (1,0,0) = 5  }

%UE in this cell wants to move but it won't be able because the destination cell is full right now. So UE will
stay here and we need to calculate next move again.
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 2 and (0,1,-1)  = 1 }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 3 and (1,0,-1)  = 1 }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 4 and (0,-1,-1) = 1 }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 5 and (-1,0,-1) = 1 }

%UE in this cell wants to move but it won't be able because the destination cell is not in range. So UE will
stay here and we need to calculate next move again.
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 2 and ((0,1,-1)  != 1 and (0,1,-1)  != 0) }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 3 and ((1,0,-1)  != 1 and (1,0,-1)  != 0)}
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 4 and ((0,-1,-1) != 1 and (0,-1,-1)  != 0)}
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 5 and ((-1,0,-1) != 1 and (-1,0,-1)  != 0)}

%UE in this cell wants to move but it won't be able because the destination cell will be occupied by another
UE with higher priority movement. So UE will stay here and we need to calculate next move again.
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 3 and (1,-1,-1) = 1  and (1,-1,0) = 2 }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 4 and ( ((0,-2,-1) = 1  and (0,-2,0) = 2) or ((-1,-1,-1) = 1
and (-1,-1,0) = 3)   ) }
rule : {randInt(4) + 2} 100 { (0,0,-1) = 1 and (0,0,0) = 5 and ( ((-1,-1,-1) = 1  and (-1,-1,0) = 2) or ((-2,0,-1) = 1
and (-2,0,0) = 3) or ((-1,1,-1) = 1  and (-1,1,0) = 4)  ) }

% if cell is not empty but it will be empty or cell is empty now and nobody wants to come here we do nt
need to calculate next movement. Put 8 on it.
rule : 8 100 { (0,0,-1) != 1 or (0,0,0) != 6 }
rule : 8 100 { (0,0,-1) != 0 or (0,-1,-1) != 1 or (0,-1,0) != 2 }
rule : 8 100 { (0,0,-1) != 0 or (-1,0,-1) != 1 or (-1,0,0) != 3 }
rule : 8 100 { (0,0,-1) != 0 or (0,1,-1) != 1  or (0,1,0) != 4  }
rule : 8 100 { (0,0,-1) != 0 or (1,0,-1) != 1  or (1,0,0) != 5  }
```

SigStrUE plane rules are:

```
[SigStrUE]
rule : 7 100 { (0,0,1) = 10 } %No need to calculate because no BS cover this cell. (Put there 7).

% if cell is empty and no ones from neighbors wants to come so there is no need to calculate SigStr.
rule : 7 100 { (0,0,-2) = 0 and (0,-1,-1) != 2 and (-1,0,-1) != 3 and (0,1,-1) != 4 and (1,0,-1) != 5 }

% if Cell is going to be empty there is no need to calculate SigStr.
rule : 7 100 { (0,0,-2) = 1 and (0,0,-1) = 2 and (0,1,-2) = 0 }
rule : 7 100 { (0,0,-2) = 1 and (0,0,-1) = 3 and (1,0,-2) = 0 and ((1,-1,-2) = 0 or (1,-1,-1) != 2) }
rule : 7 100 { (0,0,-2) = 1 and (0,0,-1) = 4 and (0,-1,-2) = 0 and ((0,-2,-2) = 0 or (0,-2,-1) != 2) and ((-1,-1,-2) = 0
or (-1,-1,-1) != 3) }
rule : 7 100 { (0,0,-2) = 1 and (0,0,-1) = 5 and (-1,0,-2) = 0 and ((-1,-1,-2) = 0 or (-1,-1,-1) != 2) and ((-2,0,-2) = 0
or (-2,0,-1) != 3) and ((-1,1,-2) = 0 or (-1,1,-1) != 4) }

% after those rules it means that somebody will be in this cell so calculate SigStr for it.
rule : 25 100 { (0,0,2) = 20 }
rule : 26 100 { (0,0,2) = 21 }
rule : 27 100 { (0,0,2) = 22 }
rule : 28 100 { (0,0,2) = 23 }
```

The cells in other two planes (plane 3 and 4) just refresh their previous values each time. Please note that in first two step different plane initialized them self so the precious result start from third step.

2.3. Modify Palette

Information about selected colors has been saved in Coverage.pal. Fig 5 shows the relation between numbers and colors.

Modify Palette

From	To	Color
0	1	
1	2	
2	3	
3	4	
4	5	
5	6	
6	7	
10	11	
11	12	
12	13	
13	14	
20	21	
21	22	
22	23	
23	24	
25	26	
26	27	
27	28	
28	29	

Fig 5: Selected colors

2.4. Sample picture of simulations

In this part there are pictures from first, second, third, fourth , a middle step, one step before last step and last step of a simulation.

Fig 6: First step

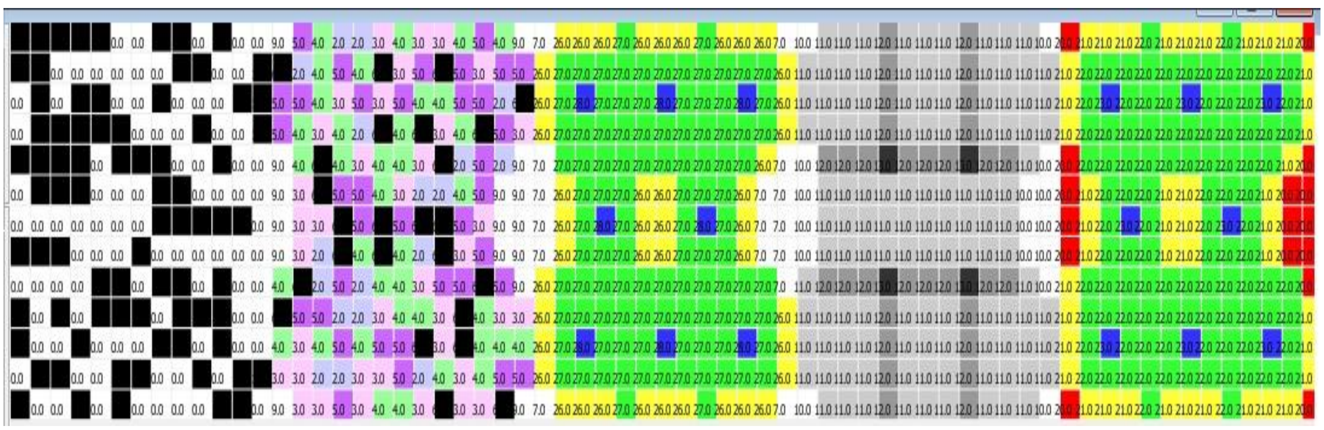


Fig 7: Second step

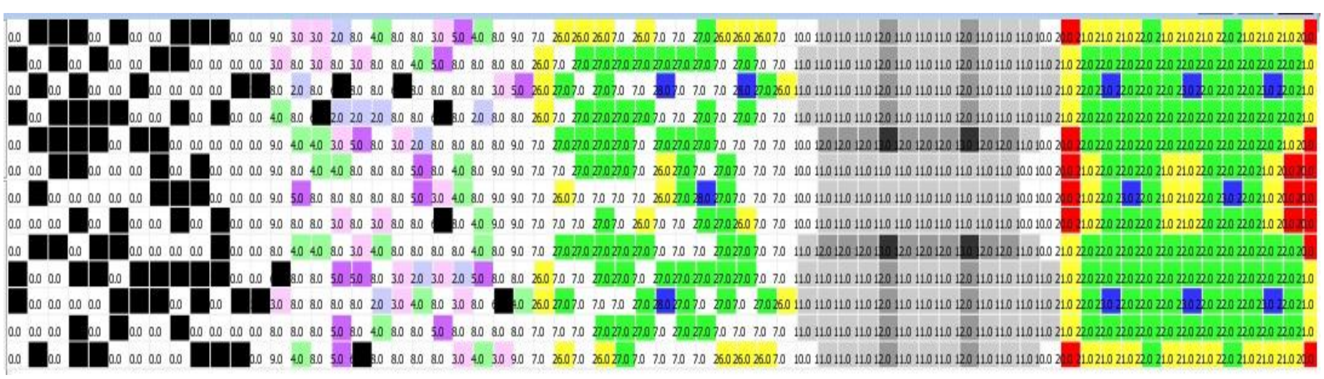


Fig 8: Third step

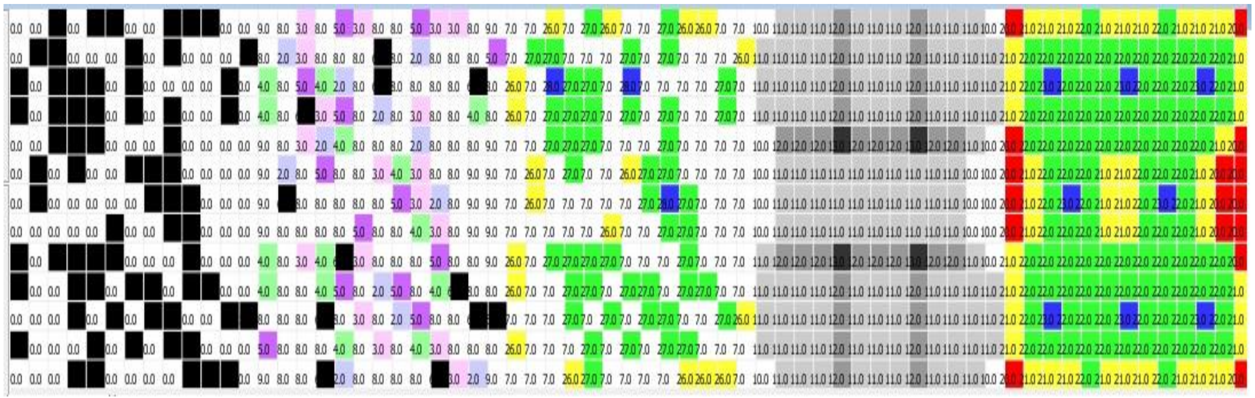


Fig 9: Fourth step

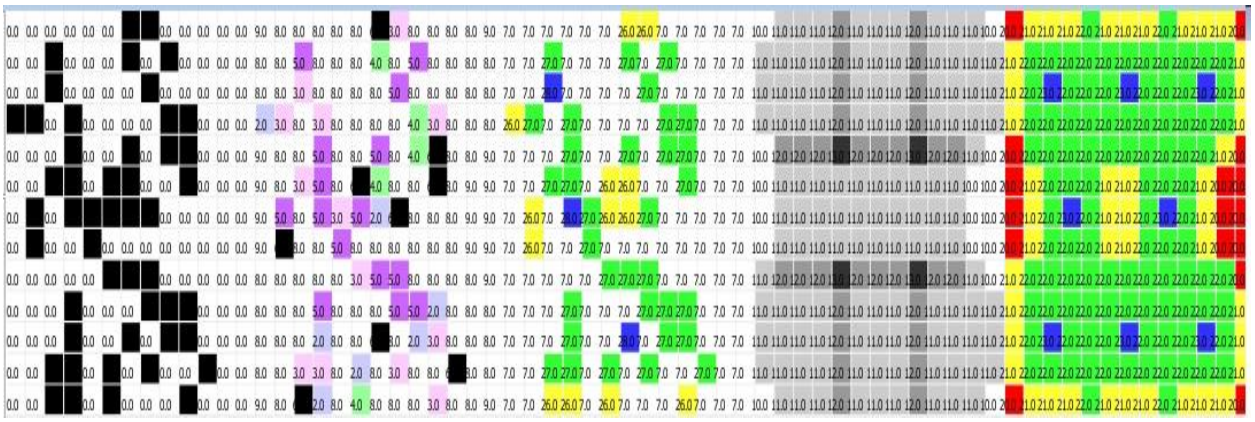


Fig 10: A middle step

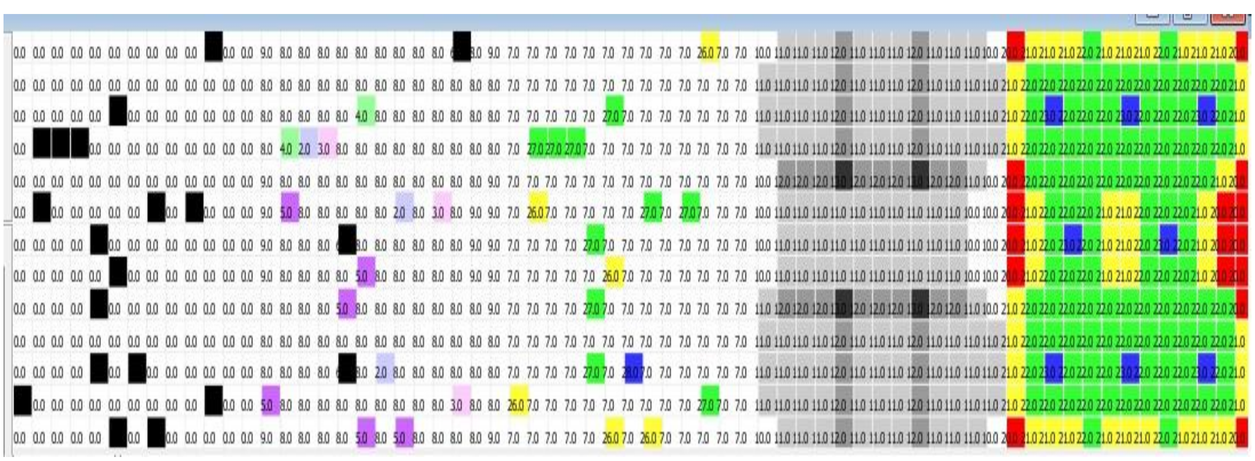


Fig 11: One step before the last step

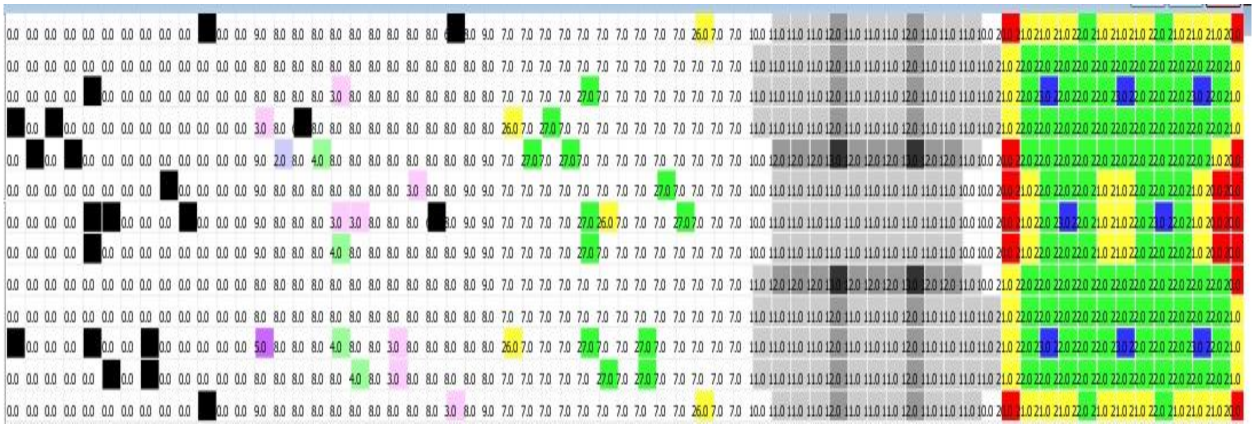


Fig 12: Last step