The nature of equations and the equations of nature

Sana Jahedi∗and James A. Yorke†

Abstract

For a system of \(N\) equations in \(N\) unknowns, \(F(X) = C\), we say a solution \(X\) is robust if for all \(C'\) sufficiently close to \(C\), there exists a solution of \(F(X') = C'\). We write \(F(X) = (f_1(X), \ldots, f_N(X))\) and \(X = (x_1, \ldots, x_N)\). We restrict the coordinate functions \(f_i\) so that they depend only on certain of the \(x_j\)'s. Which set of dependencies allow robust solutions to exist has been well understood when \(F\) is linear. Define the “structure” graph \(G(F)\) with \(N\) nodes with an edge from node \(j\) to node \(i\) if \(f_i\) is allowed to depend upon \(x_j\). For some such graphs \(G\), there are no robust solutions for any \(F\) for which \(G = G(F)\).

We adopt a simple “cycle-covering” criterion from linear systems theory and extend its use to all \(C^1\) functions and piecewise–\(C^1\) functions. We show the following. (1) If the nonlinear system \(F(X) = C\) has a robust solution, then its graph \(G(F)\) must be cycle coverable. (2) If a graph \(G\) is cycle coverable, then “almost every” \(F\) with \(G(F) = G\) has robust solutions. (3) For a large class of functions \(F\), such as when each \(f_i\) is polynomial of the \(x_j\)'s – or even when each \(f_i\) is a ratio of such polynomials – if there is one robust solution, then almost every solution for any \(C\) is robust. There are many networks in nature to which these ideas might apply. For instance; social, neural, transportation, energy flows, electrical and ecological. As an application, here we focus on ecological networks.

Keywords: structure systems; nonlinear systems; robust solutions; missing links; link prediction; complex networks; generic rank; prevalence; dilation

1 Introduction

Our understanding of the structure of complex networks is inadequate in many areas. We need tools to explain why some networks are robust to perturbations, but some are vulnerable. Differential equations and graph theory has been used as ubiquitous tools by scientists in the studying a vast group of networks[1, 2, 3, 4, 5, 6]. Even though there is an extensive literature available on studying linear system of equations using graphs and matrices, the interactions between community members in most cases are nonlinear. In the following, first we will recall some known terminology of graph theory and then we will relate it to system of equations. We report here some criteria for

∗Department of Mathematics and Statistics, University of New Brunswick. Corresponding author’s email: s.jahedi@unb.ca
†IPST, Mathematics, and Physics, University of Maryland College Park.
guaranteeing when a system of equations has a “robust” solution. A robust solution is a solution of Eq. 1 which exists despite any tiny changes in any constants appearing in the Eq. 1.

\[ F(X) = C. \] (1)

where \( X = (x_1, \ldots, x_N) \), \( F(X) = (f_1(X), \ldots, f_N(X)) \) and \( C = (c_1, \ldots, c_N) \). Throughout this paper \( F \) is a continuously differentiable function (i.e., it is \( C^1 \)) on some open region in \( \mathbb{R}^N \). Alternatively, we can assume the equations and variables are complex valued. Then \( \mathbb{R}^N \) is replaced everywhere by \( \mathbb{C}^N \), and the results of this paper remain true.

Note that not all the \( f_i \)'s in the system of equations in Fig. 1(a) are allowed to depend on all the \( x_j \)'s. For example, \( f_1 \) depends on \( x_2, x_3, \) and \( x_4 \) but not \( x_1 \). We call such a system a structured system: we are told of the dependencies that are allowed, and we must consider all such smooth systems. Such structured systems are common in linear control theory and in other fields[7, 8]. These restrictions can be visualized graphically by assigning a graph to the equation. In the graph Fig. 1(c) there is one node for each equation of the system in Fig. 1(a). There is an edge from node \( j \) to node \( i \) if \( f_i \) is allowed to depend on the \( x_j \).

\[
\begin{align*}
f_1(x_2, x_3, x_4) &= c_1 \\
f_2(x_1, x_2, x_3, x_4) &= c_2 \\
f_3(x_2) &= c_3 \\
f_4(x_2, x_3) &= c_4
\end{align*}
\]

\[ S := \begin{pmatrix}
0 & f_{12} & f_{13} & f_{14} \\
0 & f_{21} & f_{22} & f_{23} & f_{24} \\
0 & f_{32} & 0 & 0 \\
0 & f_{42} & f_{43} & 0
\end{pmatrix} \]

Figure 1: A robust structured system of equations (a) with its structure matrix (b) and its directed graph (c). This figure represents a family of systems. The equations show which \( x_j \) each \( f_i \) is allowed to depend on. Hence the form \( f_1(x_2, x_3, x_4) \) means \( f_1 \) cannot depend upon \( x_1 \), and it may or may not depend on \( x_2, x_3, \) and \( x_4 \). Almost every system of this form will have “robust” solutions for an open set of \( C \) values; see Sec. 2. The matrix \( S \) in (b) is an alternative representation of the information in (a); whenever \( f_i \) is allowed to depend on \( x_j \) in (a), there is a variable \( f_{ij} \) with an unassigned value in \( S \) in (b) and there is an edge from node \( j \) to node \( i \) in (c). The shaded \( f_{ij} \) in (b) and shaded edges (red in the online version) in (c) constitute a cycle \( (1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \rightarrow (1) \) that passes through all nodes in the graph. Hence the graph is “cycle coverable”, which is defined in the Introduction. If any of the shaded edges in (c) is eliminated, the corresponding graph will not be cycle coverable and there cannot be any robust solutions for any \( C \). The rest of the edges are inconsequential.

We give a simple criterion, that the graph must be “cycle coverable”, that completely describes which graphs represent systems that can have “robust” solutions. A cycle is a path in a graph that follows directed edges which starts and ends at the same node, with no node occurring more than once. Two cycles are disjoint if they have no nodes in common. We say a graph \( G \) has a cycle cover or is cycle coverable if there is a collection of pairwise disjoint cycles that together cover all the nodes. For example the graph in Fig. 2(c) is an example of a cycle coverable graph while the graph in Fig. 2(c) is not. For basic references on the graph theory refer to [9, 10, 11].
Recognizing all the community connections in a complex network is not always possible\[12,\ 13\]. For example, considering protein-protein interaction networks, gene regularity networks, or metabolism networks, not all the connections are known\[14\]. Therefore, when modeling interactions in a complex network, we might face their corresponding graphs are not cycle coverable even though we know from experiment that those networks are robust, which suggest that there are some connections in the community which have not been captured well, in other words, there are some missing links. This motivated us to investigate what we call bottlenecks (or “dilations” in control theory\[15,\ 16,\ 17,\ 18\]).

Bottlenecks are specific subgraphs that prevent cycle coverage. We thus have another goal: finding the bottlenecks and determining which new added edges to the graph can alleviate the bottleneck. We introduce the concept of bottleneck nodes, the nodes whose shortage of edges make a graph fragile. To identify bottleneck nodes we propose a method in which we use structure matrix of a graph. The structure matrix $S = (f_{ij})$ of a graph is a matrix where the entry $f_{ij}$ is set to 0 if there is no edge from node $j$ to node $i$; otherwise it is unspecified.

For any graph, almost every choice of entries $f_{ij}$ in its structure matrix gives a matrix with same rank. That rank $r$ is called the generic rank of the structure matrix of the graph.

It is well known that a graph of a linear system is cycle coverable if and only if the generic rank of its structure matrix is $N$ (the same as the number of nodes)\[10,\ 19\]. However, this fact does not tell us which nodes are the key nodes or which edges are the key connections in the robustness of the graph. The $4 \times 4$ matrix in Fig. 1(b) has rank 4 since it has determinant which is non-0 for almost every choice of the four numbers in the product. It is independent of the remaining 6 unspecified numbers. If a graph is not cycle coverable, what is the smallest number of edges that must be added and where must they be to make a fragile structure robust? We answer these questions in sections 3 and 4.

## 2 Robust Solutions

The function $F$ in Eq. 1 may depend continuously on additional parameters $\hat{C}$ in addition to the constants $C$. Consider the continuously differentiable system of $N$ equations in $N$ unknowns,

$$F(X, \hat{C}) = C. \tag{2}$$
where $C$ and $X$ are in $\mathbb{R}^N$ and $\hat{C} = (c_{N+1}, c_{N+2}, \ldots, c_{N^*})$ for some $N^*$. If $X_0$ is a robust solution for some $C_0$ and some $\hat{C}_0$, then a solution always exists for all sufficiently small changes in $C$.

The matrix $DF(X) = \left(\frac{\partial f}{\partial x}(X)\right)$ denotes the Jacobian of $F$ at $X$. If $\det DF(X_0) \neq 0$, then the Implicit Function Theorem says that a solution always exists for sufficiently small changes in both $C$ and $\hat{C}$. The reader may imagine any number of parameters that are free to vary. We leave it to the reader to remember that $F$ can depend on additional parameters.

The nonsingularity of the Jacobian matrix at a point $X$ is a sufficient condition for robustness of $X$, but not necessary. For example, for the scalar equation $x^3 = c$, the solution $x = 0$ for $c = 0$ is robust even though its derivative is 0.

A solution $X_0$ of Eq. (I) is called linearly robust if $\det DF(X_0) \neq 0$. We say $F$ is robust if for some $C$ there is a robust solution $X$. If there is no $X$ at which $F$ is robust, we say $F$ is fragile. We say $F$ is densely robust if $F$ is robust for an open dense set of its domain. If $F$ is robust on a countable dense set $\{X_i\}$, then Sard’s Theorem (below) implies it is linearly robust on a dense set of $X$ and the dense set is open since $\det DF(X)$ is continuous.

Write $G(F)$ for the (structure) graph associated with $F$ in Eq. (I); it has $N$ nodes, and it has a (directed) edge from the $j^{th}$ node to the $i^{th}$, written as $j \rightarrow i$, precisely when $f_i$ depends on $x_j$.

**Assumption.** Throughout this paper we assume each graph has $N$ nodes, and each node has an outgoing edge and an incoming edge. Furthermore, there is at most one edge between any pair of nodes.

When something, either a solution, or a function, or a graph, is not robust, we say it is fragile. For some examples of robust and fragile networks refer to Fig. 3.

We say $F$ respects (the structure of) a graph $G$ and write $F \in \mathcal{F}(G)$ if $G$ and $G(F)$ have the same number of nodes and $G(F)$ is a subgraph of $G$. That is, $G$ has an edge $j \rightarrow i$ whenever $f_i$ depends on $x_j$, and $G$ may have additional edges. $\mathcal{F}(G)$ represents the set of functions that respect $G$. For example, the identically 0 function respects every graph.

In part (I) of the next theorem, “almost every” is meant in the sense of “prevalence” which is described after the theorem.

**Theorem 1. Cycle Covering Theorem (CCT).**

(I) If a graph $G$ is cycle coverable, then almost every $F \in \mathcal{F}(G)$ is densely robust.

(II) If $F$ is $C^1$ and $G(F)$ is not cycle coverable, then for almost every $C$, Eq. (I) has no solutions.

Perhaps this might be called the Fundamental Theorem of Structured Systems.

Part (II) of Thm. (I) is an application of our Bottleneck Lemma, so we postpone its proof. It is in the Appendix. The concept of “prevalence” is a replacement for infinite dimensional spaces of the concept “almost everywhere” for finite dimensional spaces [20].

Let $\mathcal{V}$ be a real topological vector space and let $S$ be a Borel-measurable subset of $\mathcal{V}$. In particular both open and closed sets are Borel sets and they are the cases we encounter here. $S$ is said to be prevalent if there exists a finite-dimensional subspace $\mathcal{P}$ of $\mathcal{V}$, such that for all $v \in \mathcal{V}$, for almost every $p$ in $\mathcal{P}$, $v + p \in S$. [21] [22].

First some notation. Let $\mathcal{L}(G)$ be the space of linear functions (or matrices) that respect $G$. Notice that $\mathcal{L}(G)$ is a finite dimensional subspace of $\mathcal{F}(G)$.
Proof of part (I) of theorem. Assume $G$ is a cycle coverable graph. Then almost every (in the sense of Lebesgue measure) matrix in $L(G)$ is nonsingular\cite{7,19}.

For any $X$, let $P(X)$ be the set of $F$ in $F(G)$ that are linearly robust at $X$. Notice that $P(X)$ is an open set. To show that $P(X)$ is prevalent, it suffices to show that for every $F \in F(G)$ and for almost every $A \in L(G)$, we have $F + A \in P(X)$. If $F \in F(G)$ and $A \in L(G)$, then $F + A \in F(G)$. Then $D[F + A](X) = DF(X) + A$ is in $L(G)$ and is nonsingular for almost every $A \in L(G)$. Hence $P(X)$ is prevalent for every $X$.

Let $X_n$ for $n = 1, 2, 3, \cdots$ be a dense set in the domain of $F$. Define the set $\mathcal{M}(X)$ to be the set of $A \in L(G)$ for which $D[F + A](X)$ is non-singular. Since each $P(X_n)$ is prevalent for each $n$, $\mathcal{M}(X_n)$ has full measure.

For any $F \in V$, $\mathcal{M}$ is a set of full measure. From measure theory, the intersection of a countable number of full measure sets has full measure. Hence the set of $F$ that are non singular at every $X_n$ is Prevalent. Notice that for each such $F$, $DF$ will be nonsingular on an open set that includes a neighborhood of each of the $X_n$. Hence almost every $F$ is nonsingular on an open dense set.□

3 Bottlenecks and a shortage of edges

To demonstrate that a graph is cycle coverable, it is sufficient to find a covering. Many graphs are not cycle coverable. Here we introduce a criterion for identifying these. Let $B$ be a set of nodes in graph $G$.

The **forward set of $B$**, denoted by $B^\rightarrow$, is the set of all nodes $g$ in $G$ for which there is an edge starting at a node in $B$ and ending at $g$.

The **backward set of $B$**, denoted by $B^\leftarrow$, is the set of all nodes $g$ in $G$ for which there is an edge starting at $g$ and ending at a node in $B$.
Write $\#(M)$ for the number of nodes in a set $M$.

For $k > 0$, we say a pair of sets of nodes $B$ and $B^\rightarrow (B^\leftarrow)$ is a $(k)$-forward (or -backward) bottleneck if $B$ has exactly $k$ nodes more than $B^\rightarrow$ (or more than $B^\leftarrow$, respectively).

We refer to $B$ as the bottle (and usually color its nodes blue) and $B^\leftarrow$ (or $B^\rightarrow$, resp.) as the neck (usually colored red). For simplicity we sometimes refer to the nodes in $B$ as bottleneck nodes. The word “Bottleneck” connects the B in bottle to the K in neck. Some nodes can be in both the bottle and the neck. These nodes are colored half blue and half red as in Fig. 5b. When all edges are bi-directional, “backward” and “forward” bottlenecks are the same.

A system that has a bottleneck will sometimes have many. For each bottle $B$ we call $\ker(B)$ $= \#(B) - \#(B^\rightarrow)$ the kernel dimension of $B$. Considering all bottles $B$, denote the maximum value of $\ker(B)$ by $k_{\text{max}}$. Figure 4(a) has a forward 6-bottleneck whose bottle $B$ consists of the 12 blue nodes. It can be split in two, with one bottle being nodes 16 and 17 and the other being $B'$, the remaining 10 nodes. Within $B'$, there is a bottle consisting of nodes 41, 42, and 43 and another consisting of 45, 46, and 47. In contrast Figure 4(b) has many more edges from blue nodes to red so that while $B$ is still the bottle of a 6-bottleneck, it has no subsets that are bottles.

We can say that $B$ is a max bottle if $\ker(B) = k_{\text{max}}$.

Of those sets $B$ whose kernel is $k_{\text{max}}$, there is a $B$ with the fewest or minimum number of nodes. It is unique. We can call it the minimax bottle: the maximum $k$ using the minimum number of nodes since it is both mini and max. In the proof of our Bottleneck Lemma, the bottle $B$ we construct is the minimax bottle. In Fig. 4(a) the blue nodes are a bottle of a 6-bottleneck. Adding node 11 to $B$ creates a bigger bottle of a 6-bottleneck that is not a minimax bottle. Adding node 11 is not useful. It is an illustration of a non max bottle. Our “Bottleneck Lemma” below is a key to why some systems have no robust solutions.

Because the forward and backward bottles play such similar roles, we generally only refer to the forward bottles to avoid repetition. A forward bottleneck for a matrix $M$ is a backward bottleneck for the transpose of $M$. They have the same graph except that the direction of each edge is reversed.

**Bottleneck Lemma.** A graph $G$ is cycle coverable if and only if there is no (forward) bottleneck.

**Proof.** To show that the existence of a bottleneck implies there is no cycle covering, assume there is a bottleneck, $B$ and $B^\leftarrow$. Consider any collection of disjoint cycles (which might not be a covering). Whenever a cycle has a node in the bottle $B$ of the bottleneck, the next node on the cycle must be in the neck, $B^\leftarrow$. Hence there be at least as many neck nodes as bottle nodes. But there are more bottle nodes than neck nodes. Therefore the collection of cycles cannot be a covering.

Now assume $G$ is not cycle coverable. We will prove there exists a bottleneck. Since $G$ is not cycle coverable, the structure matrix $S$ associated with $G$ is singular. Let $r$ denote the generic rank of the structure matrix $S$, where $r < N$. We will show there is a forward $k$-bottleneck where $k = N - r$. By the symmetry of the argument there will also be a backward $k$-bottleneck.

Theorem 3.2 of Kang Li et al shows that for an $N \times N$ structure matrix, the generic rank is $r$ if and only if there exists a set $E$ of $r$ edges $b_i \rightarrow c_i$ for $i = 1, 2, \cdots, r$ such that the $b_i$ are distinct and the $c_i$ are distinct, and furthermore there is no such set of $r + 1$ edges. $S$ is singular if and only if $r < N$.

We write $E_{\text{left}} = \{b_i\}$ and $E_{\text{right}} = \{c_i\}$ for the sets of left and right ends of the edges in $E$, and $N$ for the set of all nodes. Define $B_0 := N - E_{\text{left}}$ to be the set of $k = N - r$ nodes that are not in $E_{\text{left}}$. 
Each \( b^* \) in \( B_0 \) has one or more edges \( b^* \rightarrow c^* \) for some \( c^* \). All such \( c^* \) must be in \( E_{\text{right}} \) since otherwise \( b^* \rightarrow c^* \) would be an \( r+1 \)th edge in \( E \). We will define an increasing set of \( B_j \).

For any set \( A \) of nodes, let \( \text{Left}(A) \) be the set of left ends for the edges in \( E \) whose right end is in \( A \). We now define \( R_j \) and \( B_{j+1} \) iteratively. Define for \( j \geq 0 \),

\[
\begin{align*}
R_{j+1} &= B_j^+, \\
B_{j+1} &= B_0 \cup \text{Left}(R_{j+1}).
\end{align*}
\]

We claim that each \( R_j \) is a subset of \( E_{\text{right}} \) for all \( j \geq 0 \). Suppose otherwise. Let \( j^* \geq 0 \) be the smallest \( j \) for which \( R_{j^*+1} \) is not a subset of \( E_{\text{right}} \). Then there is some node \( r_{j^*+1} \in R_{j^*+1} \) that is not in \( E_{\text{right}} \). There is \( b_{j^*} \in B_{j^*} \) that has an edge \( b_{j^*} \rightarrow r_{j^*+1} \).

If \( j^* = 0 \), then \( b_{j^*} \rightarrow r_{j^*+1} \) is an edge in the graph, neither of whose ends are in \( E_{\text{left}} \) and \( E_{\text{right}} \). Hence it can be added to \( E \). But \( E \) was chosen as large as possible. Hence \( j^* > 0 \).

By definition of \( B_j \) for \( j \geq 1 \), there is an edge \( b_{j^*} \rightarrow r_{j^*} \) in \( E \). In particular, \( r_{j^*} \) is in \( R_{j^*} \). Hence there is a node that we will call \( b_{j^*} \) in \( B_{j^*+1} \). Repeating this procedure yields a zig-zag path from \( r_{j^*+1} \) back to some node \( b_0 \) in \( B_0 \), yielding zig-zag path of \( 2j^*+1 \) edges in which the directions \( \rightarrow \) and \( \leftarrow \) alternate:

\[
b_0 \rightarrow r_1; r_1 \leftarrow b_1; b_1 \rightarrow r_2; \ldots; r_{j^*} \leftarrow b_{j^*}; b_{j^*} \rightarrow r_{j^*+1}
\]

The \( j \) backward \( b_i \rightarrow r_i \) edges of this path are all in \( E \). Let \( E' \) be the collection of edges in \( E \) except that we remove the \( j \) edges \( b_i \rightarrow r_i \) from \( E \) and substitute the remaining \( j+1 \) edges

\[
b_0 \rightarrow r_1; b_1 \rightarrow r_2; \ldots; b_j \rightarrow y,
\]

Now we have a set \( E' \) of \( r+1 \) edges with distinct left ends and distinct right ends, which contradicts the definition of \( r \). Hence there is no such \( j^* \), proving the claim.

Therefore, the increasing sequence \( B_i \) eventually stops increasing as do the \( R_i \). Define \( B \) to be the largest \( B_i \) and define \( R \) to be the final and largest \( R_i \). Then \( B = B_0 \cup \text{Left}(R) \). Since \( B_0 \) has \( k \) nodes, \( B \) has \( k \) more nodes than \( R \), and \( B^\rightarrow = R \). So \( B \) is the bottle and \( R \) is the neck of a \( k \)-bottleneck. \( \square \)

Existence of a forward \( k \)-bottleneck means the system can be made robust if \( k \) edges are carefully added, edges \( b \rightarrow c \) where \( b \) is a node in \( B \) and \( c \notin B^\rightarrow \). The edges must be added so that \( B \) and \( B^\rightarrow \) have equal numbers of nodes.

This demonstration that there is bottleneck does not make it easy to find the bottle. For example we have not told how to find the set \( E \) of edges.

The next section will provide a simple-to-code approach for finding the bottle \( B \).

4 Identifying bottleneck nodes

Finding out where a bottleneck is located or which species are included in a bottleneck is not always straightforward. Next result will help us to exactly locate the bottleneck.

Definition 1. A vector \( V \) is called a null-vector for the graph if it is an eigenvector corresponding to zero eigenvalue of the structure matrix associated with the graph.

Node \( i \) is called a null node if there exists a null-vector \( V = (v_k)_{k=1}^n \) such that \( v_i \neq 0 \). For a graph \( G \), we write \( \text{Null}(G) \) for the dimension of the null space of the structure matrix of the graph.
Theorem 2. Let $B_{null}$ be the set of null nodes of a graph. Then the pair $B_{null}$ and $B_{null} \rightarrow$ is the forward minimax bottleneck.

Another remarkable result is that the pair of sets $B_{null}$ and $B_{null} \rightarrow$ is a $Null(G)$-forward bottleneck.

We leave it to the reader to show that for every node $b$ of the minimax bottle, there is a null vector whose $b$ coordinate is non-zero.

Next an example is provided to show how Thm. 2 helps us to find bottleneck using simple commands.

This is a simple example to show how to use Thm. 2 to find out the bottlenecks. This example is done for the network represented in fig 3c.

In Matlab
\begin{verbatim}
syms 'f%d%d' [5 5] % write this to define free parameters
S=[0, f12, 0, f14, 0; f21, f22, f23, 0, f25; 0, f32, 0, f34, 0; f41, 0, f43, f44, f45; 0, f52, 0, f54, 0];
null(S) % This command calculates the basis for null space of S
\end{verbatim}

In Maple
\begin{verbatim}
with(LinearAlgebra):
S=Matrix(5,5,[0,f12,0,f14,0,f21,f22,f23,0,f25,0,f32,0,f34,0,f41,0,f43,f44,f45,0,f52,0,f54,0]):
NullSpace(S) # this command calculates the basis for null space of S
\end{verbatim}

output:
\[
\begin{pmatrix}
-f_23 & f_45 & 0 & f_24 & f_34 \\
-f_14 & f_23 & 0 & f_45 & f_21 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

The result declares that null space is one dimensional. Therefore, there exists a one-bottleneck. The nodes 1,3 and 5 are the bottle of the bottleneck, notice components 1,3 and 5 of the null-vector can be nonzero.

5 What bottlenecks tell us about equations

The set of all solutions $\{(X,C)\}$ of $\{X : F(X) = C\}$ is a nonlinear copy of $\mathbb{R}^N$ because it can be written as $\{(X,F(X)) : X \in \mathbb{R}^N\}$. But what does that tell us about the set $\{X : F(X) = C\}$ for each fixed $C$?

When there is a cycle covering, for almost every $C$, if there is a solution of $F(X) = C$, each such $X$ is isolated. For example in the 1-dimensional equation $\sin(x) = c$, there are robust (isolated) solutions for $-1 < c < +1$.

When there is a bottleneck, the system can be split into parts, one part of which is overly determined so there are usually no solutions. Another part is under-determined. If the first part has a solution, we can expect the second part to have surfaces of solutions. (There can be a third part which is has isolated solutions.)

**Backward bottlenecks.** Suppose a graph of equations has a backward bottleneck where the bottle $B$ has $n+k$ nodes and the neck $K = B^\leftarrow$ has $n$. If $n = k = 1$, we can number the nodes in $B$ as 1 and 2 and the only node in $K$ as 3. All edges coming to the bottle $B$ must come from the single node of $K$. The equations then are restricted to the form.

\[ f_1(x_3) = c_1, \]  \hspace{1cm} (3)  
\[ f_2(x_3) = c_2, \]  \hspace{1cm} (4)  
\[ f_i(x_1, \ldots, x_N) = c_i \text{ for } i = 3, \ldots, N \]  \hspace{1cm} (5)
The variable $x_3$ is over-determined (more equations than variables), and the others are under-determined.

As we vary $x_3$ we get a path in $\mathcal{R}^2$. Almost all $(c_1, c_2)$ are not on the path and therefore have no solutions.

But suppose we choose $(c_1, c_2)$ on the path so that there is a solution, $x_3$. Insert this fixed value into the remaining equations in (5), yielding $N - k - n$ equations in $N - n$ unknowns. Hence there are $k$ more variables than equations. Assuming now that $F$ is very smooth, say $C^\infty$, Sard’s Theorem says that for almost every choice of the remaining $N - k - n$ $c_i$’s, the selection of $c_i$’s is a regular value. For such a regular value, the solutions (if there are any) lie on a $k$-dimensional smooth surface. In summary, in the case where there is no loop covering, there are almost no choices of $C$ for which there are solutions, but if $C$ is chosen to be a special value where there is a solution, then for almost all choices of the remaining $c_i$, there is a $k$-dimensional surface of solutions if there are any. This $k$-dimensional set of solutions came as one of the many surprises we had in developing this theory.

**Forward Bottlenecks.** When there is a backward $k$-bottleneck, there is also a forward $k$-bottleneck. If $n = k = 1$, we can renumber the equations and variables so that $x_N$ and $x_{N-1}$ appear in only one equation, Eq. (7). Then the other $N - 2$ variables are required to satisfy $N - 1$ equations, which is over determined.

$$f_i(x_1, \ldots, x_{N-2}) = c_i \text{ for } i = 1, 2, \ldots, N - 1,$$

$$f_N(x_1, x_2, x_3, \ldots, x_N) = c_N,$$

(6) (7)

### 6 Ecological networks

One of the key challenges ecologists confront today, is to understand how interactions between species impact community structure, species coexistence, and biodiversity [23, 24, 25]. Ecologists have attempted to respond to such questions by visualizing a complex ecological community as a network [26, 27, 28], mathematically represented as a directed graph; see Figs. 2 and 4.

Our initial motivation for this investigation was to determine which ecological networks have one or more robust steady-states (i.e., solutions). In most ecological networks each species may be connected to, i.e., may be directly influenced by, many other species [29], though some of those influences might be quite weak. There has been no general criterion in the ecological literature for determining when a robust steady state exists.

Our results allow investigators to create “knock-out” versions of networks with many fewer links to test the effect of adding or removing interactions, testing which network interactions have no effect on the existence of robust steady states. Our results provide some answers.

When there are $N$ species in an ecological network, their interactions can often be modeled as

$$\frac{1}{x_i} \frac{dx_i}{dt} = -c_i + f_i(x_1, \ldots, x_N), \quad i = 1, 2, \ldots, N$$

(8)

where $x_i > 0$ is the population density of the $i^{th}$ species, $c_i$ is a constant, and $f_i(x_1, \ldots, x_N)$ expresses how the growth rate of the $i^{th}$ species depends upon the network [1].

We say $X$ is a steady state if all species remain at a constant level, i.e., $\frac{dx_i}{dt} = 0$ for all $i$, so $X$ is a solution of Eq. (1).
In a graph representing biological networks, each node represents a species. An arrow \( j \to i \) from species \( j \) to species \( i \) means species \( j \) “directly influences” the density of species \( i \). Therefore most edges are bi-directional. For example an edge which represents interactions between predator-prey is bi-directional. The reader might be interested in what happens when some of the edges are uni-directional. Our results cover all the cases. The mere presence or even the smell of a top predator may have a major effect on prey species while the prey species has almost negligible impact on the predator. Such situations could be modeled using uni-directional edges.

**The Competitive Exclusion Principle (CEP)** is a statement that two predators that don’t interact directly with each other and depend on the same resource cannot coexist.

The Competitive Exclusion Principle asserts that if there are more predator species than resource species, then there is no robust steady state. This heuristic statement is easily criticized as being unclear. To be precise we resort to a mathematical state. In other words, if they are coexisting in a natural environment, then some other factor is involved and the ecologist may want to figure out what that factor is. If the steady-state of the network has equations of the form in Fig. 2(a), then there is no robust steady state. Figure 2(b) is a graphical equivalent of (a). We now know that having no robust steady state is equivalent to saying (b) is not cycle coverable by Thm. 1.

In his paper [30] on CEP, ecologist S. A. Levin described a biological network that has the graph (b) where there is only one predator, species (or node) 2, that is shared by two prey species, species 1 and 3. He found again there can be no robust steady state. The equations and graphs are the same for Levin’s case and the CEP. The cases are different however. If species \( i \) is a predator and \( j \) is a prey, increasing the prey will increase the predator, i.e. \( f_{ij} = \frac{\partial f_j}{\partial x_i}(X) > 0 \), but increasing predator will decrease the prey, that is \( f_{ji} = \frac{\partial f_j}{\partial x_i}(X) < 0 \). Hence for the CEP, \( f_{12} \) and \( f_{32} \) are positive while \( f_{21} \) and \( f_{23} \) are negative. For Levin’s case, the signs are reversed.

We can extend Levin’s insight instead to every graph including those that are robust. Every edge can be assigned a sign, positive or negative, and each choice for all the edges yields different biology.

For this fragile graph 2, there are additional possibilities. Two of the species 1 and 2 might be mutualistic, with both terms positive, or antagonistic with both negative. Every choice of positive (+) and negative (-) is a case where there is no robust steady state. There are 10 different choices of these terms if we equate cases that are left-right symmetric. We leave the possible choices for the term \( f_{22} \) to the reader.

\[
S = \begin{pmatrix}
0 & \pm & 0 \\
\pm & \pm & \pm \\
0 & \pm & 0
\end{pmatrix}
\]

Substantial research aims at discovering what stabilizes ecological networks. Some have found ways to promote robustness [31, 32], but there is still a gap; is a web robust? Fig. 4a is “schematic representation of an ecological graph” from Fig. 1 in Solé et al. [33] though they do not discuss whether there are equilibria. We show their graph is fragile. Our methods show that there is a 6-species bottleneck.

Gross et al. [31] suggested two “universal” rules: Food-web stability is enhanced when (i) “species at a high trophic level feed on multiple prey species”, and (ii) “species at an intermediate trophic level are fed upon by multiple predator species.” Stability is hard to assess. Robert T. Paine [34] showed that eliminating one “keystone” species (starfish) can cause the ecosystem to collapse. These rules suggest adding edges might enhance the robustness of the ecological network, but it
is not the full story. Fig. 4b is made by adding 22 bi-directional (green) edges to Fig. 4a, adding edges throughout the network, but the resulting network, Fig. 4b is not yet robust. Our theory suggests where the edges must be added to make a system robust.

Write $B$ for the set of blue nodes and $B \rightarrow$ the red nodes. Since $|(B)| = 12$ and $|(B \rightarrow)| = 6$, $B$ is a 6-bottleneck so the graph cannot be robust. At least six forward edges starting in $B$ and not ending in $B \rightarrow$ must be added before the reduced graph is robust.

Figure 4: A bottleneck makes a network fragile. The six X’s mean six species must be eliminated before it becomes robust – unless carefully chosen edges are added. (a): This graph is reproduced from Fig. 1 of Solé et al. [33] (b): Let $B$ denote the 12 nodes shaded blue. Then the forward nodes $B \rightarrow$ consists of 6 nodes shaded red. Hence $B$ is a 6-species bottleneck. At least six edges that increase the set $B \rightarrow$ must be added to make the graph robust. 22 bi-directional edges have been added, shown in green, adding at least one edge to every node, and edges between every pair of layers, and edges within each layer except the lowest. These make no difference to the robustness. $B$ is still a 6-species bottleneck.

Fig. 3d is representing interactions between species in four different trophic levels. In this figure each blue node species on each level is connected only to the species in the adjacent trophic level, though some edges connect reds to reds. Let $N_1, N_2, N_3, N_4$ denote the number of species in each of these “trophic” levels, listing from the bottom to the top. This type of “trophic” graph cannot be cycle coverable unless the total number of species in the odd-numbered levels, $N_{\text{odd}} = N_1 + N_3$, equals the total of species in the even-numbered levels, $N_{\text{even}}$. Otherwise, let $B$ consist of all the species either in the odd-numbered layers or in the even-numbered layers, whichever is numerically greater. Then $B$ is the bottle of a bottleneck, and $B \rightarrow$ is the neck. Here it consists of the remaining species. Here $N_{\text{odd}} = 8 > N_{\text{even}} = 5$, so by our Bottleneck Lemma at least 3 edges must be added. There are other requirements too, which we leave to the reader to discover, such as if $N_1 > N_2$, then the graph is not cycle coverable.

In biological networks often the densities of some species are dependent on an intermediate
species, which often modelers incorporate the intermediate variable implicitly. For example consider the structured equations given in Fig. 5a. These steady state equations demonstrate that the densities of species $x_3$ and species $x_4$ both depend on total number of $x_1 + x_2$. The structure graph given in Fig. 5a is not robust even though the graph is cycle coverable. This type of implicit structure is not supported by our theory. Hence, when we introduce an intermediate species, we first rewrite the explicit structure by introducing a new variable and we are forced to add a new constant $c_5$. For example, in Fig. 5b we add the variable $x_5 = x_1 + x_2$ and the constant, $c_5$. The structure graph in Fig. 5b is not cycle coverable and it agrees by our theory that reports the given structured equation in Fig. 5b does not have a robust solution.

\[
\begin{align*}
  f_1(x_3) &= c_1 \\
  f_2(x_4) &= c_2 \\
  f_3(x_1 + x_2) &= c_3 \\
  f_4(x_1 + x_2) &= c_4 \\
\end{align*}
\]

\[
\begin{align*}
  f_1(x_3) &= c_1 \\
  f_2(x_4) &= c_2 \\
  f_3(x_5) &= c_3 \\
  f_4(x_5) &= c_4 \\
  f_5(x_1, x_2, x_5) &= c_5 \\
\end{align*}
\]

Figure 5: An example with a forbidden implicit structure. Suppose we want to consider only equations of the form in (a). Its structure graph is loop coverable, but there are no robust solutions since we have two equations that involve a function $(x_1 + x_2)$ that occurs more than once. Such a restriction is not allowed in our formulation of structured systems. Here $x_1 + x_2$ is over-determined since it must satisfy 2 equations. There cannot be a robust solution for almost any choice of $f_3$ and $f_4$. If the $c_3$ and $c_4$ equations are satisfied, a small change in either constant results in no solutions for almost any choice of $f_3$ and $f_4$. We convert the system in (a) into an (explicit) structured system in (b) by adding the variable $x_5$ and an extra equation, $x_1 + x_2 - x_5 = c_5$. This requires the addition of the constant, $c_5$. In (b) there is a bottleneck so there are no robust solutions. Nodes 1,2,3, and 4 constitute a (forward) bottle of a 1-bottleneck. The forward minimax bottle however consists of nodes 1 and 2. Nodes 3 and 4 are a backward minimax bottle. This example was motivated by the systems [5].

Discussion

A wide part of literature has been devoted to analyze stability of solutions and structural stability. For instance ecologist Richard Levins in [1] uses graph theoretical approach to give a criterion for stability of a structure. However, there is no general criterion for existence of solutions, and before discussing under what conditions stability of solutions will be robust to perturbations, one need to know if there is a solution to a system, and if existence is a structural property of the system, i.e. solutions would persist despite tiny changes in the structure. The results of this paper is on the existence and robustness of existence of solutions.

We say a system of structured equations is linearly robust if the structure matrix associated by that has nonzero determinant. One lesson to be learned from this would be even though the change
in the order of equations changes the graph dramatically, but its rank does not change and whether
the graph is robust does not change. An alternative to our use of the determinant of structure
matrix would be to use the “permanent” of a matrix. If the structure matrix of a graph has only 0
and 1 entries, the graph is cycle coverable if and only if the “permanent” of the matrix is nonzero.

Using different terminology, Yorke and Anderson [35] introduced the robustness criterion of
whether a network’s graph is cycle coverable. It was similarly based on the terms in a determinant
of a matrix $M$, but only for the case where $F(X) = MX$ and $M$ is an anti-symmetric matrix.

Our cycle covering Thm. 1 determines the structures that can have robust solutions. According
to our result if a structure graph is cycle coverable then almost every $C^1$ function that respect
that graph has a robust solution. In other words our result states that robustness is a prevalent
property in the space of all $C^1$ functions that respect a graph. The result are also true for all the
piecewise smooth functions.

In addition, we showed that a fragile structure has almost no solutions. Our Bottleneck Lemma
shows when a structure is fragile there may exist some bottleneck subsets of nodes. But one of
them is unique. We call it the minimax bottleneck.

Given a fragile graph $G$. Show the set of all the nodes of graph $G$ by $A$. The set $A$ cannot be the
bottle of a bottleneck. Because, according to our assumptions only restrictions in which every node
has an incoming edge and an outgoing edge is allowed. Hence, $A^+ = A$. Thus we always need a
method for finding the bottle of the bottleneck in a given fragile graph. Our Thm. 2 gives a easy to
code approach to find the bottleneck. Another highlight of Thm. 2 is that the located bottleneck is
the minimax bottle, meaning it is a Null(G)-bottleneck, where Null(G) is the dimension of the null
space of the structure matrix associated with the graph. The ideas here can be applied anywhere
there are equations and robust solutions are desirable, including problems such as equilibria of
dynamical systems, but the ecologists’ use of graphs makes it particularly applicable to ecology.

We believe that our results will provide both ecologists and students with a better understanding
of the stability of ecosystems, of which robust equilibria are a prerequisite. Their application might
be broad. They could even be used in high school algebra as a criterion for determining if a sparse
system of linear (or higher order) equations can have a robust steady state.

**Appendix**

**Sard’s Theorem and More**

**Sard’s Theorem** [36]. Assume $F : \mathbb{R}^N \rightarrow \mathbb{R}^{N'}$ is $C^m$ where $m \geq 1 + N - N'$ if $N > N'$ and
otherwise $m = 1$. For almost every $C$, whenever $F(X) = C$, $DF(X)$ has full rank.

We apply the following three special cases or consequences in different situations. The first two
only require that $F$ is $C^1$. For our third, we include the Implicit Function Theorem.

**Describing solutions.** Assume $F : \mathbb{R}^N \rightarrow \mathbb{R}^{N'}$ is $C^m$ where $m \geq 1 + N - N'$ if $N > N'$ and
otherwise $m = 1$. We will use the following results.

$(S_I)$ Assume $N = N'$. Then for almost every $C \in \mathbb{R}^N$, whenever $X$ satisfies $F(X) = C$, $F$ is
linearly robust at $X$.

$(S_{II})$ Assume $N < N'$. Then $F(\mathbb{R}^N)$ has measure 0 in $\mathbb{R}^{N'}$. Hence for almost every $C \in \mathbb{R}^{N'}$, there
are no solutions of $F(X) = C$.

$(S_{III})$ Assume $N > N'$. For almost every $C \in \mathbb{R}^{N'}$, the set of $X$ satisfying $F(X) = C$ is empty or
is a smooth surface of dimension $N - N'$.

Cases (S_I) and (S_III) are implied by Sard’s Theorem while (S_{II}) is a lemma used in proving Sard’s Theorem. The Implicit Function Theorem is used with Sard’s Theorem to prove (S_{II}). Note that Sard’s Theorem says only that for almost each $C$, if a solution exists, it is linearly robust. Sard’s Theorem implies that if $F$ is robust at $X$, then arbitrarily close to $X$ there are points $X'$ at which $F$ is linearly robust.

**Cycle coverable, the linear case**

A proof for following Lemma can be found in [10], but for convenience of the reader we prove it here.

**Lemma.** A graph is cycle coverable if and only if its structure matrix is nonsingular.

**Proof.** Consider the graph $G$ is given, denote its structure matrix by $S$. First we will prove is $S$ is nonsingular, then $G$ is cycle coverable.

$$\det S$$

is the sum of the products of $N$ terms, each of the form

$$\pm s_{1,\pi(1)} \times s_{2,\pi(2)} \times \ldots \times s_{N,\pi(N)},$$

where $\pi$ is a permutation of the numbers $1, \ldots, N$, i.e., Hence, $\det A \neq 0$ implies that at least one such product is non zero, so we can assume $s_{1,\pi(1)} \times s_{2,\pi(2)} \times \ldots \times s_{N,\pi(N)}$ is such a non-zero term. Therefore, in this product $s_{i,\pi(i)} \neq 0$ for every $i$. The $N$ edges $\pi(i) \rightarrow i$ together form a cycle covering of the graph. Specifically, we can follow the edges $1 \rightarrow \pi(1) \rightarrow \pi(\pi(1)) \ldots$ until encountering an $n$ for which $\pi^n(1) = 1$, yielding a cycle. Here $1 \leq n \leq N$.

If $n < N$, the remaining integers from 1 to $N$ that are not in this cycle are in other such cycles. Consequently, the product $s_{1,\pi(1)} \times s_{2,\pi(2)} \times \ldots \times s_{N,\pi(N)}$ corresponds to and specifies a set of pairwise disjoint cycles in $G$ that includes all nodes $1, 2, \ldots, N$. As a result, the graph $G$ is cycle coverable.

Now assume $G$ is cycle coverable, we will prove $S$ is nonsingular. Suppose a graph $G$ with $N$ nodes has a cycle covering. Since all the nodes are covered, for each node $i$, there is a unique $j$ (which we will denote by $\pi(i)$) for which the edge $j \rightarrow i$ is in one of the cycles in the cover. For such a pair $i$ and $j$, we set $s_{i,j} = s_{i,\pi(i)} = 1$. For every other pair $(i, j)$, if there is no edge from node $j$ to node $i$ in the graph, set $m_{i,j} = 0$, but if $j \rightarrow i$ is an edge in the graph but not in any of the chosen cycles, set $m_{i,j} = \pm \varepsilon$, for some fixed $\varepsilon$. Then the determinant will have exactly one term of the form (9) where all of the $s_{i,\pi(i)}$ are $\pm 1$, so the term is $\pm 1$.

For $\varepsilon = 0$, $\det M = \pm 1$. Since $S$ depends continuously on $\varepsilon$, for $\varepsilon$ sufficiently small but non-zero, $\det S \neq 0$. \qed

**Proof of part (II) of Theorem [1].**

Assume the graph $G$ is not cycle coverable. Then according to our Bottleneck lemma, for some $k \geq 1$, there exists a backward $k$-bottleneck $B$ such that $B$ has $k$ more nodes that $B^{-}$. Let us assume that $B^{-}$ has $n$ nodes, so there would be a subset of structured equations such that

$$f_{jm}(x_{i_1}, \ldots, x_{i_n}) = c_{jm}, \quad m = 1, \ldots, n + k.$$  (10)
Define $F^B = (f_{j_1}, \ldots, f_{j_{n+k}})$. Then $F^B$ is a $C^1$ map from an $n$-dimensional space to an $n + k$-dimensional space. Hence $F^B(R^k)$ has measure 0 in $R^{n+k}$. Therefore, for almost every choice of $(c_{j_1}, \ldots, c_{j_{n+k}}) \in R^{n+k}$, there are no solutions of Eq. [10]. Hence for almost all $C \in R^{n+k}$, there are no solutions of $F(X) = C$.

**Acknowledgment** We thank Richard Solè and Jose M. Montoya for their permission for using a version of one of their figures in our work.

**References**

[1] Richard Levins. Discussion Paper: The Qualitative Analysis of Partially Specified Systems. *Annals of the New York Academy of Sciences*, 231(1):123–138, 1974.

[2] Ernesto Estrada. *The Structure of Complex Networks: Theory and Applications*. OUP Oxford, 2012.

[3] Donald J. Jacobs, A. J. Rader, Leslie A. Kuhn, and M. F. Thorpe. Protein flexibility predictions using graph theory. *Proteins: Structure, Function, and Bioinformatics*, 44(2):150–165, 2001.

[4] Saraswathi Vishveshwara, K. V. Brinda, and N. Kannan. protein structure: insights from graph theory. *Journal of Theoretical and Computational Chemistry*, January 2012.

[5] Naghmeh Akhavan and James A. Yorke. Population Collapse in Elite-Dominated Societies: A Differential Equations Model without Differential Equations. *SIAM Journal on Applied Dynamical Systems*, pages 1736–1757, January 2020.

[6] Stefano Allesina and Mercedes Pascual. Googling Food Webs: Can an Eigenvector Measure Species’ Importance for Coextinctions? *PLOS Computational Biology*, 5(9):e1000494, September 2009.

[7] Jean-Michel Dion, Christian Commault, and Jacob van der Woude. Generic properties and control of linear structured systems: a survey. *Automatica*, 39(7):1125–1144, July 2003.

[8] Fengjiao Liu and A. Stephen Morse. A Graphical Characterization of Structurally Controllable Linear Systems With Dependent Parameters. *IEEE Transactions on Automatic Control*, 64(11):4484–4495, November 2019.

[9] Mark R.T. Dale. *Applying Graph Theory in Ecological Research*. Cambridge University Press, Cambridge, 2017.

[10] W K Chen. *Graph Theory and Its Engineering Applications*, volume Volume 5 of *Advanced Series in Electrical and Computer Engineering*. World Scientific, February 1997.

[11] J. A Bondy and U. S. R Murty. *Graph theory with applications*. New York, Elsevier, 1976.

[12] Iftikhar Ahmad, Muhammad Usman Akhtar, Salma Noor, and Ambreen Shahnaz. Missing Link Prediction using Common Neighbor and Centrality based Parameterized Algorithm. *Scientific Reports*, 10(1):364, December 2020.
[13] Simachew Abebe Mengiste, Ad Aertsen, and Arvind Kumar. Effect of edge pruning on structural controllability and observability of complex networks. *Scientific Reports*, 5(1):18145, December 2015.

[14] Linyuan Lu and Tao Zhou. Link Prediction in Complex Networks: A Survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, March 2011.

[15] Ching-Tai Lin. Structural Controllability. *IEEE Transactions on Automatic Control*, AC-19, 1974.

[16] R. D. Johnston, G. W. Barton, and M. L. Brisk. Determination of the generic rank of structural matrices. *International Journal of Control*, 40(2):257–264, August 1984.

[17] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-Laszl Barabasi. Controllability of complex networks. *Nature*, 473(7346):167–173, May 2011.

[18] Lin Wu, Min Li, Jian-Xin Wang, and Fang-Xiang Wu. Controllability and Its Applications to Biological Networks. *Journal of Computer Science and Technology*, 34(1):16–34, January 2019.

[19] Kang Li, Yugeng Xi, and Yu Zhang. Some novel characterizations of generic rank of structured matrix. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*, pages 2510–2514 vol.4, Philadelphia, PA, USA, 1998. IEEE.

[20] Tim Sauer, James A. Yorke, and Martin Casdagli. Embedology. *Journal of Statistical Physics*, 65(3):579–616, November 1991.

[21] Brian R. Hunt, Tim Sauer, and James A. Yorke. Prevalence: a translation-invariant almost every on infinite-dimensional spaces. *Bulletin of the American Mathematical Society*, 27(2):217–238, 1992.

[22] William Ott and James Yorke. Prevalence. *Bulletin of the American Mathematical Society*, 42(3):263–290, 2005.

[23] Melanie J. Hatcher, Jaimie T. A. Dick, and Alison M. Dunn. How parasites affect interactions between competitors and predators. *Ecology Letters*, 9(11):1253–1271, 2006.

[24] Jonathan M. Levine, Jordi Bascompte, Peter B. Adler, and Stefano Allesina. Beyond pairwise mechanisms of species coexistence in complex communities. *Nature*, 546(7656):56–64, June 2017.

[25] Jennifer M. Lang and M. Eric Benbow. Species Interactions and Competition. *Nature Education Knowledge*, page 4(4):8, 2013.

[26] Atsushi Mochizuki, Bernold Fiedler, Gen Kurosawa, and Daisuke Saito. Dynamics and control at feedback vertex sets. II: A faithful monitor to determine the diversity of molecular activities in regulatory networks. *Journal of Theoretical Biology*, 335:130–146, October 2013.

[27] Antonio J. Golubski, Erik E. Westlund, John Vandermeer, and Mercedes Pascual. Ecological Networks over the Edge: Hypergraph Trait-Mediated Indirect Interaction (TMII) Structure. *Trends in Ecology & Evolution*, 31(5):344–354, May 2016.
[28] Jennifer A. Dunne, Richard J. Williams, and Neo D. Martinez. Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecology Letters*, 5(4):558–567, 2002.

[29] Jos M. Monteoya, Stuart L. Pimm, and Ricard V. Sol. Ecological networks and their fragility. *Nature*, 442(7100):259–264, July 2006.

[30] Simon A. Levin. Community Equilibria and Stability, and an Extension of the Competitive Exclusion Principle. *The American Naturalist*, 104(939):413–423, 1970.

[31] Thilo Gross, Lars Rudolf, Simon A. Levin, and Ulf Dieckmann. Generalized Models Reveal Stabilizing Factors in Food Webs. *Science*, 325(5941):747–750, August 2009.

[32] Daniel S. Maynard, Carlos A. Servn, Jos A. Capitn, and Stefano Allesina. Phenotypic variability promotes diversity and stability in competitive communities. *Ecology Letters*, 22(11):1776–1786, 2019.

[33] Ricard V. Sol and M. Montoya. Complexity and fragility in ecological networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1480):2039–2045, October 2001.

[34] Robert T. Paine. Food Web Complexity and Species Diversity. *The American Naturalist*, 100(910):65–75, 1966.

[35] J. A. Yorke and W. N. Anderson. Predator-Prey Patterns. *Proceedings of the National Academy of Sciences*, 70(7):2069–2071, July 1973.

[36] Arthur Sard. The measure of the critical values of differentiable maps. *Bulletin of the American Mathematical Society*, 48(12):883–890, 1942.