This can be obtained measuring the timing advance of cell (TA) then moving the other neighbour cells.
For moreinfo about TA: http://en.wikipedia.org/wiki/Timing_advance

the behaviour of the AT#EQCELL is as follow:

AT#EQCELL=0 //reset all
AT#EQCELL=1,<y>,<lev>

It means that you add <lev> in dBm from the measurement of the channel <y> so if you set <lev> to a negative value (e.g. -60) you decrease the real value of the cell of -60dBm. E.g. real value = -70dBm; <lev> = -60dBm; new value = -130dBm

so:

AT#EQCELL=1,<first cell ch>,-60
now you log on the second cell of the list ...

AT#EQCELL=1,<second cell ch>,-60
so you log on the third cell of the list
...
remember at the end of all to reset the system
AT#EQCELL=0

An example of algorithm is the following:

AT#MONI= 7
AT#MONI to detect the serving and neigh list

```
for (i = 0; i < 3; i++)
 {
    AT#EQCELL=1,freq(i),pwr          //With this command you change cell (see file.doc attached)
    //Send SMS/Call to calcolate TA

    AT#EQCELL=1,freq(i),pwr
    AT#MONI
    //Send SMS/Call to calcolate TA

    AT#EQCELL=1,freq(i),pwr
    AT#MONI
    //Send SMS/Call to calcolate TA
}
```

To measure TA value use the command AT#MONI. With AT#MONI=7 you can get all the parameters of the serving cell and neighbour cells. Moreover TA info is updated only during a call voice/data or SMS transmision/reception.

You can calculate TA for neighbour cells forcing the module to register to a new cell with the AT#EQCELL command. EQCELL is an hidden command that allows to equalize the cell power increasing or decresing the RXlev