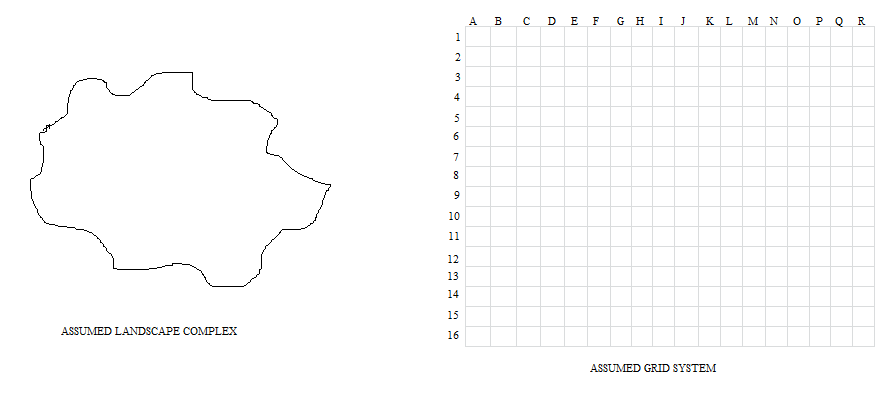
**Work Model**

We divide the work into few modules and try to achieve the goal for our work in the central Indian Landscape.

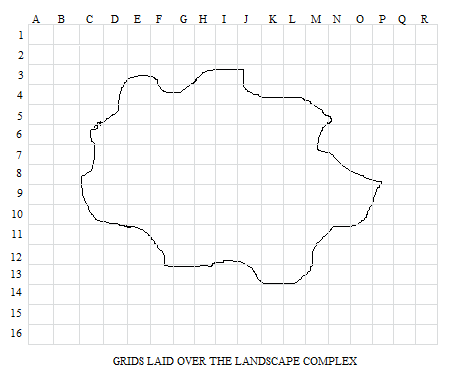
* Module 1: Setting up the grids over the focal landscape.
* Module 2: Finding the complete and partial occupancy matrix.
* Module 3: Area occupancy matrix over the grid system.
* Module 4: Parameter/Decision Variables judgement for the given landscape.
* Module 5: Payoff Calculation using the Evolutionary game model of Hawk and Dove.
* Module 6: Ranking of the grids.
* Module 7: Color Coding the cluster of rankings
* Module 8: Designing an Algorithm to identify habitat patches within the landscape.
* Module 9: Connectivity optimization based on results of Module 6, Module 7 and Module 8.

Let us consider that the following landscape needs to be studied under the assumed grid system:



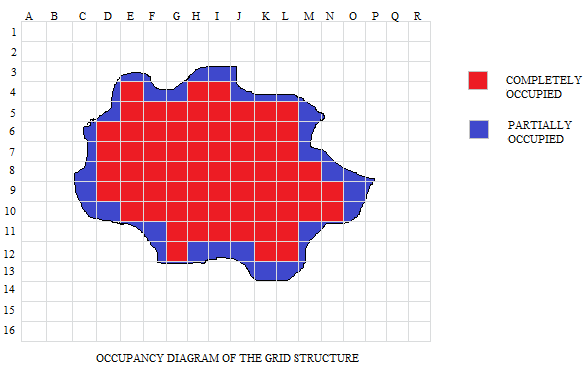
**Module 1: Setting up the grids over the focal landscape.**

We set the layer of the assumed landscape over the assumed grid system *(which has to be done with the actual data but here the word assumed is used as I do not have the actual datasets)*.



**Module 2: Finding the complete and partial occupancy matrix.**

We define in the form of a matrix all the grids with their ids which lie inside the landscape completely as well as partially. We do this in order to obtain the correct membership of the various features and their contributions to calculate the score of a grid.



**Module 3: Area occupancy matrix over the grid system.**

For the section of the area which intersects between the grid system and the focal landscape complex, we calculate the fraction occupancy on the scale of one.

This can be obtained using the “*identity”* function in ARCGIS.

For this module we use the following algorithm:

Algorithm Module\_3 (L, G)

//Input: 2D matrix of Grid.

2D matrix of focal Landscape

//Output: 2D matrix showing the occupancy matrix

n = no. of rows of L

m = no. of columns of L

p = no. of rows of G

q = no. of columns of G

for (i = 0, i < n, i++)

for (j = 0, j < m, i++)

for (k = 0, k < p, i++)

for (l = 0, l < q, i++)

if (L[i][j] ! = G[k][l]) then

T[p][q] = 0

Else if (L[i][j] = G[k][l]) then

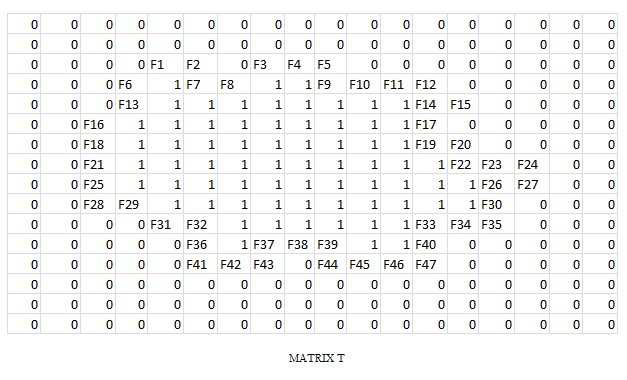
T[p][q] = 1

Else

T[p][q] = identity (L[i][j], G[k][l])

Return T.

Using the above algorithm, for the case taken by us, we obtain the following result.



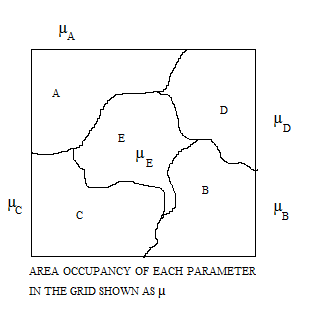
**Module 4: Parameter/Decision Variables judgement for the given landscape.**

We classify the parameters/decision variables which would act as the factors responsible for supporting or demoting the tiger population in the region.

For e.g. – Forest Cover, Prey Base, etc.

We also check the area occupancy of each parameter in each and every grid.

For our sample model let us consider 5 parameters: A, B, C, D and E and the area occupancy in each group obtained as the membership values for each parameter as the contribution in the total score of the grid.



For this module we use the following algorithm:

Algorithm Module\_4 (T, A, B, C, D, E)

//Input: 2D matrix of Obtained through module 3.

2D matrix of parameters in the focal Landscape

//Output: 2D matrix showing the area occupancy matrix in each grid (updating existing parameters matrix)

n = no. of rows of T

m = no. of columns of T

for (i = 0, i < n, i++)

for (j = 0, j < m, i++)

if (T[i][j] = 0) then

-------------------------------------

Else

A[i][j] = identity (T, A)

B[i][j] = identity (T, B)

C[i][j] = identity (T, C)

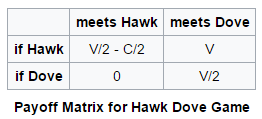
D[i][j] = identity (T, D)

E[i][j] = identity (T, E)

Return A, B, C, D, and E.

**Module 5: Payoff Calculation using the Evolutionary game model of Hawk and Dove.**

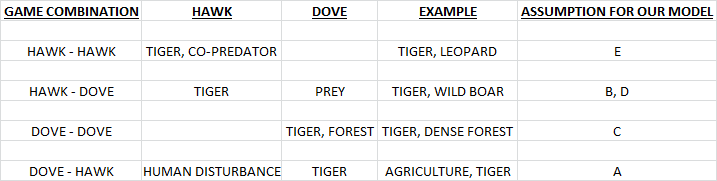
The game model we intend to use is the Hawk and Dove Evolutionary game. The contestants can be either Hawk or Dove. These are two subtypes or morphs of one species with different strategies. The Hawk first displays aggression, then escalates into a fight until it either wins or is injured (loses). The Dove first displays aggression, but if faced with major escalation runs for safety. If not faced with such escalation, the Dove attempts to share the resource.



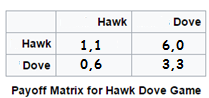
We need to classify the parameters as:

1. In which cases the tiger acts as a Hawk.
2. In which cases the tiger acts as a Dove.

For our case the following assumptions are made based on the following table:



So using the following payoff matrix we calculate the scores for each grid.



Algorithm Module\_5 (T, A, B, C, D, E)

//Input: 2D matrix of Grid.

2D matrix of updated parameters in the focal Landscape

//Output: 2D matrix showing the score matrix of grids S

n = no. of rows of T

m = no. of columns of T

for (i = 0, i < n, i++)

for (j = 0, j < m, i++)

S[i][j] = T[i][j]\*A[i][j]\* (-6) + T[i][j]\*B[i][j]\*6 + T[i][j]\*C[i][j]\*3 +

T[i][j]\*D[i][j]\*6 + T[i][j]\*E[i][j]\* 1

Return S.

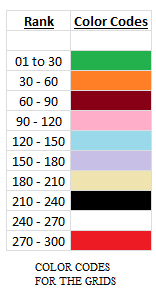
**Module 6 & 7: Ranking of the grids and Color Coding the cluster of rankings.**

We rank the grids based on the score matrix obtained from module 5. (S).

The ranking of the grids can be done in various ways, based on our suitability, like:

1. On the basis of class size.
2. On the basis of Centrality measures.
3. On the basis of ground data.
4. On the basis of any pre dominant factor, etc.

Here preferably we go through the Centrality measures and out of the present 288 grids for our model we rank them based on their respective scores and then dividing it into broad 10 categories decide the color code as the following:



Based on the above color codes let us suppose we run our following algorithm and then process the codes in our sample.

Algorithm Module\_6\_7 (S)

//Input: 2D matrix S

//Output: Color Coded grid system and color matrix

n = no. of rows of S

m = no. of columns of S

z = centrality index

for (i = 0, i < n, i++)

for (j = 0, j < m, i++)

t = max (S[i][j])

r = min (S[i][j])

sample = (t – r) / z

for (i = 0, i < n, i++)

for (j = 0, j < m, i++)

for (k = 0; k = t)

R[i][j] = count (sample multiplication)

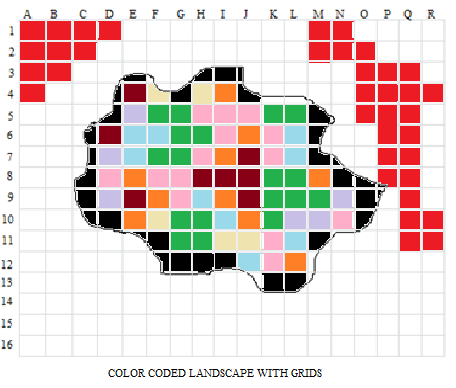
for (i = 0, i < n, i++)

for (j = 0, j < m, i++)

for (k = 0; k = t)

R[i][j] = color\_fill (sample multiplication)

Return R.



**Module 8: Designing an Algorithm to identify habitat patches within the landscape.**

The habitat patches in the focal landscape will be found out using the following algorithm:

Algorithm Module\_8 (R, z, a, b)

//Input: 2D matrix R

//Output: Color Coded patches in the grid system.

n = no. of rows of R

m = no. of columns of R

z = Best ranked grid color (Dark Green for our case)

R[a][b] = position of first z

If R[a][b] = partial grid then

STOP and START NEW SEARCH FOR z

If R[a][b] = 1 then

Record = R[a][b]

Module\_8 (R, z, a+1, b)

Module\_8 (R, z, a, b+1)

Module\_8 (R, z, a-1, b+1)

Module\_8 (R, z, a+1, b-1)

Module\_8 (R, z, a+1, b+1)

Module\_8 (R, z, a-1, b-1)

Module\_8 (R, z, a-1, b)

Module\_8 (R, z, a, b-1)

Return Record.