**Modelling:**

For the purpose of the present work, we assume that the wild tiger habitat patches in the landscape complex constitute the vertices and the collection of connections within these patches constitute the edges, comprising the focal landscape complex as a graph.The existence of an edge between any two vertices represents some population flux between the adjacent vertices.

An occupancy matrix is constructed using the ‘identity’ function of ARCGIS algorithm to indicate the cost incurred by the tiger for passage between the habitat patches in the landscape. Analysis of the graph is done using hawk and dove game theory algorithm, graph theory and minimum spanning tree (Kruskal) algorithm, in order to identify and focus on potentially important habitat patches, their potential community structure, and the possible movement patterns and the mobility rate of the tigers in between these patches.

Correlation analysis is performed on the centrality indices to draw out interesting trends in the network.

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

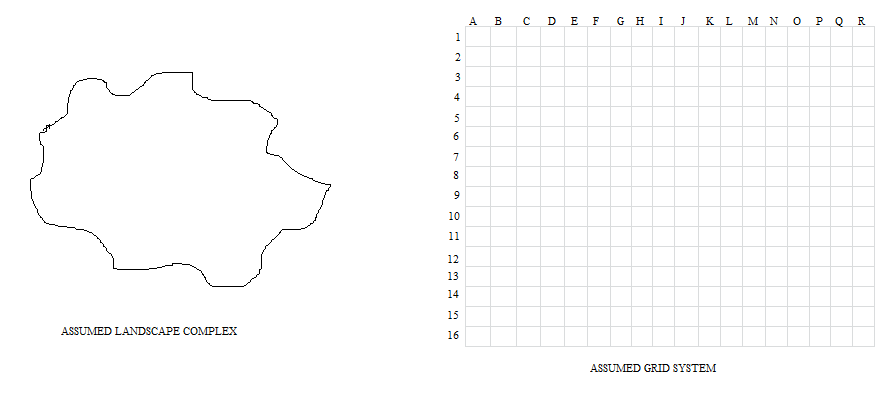


Fig 1: A diagrammatic representation of the landscape needs to be studied under the assumed grid system.

Firstly a landscape has been assumed over which the grids need to be projected as represented in Figure 1. The grid distribution system is comprehended as latitudes and longitudes. Latitudinal rows are symbolized by numbers. Longitudinal columns are symbolized by alphabets. The focal landscape (preferably the central Indian landscape region) is then divided into smaller habitat patches by setting up the grid over the landscape as represented by Figure 2.

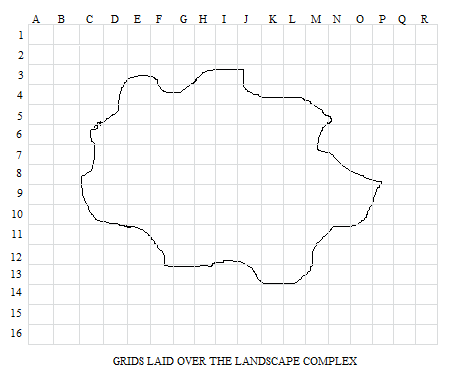


Fig.2: A diagrammatic representation of the set up landscape.

Module 2: Finding the complete and partial occupancy matrix.

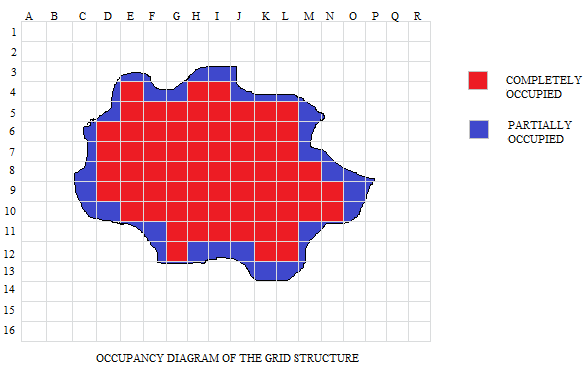


Fig 3. A diagrammatic representation of the landscape with the partial and maximum occupancy.

In Figure 3. We define the partial and complete occupancy of the landscape projected on the grids with their defined ids, if they lie inside the landscape. This has been done 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.

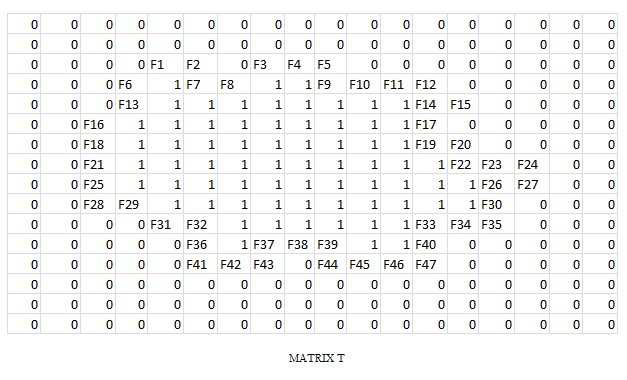


Fig. 4: A diagrammatic representation of the occupancy area of the matrix with their id.

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.

ARCGIS “identity” predicate is a default function required by all rasters in a mosaic dataset if there is no other suitable function. It is used to define the source raster as part of the default tiling behavior of the mosaic dataset and computes a geometric intersection of the input features and identity features. The input features or portions thereof that overlap identity features will get the attributes of those identity features.

The processing is done incrementally on subdivisions of the original landscape. Grids that straddle the edges of these subdivisions (also called tiles) are split at the edge of the tile and reassembled into a single grid during the processing. The vertices introduced at these tile edges will remain in the output features. Tile boundaries can be excluded in the output matrix when being processed.

The output of the algorithm will be an id coded matrix with labelling them:

* “0” for outside landscape
* “1” for inside the landscape
* “identity” for boundary of the landscape

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.

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

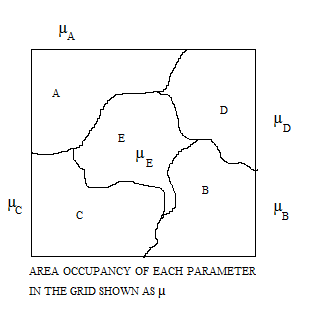


Fig. 5 A diagrammatic representation of the decision variables affecting the different areas within the landscape.

Decision variables are those parameters or the factors that elucidate the governing criteria for the persistence and the intensity of the supporting and demoting rationales of metapopulation strength for a region. The factors for each region may vary in a temporal manner and may range from the parameter brackets of habitat improvement, land acquisition, forest cover, prey base and other coordination activities.

We also check the area occupancy of each parameter in each and every grid. This is done by a linear programming model of the following constitution:

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.

For our sample model we have considered the 5 different 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.

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

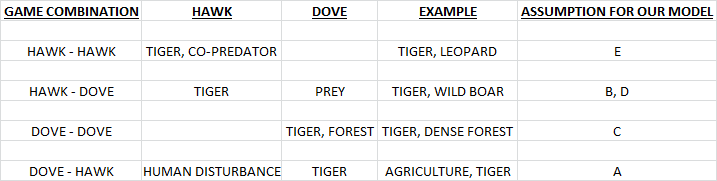
The game model we intend to use is the Hawk and Dove Evolutionary game. The contestants can be either Hawk or Dove. These are two subtypes or morphs of one species with different strategies. The Hawk first displays aggression, then escalates into a fight until it either wins or is injured (loses). The 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 are using the concept of hawk and dove game theory because it gives a better payoff for the quantum games using a random strategy and maximum payoff for pure strategy. It uses both pareto optimality and nash equilibrium concept to maximize the payoff. It removes the local correlations where both the players are unaware of the fact that an entangled state has been distributed amongst them.

In our model, we need to classify the players acting as hawk or dove as these are two different types of the same species with different outlook and strategies.

Conditions of the game to be played as:

* Hawk being aggressive
* Dove trying to share the available resources.



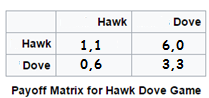
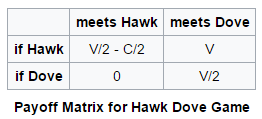


Fig. 6: A tabular representation of the hawk and dove game theory.

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: Ranking of the grids and Color Coding the cluster of rankings.

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 given in Figure 6.

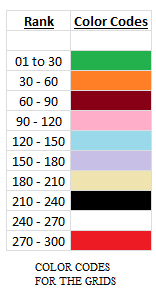
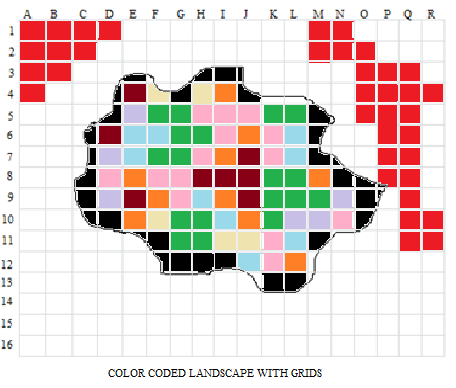
 

Fig. 7: A diagrammatic representation of the color coded patches of the original land scape and the rank matrix of each color.

Flood fill [algorithm](https://en.wikipedia.org/wiki/Algorithm) determines the area [connected](https://en.wikipedia.org/wiki/Glossary_of_graph_theory#Connectivity) to a given node in a multi-dimensional [array](https://en.wikipedia.org/wiki/Array_data_structure). It is used in the "bucket" fill tool to fill connected, similarly-colored areas with a different color for determining which pieces are distinguishable.  The algorithm looks for all nodes in the array that are connected to the start node by a path of the target color and changes them to the replacement color. Thereafter ranking of the grids are done on the basis of the color fill as shown in Figure 7. Based on the above color codes 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.

