Partitioning a reaction–diffusion ecological network for dynamic stability

Dinesh Kumar, Jatin Gupta and Soumyendu Raha

Department of Computational and Data Sciences, Indian Institute of Science, Bangalore 560012, India

The loss of dispersal connections between habitat patches may destabilize populations in a patched ecological network. This work studies the stability of populations when one or more communication links is removed. An example is finding the alignment of a highway through a patched forest containing a network of metapopulations in the patches. This problem is modelled as that of finding a stable cut of the graph induced by the metapopulations network, where nodes represent the habitat patches and the weighted edges model the dispersal between habitat patches. A reaction–diffusion system on the graph models the dynamics of the predator–prey system over the patched ecological network. The graph Laplacian’s Fiedler value, which indicates the well-connectedness of the graph, is shown to affect the stability of the metapopulations. We show that, when the Fiedler value is sufficiently large, the removal of edges without destabilizing the dynamics of the network is possible. We give an exhaustive graph partitioning procedure, which is suitable for smaller networks and uses the criterion for both the local and global stability of populations in partitioned networks. A heuristic graph bisection algorithm that preserves the preassigned lower bound for the Fiedler value is proposed for larger networks and is illustrated with examples.

1. Introduction

The distribution of populations over a range of spatially discrete patches is a critical aspect of their interactions in an ecological system. Sets of spatially isolated populations, which are linked by the dispersal of multiple potentially interacting species, are called a metapopulation [1,2]. The destruction of a habitat is
a dominant cause of extinction for various species. Tilman et al. [3] have shown that even a dominant species will eventually become extinct if its habitat is destroyed. Human activities, such as construction of roads, railway lines or fencing, affect the dispersal of species among their habitat patches. Construction of new infrastructure through continuous habitat causes fragmentation of that habitat into small patches and reduces both the quantity and quality of the habitat [4]. Large continuous habitats become highly fragmented and leave small habitat patches dispersed throughout the landscape. Populations in small, isolated habitats have a higher risk of becoming extinct and a reduced chance of recolonization [5,6], meriting the study of the existence and computation of cuts in the network, that is, the removal of the dispersal connections between habitat patches, which have a minimal impact.

The persistence of metapopulations depends upon the interactions between local density, dispersal and spatial heterogeneity [7–9]. Amarasekare [10] examined the influence of density-dependent and -independent dispersal in local dynamics by considering a spatially heterogeneous environment and mortality during dispersal. That study found that, with logistic dynamics, dispersal affects the population dynamics within patches, while Allee dynamics creates between-patch effects. The role of space and diffusion in the dynamics, stability and persistence of populations was studied by Bascompte & Solé [11], and it has been shown that the larger the spatial domain or diffusion, the more unstable the dynamics. The effect of connectivity or species dispersal movement on the long-term survival (stability) of metapopulations remains a research issue. The dispersal of species in metapopulations network can both stabilize and destabilize the metapopulations and depends on the dispersal intensity [12]. The mode of density-dependent dispersal is a key factor for the viability of sources and the long-term persistence of source–sink systems [13]. In [11,14], the authors show that increased dispersal can destabilize the previously stable systems, whereas studies in [15–17] conclude that density-independent dispersal does not affect the stability. The unstable populations of the butterfly Melitaea cinxia in every patch can be stabilized by the dispersal movement (diffusion) of populations among habitat patches in the network; this affects the dynamics substantially and makes the existence of alternative stable equilibrium points [18] possible.

As cutting a complex network interferes with the dispersal (diffusion) and, as in the present work, we study partitioning while trying to achieve stability in the daughter networks, the literature reviewed with regard to stability and dispersal empirically builds a motivating case for the mathematical rendering of the hitherto empirically studied role of dispersion in terms of the stable partitioning of the complex network dynamics. Inspired by the above studies and examples, we consider the stabilization of spatially structured whole populations owing to the dispersal movement of individuals among habitat patches. This is akin to the reverse of the Turing instability condition [19]. The present work, where appropriate, studies, using a combination of qualitative theory of differential equations and graph theory, the conditions and methods for finding partition(s), i.e. human-made cut(s) of an ecological metapopulation network such that the populations remain stable after partitioning. A classic approach to the metapopulations dynamics are the patch occupancy models, where the fraction of occupied patches is considered and explicit local population dynamics is ignored [20]. Within-patch dynamics along with the populations’ dispersal among the patches have been considered in numerous studies, such as [21–24]. The regional persistence of predator–prey interactions has been considered in patch occupancy models in [25,26] and also in models with explicit local dynamics [27]. There are many applications of graph theory in conservation biology and landscape ecology. A theoretical analysis of the stability and persistence of ecological metapopulations especially focusing on a marine system is found in [28], where the authors determined the conditions of persistence of metapopulations (which are age-structured patch populations with patch sub-populations connected by larval dispersal) by graph theory methods and found that, among various factors, the migration of individuals between patches is a very critical factor for the overall stability of metapopulations. Using the graph theory approach in [29], it is shown that the population can persist despite substantial losses of habitat area, as long as there exists a spanning tree.
Metapopulations constitute a complex network and the spectra of the complex network (in particular, the second smallest eigenvalue of the Laplacian matrix, called the Fiedler value) are associated with the connectivity of the graph (network). The master stability function (MSF) technique analyses the stability of the synchronized state of coupled oscillators and depends upon the ratio of the largest to the second-smallest eigenvalue of the Laplacian [30]. In networked dynamical systems, the Fiedler value is found to be a measure of stability and robustness [31,32]. The spectrum of the Laplacian of the graph of certain molecules can be used to predict their chemical properties [33,34] in a similar way. Our work finds that the Fiedler value plays a significant role in the stability of metapopulations.

Graph partitioning has many significant practical uses in science and engineering. Arranging the electronic components in very-large-scale integrated circuits in such a way that the number of connections between two partitions is minimal [35] is one such example. Unfortunately, graph partitioning itself is an NP-hard (a complexity class of decision problems that are intrinsically harder than those which can be solved in polynomial time) problem [36], and therefore it is not always possible to compute the optimal partition.

Graph partitioning studies in the existing literature to date have been for static graphs, none of which involves the dynamics on the graph nodes. When feasible, we find a suitable graph partition of the metapopulations network, so that the populations in the resultant smaller sub-networks are stable. Towards this, a heuristic bisection algorithm, which produces graph components with a Fiedler value greater than some threshold, is formulated. The constraint on the Fiedler value of the graph network is derived by linear stability analysis of the metapopulation dynamics. Also, the necessary and sufficient conditions for the safe deletion of one or more edges are determined. Alternating and edge principles (due to Merris [37]) are exploited for safe deletion of an edge from a graph. The global stability of the populations in the network requiring a condition on the nodes on which the minimum weight edges are incident and some restrictions on the internal patch population dynamics are also derived in this work. An exhaustive approach for smaller sized graphs gives the set of ‘desired’ partitions and analyses all of them for the ‘best’ choice.

2. Model and linearized stability analysis

In our model, prey and predator populations both occupy the spatially discrete patches of an ecological network and diffuse over the corridors that link them. Each patch is represented as a node of the graph $G$. At each node of $G$, the dynamics of the prey and predator populations is governed by a set of two differential equations (e.g. Lotka–Volterra model equations). The dispersal movement of the species among the patches occurs along the links or corridors which are the edges of the graph. The amount of dispersal between patches is proportional to the difference in population densities [21], and the proportionality constant (i.e. the dispersal rate) is given by the weight on the corresponding edge. The dynamics of the predator–prey population in the $i$th habitat patch with passive migration from/to other patches of the populations is given by

$$
\begin{align*}
\dot{x}_i &= f_i(x_i, y_i) + \sum_{j=1, j \neq i}^{n} w_{ij}^x (x_j - x_i) \\
\dot{y}_i &= g_i(x_i, y_i) + \sum_{j=1, j \neq i}^{n} w_{ij}^y (y_j - y_i),
\end{align*}
$$

(2.1)

where $i = 1, 2, \ldots, n$ and $x_i(t)$ and $y_i(t)$ are the prey and predator densities, respectively, at time $t$ in patch $i$. Real-valued functions $f_i \in C^1([0, \infty)^2)$ and $g_i \in C^1([0, \infty)^2)$ represent the dynamics of the prey and predator species, respectively, within the $i$th patch, and they are assumed to be arbitrary non-negative continuously differentiable functions over a feasible domain. The ecological network is assumed to be an undirected weighted graph so that the net diffusion rates
$w_{ij}^x$ and $w_{ij}^y$ for the prey and predator populations between patch $i$ and patch $j$, respectively, are the same in both directions, i.e. $w_{ij}^x = w_{ji}^x$ and $w_{ij}^y = w_{ji}^y$.

The populations are assumed to be (locally) stable in the connected network, while there is no population stability (including oscillatory behaviour) within an isolated habitat patch. Instability at the individual graph node leads to unstable population dynamics. If either of the two following distinct conditions, marked as type I and type II instability, respectively, holds true at the co-existing equilibrium point of the reaction part of the system (2.1), i.e. $x_i = f_i(x_i, y_i), y_i = f_i(x_i, y_i), i = 1, 2, \ldots, n$, then the system is indicated as unstable at each node of the graph:

1. $\partial f_i / \partial x_i + \partial g_i / \partial y_i > 0$ (type I)
2. $\partial f_i / \partial x_i + \partial g_i / \partial y_i < 0$, $\partial f_i / \partial x_i \partial g_i / \partial y_i - \partial f_i / \partial y_i \partial g_i / \partial x_i < 0$ (type II).

The above conditions are a direct consequence of the Dulac–Bendixson criterion [38].

Generally, we shall deal here with connected, weighted, undirected and simple (no self-loops) graphs unless otherwise stated. We now recall that the Laplacian matrix of the graph $G$, which is denoted by $L(G)$ or simply by $L$, is defined as follows:

$$L_{ij} = \begin{cases} d_i, & \text{if } i = j, \\ -w_{ij}, & \text{if } i \sim j \text{ (i.e. } i \text{ and } j \text{ are adjacent),} \\ 0, & \text{otherwise,} \end{cases}$$

where $d_i = \sum_{i \sim j} w_{ij}$ is the sum of the weights of the edges incident on node $i$. The system (2.1) may be written in vector form as follows:

$$\dot{X} = F - L^x X, \quad \dot{Y} = G - L^y Y,$$

and

$$X = (x_1, x_2, \ldots, x_n)^T, \quad Y = (y_1, y_2, \ldots, y_n)^T, \quad F = (f_1, f_2, \ldots, f_n)^T, \quad G = (g_1, g_2, \ldots, g_n)^T,$$

where $L^x$ and $L^y$ are the Laplacians of the graph corresponding to prey and predator dispersal, respectively. $L^x$ and $L^y$ are referred to as prey Laplacian and predator Laplacian, respectively, and their respective Fiedler values $\lambda_2^x$ and $\lambda_2^y$ are referred to as the prey Fiedler value and the predator Fiedler value, respectively. Note that the Fiedler value of the Laplacian matrix of graph $G$ is also called the Fiedler value of graph $G$ and is denoted by the symbol $\lambda_2(G)$ or simply by $\lambda_2$.

Linearizing system (2.2) around its co-existing equilibrium solution, we obtain

$$
\begin{pmatrix}
\dot{x}_i \\
\dot{y}_i
\end{pmatrix}
= \begin{pmatrix}
\frac{\partial F}{\partial x_i} & \frac{\partial F}{\partial y_i} \\
\frac{\partial G}{\partial x_i} & \frac{\partial G}{\partial y_i}
\end{pmatrix}
\begin{pmatrix}
x_i \\
y_i
\end{pmatrix}
- \begin{pmatrix}
L^x & 0 \\
0 & L^y
\end{pmatrix}
\begin{pmatrix}
x_i \\
y_i
\end{pmatrix}.
$$

Let $X = \sum_j \xi_j^x(t) \Phi_j^x$ and $Y = \sum_j \xi_j^y(t) \Phi_j^y$, where $\Phi_j^x$ and $\Phi_j^y$ are eigenvectors of the Laplacian matrices $L^x$ and $L^y$ of graph $G$ corresponding to the $j$th smallest eigenvalues $\lambda_j^x$ and $\lambda_j^y$, respectively. Functions $\xi_j^x(t)$ and $\xi_j^y(t)$ are scalar valued functions of time. Term wise, the above system can be written as

$$
\begin{pmatrix}
\xi_j^x(t) \\
\xi_j^y(t)
\end{pmatrix}
= \begin{pmatrix}
\frac{\partial f_1, f_2, \ldots, f_n}{\partial(x_1, x_2, \ldots, x_n)} & \frac{\partial f_1, f_2, \ldots, f_n}{\partial(y_1, y_2, \ldots, y_n)} \\
\frac{\partial g_1, g_2, \ldots, g_n}{\partial(x_1, x_2, \ldots, x_n)} & \frac{\partial g_1, g_2, \ldots, g_n}{\partial(y_1, y_2, \ldots, y_n)}
\end{pmatrix}
\begin{pmatrix}
\xi_j^x(t) \\
\xi_j^y(t)
\end{pmatrix}
- \begin{pmatrix}
\lambda_j^x I_n & 0 \\
0 & \lambda_j^y I_n
\end{pmatrix}
\begin{pmatrix}
\xi_j^x(t) \\
\xi_j^y(t)
\end{pmatrix},
$$

that is,

$$
\begin{pmatrix}
\xi_j^x(t) \\
\xi_j^y(t)
\end{pmatrix}
= (A - D_j)
\begin{pmatrix}
\xi_j^x(t) \\
\xi_j^y(t)
\end{pmatrix},
$$

(2.3)
where

\[
A := \begin{bmatrix}
\frac{\partial (f_1, f_2, \ldots, f_n)}{\partial (x_1, x_2, \ldots, x_n)} & \frac{\partial (f_1, f_2, \ldots, f_n)}{\partial (y_1, y_2, \ldots, y_n)} \\
\frac{\partial (g_1, g_2, \ldots, g_n)}{\partial (x_1, x_2, \ldots, x_n)} & \frac{\partial (g_1, g_2, \ldots, g_n)}{\partial (y_1, y_2, \ldots, y_n)}
\end{bmatrix},
\]

\[
D_j := \begin{bmatrix}
\lambda_j^x I_n & 0 \\
0 & \lambda_j^y I_n
\end{bmatrix},
\]

\(I_n\) is the identity matrix of order \(n\).

Noting that \(\lambda_1^x = 0, \lambda_1^y = 0\) and \(\lambda_2^x \leq \lambda_j^x, \lambda_2^y \leq \lambda_j^y\ \forall j \geq 2\), and summing the terms (2.3) over \(j\), we obtain

\[
\begin{pmatrix}
\dot{X} \\
\dot{Y}
\end{pmatrix} \leq (A - D_2) \begin{pmatrix}
X \\
Y
\end{pmatrix},
\]

where the inequality is a matrix entry-wise and dispersal loss is assumed. Let the eigenvalues of the coefficient matrix \(A - D_2\) of inequality system (2.4) be denoted by \(\sigma\). These eigenvalues determine the temporal growth and are given by the roots of the characteristic polynomial of the coefficient matrix \(A - D_2\), i.e.

\[
\det (A - D_2 - \sigma I) = 0.
\]

Then, at the \(i\)th patch, the following holds:

\[
\sigma^2 - \sigma \left( \text{tr} J_i - \lambda_2^x - \lambda_2^y \right) + \left( \lambda_2^x \lambda_2^y - \lambda_2^x \frac{\partial g_i}{\partial x_i} - \lambda_2^y \frac{\partial f_i}{\partial x_i} + \det J_i \right) = 0,
\]

(2.5)

where \(J_i\) is the Jacobian of the reaction terms of the prey and predator populations in the \(i\)th patch calculated at the co-existential equilibrium point of system (2.1), that is,

\[
J_i := \begin{bmatrix}
\frac{\partial f_i}{\partial x_i} & \frac{\partial f_i}{\partial y_i} \\
\frac{\partial g_i}{\partial x_i} & \frac{\partial g_i}{\partial y_i}
\end{bmatrix}.
\]

**Remark.** The matrix \(A - D_2\) consists of four blocks, each of which is a diagonal matrix of order \(n\). This peculiarity of the matrix \(A - D_2\) results in \(n\) quadratic factors of its characteristic polynomial and the eigenvalues can be computed by determining the eigenvalues of a \(2 \times 2\) matrix, which is made up of the \((i, i)\)th entries \((i = 1, \ldots, n)\) of each block matrix, and the positions of the entries are the same as their respective block positions in the original matrix. This can be concluded by noting that if \((x_1, x_2)^T\) is an eigenvector of the above-mentioned \(2 \times 2\) matrix, then \((0, \ldots, 0, x_1, 0, \ldots, 0, x_2, 0, \ldots, 0)^T\) is an eigenvector of the matrix \(A - D_2\), where \(x_1\) is at the \(i\)th position and \(x_2\) is at the \((i + n)\)th position.

As the graph network for both populations is the same, the only difference, if any, is in their diffusion rates \(w_i^x\) and \(w_j^y\) between \(i\) and \(j\) patches. This means that \(L^x\) and \(L^y\) are of the same order but differ in their non-zero entries. Without loss of generality, we write \(\lambda_2^x = \lambda_2^y + \mathcal{E}\), \(\mathcal{E} \geq 0\), put it in equation (2.5) and simplify to obtain

\[
\sigma^2 - \sigma \left( \text{tr} J_i - 2\lambda_2^y - \mathcal{E} \right) + \left( \lambda_2^y \left( \text{tr} J_i - \mathcal{E} \right) + \det J_i - \mathcal{E} \frac{\partial g_i}{\partial y_i} \right) = 0, \quad 1 \leq i \leq n.
\]

(2.6)
The roots of the above equations have non-positive real parts (which indicates the linear stability of system (2.1)) if and only if

\[ \text{(i) } \operatorname{tr} J_i - 2\lambda_i^y - \mathcal{S} \leq 0 \iff \lambda_2^x + \lambda_2^y \geq \operatorname{tr} J_i \quad \iff \lambda_2^x + \lambda_2^y \geq \max_i \operatorname{tr} J_i \quad \text{and} \]

\[ \text{(ii) } \lambda_i^y \geq 0 \iff \det J_i - \mathcal{S} \frac{\partial f_i}{\partial y_i} \geq 0 \iff (\operatorname{tr} J_i - \mathcal{S})^2 - 4 \left( \det J_i - \mathcal{S} \frac{\partial f_i}{\partial y_i} \right) \leq 0 \]

\[ \iff \mathcal{S}^2 - 2\mathcal{S} \left( \operatorname{tr} J_i - \frac{\partial f_i}{\partial y_i} \right) + \operatorname{tr} J_i^2 - 4 \det J_i \leq 0 \]

\[ \iff (\mathcal{S} - \alpha_1)(\mathcal{S} - \alpha_2) \leq 0, \quad \alpha_i = \operatorname{tr} J_i - \frac{\partial f_i}{\partial y_i} + \left(-1\right)^{i+1} \sqrt{\left(\frac{\partial f_i}{\partial y_i}\right)^2 - \operatorname{tr} J_i \frac{\partial f_i}{\partial y_i} + \det J_i} \]

\[ \iff \left(\frac{\partial f_i}{\partial y_i}\right)^2 - \operatorname{tr} J_i \frac{\partial f_i}{\partial y_i} + \det J_i \geq 0, \quad \alpha_1 \leq \mathcal{S} \leq \alpha_2. \]

In other words, if the sum of the Fiedler values of the Laplacians is greater than some threshold value and the gap between these eigenvalues is within an appropriate range, then the reaction diffusion system (2.1) is linearly asymptotically stable despite the instability of populations at each patch of the network.

If there is a type II instability in the network patches, then the gap \( \mathcal{S} \neq 0 \). If \( \mathcal{S} = 0 \), then from the second line of condition (ii) above, \( \det J_i \geq 0 \), which contradicts the definition of type II instability. Condition (i) is automatically fulfilled under type II instability. The gap \( \mathcal{S} \) between the Fiedler values \( \lambda_2^x \) and \( \lambda_2^y \) needs to be addressed to stabilize the populations in the components while partitioning the network under the type II instability condition. Under the type I instability condition, both the sum of the Fiedler values and the gap between them affect the partitioning.

Consider the special case \( L^y = \mu L^y, \mu > 0 \). This means that, if \( w_{ij}^y \) is the diffusion rate of the predator population between the \( i \)th patch and the \( j \)th patch, the prey’s diffusion \( w_{ij}^y = \mu w_{ij}^y \) between patches \( i \) and \( j \). Ecologically, this may mean that one species is uniformly diffusely driven in the network by the other, as the prey species escapes the predators or predators chase the prey. Under this special case, (2.5) becomes

\[ \sigma^2 - \sigma \left( \operatorname{tr} J_i - (\mu + 1)\lambda_i^y \right) + \mu \left( \lambda_i^y \frac{\partial f_i}{\partial x_i} + \frac{1}{\mu} \frac{\partial f_i}{\partial x_i} \right) + \frac{\det J_i}{\mu} = 0, \quad (2.7) \]

from which the conditions

\[ \text{(a) } \lambda_i^y \geq \frac{1}{\mu + 1} \max_i \operatorname{tr} J_i = \tau \text{ (say)} \]

\[ \text{(b) } \lambda_i^y \geq 0 \iff \left( \frac{\partial f_i}{\partial x_i} + \frac{1}{\mu} \frac{\partial f_i}{\partial x_i} \right)^2 - 4 \frac{\det J_i}{\mu} \leq 0, \text{ } 1 \leq i \leq n \]

follow and must hold for the linear stability of system (2.1). Note that condition (b) above does not occur under type II instability. Thus in the special case \( L^y = \mu L^y \), type II instability is ruled out. Under type I instability, the above conditions (a) and (b) show that the populations are stable in a network if the predator Fiedler value \( \lambda_i^y \) is at least some real number \( \tau \) and the internal patch population dynamics satisfies condition (b) at the co-existential equilibrium point.

For brevity, we define the following. A stable network is a network where both populations (prey and predator) co-exist, that is, the conditions (a) and (b) above hold true. Note that if a network is stable, then any partition of it gives components satisfying condition (b). A stable partition is a partition of a graph if all the components obtained after partitioning are stable.

Hence, to determine the stable partition of a stable network one needs only to verify that the predator Fiedler value in the partitioned components is at least \( \tau \). In ecological terms, sustainable separations of the patches are possible if the populations’ dispersal movement rates are sufficient as, by Weyl’s monotonicity theorem [39], the larger the entries (corresponding to higher diffusion rates among patches) in the Laplacians, the greater the eigenvalues. After the partition, if the populations start increasing their diffusion rates among the patches in the partitioned components, then an arbitrary partition can become stable. Under the assumption that the diffusion rates are time invariant, it is necessary to examine whether it is possible to obtain a stable partition of a stable network.
The Fiedler value of special graphs of size $n$. Here, $D$ is the diameter of the graph, $d$ is the mean distance, $d_i$ is the sum of weights $w_i$ of edges that are incident on node $i$, $V$ is the set of nodes of $G$, $\delta(G)$ is the minimum sum of the weights of edges that are incident on a node of $G$, $|S|$ is the size of subset $S$ of $V$, $n - |S|$ is the size of the complement of $S$ in $G$ and $E(S, \overline{S})$ is the set of edges whose one end node is in $S$ and the other is in $n-S$.

<table>
<thead>
<tr>
<th>lower bounds</th>
<th>upper bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{4}{nD}$</td>
<td>$\frac{4}{(2(n-1)d - n + 2)}$</td>
</tr>
<tr>
<td>$\max_{i \neq j}(d_i + d_j) - (n-2)$</td>
<td></td>
</tr>
<tr>
<td>$\frac{1}{2} \left( \sum_{k \in V} w_k + \sum_{i \in V} w_i \right)$</td>
<td>$\frac{n}{n-1} \delta(G)$</td>
</tr>
<tr>
<td>$n \frac{E(S, \overline{S})}{</td>
<td>S</td>
</tr>
</tbody>
</table>

$i$ and $j$ are non-adjacent nodes

Table 2. The Fiedler value of special graphs of size $n$.

<table>
<thead>
<tr>
<th>graph (G)</th>
<th>path</th>
<th>cycle</th>
<th>cube</th>
<th>complete</th>
<th>star</th>
<th>tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_2(G)$</td>
<td>$2 \left( 1 - \cos \left( \frac{\pi}{n} \right) \right)$</td>
<td>$2 \left( 1 - \cos \left( \frac{2\pi}{n} \right) \right)$</td>
<td>$2$</td>
<td>$n$</td>
<td>$1$</td>
<td>$\leq 1$</td>
</tr>
</tbody>
</table>

The sensitivity of the Fiedler value is a function of the partition of the graph or of any deletion of an edge from the graph or of any change in the weights of the graph edges. If any small change in the edge weights leads to an abrupt change in the Fiedler value, then there is no point in partitioning the network. However, Weyl’s perturbation theorem [39] applied in our context gives $|\lambda_2(L+P)-\lambda_2(L)| \leq \|P\|$, where $L$ is the Laplacian matrix, $P$ is the perturbation matrix and $\lambda_2$ is the second smallest eigenvalue of a matrix. This means that, in the context of the patched ecological system, a small change in the Laplacian matrix implies a small change in the Fiedler values.

There are many easily computable lower and upper bounds for the Fiedler values for graphs in the existing literature [33,40,41]. Some of these bounds are listed in Table 1. If $\tau$ is compared with these bounds (see theorems 2.1 and 2.2), then the bounds provide an easy way to an otherwise difficult check of stability of the populations on the graph nodes of the patch network. Table 2 lists the Fiedler values of some special graphs with edges having unit weight.

The following theorems give our results on removal of edge(s) in the network.

**Theorem 2.1.** Let $C$ be any partitioned component of a graph $G$. If the stability threshold value $\tau$ for stability is greater than the upper bound of the predator Fiedler value $\lambda_2^V(C)$, then the partitioned component $C$ is unstable.

**Theorem 2.2.** If the stability threshold value $\tau$ is less than the lower bound of the predator Fiedler value $\lambda_2^V(C)$, then the partitioned component $C$ of graph $G$ is stable.

**Theorem 2.3 (necessary condition for removal of some edges).** Let $G(V, E)$ be a graph with node set $V$, edge set $E$ and size $|V| = n(\geq 3)$. Let $G'$ be a stable subgraph of $G$ obtained by removing some edges. Then the average weight of edges (k) incident on a node of $G'$ is at least $((n-1)/n)\tau$.

**Proof.** We know that $L_{ii} = d_i = \sum_{i \neq j} w_{ij}$ is the sum of the weights of edges incident on node $i$. Thus, $\sum_{i=1}^n L_{ii} = \sum_{i=1}^n d_i$ implies that

$$\sum_{i=1}^n d_i = \text{tr}L = \lambda_1 + \cdots + \lambda_n \geq (n-1)\tau,$$

from which it follows that the average weight of the edges incident on any node must be greater than or equal to $((n-1)/n)\tau$.

From the stability conditions, it is clear that if the difference between the predator Fiedler values of the graph Laplacian $L(G)$ and edge-cut-induced graph component Laplacian $L(C)$ is not
more than the difference between the predator Fiedler value of \( L(G) \) and \( \tau \), then the populations in the graph component \( C \) will be stable. Although the interlacing theorem [42] places each Laplacian eigenvalue of a subgraph (obtained after the removal of an edge from the original graph) in between the two consecutive Laplacian eigenvalues of the original graph, it is hard to estimate the exact difference between these eigenvalues. As a special case, we consider the following.

**Theorem 2.4 (sufficient condition for removal of an edge).** Let \( G(V,E) \) be a graph with \(|V| = n (\geq 3)\), and let \( i \) and \( j \) be two adjacent nodes. Let \( \epsilon_i \) be a unit vector in \( \mathbb{R}^n \), whose \( i \)th component is 1 and 0 otherwise. Let the difference of the \( i \)th and \( j \)th columns of the Laplacian matrix \( L(G) \) of \( G \) be \( \lambda \mathbf{v}, \mathbf{v} = \epsilon_i - \epsilon_j \). Then the resultant subgraph \( G' \) obtained by deleting the edge of weight \( w_{ij} \neq \lambda \) that links the \( i \)th and the \( j \)th node of graph \( G \) is stable provided \( w_{ij} \leq (\lambda(G) - \tau)/\mathbf{v}^\top \mathbf{v} = (\lambda(G) - \tau)/2 \).

**Proof.** We know that a change in the labelling of nodes does not affect the properties of \( L \), hence we relabel the nodes \( i \) and \( j \) as nodes 1 and 2, respectively. The Laplacian matrix of a subgraph \( G' \) that is induced from a graph \( G \) by deleting an edge is a symmetric rank 1 updated Laplacian matrix of graph \( G \). This can be written as

\[
L(G') = L(G) - \begin{bmatrix}
  w_{12} & -w_{12} & 0 & \ldots & 0 \\
  -w_{12} & w_{12} & 0 & \ldots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & 0 & \ldots & 0
\end{bmatrix} = L(G) - w_{12} \mathbf{v} \mathbf{v}^\top, \tag{2.8}
\]

where \( \mathbf{v} \mathbf{v}^\top = (\epsilon_1 - \epsilon_2)(\epsilon_1 - \epsilon_2)^\top \) is a symmetric rank 1 perturbation matrix with entries \( a_{11} = a_{22} = 1, a_{12} = a_{21} = -1 \) and the rest being zero. \( w_{12} \) is the weight of the edge that connects nodes 1 and 2. From equation (2.8), it is easy to see that \( \mathbf{v} \) is an eigenvector of \( L(G') \) and the corresponding eigenvalue is \( \lambda = w_{12} \mathbf{v} \mathbf{v}^\top \). Thus \( \lambda \) is reduced by \( w_{12} \mathbf{v} \mathbf{v}^\top \), and by theorem 2.1 of [43] the rest of the eigenvalues do not change. For stability of the subgraph \( G' \), we then require \( \lambda(G) - w_{12} \mathbf{v} \mathbf{v}^\top \mathbf{v} \geq \tau \) .

For stability after deleting an edge, theorem 2.4 provides an upper bound on the weight of the edge when the graph Laplacian matrix has a special form as in the theorem. One example of such a graph with its Laplacian having eigenvectors of the form \( \epsilon_i - \epsilon_j, 1 \leq i \leq n, i \neq j \), is a complete graph with equal weight on its edges. Figure 1a shows a complete graph \( K_4 \) with all edges having equal weight 1; its only non-zero eigenvalue is 4 with multiplicity 3, and the corresponding eigenvectors are in the form of \( (0, -1, 0, 1) = \epsilon_4 - \epsilon_2, (0, -1, 1, 0) = \epsilon_3 - \epsilon_2, (1, -1, 0, 0) = \epsilon_1 - \epsilon_2 \) and their linear combinations. By theorem 2.4, we can delete any edge of the graph \( K_4 \) safely as long as \( \tau \leq 2 \).

Also note that, for \( L(G) \) to have an eigenvector in the form of \( \epsilon_i - \epsilon_j \), the entry \( a_{ii} \) and \( a_{jj} \) in \( L(G) \) must be equal as \( a_{ii} - a_{jj} = -(a_{jj} - a_{ii}) \). This information is helpful in deciding whether the matrix has an eigenvector in the desired form or not. If \( a_{ii} \neq a_{jj} \) for any pair \( (i,j), i \neq j \), then the matrix cannot have eigenvectors of the form \( \epsilon_i - \epsilon_j \), and if \( a_{ii} = a_{jj} \), then we check the difference between the \( i \)th and \( j \)th columns. Theorem 2.4 can be generalized in a straightforward manner by using theorem 3.1 from [44], as follows.
Theorem 2.5 (sufficient condition for removal of $r$ edges). Let $G(V, E)$ be a graph with $|V| = n(\geq 3)$, and let there be $r$ edges $e(i_k,j_k)$, $1 \leq k \leq r$, that link nodes $i_k$ and $j_k$, respectively. Let \( \{v_k = \varepsilon_{i_k} - \varepsilon_{j_k}, i_k \neq j_k, 1 \leq k \leq r\} \) be a list of eigenvectors of $L(G)$ that correspond to the eigenvalues $\lambda_k \neq 0$, $1 \leq k \leq r$. Then the subgraph $G'$ obtained by deleting the $r$ edges $e(i_k,j_k)$, $1 \leq k \leq r$, of weights $w_{i_k,j_k} \neq \lambda_k$ yields the Laplacian $L(G')$, which is a rank-$r$ perturbation to the Laplacian matrix of graph $G$, i.e. $L(G') = L(G) - \sum_{k=1}^{r} w_{i_k,j_k} v_k v_k^T$, and the subgraph $G'$ is stable if

$$\min \{\nu_1, \nu_2, \ldots, \nu_r \} \geq \tau,$$

where $\nu_k, 1 \leq k \leq r$, are eigenvalues of $r \times r$ matrix $\text{diag}(\lambda_1, \ldots, \lambda_r) + (w_{i_k,j_k} v_k v_k^T)(v_1, \ldots, v_r)$.

In theorem 2.5, if all the $r$ edges are non-adjacent, then all the pairs of the nodes $(i_k,j_k)$, $1 \leq k \leq r$, are distinct. This ensures that the list $\{v_k, 1 \leq k \leq r\}$ is a set of orthogonal eigenvectors of $L$. The list of eigenvalues of the rank-$r$ updated matrix, i.e. $L(G) - \sum_{k=1}^{r} w_{i_k,j_k} v_k v_k^T$, becomes $[\lambda_1 - 2w_{i_1,j_1}, \lambda_2 - 2w_{i_2,j_2}, \ldots, \lambda_r - 2w_{i_r,j_r}, \lambda_{r+1}, \ldots, \lambda_n]$. Thus, the stability of populations in the graph $G'$ then simply requires $\min_{k}[\lambda_k - 2w_{i_k,j_k}, 1 \leq k \leq r] \geq \tau$.

For particular cases, we can apply two principles by Merris [37] (given below) for deciding the candidate edge to be deleted from the graph safely. We consider an eigenvector corresponding to the eigenvalue $\lambda_2(G)$. By the alternating principle figure 1b is obtained from figure 1a by deleting edge $e(2,3)$ at a cost of reduction in $\lambda_2(K_4)$ of 2, as $x(2) = -x(3)$ in eigenvector $v_2$. Similarly, figure 1c is obtained from figure 1b by deleting edge $e(1,4)$ without any change in the Fiedler value as $x(1) = x(4)$ in $v_2$.

Alternating principle. Let $G$ be a graph and $x$ an eigenvector of $L(G)$ corresponding to the eigenvalue $\lambda_2(G)$. Let the adjacent nodes $i$ and $j$ of $G$ be such that $x(i) = -x(j)(\neq 0)$. Let $G'$ be the graph obtained by deleting the edges between all such paired nodes $i$ and $j$. Then $x$ is an eigenvector of $G'$ corresponding to the eigenvalue $\lambda_2 - 2$.

Edge principle. Let $G$ be a graph and $x$ an eigenvector of $L(G)$ corresponding to the eigenvalue $\lambda$, such that $x(u) = x(v)$ for some adjacent nodes $u$ and $v$. Let $G'$ be the graph obtained by removing edge $e(u,v)$. Then $x$ is an eigenvector of $G'$ corresponding to the eigenvalue $\lambda$.

3. Global stability

Here, we determine some parametric conditions for the global stability of system (2.1) by using a contraction analysis [45] based on the idea of exponential convergence of nearby trajectories. If all of the neighbouring solution trajectories of the system show a contraction behaviour at each point of the feasible domain, then the system $\dot{z} = h(z,t), z \in \mathbb{R}^n$ is globally stable. The required condition for this is that the largest eigenvalue of the symmetric part of the Jacobian of this system must be uniformly negative. The symmetric part of the Jacobian of the vector field of system (2.1) is obtained as

$$
\begin{bmatrix}
\frac{\partial f_1}{\partial x_1} - L^{(1)}_{11} & \frac{\partial f_1}{\partial x_2} - L^{(2)}_{12} & \cdots & \frac{\partial f_1}{\partial x_n} - L^{(n)}_{1n} & \frac{1}{2} \left( \frac{\partial g_1}{\partial y_1} + \frac{\partial g_1}{\partial y_2} \right) & 0 & \cdots & 0 \\
\frac{\partial f_1}{\partial x_2} - L^{(1)}_{21} & \frac{\partial f_1}{\partial x_2} - L^{(2)}_{22} & \cdots & \frac{\partial f_1}{\partial x_n} - L^{(n)}_{2n} & \frac{1}{2} \left( \frac{\partial g_2}{\partial y_1} + \frac{\partial g_2}{\partial y_2} \right) & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_1}{\partial x_n} - L^{(1)}_{n1} & \frac{\partial f_1}{\partial x_n} - L^{(2)}_{n2} & \cdots & \frac{\partial f_1}{\partial x_n} - L^{(n)}_{nn} & \frac{1}{2} \left( \frac{\partial g_n}{\partial y_1} + \frac{\partial g_n}{\partial y_2} \right) & 0 & \cdots & 0 \\
\frac{1}{2} \left( \frac{\partial g_1}{\partial y_1} + \frac{\partial g_1}{\partial y_2} \right) & 0 & \cdots & 0 & \frac{\partial g_1}{\partial y_1} - L^{(1)}_{11} & -L^{(2)}_{12} & \cdots & -L^{(n)}_{1n} \\
0 & \frac{1}{2} \left( \frac{\partial g_2}{\partial y_1} + \frac{\partial g_2}{\partial y_2} \right) & \cdots & 0 & -L^{(2)}_{21} & \frac{\partial g_2}{\partial y_2} - L^{(2)}_{22} & \cdots & -L^{(n)}_{2n} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{1}{2} \left( \frac{\partial g_n}{\partial y_1} + \frac{\partial g_n}{\partial y_2} \right) & -L^{(n)}_{n1} & -L^{(n)}_{n2} & \cdots & \frac{\partial g_n}{\partial y_n} - L^{(n)}_{nn}
\end{bmatrix}
$$
According to the Gershgorin disc theorem [46], all the eigenvalues of a matrix $A$ of order $n$ lie in the union of Gershgorin discs centred at $a_{ii}$ with radius $\sum_{j\neq i} |a_{ij}|$, $i = 1, \ldots, n$. For the global stability of system (2.1), we obtain the following sufficient conditions, which ensure that the union of Gershgorin intervals for the matrix (3.1) lies in the left half of the real number line so that the largest eigenvalue of the symmetric part of the Jacobian matrix is uniformly negative. For each $i \in \{1, \ldots, n\}$,

1. $\partial f_i/\partial x_i - L^{(x)}_{ii} < 0$, and $\partial g_i/\partial y_i - L^{(y)}_{ii} < 0$, $\forall(x_i, y_i)$ in the feasible domain. (This condition ensures the centres of the discs lie on the negative axis).
2. $\sum_{j \neq i} |g_{ij}| + \frac{1}{2} |\partial g_i/\partial x_i + \partial f_i/\partial y_i| < |\partial f_i/\partial x_i - L^{(x)}_{ii}|$ and $\sum_{j \neq i} |g_{ij}| + \frac{1}{2} |\partial g_i/\partial x_i + \partial f_i/\partial y_i| < |\partial g_i/\partial y_i - L^{(y)}_{ii}|$, $\forall(x_i, y_i)$ (which ensures that the radius of each disc is not more than the distance of its centre from the origin).

That is,

1. $L^{(x)}_{ii} > \partial f_i/\partial x_i$ and $L^{(y)}_{ii} > \partial g_i/\partial y_i$, $\forall(x_i, y_i)$.
2. $\sum_{j \neq i} |g_{ij}| + \frac{1}{2} |\partial g_i/\partial x_i + \partial f_i/\partial y_i| < |\partial f_i/\partial x_i + \partial g_i/\partial y_i| < |\partial f_i/\partial x_i|$, $\forall(x_i, y_i)$, since $L^{(y)}_{ii} = \sum_{j \neq i} |L^{(y)}_{ij}|$. Similarly, $\frac{1}{2} |\partial g_i/\partial x_i + \partial f_i/\partial y_i| < |\partial g_i/\partial y_i|$, $\forall(x_i, y_i)$.

After further simplifications, we have

1. $\min_{i} L^{(x)}_{ii} > \max_{(x_i, y_i) \in D_i} \partial f_i/\partial x_i = \tau_1$ (say) and $\min_{i} L^{(y)}_{ii} > \max_{(x_i, y_i) \in D_i} \partial g_i/\partial y_i$, $\forall(x_i, y_i) = \tau_2$ (say),
2. $\max_{(x_i, y_i) \in D_i} \frac{1}{2} |\partial g_i/\partial x_i + \partial f_i/\partial y_i| < \min_{(x_i, y_i) \in D_i} \partial f_i/\partial x_i$ and $\max_{(x_i, y_i) \in D_i} \frac{1}{2} |\partial g_i/\partial x_i + \partial f_i/\partial y_i| < \min_{(x_i, y_i) \in D_i} |\partial g_i/\partial y_i|$, where the compact subset $D_i$ of $\mathbb{R}^2_+$ is the feasible domain for the $i$th patch populations.

Once again it is clear from these conditions that the stability of the population in the patch network depends upon both the local patch dynamics of the populations and their connectivity to the rest of the network. Condition (ii) above, if satisfied in a network, will be satisfied after the partition, too. It is clear from this analysis that the global stability conditions are stronger than the local stability conditions. For a connected graph, if each diagonal entry of the Laplacian is large, than naturally the Fiedler value of the Laplacian is also correspondingly large; but the converse is not always true. An example can easily be constructed, considering two path graphs $P_3$; one with unit edge weights and the other with edge weights 2 and 1. Their respective Laplacian matrices are

\[
\begin{pmatrix}
1 & -1 & 0 \\
-1 & 2 & -1 \\
0 & -1 & 1
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
2 & -2 & 0 \\
-2 & 3 & -1 \\
0 & -1 & 1
\end{pmatrix}
\]

The latter graph’s Laplacian has a larger Fiedler value (1.268) than that of the former graph (1), whereas not all three diagonal entries of the former graph’s Laplacian have been increased.

4. Exhaustive partitioning

In this section, we present a method exploring all possible partitions for smaller sized graphs such that the stability conditions are satisfied by the graph components after the cut. First, we give some common definitions (for example, see [47]) related to this procedure.

**Definition 4.1.** A **spanning tree** in a graph $G$ is a minimal subgraph connecting all nodes of the graph $G$.

**Definition 4.2.** A **cut-set** of a graph $G$ is a set of edges whose removal from $G$ leaves $G$ disconnected, provided no proper subset of these edges disconnects (in the same way) the graph.
Algorithm 1 Exhaustive algorithm.

1: Determine all $2^n - 1$ cut-sets of a graph.
   (i) select a spanning tree $T$ of the given connected graph $G$,
   (ii) determine all $n - 1$ fundamental cut-sets with respect to spanning tree $T$,
   (iii) generate all remaining cut-sets by applying result [d], as result [b] says that the set of
   fundamental cut-sets is the basis of cut-subspace $W_s$.

2: Reject all the cut-sets which give the isolated node as a component (as we want a stable
   population in each component). To decide whether a cut-set gives an isolated patch or not,
   we consider the nodes that are connected by an edge corresponding to the first non-zero
   component in the cut-set vector. If the first 1 occurs at the $i$th place in a cut-set vector, then
   this corresponds to the edge $e_i$ that connects nodes $v_p$ and $v_q$ (say). If all the edges that are
   incident on either $v_p$ or $v_q$ fall in the considered cut-set vector, then one or both nodes are
   isolated by the cut-set vector and we drop that cut-set from the list. If not, then we go to the
   next non-zero component in the same cut-set vector and repeat the above procedure until all
   nodes are checked for isolation by this cut-set.

3: Corresponding to the remaining cut-sets, determine the predator Fiedler value $\lambda^y_2$ of each
   component for linear stability and determine the nodes (say $n^x_k$ corresponds to the prey
   network and $n^y_l$ corresponds to the predator network) of each partitioned component with
   minimum edge weights incident on it for global stability.

4: List all the cut-sets that give stable partitions (i.e. $\lambda^y_2 \geq \tau$ and $n^x_k > \tau_1$, $n^y_l > \tau_2$ for each
   partitioned component, for linear and global stability, respectively). Note that a global stable
   partition is also a linear stable partition.

Definition 4.3. A fundamental cut-set with respect to spanning tree $T$ is a cut-set containing
exactly one branch of spanning tree $T$.

Definition 4.4. A ring sum of two graphs $G_1(V_1,E_1)$ and $G_2(V_2,E_2)$ is a graph consisting of the
node set $V_1 \cup V_2$ and edges that are either in $G_1$ or in $G_2$, but not in both.

Definition 4.5. Vector space associated with a graph. Let us consider the graph $G$ with edges
$e_1, e_2, \ldots, e_n$. Any subset $g$ of these $n$ edges can be represented by an $n$-tuple $X = (x_1, x_2, \ldots, x_n)$,
such that $x_i = 1$ if $e_i$ is in $g$ and $x_i = 0$ otherwise. There exists a vector space $W_G$ over Galois
field modulo 2 associated with every graph $G$, where vector addition is taken as the ring
sum of corresponding graphs, defined as $X \oplus Y = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$ and the scalar
multiplication is defined as $c \cdot X = (c \cdot x_1, c \cdot x_2, \ldots c \cdot x_n)$.

Definition 4.6. A cut-set vector is a vector in $W_G$ representing either a cut-set or a union of
degree-disjoint cut-sets in $G$.

Definition 4.7. The rank of a graph is the number of edges in any spanning tree of a connected
graph $G$.

Result [a]: The set of all cut-set vectors in $W_G$ forms a subspace $W_s$.
Result [b]: The set of cut-set vectors corresponding to the set of fundamental cut-sets with respect
to any spanning tree forms a basis for the cut-subspace $W_s$.
Result [c]: The dimension of the cut-set subspace $W_s$ is equal to the rank $r$ of the graph, and the
number of cut-set vectors (excluding $0$) in $W_s$ is $2^r - 1$.
Result [d]: The ring sum of any two cut-sets in a graph is either a third cut-set or an edge-disjoint
union of cut-sets.

The exhaustive partitioning procedure is given as algorithm 1 to obtain the desired partition
of the graph such that each resultant component has stable populations.
If we minimize (or maximize) the cost of cutting the edge(s), then we choose the cut-set which has the minimum (respectively, maximum) weight among all the cut-sets obtained in step 4. This exhaustive procedure has exponential complexity, which increases with the number of nodes in the graph but determines all the desired partitions. An illustrative example follows.

**Example.** Consider the graph $G(5,6)$ with edge weights as shown in figure 2a.

— **STEP 1:** To determine all the cut-sets, first we determine the basis elements of the cut-subspace $W_6$ of $W_G$, which is the set of fundamental cut-sets with respect to a spanning tree. There are four fundamental cut-sets of the given graph (shown by red dashed lines in figure 2b) w.r.t. the spanning tree $T$ (highlighted by blue solid lines). Thus, a basis of cut-subspace $W_6$ is given by $B = \{S_1, S_2, S_3, S_4\}$, where

$$S_1 = (1,0,0,0,1,0)^\top, \quad S_2 = (1,0,0,1,0,1)^\top,$$

$$S_3 = (0,1,0,1,0,0)^\top, \quad S_4 = (0,0,1,1,0,0)^\top.$$ 

Now the remaining cut-sets can be generated by taking the ring sum (modulo 2) of these four fundamental cut-sets. The total number of cut-sets generated by the fundamental cut-sets is equal to the number of ring-sums of the fundamental cut-sets, which is equal to $4^2 + 4^3 + 4^4 = 11$ and these are given as

1. $S_1 \oplus S_2 = (0,0,0,1,1,1)^\top$, 2. $S_1 \oplus S_3 = (1,1,0,1,1,0)^\top$, 3. $S_1 \oplus S_4 = (1,0,1,1,1,0)^\top$, 4. $S_2 \oplus S_3 = (1,1,0,0,0,1)^\top$, 5. $S_2 \oplus S_4 = (1,0,1,0,0,0)^\top$, 6. $S_3 \oplus S_4 = (0,1,1,0,0,0)^\top$, 7. $(S_1 \oplus S_2) \oplus S_3 = (0,1,0,0,1,1)^\top$, 8. $(S_1 \oplus S_2) \oplus S_4 = (0,0,1,0,1,1)^\top$, 9. $(S_1 \oplus S_3) \oplus S_4 = (1,1,1,0,0,1)^\top$, 10. $(S_2 \oplus S_3) \oplus S_4 = (1,1,1,1,1,0)^\top$, 11. $(S_1 \oplus S_2) \oplus (S_3 \oplus S_4) = (0,1,1,1,1,1)^\top$.

All the cut-sets and components of the graph induced by them are shown in table 3 (see appendix A).

— **STEP 2:** Consider the cut-set $S_1$. The first 1 is at first place, which is associated with the edge $e_1$ and nodes $v_1$ and $v_2$ are connected by $e_1$. Note that $e_5$ is the only edge other than $e_1$ that is incident on node $v_1$ and it is also in the cut-set $S_1$. Hence cut-set $S_1$ separates node $v_1$ from the remaining nodes. So, we drop cut-set $S_1$ from the list of our potential cut-sets. Similarly, we reject cut-sets $S_4, S_1 \oplus S_2, S_1 \oplus S_3, S_1 \oplus S_4, S_2 \oplus S_3, S_3 \oplus S_4, (S_1 \oplus S_2) \oplus S_3, (S_2 \oplus S_3) \oplus S_4$ and $(S_1 \oplus S_2) \oplus (S_3 \oplus S_4)$.

— **STEP 3:** Below is the list of cut-sets not yielding components that have an isolated patch.

1. $S_2 = (1,0,0,1,0,1)^\top$, 2. $S_3 = (0,1,0,1,0,0)^\top$, 3. $S_2 \oplus S_4 = (1,0,1,0,0,1)^\top$, 4. $(S_1 \oplus S_2) \oplus S_3 = (0,1,0,0,1,1)^\top$, 5. $(S_1 \oplus S_2) \oplus S_4 = (0,0,1,0,1,1)^\top$.

**Figure 2.** (a) A weighted graph $G(5, 6)$ and (b) a spanning tree $T$ and the fundamental cut-sets w.r.t. it. (Online version in colour.)


For each of these cut-sets, the Laplacian matrix pairs corresponding to the components are given as follows:

1. \[
\begin{bmatrix}
2 & -2 \\
-2 & 2
\end{bmatrix},
\begin{bmatrix}
2 & -2 & 0 \\
-2 & 5 & -3 \\
0 & -3 & 3
\end{bmatrix}
\]

2. \[
\begin{bmatrix}
3 & -3 \\
-3 & 3 \\
4 & -3 & -1
\end{bmatrix},
\begin{bmatrix}
-3 & 5 & -2 \\
-1 & 2 & 3
\end{bmatrix}
\]

3. \[
\begin{bmatrix}
2 & -2 \\
-2 & 2
\end{bmatrix},
\begin{bmatrix}
2 & -2 & 0 \\
0 & 5 & -5 \\
-2 & -5 & 7
\end{bmatrix}
\]

4. \[
\begin{bmatrix}
3 & -3 \\
-3 & 3
\end{bmatrix},
\begin{bmatrix}
3 & -3 & 0 \\
-3 & 8 & -5 \\
0 & -5 & 5
\end{bmatrix}
\]

5. \[
\begin{bmatrix}
5 & -5 \\
-5 & 5
\end{bmatrix},
\begin{bmatrix}
3 & -3 & 0 \\
-3 & 5 & -2 \\
0 & -2 & 2
\end{bmatrix}
\]

and the Fiedler values of these Laplacian matrix pairs are given by

1. \((4, 2.35)\),
2. \((6, 4.26)\),
3. \((4, 2.64)\),
4. \((6, 3.64)\),
5. \((10, 2.35)\),

respectively.

— STEP 4: We need \(\tau\) to be less than \(\lambda_2(G(5, 6)) = 3.625\) to ensure the stability of the given network \(G(5, 6)\). In particular, if \(\tau = 3\), then the revised list of potential cut-sets that satisfy both the stability conditions is

\[S_3 = (0, 1, 0, 1, 0, 0)^T, \quad (S_1 \oplus S_2) \oplus S_3 = (0, 1, 0, 0, 1, 1)^T.\]

The weight associated with these each of these cut-sets is 7 and 5. Thus, the maximum weighted cut-set that the system can tolerate is 1 and only \(S_3\) qualifies. If the objective is to minimize the cost of cutting the edges along the sustainable cut-set, then the lowest weighted cut-set is \((S_1 \oplus S_2) \oplus S_3\). From these two cut-sets \(S_3\) and \((S_1 \oplus S_2) \oplus S_3\), either both will be globally stable or neither of them. As in both cases the lowest edge weight incident on a node is 3, and if the global threshold values \(\tau_1\) and \(\tau_2\) are set as less than 3, then both the cut-sets will be globally stable.

Another example to consider is an Erdős–Rényi graph \([23]\) of 25 nodes, shown in figure 3. The graph is generated by starting with a set of distinct nodes and adding successive edges.
between them at random with probability $p = 0.5$. Using the exhaustive procedure, the most linearly stable components (corresponding to the maximum possible Fiedler value) obtained from the partitioning of the Erdős–Rényi graph are shown in figure 4.

5. Partitioning by the heuristic bisection method

For large networks, exhaustive partitioning is computationally too expensive. Thus we need a trade-off between a useful result and computational complexity. Existing min-cut algorithms, such as the spectral bisection partitioning method [48] and Kernighan–Lin algorithm [35], which tend to cut the graph with minimum possible edges, do not give the desired linearly asymptotic partitions. In this section, we provide a heuristic graph bi-partitioning algorithm, which starts with a random partition and searches locally around it for an appropriate partition that provides linearly stable components. This algorithm also sets up the basis for general partitioning problems such as $k$-way partitioning and partitioning into unequal size components.

Let $G$ be a stable graph with $n$ nodes and let it be randomly partitioned into two equally sized components $C_1$ and $C_2$ with at most one node difference in their sizes (differences occur when $n$ is odd). Now the appropriate partition with $\lambda_2^y(C_i) \geq \tau$, $i = 1, 2$, can be achieved by pushing the larger weighted edges which lie on the graph cut-set into the graph components. The motivation for doing this way is Weyl’s monotonicity theorem, which guarantees the increment in the Fiedler value if we increase the weights on the graph edges. This is done by swapping some subsets of nodes of $C_1$ and $C_2$, associated with larger weighted edges on the cut-set, while making sure that these nodes take the minimal weights away from their respective original components to the graph cut.

Let $A$ be the set of nodes from component $C_1$ and $B$ the set of nodes from component $C_2$, by swapping which we obtain the desired partition. Then the heuristic algorithm will identify the sets $A$ and $B$ sequentially in a finite ($O(n)$ node swap computations) number of steps, without considering exhaustively all possible partitions.
Algorithm 2 Heuristic bisection algorithm.

1: Obtain two equally sized (with at most one node difference) components $C_1$ and $C_2$ by an arbitrary partition of graph $G$ of size $n$.
2: Determine the difference between the external cost and the internal cost for each node in both the components $C_1$ and $C_2$.
3: Identify $a_1 \in C_1$ and $b_1 \in C_2$ such that $a_1 = \max_{s \in C_1} \{E_s - I_s\}$ and $b_1 = \max_{t \in C_2} \{E_t - I_t\}$.
4: Repeat step 1 and step 2 for the components $C_1 - \{a_1\}$ and $C_2 - \{b_1\}$ and identify the pair $(a_2, b_2)$. To avoid repetition of the nodes in the identification, ignore all previously identified $a_i$ and $b_i$, and continue until the list of pairs $\{(a_1, b_1), \ldots, (a_l, b_l)\}$ where $l$ is at most $\lceil n/2 - 1 \rceil$ is identified.
5: Interchange the first $k$ identified nodes of a component with the other component. That is, nodes $a_1, \ldots, a_k$ in $C_1$ are now in $C_2$ and nodes $b_1, \ldots, b_k$ in $C_2$ are now in $C_1$.
6: Determine the Fiedler values of both updated components (say, $C_1^k$ and $C_2^k$) obtained in step 4.
7: Choose the number (not necessarily the least) $k$ for which $\lambda_2^{\psi_i}(C_i^k) \geq \theta$, $i = 1, 2$, $\theta = \max_{r} \{r + r : \min_{i} \{\lambda_2^{\psi_i}(C_i^k) - \tau = r, \ r \geq 0\}\}$. Components $C_1^k$ and $C_2^k$ will be the desired partition of $G$ with $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k\}$.

Figure 5. A graph partition of an Erdős–Rényi graph (figure 3) with a spectral bisection algorithm. (Online version in colour.)

The external and internal costs of each node of the component subgraphs are defined as follows.

External cost of node $x$: $E_x = \sum_{x \sim y} w_{xy}$, where $x$ and $y$ belong to different components.

Internal cost of node $x$: $I_x = \sum_{x \sim y} w_{xy}$, where $x$ and $y$ both belong to the same component.

The heuristic method is given by algorithm 2. Before arriving at subsets $A$ and $B$, algorithm 2 swaps the set of nodes between $C_1$ and $C_2$ by a finite number of times and each time the size of the set varies by 1. It starts with a single node pair and then adds one node pair each time until the list of at most $(n/2 - 1)$ node pairs is exhausted, with $n$ being the graph size. At each step, we need to calculate the Fiedler value of the updated components. The maximum Fiedler valued pair among the $(n/2 - 1)$ pairs gives the
The heuristic method in algorithm 2 intrinsically tries to maximize the Fiedler values in its post-partition components. This imparts robustness to the algorithm in the sense that well-connectedness in a patched ecological network is a measure of its robustness. The present algorithm is computationally compared with the min-cut spectral bisection and Kernighan–Lin algorithms. Partitioning by the heuristic bisection method of algorithm 2 produces higher Fiedler valued components than by the min-cut spectral bisection and Kernighan–Lin algorithms. The
The graph in Figure 3 is partitioned by the Kernighan–Lin algorithm into two components (using METIS [49]) with Fiedler values 26.66 and 11.74; spectral bisection gives Fiedler values 34.88 and 37.74 (Figure 5), while algorithm 2 gives Fiedler values 37.03 and 45.77 (Figure 6).

The heuristic method in algorithm 2, starting with distinct and randomly chosen initial partitions, provides a way to generate ecologically sustainable partitions, when such stable partitions exist. From partitions generated by this method starting with 100 initial random and distinct partitions of the graph in Figure 3 and the threshold value set to $\tau = 28$, the most stable partition obtained from the initial partition shown in Figure 7 is depicted in Figure 6.

6. Conclusion and discussion

The problem of partitioning a patched ecological network into stable components has been addressed in this work by combining dynamical stability criteria with spectral graph partitioning. The threshold conditions for the existence of a cut that crosses the metapopulations network without causing local and global instabilities in the populations have been derived.

A significant role of the Fiedler value (i.e. the role of diffusion rates in the ecological sense) of the graph network and its importance is observed in deciding the metapopulations stability. We find that for stable cuts to exist the Fiedler value must be greater than some threshold value which is based on the populations’ individual patch dynamics. In ecological terms, the sustainable separation of the patches into two components is possible provided the populations’ internal patch dynamics and dispersal movement among the patches compensate each other. Global stability conditions for the metapopulations model have been derived by a contraction analysis. We find that the node with the minimum edge weight incident on it decides the global stability in the network apart from local patch dynamics of populations.

As exhaustive partitioning is too costly for practical networks, the heuristic bisection algorithm has been constructed to find the best possible cut in the neighbourhood of an initial arbitrary cut (usually obtained by a random process). The heuristic algorithm tries to maximize intra-component Fiedler values and is based on the idea of bringing the larger weighted edges from cut-sets into components by way of swapping some nodes which have high external cost and low internal cost. Using different starting cuts for the heuristic bisection algorithm, many ecologically sustainable partitions can be obtained from which the most stable partition is output. A computational comparison with the existing static min-cut algorithms (spectral bisection and Kernighan–Lin) finds that the proposed heuristic algorithm provides a better stable partition than the static min-cut algorithms.

The basic idea contained in the present work may be the basis of similar problems on networks that are directed graphs and on networks containing delays in reaction–diffusion terms.

The present study can be extended to patched ecological networks driven by reaction–diffusion dynamics containing delays, as delay differential equations can be expanded into a larger number of ordinary differential equations proportional to the number of delay intervals by the Bellman method of steps [50]. However, derivation of the specific conditions and efficient heuristic algorithms for producing stable cuts for such systems exploiting the particular topology of the network (e.g. for neural networks) and the particular structure of the delay differential equations over the network is outside the scope of the present work. We observe that dynamically stable partitioning of network systems with delays is important from a synchronization point of view [51,52].

Data accessibility. Matlab codes are supplied as electronic supplementary material to generate figures 3–7.

Authors' contributions. D.K. carried out the research, J.G. did all the computational work and S.R. guided and suggested the research problem. D.K. and S.R. contributed to writing this manuscript.

Competing interests. We declare that we have no competing interests.

Funding. D.K.'s work was partially supported by a post-doctoral fellowship awarded by the National Board for Higher Mathematics of India.

Acknowledgements. We thank the anonymous reviewers for their valuable comments that have helped in improving the article.
Table 3. All the cut-sets of graph $G(5, 6)$ and their respective components. Cut-sets highlighted in red show no isolation of a patch.

<table>
<thead>
<tr>
<th>s. no.</th>
<th>cut-sets</th>
<th>graph components</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$S_1 = (1, 0, 0, 0, 1, 0)^T = {e_1, e_5}$</td>
<td><img src="image1" alt="Graph 1" /></td>
</tr>
<tr>
<td>2.</td>
<td>$S_2 = (1, 0, 0, 1, 0, 0)^T = {e_2, e_6}$</td>
<td><img src="image2" alt="Graph 2" /></td>
</tr>
<tr>
<td>3.</td>
<td>$S_3 = (0, 1, 0, 1, 0, 0)^T = {e_2, e_4}$</td>
<td><img src="image3" alt="Graph 3" /></td>
</tr>
<tr>
<td>4.</td>
<td>$S_4 = (0, 0, 1, 1, 0, 0)^T = {e_5, e_4}$</td>
<td><img src="image4" alt="Graph 4" /></td>
</tr>
<tr>
<td>5.</td>
<td>$S_1 \oplus S_2 = (0, 0, 0, 0, 1, 1)^T = {e_6, e_5, e_4}$</td>
<td><img src="image5" alt="Graph 5" /></td>
</tr>
<tr>
<td>6.</td>
<td>$S_1 \oplus S_3 = (1, 1, 0, 1, 1, 0)^T = {e_1, e_5} \cup {e_2, e_4}$</td>
<td><img src="image6" alt="Graph 6" /></td>
</tr>
<tr>
<td>7.</td>
<td>$S_1 \oplus S_4 = (1, 0, 1, 1, 1, 0)^T = {e_1, e_5} \cup {e_3, e_4}$</td>
<td><img src="image7" alt="Graph 7" /></td>
</tr>
<tr>
<td>8.</td>
<td>$S_2 \oplus S_3 = (1, 1, 0, 0, 0, 1)^T = {e_1, e_2, e_6}$</td>
<td><img src="image8" alt="Graph 8" /></td>
</tr>
<tr>
<td>9.</td>
<td>$S_2 \oplus S_4 = (1, 0, 1, 0, 0, 1)^T = {e_1, e_3, e_6}$</td>
<td><img src="image9" alt="Graph 9" /></td>
</tr>
<tr>
<td>10.</td>
<td>$S_3 \oplus S_4 = (0, 1, 1, 0, 0, 0)^T = {e_1, e_2}$</td>
<td><img src="image10" alt="Graph 10" /></td>
</tr>
<tr>
<td>11.</td>
<td>$(S_1 \oplus S_2) \oplus S_3 = (0, 1, 1, 0, 1, 1)^T = {e_2, e_5, e_6}$</td>
<td><img src="image11" alt="Graph 11" /></td>
</tr>
<tr>
<td>12.</td>
<td>$(S_1 \oplus S_2) \oplus S_4 = (1, 0, 0, 1, 0, 1)^T = {e_5, e_6}$</td>
<td><img src="image12" alt="Graph 12" /></td>
</tr>
<tr>
<td>13.</td>
<td>$(S_1 \oplus S_3) \oplus S_4 = (1, 1, 1, 0, 1, 0)^T = {e_1, e_5} \cup {e_2, e_3}$</td>
<td><img src="image13" alt="Graph 13" /></td>
</tr>
<tr>
<td>14.</td>
<td>$(S_2 \oplus S_3) \oplus S_4 = (1, 1, 1, 1, 0, 1)^T = {e_1, e_4, e_6} \cup {e_2, e_3}$</td>
<td><img src="image14" alt="Graph 14" /></td>
</tr>
<tr>
<td>15.</td>
<td>$(S_1 \oplus S_2) \oplus (S_3 \oplus S_4) = (0, 1, 1, 1, 1, 1)^T = {e_2, e_5, e_6} \cup {e_3, e_4}$</td>
<td><img src="image15" alt="Graph 15" /></td>
</tr>
</tbody>
</table>


47. Deo N. 2017 *Graph theory with applications to engineering and computer science*. Mineola, NY: Courier Dover Publications.