1 Introduction

An undirected network with all the eigenvectors associated to an adjacency matrix \( A \) can say to be most localized when a wave-like information, given as an input to a node in the network, is localized or trapped to that node only \[1\]. In other words, information does not propagate through the network. For instance, if we consider a network where each node is isolated and has only self-loops, the corresponding adjacency matrix will have all its eigenvectors with one entry taking value 1 whereas rest of the entries taking value zero, such as \([1, 0, \ldots, 0]\). An eigenvector having such entries is referred to as the most localized eigenvector. Similarly, an eigenvector represented by \([1/\sqrt{N}, 1/\sqrt{N}, \ldots, 1/\sqrt{N}]\) is said to be a completely delocalized eigenvector. These are two extreme cases for the eigenvector localization. A detailed study on the eigenvector localization and different measurements of localization can be found in Refs. \[2, 3\].

Furthermore, we can think about the eigenvector localization from a different point of view. We can define a standard basis, i.e. \(\{e_1, e_2, \ldots, e_N\}\) in the \(N\) dimensional real vector space where \(e_i \in \mathbb{R}^{N \times 1}\), with \(i^{th}\) component being equal to 1 and all others being zero. If any eigenvector \(X_j \in \mathbb{R}^{N \times 1}\), \(j \in \{1, 2, \ldots, N\}\) corresponding to \(A \in \mathbb{R}^{N \times N}\) is aligned with any standard basis vector \((e_i)\), we can say that this eigenvector is the most localized one. However, the current investigation considers simple (without self-loop and multiple edges), connected, undirected, and unweighted networks. Therefore, as we know, from the Perron-Frobenius theorem \[4\], that all the entries of the principal eigenvector (PEV) are positive and it is not possible to achieve an eigenvector which is the most localized. To measure eigenvector localization, we use inverse participation ratio (IPR) value. We focus on studying the localization properties of the largest (principle) eigenvalue and its corresponding eigenvector viz. principal eigenvector and whenever necessary we mention separately about other eigenvectors.

For a particular value of \(N\) (number of nodes) and \(M\) (number of connections), if we can enumerate all the possible network configurations, the network corresponding to the maximum IPR value will be our desired network. The number of possible network configurations for a given \(N\) and \(M\) is of the order \(O(N^{2M})\) \[5\]. Therefore, we formulate this problem through an optimization technique as follows. Given an input graph \(G\) with \(N\) vertices, \(M\) edges and a function \(\zeta: \mathbb{R}^{N \times 1} \rightarrow \mathbb{R}\), we wish to attain the maximum possible value of \(\zeta(G)\) over all the simple, connected, undirected, and unweighted graph \(G\). Thus, we maximize the objective function \(\zeta(G) = IPR(X_1) = x_1^4 + \ldots + x_N^4\) subject to the constraints that \(x_1^2 + \ldots + x_N^2 = 1\), and \(0 < x_i < 1\).

We refer the initial network as \(G_{\text{init}}\) and the optimized network as \(G_{\text{opt}}\). In the current study, our aim is to get a network having maximum possible IPR value for PEV for a given network size. To the best of our knowledge this is the first piece of work which attempts to construct a network structure having a very high localized PEV for the given network parameters.
We organize the supplementary materials as follows: Section 2 describes the notations and definitions used in the later discussion. In addition, it contains a brief explanation of the optimization procedure used in our work. Section 3 illustrates various numerical results including degree and eigenvector entry distribution of the initial as well as of the optimized network. Moreover, this section exhibits the results for the initial network taken as an SF network. Section 4 consists of analytical derivations of the changes in the IPR value as a function of edge rewiring for discrete as well as continuous cases. Finally, in section 5, we summarize the current study and discuss the open problems for further investigations.

![Cytoscape image from numerical data](image1)

**Fig. S 1**: Cytoscape image from numerical data; Network at (a) intermediate stage of the optimization process (b) the optimized network.

### 2 Methods

We represent our network as \( G = \{V, E\} \), for the set of nodes \( V = \{v_1, v_2, \ldots v_N\} \) and edges \( E = \{e_1, e_2, \ldots e_M\} \), where \( N \) and \( M \) denote the size of \( V \) and \( E \), respectively. We refer \( E^c = \{e_1^c, e_2^c, \ldots, e_{N(N-1)/2-M}^c\} \) as the set of edges which are not present in \( G \). The adjacency matrix corresponding to \( G \) is defined as,

\[
a_{ij} = \begin{cases} 
1 & \text{if } i \sim j \\
0 & \text{Otherwise}
\end{cases}
\]

Further, \( d_i = \sum_{j=1}^{N} a_{ij} \) denotes the degree of node \( v_i \) and \( \{d_i\}_{i=1}^{N} \) denotes the degree sequence of \( G \). We refer \( d \) to be the degree vector and \( \bar{d} = \frac{d}{\sqrt{\sum d_i^2}} \) as the normalized degree vector [6]. The spectrum of \( G \) is a set of eigenvalues \( \{\lambda_1, \lambda_2, \ldots, \lambda_N\} \) of \( A \) where \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \) and corresponding eigenvectors are \( X_1, X_2, \ldots, X_N \), respectively. The new network and the corresponding adjacency matrix after a single edge rewiring can be denoted as \( G' \) and \( A' \), respectively. The eigenvalues and eigenvectors of \( A' \) are denoted as \( \{\lambda'_1, \lambda'_2, \ldots, \lambda'_N\} \) and \( \{X'_1, X'_2, \ldots, X'_N\} \) respectively. We quantify the eigenvector localization using the IPR [7] as follows,

\[
IPR(X_k) = \sum_{i=1}^{N} x_i^4
\]

\[
\sum_{i=1}^{N} x_i^2 = 1
\]

where \( x_i \) is the \( i^{th} \) component of the eigenvector \( X_k \) for \( i, k \in \{1, 2, \ldots, N\} \). Here, \( X_k \)'s are normalized eigenvector in the Euclidean norm.
Algorithm 1: IPR-Optimization

1. \( \text{ipr} \leftarrow \text{IPR}(X_1) \) of \( A \)

2. \textbf{while} IPR\((X_1)\) not optimized \textbf{do}

3.   rewire an edge uniformly and independently at random in \( G \) and denotes the new

4.   check for connectedness of \( G' \)

5.   \( \text{ipr}' \leftarrow \text{IPR}(X_1') \) of \( A' \)

6.   \textbf{if} \( \text{ipr}' > \text{ipr} \) \textbf{then}

7.     \( A \leftarrow A' \)

8.     \( \text{ipr} \leftarrow \text{ipr}' \)

9. \textbf{end}

10. store \( \text{ipr} \), \( \text{ipr}' \), and \( A \), separately

11. \textbf{end}

\[ 
\begin{matrix}
\begin{array}{c}
\text{(a) ER network as initial network}
\\
\text{(b) SF network as initial network.}
\end{array}
\end{matrix}
\]

Fig. S 2: Degree distribution of (i) initial network (ii) optimized network; PEV entry distribution
of (iii) initial network (iv) optimized network.

The Monte Carlo based optimization process can be summarized as follows. We find PEV
\((X_1)\), of the adjacency matrix, \( A \) of a graph \( G \) and calculate the IPR value of \( X_1 \). We rewire one
edge uniformly and independently at random in \( G \) to obtain another graph \( G' \). We check whether
\( G' \) is connected, if not the edge rewiring step is repeated till we get another \( G' \) which is a connected
network. We find out PEV of \( A' \) matrix and calculate the IPR value of \( X_1' \). We replace \( A \) with
\( A' \) if \( IPR(X_1') > IPR(X_1) \). Steps from third to ten are repeated until IPR value gets saturated
which corresponds to the optimized network. We remark that to check the connectedness during
an edge rewiring, we use depth-first search (DFS) algorithm \[8\]. The recorded value of \( \text{ipr} \) variable
during the optimization process gives the increment of IPR value which is depicted in Fig 1(a)
and Fig. S 3(a)(i). Whereas, the recorded value of \( \text{ipr}' \) variable during the evolution gives the
rewiring of all the edges including drops of the IPR value which is depicted in Fig 2 and Fig.
S 3(b). We depict a network at an intermediate evolution stage and the final optimized one in
Fig. S 1.

We consider two different network models to create the initial network. One is Erdős-Rényi(ER)
model and another is the scale-free (SF) model \[9\]. The ER random network is generated for edge
probability \( p = \langle k \rangle / N \), where \( \langle k \rangle \) is the average degree of the network. Therefore, in each realization,
the total number of edges will fluctuate around \( N \langle k \rangle / 2 \), however it does not affect the properties of
the final networks evolved through the optimization process. To generate an SF network, we use
the Barabasi-Albert model \[10\].
Table S 1: (a) Results are shown for the average over 31 realizations with 6,00,000 edge rewiring steps. \(ER_{opt}\) denotes the optimized network achieved through the network evolution from the initial base network being \(ER\) network. (b) \(ER_{conf}^{opt}\) refers to the network constructed from the configuration model having the same degree sequence as of the \(ER_{opt}\) network. \(ER_{cc}^{opt}\) refers to the network constructed with the same \(\langle CC \rangle\) and degree sequence as in the \(ER_{opt}\) network. Finally, \(ER_{deg-deg}^{opt}\) refers to the network having the degree-degree correlation as in the \(ER_{opt}\) network.

3 Numerical Results

We have analyzed the degree distribution and PEV entries distribution of the initial as well as optimized networks. During the evolution process, it is observed that there is a drastic change in the degree distribution and distribution of the PEV entries. Moreover, irrespective of whether we start the network optimization process taking \(ER\) random network or SF network as an initial network, finally we reach to the network having the same heterogeneous degree distribution (Fig. S 2).

We summarize the ensemble average results for the initial network taken as an \(ER\) random network and the optimized network (\(ER_{opt}\)) in the Table S 1(a). To check the plausible correlation between the degree sequence of the final optimized network and PEV localization, we construct a network which has the same degree sequence as of the optimized network. For this purpose, we use the configuration model [11], which we denote as \(ER_{conf}^{opt}\). We observe that \(ER_{conf}^{opt}\) has an IPR value which is much lesser than that of the \(ER_{opt}\) network (Table S 1(b)). It concludes that having a specific degree distribution is important but is not the only factor for the PEV localization. Nevertheless, there exist other structural properties which are acquired by the evolved network during the optimization process. In the following, we discuss the impact of other structural properties on the localization of PEV.

Further, we find the clustering coefficient [12] of a node as,

\[
C_i = \frac{2\Delta_i}{d_i(d_i - 1)}
\]

where \(d_i\) denotes the degree of node \(i\), and \(\Delta_i\) is the number of triangles the node \(i\) is participating in. Subsequently, the average clustering coefficient [12] can be defined as,

\[
\langle CC \rangle = \frac{1}{N} \sum_{i=1}^{N} C_i
\]

During the optimization process, we keep a record of the average clustering coefficient \(\langle CC \rangle\) which indicates an increase in \(\langle CC \rangle\). To check a correlation between \(\langle CC \rangle\) and IPR, we use an algorithm given in Ref. [13] which takes a degree sequence and average clustering coefficient as an input and creates a random network having the exactly same degree sequence and \(\langle CC \rangle\) as of \(ER_{opt}\). We denote the network generated using this algorithm as \(ER_{cc}^{opt}\). It is interesting to observe that
Fig. S 3: (a) Changes of various network properties during the network evolution when initial network is taken as SF network with $N = 500$, $\langle k \rangle = 10$. We perform the rewiring process for 2,00,000 edge rewiring steps and store the network after each 100th step. Saturated value of IPR fluctuate as $(0.2 \pm 0.007)$. (b) In the optimized network, there exists few edges rewiring which lead to a sudden drop in the IPR value.

though $ER_{\text{opt}}$ and $ER_{\text{opt}}^{cc}$ have the same degree sequence and $\langle CC \rangle$, they are having different IPR value (Table S1(b)). It asserts that by tuning $\langle CC \rangle$, we cannot achieve a network structure corresponding to $ER_{\text{opt}}$ network. Further, we track the degree-degree correlation coefficient ($r_{\text{deg-deg}}$) during the evolution process. The $r_{\text{deg-deg}}$ is measured as follows [14],

$$\begin{align*}
    r_{\text{deg-deg}} &= \frac{\sum_{i=1}^{M} j_i k_i - \sum_{i=1}^{M} \frac{1}{2} (j_i + k_i)^2}{\sum_{i=1}^{M} \frac{1}{2} (j_i^2 + k_i^2) - \sum_{i=1}^{M} \frac{1}{2} (j_i + k_i)^2} \\
    M &= \text{the total number of edges in the network and } j_i, k_i \text{ are the degrees of nodes with } i^{th} \text{ connection. As a consequence of the evolution process, the } r_{\text{deg-deg}} \text{ of the optimized network becomes negative indicating the presence of disassortativity in the optimized network. To check how evolution of disassortativity has an impact on the enhancement in the IPR value, we use Sokolov algorithm [15] to construct a network with the same number of nodes with the same IPR pattern as observed for the initial network taken as ER random networks. In fact, on separating the two components of the optimized network, these do not manifest localized PEV individually. However, these two components taken as $G_{\text{init}}$ separately, the evolution leads to two optimized networks which have the same structure as in (Fig. 3(right)). However, to achieve a better normalized PEV, the network should be sparse. For the dense network, the evolution process}
\end{align*}$$

Further, we measure the correlation between the clustering coefficient sequence and the PEV entries, using Pearson product-moment correlation coefficient [16]. We measure correlation coefficient between the degree vector and clustering coefficient (cc) vector denoted as $r_{\text{deg-cc}}$ and the correlation between the PEV entries and the cc vector denoted as $r_{\text{pev-cc}}$.

$$\begin{align*}
    r &= \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2}} \\
    \text{To calculate } r_{\text{deg-cc}} \text{ and } r_{\text{pev-cc}}, \text{ we normalize the degree vector in the Euclidean norm as in [6]. These measures provide an insight to the network structure of the most optimized network.}
\end{align*}$$

In the main article, we have discussed the results of various network properties during the evolution process with an initial network being taken as ER random network. Additionally, herein we consider SF network as an initial network and describe the result in Fig. S3(a) for changes in various network properties during the evolution with $N = 500$ and $\langle k \rangle = 10$. Fig. S3(b) exhibits the same IPR pattern as observed for the initial network taken as ER random networks. In fact, if we separate the two components of the optimized network, these do not manifest localized PEV individually. However, these two components taken as $G_{\text{init}}$ separately, the evolution leads to two optimized networks which have the same structure as in (Fig. 3(right)). However, to achieve a better localized PEV, the network should be sparse. For the dense network, the evolution process
does not lead to a significant increase in the IPR value. The IPR value saturates for a dense network without being appreciably different from the initial network (Fig. 6(b)(i)).

Fig. S 4: Colormap of eigenvector entries during the network evolution of 50,000 iterations. The black color line near 400th entry (y-axis) represents the maximum PEV entry weight corresponding to the hub node.

Fig. S 5: Sorted PEV for (i) and (iv) initial ER random and SF network; (ii) and (v) optimized network; (iii) and (vi) after removing an edge connected to hub node.

Furthermore, we scan the eigenvector entries and they provide interesting insight into the final network structure. We depict the changes of the PEV entries during the network evolution using a color map (Fig. S 4). The pictorial representation describes the changes in the PEV entries and formation of the hub node during the evolution. We can see that during the evolution, formation of the hub node happens much before the IPR value gets saturated. Further, we check the sorted PEV entries of $G_{\text{init}}$, $G_{\text{opt}}$ and $G_{\text{opt} - \text{edge}}$ whereas $G_{\text{opt} - \text{edge}}$ is the network obtained after removal of an edge connected to the hub node. After completion of the optimization process, we find a network which has (1) one PEV entry with very high value (2) there exist few relatively smaller entries and, (3) there exists a large chunk of nodes with very small entries (Fig. S 5(ii)). Furthermore, the rewiring which leads to a drop in the IPR value causes a large changes in the PEV entries (Fig. S 5(iii)). The optimized network (localized PEV) and the network obtained after single rewiring (delocalized PEV) differ in only one edge. The sorted PEV corresponding to the initial SF network which has the same pattern as reflects in Fig. S 5(b).

The main article contains the numerical results for several realizations of ER random network with $N = 500$ and $\langle k \rangle = 10$. Here, we provide the results for changes in the IPR values with
4 Analytical Derivation

In this section, we present the analytical derivation of the changes in the IPR value as edges are rewired. Our aim is to investigate the changes in the IPR value of a network with respect to an edge rewiring and find out a condition which leads to an increase in the IPR value. Each edge rewiring is a two-step process, (i) removal of an edge followed by (ii) addition of an edge (Fig. S7). Hence, the removal of an edge causes changes in two entries in the adjacency matrix from 1 to 0, and later edge addition changes another two entries from 0 to 1. Thus each edge rewiring changes four entries of the adjacency matrix. Therefore, we consider each $a_{ij}$ as a discrete variable, however, to make our analysis for a broader class of problems, instead of removing and addition of an edge, we propose to reduce the edge weight by some amount followed by enhancement of similar amount of weight to another edge. For this case, $a_{ij}$ will not be a discrete variable, rather it will be a continuous variable. First, we consider the case for $a_{ij}$ being discrete variable and then we consider the continuous case.

4.1 Derivation for discrete matrix elements

In this section, we analyze how a single edge rewiring lead to a changes in the IPR value. For all the subsequent analysis we consider the symmetric matrix $A$, with $a_{ij} \in \{0, 1\}$. Given a matrix at $t^{th}$ step as $A^t$, we refer the matrix after a single edge rewiring as $A^{t+1}$, $t = 0, 1, \ldots, T$, where

Fig. S 6: Initial network is ER random network; (■) and (●) refer values before and after optimization, respectively. (a) Depicts the results for fixed $\langle k \rangle = 10$ with different values of $N$. (b) Plots for fixed $N = 500$ with different $\langle k \rangle$. Furthermore, we observe that as the average degree of the network increases with a fixed $N$, IPR value of the final optimized network decreases (Fig. S6(b)).
$T$ is the total number of edge rewiring during the network evolution or total time step for which the evolution happens. We are interested in the changes of IPR values and hence the eigenvector entries due to a single edge rewiring. We consider four distinct arbitrary nodes $p$, $q$, $r$ and $s$ such that there exists an edge between $p$ and $q$ and no edge between $r$ and $s$. Therefore, we can represent each edge rewiring process as $A^t \xrightarrow{\Delta a_{pq}} A' \xrightarrow{\Delta a_{rs}} A^{t+1}$, where $A'$ is the intermediate stage after an edge removal and $\Delta a_{ij}$ is the amount of changes caused by a single edge rewiring. Note that during the rewiring, entries corresponding to $p$, $q$, $r$ and $s$ nodes will change whereas all other entries of $A^{t+1}$ matrix will remain unchanged. Additionally, in our case,

$$\Delta a_{ij} = \begin{cases} 
-1 & \text{for edge removal} \\
1 & \text{for edge addition} 
\end{cases}$$

But each step will lead to a change in the eigenvalues $\lambda$ and the corresponding eigenvector entries, $x_i$’s. The eigenvalues and eigenvectors of $A_t$ are denoted as $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ and $\{X_1, X_2, \ldots, X_N\}$, where $x_i$ is the $i^{th}$ component of the eigenvector $X_k$ for $k, i \in \{1, 2, \ldots, N\}$. We do not make any assumption in the derivation for a particular eigenvector and the corresponding eigenvalue. Hence, without the loss of generality, we drop $X_k$ from the IPR notation in Eq. 1 for the rest of the analysis.

Hence, removing an edge lead to a change in two entries of $A_t$ and produce a new matrix $A'_t$. We can denote the amount of changes of the eigenvector entries as $\Delta_0 x_i$, changes in eigenvalue as $\Delta_0 \lambda_i$ and changes in IPR as $\Delta_0 IPR_i$. Hence, changes in $A_t$ will reflect as $x_i = x_i + \Delta_0 x_i$, $\lambda_i = \lambda_i + \Delta_0 \lambda_i$ and $IPR = IPR + \Delta_0 IPR$ respectively. Subsequently after adding an edge in $A_t$ produces $A^{t+1}$, and we get another changes as $\Delta_1 x_i''$, $\Delta_1 \lambda_i''$ and $\Delta_1 IPR''$. Hence, the eigenvector entries of $A^{t+1}$ will be $x_i'' = x_i + \Delta_1 x_i''$, similarly $\lambda_i'' = \lambda_i + \Delta_1 \lambda_i''$ and $IPR'' = IPR + \Delta_1 IPR''$. Therefore after the rewiring we have,

$$x_i^{t+1} = x_i + \Delta x_i^{t+1}, \text{ where } \Delta x_i^{t+1} = \Delta_0 x_i + \Delta_1 x_i''$$

$$\lambda_i^{t+1} = \lambda_i + \Delta \lambda_i^{t+1}, \text{ where } \Delta \lambda_i^{t+1} = \Delta_0 \lambda_i + \Delta_1 \lambda_i''$$

$$\begin{align*}
IPR^{t+1} &= IPR^t + \Delta IPR^{t+1}, \text{ where } \Delta IPR^{t+1} = \Delta_0 IPR^t + \Delta_1 IPR'' \\
a_{ij}^{t+1} &= a_{ij} + \Delta a_{ij}^{t+1}
\end{align*}$$

Note that perturbation theory has been used to measure changes in the eigenvalue and eigenvector of the adjacency matrix of the network with respect to the edge removal and addition \[17\]. We use the above formulation which gives a way for the subsequent analysis. Our aim is to track the changes in IPR value of a network as edges are rewired and to find out the condition which leads to an increase in the IPR values. The IPR function is given as:

$$IPR = \sum_{i=1}^{N} (x_i)^4$$

Hence, using Eq. 4 we get,

$$\Delta IPR^{t+1} = IPR^{t+1} - IPR^t$$

$$= \sum_{i=1}^{N} (x_i^{t+1})^4 - \sum_{i=1}^{N} (x_i)^4$$

$$= \sum_{i=1}^{N} (x_i + \Delta x_i^{t+1})^4 - \sum_{i=1}^{N} (x_i)^4$$

$$= \sum_{i=1}^{N} [4(x_i)^3 \Delta x_i^{t+1} + 6(x_i)^2 (\Delta x_i^{t+1})^2 + 4x_i (\Delta x_i^{t+1})^3 + (\Delta x_i^{t+1})^4]$$

$$= \sum_{i=1}^{N} [4(x_i)^3 \Delta x_i^{t+1} + 6(x_i)^2 (\Delta x_i^{t+1})^2 + 4x_i (\Delta x_i^{t+1})^3 + (\Delta x_i^{t+1})^4]$$
Therefore, any changes in the IPR value (Eq. 5) is governed by the changes in the eigenvector entries. We first measure changes in the eigenvector entries and for that purpose we use the eigenvalue equation as follows,

\[ AX = \lambda X \]

At \( t^{th} \) iterations, we can write using the above equation as follows

\[ \sum_{j=1}^{N} a_{ij}^t x_j^t = \lambda^t x_i^t, \ \forall \ i = 1, 2, \ldots, N \]  \hspace{1cm} (6)

Similarly at \((t + 1)^{th}\) iterations, it will be,

\[ \sum_{j=1}^{N} a_{ij}^{t+1} x_{j}^{t+1} = \lambda^{t+1} x_{i}^{t+1}, \ \forall \ i = 1, 2, \ldots, N \]  \hspace{1cm} (7)

Now, we rewire an edge and measure the changes in the eigenvector entries. Here, we consider separately for \( i = p, q, r, \) and \( s \) nodes and for the rest of other nodes. From Eq. 6 and Eq. 7 we get,

**Case1: \( i = p \)**

\[ \sum_{j=1}^{N} a_{pj}^t x_j^t = \lambda^t x_p^t \]  \hspace{1cm} (8)

\[ \sum_{j=1}^{N} a_{pj}^{t+1} x_{j}^{t+1} = \lambda^{t+1} x_{p}^{t+1} \]  \hspace{1cm} (9)

Now we use Eq. 3 in Eq. 9 and get,

\[ \sum_{j=1}^{N} (a_{pj}^t + \Delta a_{pj}^{t+1})(x_j^t + \Delta x_j^{t+1}) = (\lambda^t + \Delta \lambda^{t+1})(x_p^t + \Delta x_p^{t+1}) \]

We know that \( a_{pj}^t + \Delta a_{pj}^{t+1} = a_{pj}^t, \ \forall \ j \neq q \) and using Eq. 8 we get

\[ \Delta x_p^{t+1} = \frac{1}{\lambda^t + \Delta \lambda^{t+1}} \left[ \sum_{j=1}^{N} a_{pj}^t \Delta x_j^{t+1} - \Delta \lambda^{t+1} x_p^t + \Delta a_{pq}^{t+1}(x_q^t + \Delta x_q^{t+1}) \right] \]

**Case2:** Similarly for \( i = q \)

\[ \Delta x_q^{t+1} = \frac{1}{\lambda^t + \Delta \lambda^{t+1}} \left[ \sum_{j=1}^{N} a_{qj}^t \Delta x_j^{t+1} - \Delta \lambda^{t+1} x_q^t + \Delta a_{qp}^{t+1}(x_p^t + \Delta x_p^{t+1}) \right] \]

**Case3: \( i = r \)**

\[ \Delta x_r^{t+1} = \frac{1}{\lambda^t + \Delta \lambda^{t+1}} \left[ \sum_{j=1}^{N} a_{rj}^t \Delta x_j^{t+1} - \Delta \lambda^{t+1} x_r^t + \Delta a_{rs}^{t+1}(x_s^t + \Delta x_s^{t+1}) \right] \]

**Case4: \( i = s \)**

\[ \Delta x_s^{t+1} = \frac{1}{\lambda^t + \Delta \lambda^{t+1}} \left[ \sum_{j=1}^{N} a_{sj}^t \Delta x_j^{t+1} - \Delta \lambda^{t+1} x_s^t + \Delta a_{sr}^{t+1}(x_r^t + \Delta x_r^{t+1}) \right] \]
As ∆x_i^{t+1} = \frac{1}{\lambda^t + \Delta \lambda^{t+1}} \left[ \sum_{j=1}^{N} a_{ij}^{t} \Delta x_{j}^{t+1} - \Delta \lambda^{t+1} x_i^{t} \right]

In general, combining all of the above five cases by Kronecker delta \([9]\) we get,

\[ \Delta x_i^{t+1} = \frac{1}{\lambda^t + \Delta \lambda^{t+1}} \left[ \sum_{j=1}^{N} a_{ij}^{t} \Delta x_{j}^{t+1} - \Delta \lambda^{t+1} x_i^{t} + \sum_{j=1}^{N} (\delta_{jq} \delta_{iq} + \delta_{jq} \delta_{ip} + \delta_{jr} \delta_{is} + \delta_{js} \delta_{ir}) \Delta a_{ij}^{t+1} (x_j^{t} + \Delta x_{j}^{t+1}) \right] \]

From the numerical simulations we observe that a single edge rewiring makes very small amount of changes in the eigenvalues and also all other entries in the eigenvectors except for \( p, q, r, \) and \( s \) entries. Thus, we assume \( \lambda^t + \Delta \lambda^{t+1} \sim \lambda^t \) and for \( \Delta x_i^{t+1} = 0, \forall i \neq p, q, r, \) and \( s \). In addition we assume that nodes \( p, q, r, \) and \( s \) are distinct. By using these assumption we get,

\[ \begin{align*}
\Delta x_p^{t+1} &= -\frac{x_q^t}{\lambda^t} \\
\Delta x_q^{t+1} &= -\frac{x_p^t}{\lambda^t} \\
\Delta x_r^{t+1} &= \frac{1}{(\lambda^t)^2 - 1} \left( x_s^t \lambda^t + x_r^t \right) \\
\Delta x_s^{t+1} &= \frac{1}{(\lambda^t)^2 - 1} \left( x_r^t \lambda^t + x_s^t \right)
\end{align*} \]  

(10)

Now, using Eqs. (10) in Eq. (3) we get,

\[ \Delta IPR^{t+1} = \left[ 4(x_p^t)^3 \Delta x_p^{t+1} + 6(x_p^t)^2 (\Delta x_p^{t+1})^2 + 4x_p^t (\Delta x_p^{t+1})^3 + (\Delta x_p^{t+1})^4 \right] + \left[ 4(x_q^t)^3 \Delta x_q^{t+1} + 6(x_q^t)^2 (\Delta x_q^{t+1})^2 + 4x_q^t (\Delta x_q^{t+1})^3 + (\Delta x_q^{t+1})^4 \right] + \left[ 4(x_r^t)^3 \Delta x_r^{t+1} + 6(x_r^t)^2 (\Delta x_r^{t+1})^2 + 4x_r^t (\Delta x_r^{t+1})^3 + (\Delta x_r^{t+1})^4 \right] + \left[ 4(x_s^t)^3 \Delta x_s^{t+1} + 6(x_s^t)^2 (\Delta x_s^{t+1})^2 + 4x_s^t (\Delta x_s^{t+1})^3 + (\Delta x_s^{t+1})^4 \right] \]

As \( \Delta x_i \) is very small, we consider only the first order term and ignore the higher order term and get,

\[ \Delta IPR^{t+1} = \left[ 4(x_p^t)^3 \Delta x_p^{t+1} + 4(x_q^t)^3 \Delta x_q^{t+1} + 4(x_r^t)^3 \Delta x_r^{t+1} + 4(x_s^t)^3 \Delta x_s^{t+1} \right] = 4 \left[ \lambda^t x_p^t x_s^t [(x_p^t)^2 + (x_s^t)^2] + [(x_p^t)^4 + (x_s^t)^4] \right] \frac{x_p^t x_q^t [(x_p^t)^2 + (x_q^t)^2]}{(\lambda^t)^2 - 1} \]

Hence IPR value will increase if the changes of the IPR value \( \Delta IPR^{t+1} > 0 \). Thus,

\[ \frac{\lambda^t x_p^t x_s^t [(x_p^t)^2 + (x_s^t)^2]}{(\lambda^t)^2 - 1} + \frac{[(x_p^t)^4 + (x_s^t)^4]}{(\lambda^t)^2 - 1} > \frac{x_p^t x_q^t [(x_p^t)^2 + (x_q^t)^2]}{\lambda^t} \]

(11)

\( f_2 > f_1 \)

where \( f_1 = \frac{x_p^t x_s^t [(x_p^t)^2 + (x_s^t)^2]}{\lambda^t} \) and \( f_2 = \frac{\lambda^t x_p^t x_s^t [(x_p^t)^2 + (x_s^t)^2]}{(\lambda^t)^2 - 1} + \frac{[(x_p^t)^4 + (x_s^t)^4]}{(\lambda^t)^2 - 1} \). We check our analytical condition (Eq. (11)) through numerical experiments. Starting with an ER random network, we
choose a pair of nodes \((p \text{ and } q)\) such that they have an edge between them and evaluate \(f_1\). Next, we scan the eigenvector entries \((x_r \text{ and } x_s)\) by evaluating \(f_2\) such that the above-mentioned inequality holds for it. We check if the nodes corresponding to \(r\) and \(s\) are connected or not. If they are connected, we again scan the other entries such that corresponding nodes are not connected. Once we find two pairs of the nodes satisfying the condition in Eq. 11 we remove the edge between \(p\) and \(q\) and introduce a connection between the nodes \(r\) and \(s\). We repeat this process for a large number of iterations. As reflected from Fig. S8 initially, satisfying this condition leads to an increment in the IPR value up to 0.16 quite fast. After this value, the rewiring performed based on the pair of eigenvector entries satisfying the condition (in Eq. 11), does not lead to an increase in the IPR value, indeed the IPR value decreases for a further iterations and then it gets saturated (Fig. S8). It probably happens due to the assumptions made by us, particularly, \(\Delta x_{i}^{t+1} = 0, \forall i \neq p, q, r,\) and \(s\) (used in Eq. 11). We have observed that our assumptions stand valid for small IPR values \((r_1\) region) and the regime where IPR values start rapidly increasing (initial part of \(r_2\) region) in Fig. 1(a) and Fig. S3(a)(i). However, during the further evolution, for which the IPR value approaches close to the saturation zone \((r_3\) region in Fig. 1(a) and Fig. S3(a)(i)), our assumptions do not hold good. Therefore, Eq. 11 is able to provide a condition for the increase in the IPR value but does not give satisfactory conclusion beyond certain regime which requires more investigations. However, it is interesting to note that using the condition in Eq. 11 we achieve 60% of the optimal IPR value within 500 iterations (Fig. S8), whereas optimization process takes approx. 20,000 iterations to reach it (Fig. 1(a)).

4.2 Derivations of Continuous variables

To make the analytical derivation for a large class of problems like instead of removing an edge and adding in another place of the network, we can reduce the edge weight in one place and increase the same amount in another place. Hence, changes will occur in the four entries of the adjacency matrix. Thus, we consider adjacency matrix entries as a continuous variable instead of a discrete variable. It will be applicable for non-negative weighted networks. Here, we reduce the edge weight between the nodes \(p\) and \(q\), at the same time increase the same amount of edge weight between the nodes \(r\) and \(s\). To measure the rate of changes in the IPR value with respect to changes in the matrix entry, we differentiate IPR function in Eq. 1 with respect to \(a_{pq}\) and get,

\[
\frac{\partial IPR}{\partial a_{pq}} = 4 \sum_{i=1}^{N} x_{i}^{3} \frac{\partial r_{i}}{\partial a_{pq}}
\] (12)
As $A$ is a symmetric matrix, we consider the change in $a_{pq}$ and $a_{qp}$ entries simultaneously to measure the rate of changes in the IPR value. From Eq. 12 we can say that rate of changes in the IPR value is governed by the rate of changes in the $x_i$ values. To get the values of $\frac{\partial x_i}{\partial a_{pq}}$, we use the eigenvalue equation of $A$,

$$AX = \lambda X$$

which can be written as,

$$\sum_{j=1}^{N} a_{ij} x_j = \lambda x_i, \ \forall i = 1, 2, \ldots N$$

(13)

Differentiating Eq. 13 with respect to $a_{pq}$ we get,

$$\lambda \frac{\partial x_i}{\partial a_{pq}} + x_i \frac{\partial \lambda}{\partial a_{pq}} = \sum_{j=1}^{N} \left[ x_j \frac{\partial a_{ij}}{\partial a_{pq}} + a_{ij} \frac{\partial x_j}{\partial a_{pq}} \right]$$

As mentioned above, $A$ is a symmetric matrix, so we use

$$\frac{\partial a_{ij}}{\partial a_{pq}} = \left( \delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right)$$

where $\delta_{ij}$ is Kronecker delta function [9]. Using the above two equations we get,

$$\lambda \frac{\partial x_i}{\partial a_{pq}} + x_i \frac{\partial \lambda}{\partial a_{pq}} = \delta_{ip} x_q + \delta_{iq} x_p + \sum_{j=1}^{N} a_{ij} \frac{\partial x_j}{\partial a_{pq}}$$

(14)

Further, to calculate the value of $\frac{\partial \lambda}{\partial a_{pq}}$, we differentiate Eq. 2 with respect to $a_{pq}$ and get,

$$\sum_{i=1}^{N} 2x_i \frac{\partial x_i}{\partial a_{pq}} = 0$$

(15)

We multiply Eq. 13 by $x_i$ and then differentiate with respect to $a_{pq}$, we get

$$2\lambda x_i \frac{\partial x_i}{\partial a_{pq}} + x_i^2 \frac{\partial \lambda}{\partial a_{pq}} = \sum_{j=1}^{N} \left[ x_i x_j \frac{\partial a_{ij}}{\partial a_{pq}} + a_{ij} x_j \frac{\partial x_i}{\partial a_{pq}} + a_{ij} x_i \frac{\partial x_j}{\partial a_{pq}} \right]$$

Summing over for $i = 1, 2, \ldots, N$ in the above equation and then use Eq. 2 and Eq. 15 we get,

$$\frac{\partial \lambda}{\partial a_{pq}} = 2x_p x_q + \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \left[ x_j \frac{\partial x_i}{\partial a_{pq}} + x_i \frac{\partial x_j}{\partial a_{pq}} \right]$$

Using symmetry in the above we get,

$$\frac{\partial \lambda}{\partial a_{pq}} = 2x_p x_q + 2 \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} x_i \frac{\partial x_j}{\partial a_{pq}}$$

(16)

Now we use Eq. 16 in Eq. 14 and get

$$\frac{\partial x_i}{\partial a_{pq}} = \frac{1}{\lambda} \left[ \delta_{ip} x_q + \delta_{iq} x_p - 2x_p x_q x_i + \sum_{j=1}^{N} a_{ij} \frac{\partial x_j}{\partial a_{pq}} - 2 \sum_{j=1}^{N} \sum_{k=1}^{N} a_{jk} x_k x_i \frac{\partial x_j}{\partial a_{pq}} \right]$$

(17)
Hence, using Eq. 17 in Eq. 12 we get,

\[
\frac{\partial \text{IPR}}{\partial a_{pq}} = \sum_{i=1}^{N} \frac{4x_i^3}{\lambda} \left[ \delta_{ip}x_q + \delta_{iq}x_p - 2x_px_q x_i - \sum_{k=1}^{N} \sum_{j=1}^{N} a_{kj}x_k x_i \frac{\partial x_j}{\partial a_{pq}} + \sum_{j=1}^{N} a_{ij} \frac{\partial x_j}{\partial a_{pq}} \right]
\]

\[
= \frac{4}{\lambda} \left[ x_p x_q (x_p^2 + x_q^2 - 2\text{IPR}) + \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} x_i \frac{\partial x_j}{\partial a_{pq}} (x_i^2 - 2\text{IPR}) \right]
\]

From the above equation, we learn that rate of change in the IPR value is inversely proportional to \( \lambda \), i.e., \( \frac{\partial \text{IPR}}{\partial a_{pq}} \propto \frac{1}{\lambda} \). Hence, the rate of change in the IPR value will be slow if a network has large \( \lambda \) value. Therefore, Eq. 16, Eq. 17, and Eq. 18 are important which tells about the rate of change in terms of \( \lambda \), \( x_i \), and IPR values.

This enables us to measure the amount of change in the IPR value during and/or after an edge rewiring as follows,

\[
\text{IPR} \frac{\partial a_{pq}}{\text{IPR}} \text{IPR} + \frac{\partial \text{IPR}}{\partial a_{pq}} \varepsilon_1 \frac{\partial \text{IPR}}{\partial a_{rs}} \text{IPR} + \frac{\partial \text{IPR}}{\partial a_{pq}} \varepsilon_2 \frac{\partial \text{IPR}}{\partial a_{rs}} \varepsilon_2 + \frac{\partial^2 \text{IPR}}{\partial a_{pq} \partial a_{rs}} \varepsilon_1 \varepsilon_2
\]

where \( \frac{\partial \text{IPR}}{\partial a_{pq}} \) and \( \frac{\partial \text{IPR}}{\partial a_{rs}} \) are the rate of changes in the IPR value caused by the change in \((p, q)\) and \((r, s)\) edges, respectively. The amount of changes in the edge weight are denoted by \( \varepsilon_1 \) and \( \varepsilon_2 \), respectively. Hence, \( \frac{\partial \text{IPR}}{\partial a_{pq}} \varepsilon_1 + \frac{\partial \text{IPR}}{\partial a_{rs}} \varepsilon_2 + \frac{\partial^2 \text{IPR}}{\partial a_{pq} \partial a_{rs}} \varepsilon_1 \varepsilon_2 \) will give the amount of changes in the IPR value for an edge rewiring with \( a_{ij} \) taken as a continuous variable.

## 5 Conclusion and Open Problem

We explore localization properties of the PEV in complex networks. We construct a network structure through the optimization process that possesses highly localized PEV quantified by IPR. We analyze several structural properties during the network optimization process. It suggests that the most localized PEV does not depend upon a single network property rather it depends on several properties simultaneously. This approach provides a comprehensive way to investigate not only the optimized network but also intermediate networks before the most optimized structure is found. In other words, we develop a learning framework to explore localization of eigenvector through a sampling-based optimized method. Additionally, we have identified a special set of edges which are essential for the (de)localization of PEV in the most optimized network structure. Rewiring any one edge belongs to the special set leads to a complete delocalization of PEV. Moreover, we have found a region where networks have a very high localized PEV which is robust against the single edge rewiring. Finally, we have analytically derived a condition for changes in the IPR value as a function of edge rewiring. The condition gives a way to a construct network which has a high localized IPR value for PEV, but using this analytical treatment we can not achieve the IPR value which was observed numerically for the most optimized network. There may exist certain other parameters which should be satisfied to exactly mimic the numerical simulations. Additionally, we have derived the rate of changes in the IPR value for the non-negative weighted symmetric matrices which might be useful to understand the localization in weighted networks.

We have seen that in the saturation time there are drops in the IPR values which leads to the complete delocalization of the PEV. Our analysis also unable to capture such rare phenomenon. We have seen that the edge rewiring leads to the complete delocalization rotating the PEV in approx. 90°. In addition, we find that PEV becomes delocalized and at the same time eigenvector...
corresponding to the second largest eigenvalue becomes localized from its delocalized state. Hence, instead of finding the cause of IPR drops, if we can find the reason when small perturbation leads to a large change in the eigen structures, it may give some insight of this IPR drops.

References

[1] M. Sade et al., Phys. Rev. E 72, 066123 (2005).
[2] P. Cizeau, J. P. Bouchaud, Phys. Rev. E 3, 1810 (1994).
[3] C. Bordenave, and A. Guionnet, Probability Theory and Related Fields 157, 885, (2013).
[4] P. V. Mieghem, Graph Spectra for Complex Networks, Cambridge University Press, (2011).
[5] F. Harary and E. Palmer, Graphical enumeration, 1st ed., Academic Press, (1973).
[6] C. Li , H. Wang and P. V. Mieghem, Networking 7289, 149, (2012).
[7] A. V. Goltsev et al., Phys. Rev. Lett. 109, 128702 (2012).
[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms, 3rd ed., MIT Press Cambridge, (2009).
[9] M. E. J. Newman, Networks: An Introduction, Oxford University Press, (2010).
[10] Réka Albert and Albert-László Barabási, Rev. Mod. Phys. 74, 47 (2002).
[11] C. I. D. Genio et al., PLOS ONE, 5, e10012 (2010).
[12] J. Saramäki et al., Phys. Rev. E 75, 027105 (2007).
[13] L. S. Heath, N. Parikh, Physica A 390, 4577 (2011).
[14] M. E. J. Newman, Phys. Rev. Lett. 89, 208701 (2002).
[15] R. X. Brunet and I. M. Sokolov, Phys. Rev. E 70, 066102 (2004).
[16] D. Freedman, R. Pisani, R. Purves, Statistics, 4th ed., W. W. Norton & Company (2007).
[17] P. V. Mieghem et. al., Phys. Rev. E 84, 016101 (2011) and references therein.