Symmetry-driven network reconstruction through pseudobalanced coloring optimization

OPEN ACCESS

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

PAPER: Interdisciplinary statistical mechanics

Symmetry-driven network reconstruction through pseudobalanced coloring optimization

Ian Leifer[1], David Phillips[2], Francesco Sorrentino[3] and Hernán A Makse[1,*]

[1] Levich Institute and Physics Department, City College of New York, New York, NY 10031, United States of America
[2] Mathematics Department, United States Naval Academy, Annapolis, MD, 21402, United States of America
[3] Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM, 87131, United States of America
E-mail: hmakse@ccny.cuny.edu

Received 2 December 2021
Accepted for publication 6 June 2022
Published 15 July 2022

Online at stacks.iop.org/JSTAT/2022/073403
https://doi.org/10.1088/1742-5468/ac7a26

Abstract. Symmetries found through automorphisms or graph fibrations provide important insights in network analysis. Symmetries identify clusters of robust synchronization in the network which improves the understanding of the functionality of complex biological systems. Network symmetries can be determined by finding a *balanced coloring* of the graph, which is a node partition in which each cluster of nodes receives the same information (color) from the rest of the graph. In recent work we saw that biological networks such as gene regulatory networks, metabolic networks and neural networks in organisms ranging from bacteria to yeast and humans are rich in fibration symmetries related to the graph balanced coloring. Networks based on real systems, however, are built on experimental data which are inherently incomplete, due to missing links, collection errors, and natural variations within specimens of the same biological species. Therefore, it is fair to assume that some of the existing symmetries were

*Author to whom any correspondence should be addressed.

J. Stat. Mech. (2022) 073403

not detected in our analysis. For that reason, a method to find pseudosymmetries and repair networks based on those symmetries is important when analyzing real world networks. In this paper we introduce the *pseudobalanced coloring* (PBCIP) problem, and provide an integer programming formulation which (a) calculates a PBCIP of the graph taking into account the missing data, and (b) optimally repairs the graph with the minimal number of added/removed edges to maximize the symmetry of the graph. We apply our formulation to the *C. elegans* connectome to find pseudocoloring and the optimal graph repair. Our solution compares well with a manually curated ground-truth *C. elegans* graph as well as solutions generated by other methods of missing link prediction. Furthermore, we provide an extension of the algorithm using Bender's decomposition that allows our formulation to be applied to larger networks.

Keywords: network reconstruction, optimization over networks, network dynamics

Contents

J. Stat. Mech. (2022) 073403

1. Introduction

Network models have become a crucial tool in the investigation of biological systems [1–4]. Examples include neural networks [5, 6], gene regulatory networks [7, 8], metabolic networks [9], and ecological networks [10]. The goal of these studies is to better understand the biological function of the system via a network model. In recent work, Morone *et al* [11] found that *automorphisms* [12] describe the symmetries and function of the neural connectome of the nematode *C. elegans* [13]. An automorphism of a graph is a permutation of nodes that preserves the link structure of the graph (we formally define automorphisms in section 2). Later, Morone *et al* and Leifer *et al* [14, 15] uncovered fibration symmetries using the algorithms for *K-balanced coloring* [12, 16, 17] from [18, 19] in biological networks spanning from transcriptional regulatory networks to signaling pathways and the metabolism. Intuitively, a K-balanced coloring (the K is omitted when implicit in context) of a graph is a way of partitioning nodes into K clusters such that nodes in the same cluster have the same number of links to every other cluster. This definition can be generalized to the case of weighted links. Section 3 provides a formal definition and a discussion of how graph automorphism, fibration symmetries and balanced coloring are related.

K-balanced coloring and resulting fibration symmetries establish a connection between the topology of the graph and its dynamics. We say two nodes in the network are synchronized if they have identical dynamics over time. More precisely, for a network, $G = (V, E)$ made of a set of V nodes, E edges (we use both links and edges to refer to two nodes that are adjacent in a network) and time-varying states, $x_i(t) \in \mathbb{R}^k$ for $i \in V$, we say two nodes $i, j \in V$ are synchronized if $x_i(t) = x_j(t)$ for all time t. In this paper we are mainly interested in cluster synchronization formed by two or more synchronized nodes, where several clusters of synchrony can coexist in the network. Complete synchronization, where all the nodes in the network are synchronized in the same state, is not considered in this paper. As shown in [11, 14, 17, 20–23], the symmetries of the network, given either by automorphisms [21], groupoids [17] or graph fibrations [14], are associated with patterns of cluster synchrony. Namely, Golubitsky *et al* [17] showed that nodes with the same color in a K-balanced coloring, or, in other words, symmetric under the fibration symmetry, form a cluster of synchronized nodes. Furthermore, Leifer *et al* [24] used available experimental data on gene co-expression in bacteria to confirm the existence of cluster synchronization predicted solely by the fibration symmetries in these networks. Existence of these synchronous solutions can considerably augment our understanding of the functionality of the modeled system [14, 15].

Symmetry and synchronization are important concepts in the field of physics. Symmetry lays in the foundation of the standard model and has broad applications in other parts of physics from Lagrangian mechanics to crystallography and nuclear spectroscopy. One of the fundamental problems in chaos theory is the search for synchronous solutions in dynamical systems [25]. In the context of biological networks, in particular in neural networks, complete synchronization is often studied by considering the dynamics of networks of coupled oscillators using the Kuromoto model [26, 27]. Methods of statistical mechanics have broad applications in the studies of networks of coupled

J. Stat. Mech. (2022) 073403

oscillators [27–29]. For example, the transition of a network of coupled oscillators to the synchronous state, called the synchronization transition, can be thought of as a phase transition (bifurcation) in the ensemble of nodes of the network of oscillators [29], which then permits using the tools of statistical mechanics developed for study of phase transitions.

Despite these advances, modeling biological systems as networks with symmetries pose additional challenges. Firstly, experimental data on networks collected by different techniques are never complete due to experimental errors and the impossibility to measure every possible link. The missing links include protein–protein interactions, binding between transcription factors and DNA, neural synaptic connections in the brain or even metabolic reactions. Secondly, natural variability across individuals of the same species results in different networks making interpretation of the data across specimens more difficult [5]. Thirdly, organisms adjust to the changing habitat using neural plasticity, epigenetics and other adaptations. For example, even two organisms that were identical at a given moment of time may become different in the future. Organisms survive and benefit from these small variations, so evolution through natural selection occurs. Therefore, networks that model these systems exhibit not only differences across individuals, but also have missing experimental links that can mask their underlying regularities. In particular, the missing experimental interactions and natural variations make the existence of perfect symmetries difficult to find in biological networks. Thus, symmetries in biological networks have been very hard to find since the time of Monod's first formulation of the problem [30]. The difficulty persisted despite the ubiquitous existence of synchronization across all biological networks, which is a manifestation of symmetries in the underlying networks supporting the biological activity [24].

A typical case study is the connectome of the nematode *C. elegans*, which was fully mapped in a landmark paper in 1986 [13]. Despite of being one of the simplest connectomes, containing only 302 neurons in the hermaphrodite, there is plenty of variability from animal to animal (about 25% of links are different between animals [5]) and modern measurements and data curation [5] keep finding new synaptic connections that were missed before [13]. Despite this inherent disorder, the function and neural activity of *C. elegans* exhibits strong regularities, as observed, for instance, in the oscillatory locomotion patterns and the neural synchronization observed in neuronal activity [31].

What is the anatomical substrate for this synchrony? Theory predicts that robust synchronization arises from the symmetries in the underlying network. However, perfect symmetries are impossible to be realized in biology. Therefore, Morone *et al* [11] introduced the concept of *pseudosymmetries* which are almost-symmetries of the incomplete individual network that, upon reconstruction of a few missing links, they reveal an underlying ideal perfect symmetry that is the unique 'blueprint' of the symmetry of the species. Each individual network can be thought of as a small variation of this 'blueprint' network containing the ideal symmetries. The observed experimental networks are all different and always pseudosymmetric and can be slightly modified to exhibit the perfect 'blueprint' of the species.

In this paper we introduce the *pseudobalanced K-coloring* problem, and an integer programming formulation (PBCIP) to find the minimum number of missing links in the network to achieve symmetry. We are interested in biological networks since they are

J. Stat. Mech. (2022) 073403

the substrate for cluster synchronization of their units. These include brain networks (connectomes), gene regulatory networks, metabolic networks and others. While the symmetries are not perfect, yet, they are close enough to ideal symmetries such that they can sustain the observed experimental cluster synchronization and functional regularities. We present an optimal integer linear programming formulation of the problem that attempts to identify the pseudosymmetries of the network based on pseudobalanced coloring, and at the same time, reconstruct the network with a minimum number of modified links to transform the pseudosymmetric network into ideal symmetry. Beyond biological networks, our symmetry-driven reconstruction can be applied to any network that supports cluster synchronization of its units.

Reconstructing a network to reveal ideal symmetries is an instance of the problem of link prediction in complex networks [32]. Link prediction is the problem of determining the probability of the existence of a link between a pair of nodes in the network that is missing from the data [32–36]. Previous work proposed several approaches to find missing links based on different statistical and Markov chain models, see review in [32]. However functional and dynamical features of the incomplete networks, such as synchronizability, are rarely taken into consideration. The approach presented here employs network symmetry as an organizing principle of the network structure to guide the link prediction and reconstruct the network from its incomplete state.

Traditionally, network symmetry is defined by using graph automorphisms or symmetry permutations. An automorphism is a permutation of nodes that preserves the adjacency structure of the graph [37, 38]. That is, before and after the permutation the nodes are connected to the same nodes. The set of all automorphisms of a graph form the symmetry group of the graph. Automorphisms are, however, sensitive to small perturbations in the graph, i.e. the removal of one link can drastically change the automorphism groups [11, 14, 39]. For an explicit example, consider the two graphs in figure 1 and their corresponding automorphisms presented on the right side. Graphs in figures 1(a) and (b) differ only by link (1, 3). Missing this link has a global effect on the symmetry of the network decreasing the size of the automorphism group from five elements $(\sigma_1, \ldots, \sigma_5)$ to only two (σ_1 and σ_2). Therefore, the symmetry of the graph can be substantially reduced by missing just a few links in the topology of a graph. Inversely, the graph in figure 1(b) can be restored to be a perfectly symmetric graph in figure 1(a) by adding just one link.

The automorphisms of the graphs are important since they lead to robust cluster synchronization of the nodes activity. The symmetry group of the network leads to synchronization of the nodes associated by the orbits of the graph. An orbit of a node is the set of nodes that are obtained by the application of all the automorphisms of the graph. The orbits are non-overlapping clusters and each orbit can sustain an independent cluster synchronization, although its stability is not guaranteed.

The pseudosymmetry formulation proposed in [11] attempts to reconstruct the graph based on the automorphism group of the network. However, automorphisms cannot predict all the possible symmetries and cluster synchronization present in the network. Instead, balanced coloring [17] and associated symmetry fibrations [14, 15] provide a more general type of symmetry. Since all orbits of the graph are also balanced colored, but not the opposite [40], a balanced coloring-based formulation captures more cluster

J. Stat. Mech. (2022) 073403

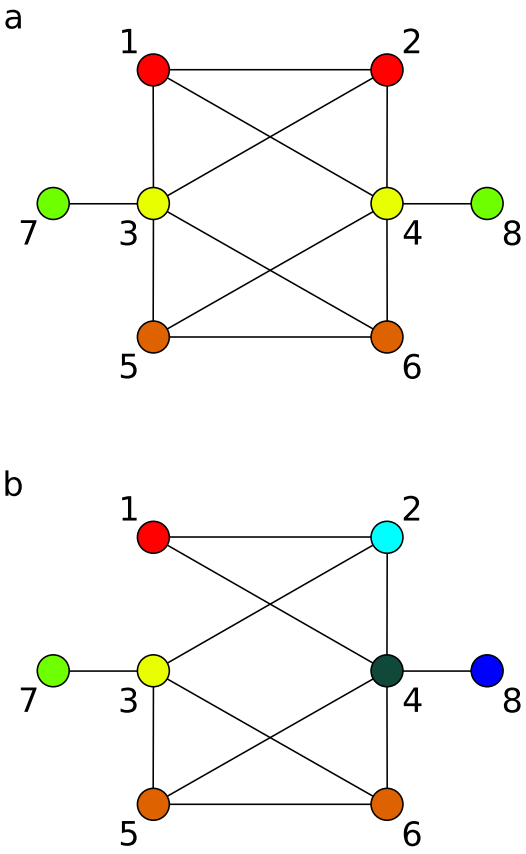

Figure 1. Differences between automorphism groups and balance coloring of two similar graphs (a) and (b) differing by only one link (1, 3). The automorphism group of graph (a) consists of 5 elements: $\sigma_1, \ldots, \sigma_5$ whereas the automorphism group of graph (b) consists of 2 elements: σ_1 and σ_5. Graph coloring is balanced in both networks, i.e. nodes of the same color have the same number of links to every other color, but graph (a) is much 'more symmetric' than (b) and requires a fewer non-trivial colors. On the contrary, missing a single link makes the graph in (b) less symmetric as seen from the abundance of trivial colors (colors assigned to just one node) needed to color the graph. The purpose of this work is to develop a formulation that would identify this missing link that makes the graph more symmetric, in a precise way.

synchronizations than those predicted by automorphisms. In this paper, we derive the symmetry reconstruction of the graph by finding the ideal balanced coloring of the graph rather than the ideal automorphisms as done in [11]. Thus, we will reconstruct the network based on *pseudobalanced coloring*. Reconstructing the network has two objectives: (1) determine the 'best' number of clusters, i.e. determine K, and (2) perform the minimum number of *perturbations* to perform on the graph. In general, we use perturbation to refer to any change to a link weight with the convention that reducing a link weight to zero removes the link altogether. For this paper, we focus on the case of unweighted links and only adding links to the graph. In particular, we are interested in the following optimization problem: find the minimum number of new links to add to

J. Stat. Mech. (2022) 073403

a graph so that it possesses a balanced K-coloring. After solving this problem for the allowable values of K, we use graph functions to determine the 'best' choice of K.

A balanced coloring with the minimal number of colors gives rise to a symmetry fibration, which is a transformation that collapses all the colors of the graph into a representative node of the base. Thus, the present formulation based on pseudobalanced coloring is related to finding a quasi-fibrations of the graphs [41]. Boldi *et al* [41] deals with this analogous graph reconstruction formulation based on quasi-fibrations.

For smaller networks, adding links manually to find symmetry is possible [11], but such a method is computationally impractical for larger networks. Moreover, such ad hoc methods do not have an optimality guarantees. Thus, an important endeavor is to develop models and algorithms that can find symmetry in the presence of small perturbations in the underlying input data. In this paper, we describe an integer programming formulation that finds the minimum number of edges to add to a given network so it has a balanced coloring [16] of a given size (balanced colorings are also referred to as equitable partitions [12]). Our complexity results in section 4 show that the directed variant of our problem is weakly NP-Hard and we conjecture the same is true of the undirected version so an efficient algorithm is unlikely to exist for our problems of interest. Therefore, integer programming is a natural method to use to solve our problem. We are able to solve the integer program for modestly sized instances (20–30 nodes and 100–120 edges) of interest using Gurobi 9.11 [42]. Most instances required seconds to solve although there were cases where hours were needed. In preliminary tests, we found that solving larger instances with approximately 280 nodes and 3000 edges required minutes to hours to solve. We emphasize that we do not draw conclusions from these results as we did not conduct a full computational study. For future work, we also describe an algorithm based on a Bender's decomposition of the integer program in appendix A.2 to help improve the computational efficiency of solving the integer program for larger scale instances.

The paper is organized as follows. In section 2, we review the previous pseudosymmetry formulation [11] and provide intuition and motivation for our method. In section 3, we give formal definitions for the problems of interest. In section 4, we prove that a simplified variant of our problem is weakly NP-Hard. In section 5, we describe the integer linear programming formulation of our problem. In section 6, we present results from our integer linear programming approach applied to experimental data on the *C. elegans* connectome. In section 7, we show how a repaired graph can be partitioned into co-functioning clusters. In section 8, we compare results obtained by using different objectives with some of the traditional link prediction methods. In section 9, we discuss implications from our results and outline future work. Appendix A.2 provides a Bender's decomposition and algorithm for our integer linear program.

2. Previous work on pseudosymmetry

Consider for the moment an undirected graph $G = (V, E)$ and let π denote a permutation of the node labels, i.e. $\pi : V \to V$. Then π is an automorphism if, and only if, for all $u, v \in V$, if $uv \in E$ then $\pi(u)\pi(v) \in E$. The automorphism group, denoted $\texttt{Aut}(G)$, is the set of all automorphisms of G along with function composition as the multiplication.

A useful characterization of automorphisms involves permutation matrices. Let A be a node-node adjacency matrix of G and P denote the permutation matrix associated with a permutation π. Then $\pi \in \texttt{Aut}(G)$ if and only if

$$[P, A] = PA - AP = \mathbf{0} \tag{1}$$

where $\mathbf{0}$ denotes the matrix of all zeros. Let $\mathcal{P}$ denote the set of $|V| \times |V|$ matrices. If the permutation matrix $P \in \mathcal{P}$ satisfies equation (1), we say that P is a *perfect symmetry*. The *trivial symmetry* is the identity matrix and is shared by all graphs.

In order to study graphs that slightly vary from those with perfect symmetry, Morone and Makse [11] suggest looking for pseudosymmetries defined by a small parameter $\varepsilon \in \mathbb{R}$:

$$\mathcal{P}_\varepsilon = \{P_\varepsilon \in \mathcal{P} : \|[P_\varepsilon, A]\| \leqslant \varepsilon\} \tag{2}$$

where $\|X\|$ is the Frobenius norm of the matrix $X = (x_{ij})$, i.e. $\|X\| = \sqrt{\sum_{i=1,j=1}^{n,m} x_{ij}^2}$. For $\varepsilon = 0$ this definition is the same as perfect symmetry. Note, the set of permutations associated with $\mathcal{P}_\varepsilon$ does not necessarily form a group. The naive way to find elements of $\mathcal{P}_\varepsilon$ is to consider all possible permutations and determine whether equation (2) is satisfied. Because $|\mathcal{P}| = (|V|)!$ this naive approach is computationally intractable and finding a more efficient method is desirable. Moreover, for a given permutation matrix that satisfies equation (2), we would like a method that finds a related graph that is symmetric. Thus, we want an alternative approach that also finds a way to 'repair' G, minimally perturbs G and results in a symmetric permutation matrix.

Because the Frobenius norm is unitarily invariant, i.e. $\|X\| = \|UXV\|$ for all matrices X and unitary matrices U and V,

$$\|[P_\varepsilon, A]\| = \|P^\varepsilon A - AP^\varepsilon\| = \left\|P^\varepsilon A (P^\varepsilon)^{-1} - A\right\|. \tag{3}$$

We focus on the case of adding edges to an undirected graph G and show that, in this case, ε is bound by four times the number of added edges. Suppose that we have a graph, G, with added edges so that P^ε is a symmetry and let the adjacency matrix of this graph be denoted by A_ε. Then $P^\varepsilon A_\varepsilon (P^\varepsilon)^{-1} - A_\varepsilon = \mathbf{0}$, equation (3), and the triangle inequality imply that:

$$\begin{aligned}
\|[P_\varepsilon, A]\| &= \left\|A - P_i^\varepsilon A (P_i^\varepsilon)^{-1}\right\| \\
&= \left\|P_i^\varepsilon A_\varepsilon (P_i^\varepsilon)^{-1} - P_i^\varepsilon A (P_i^\varepsilon)^{-1} - A_\varepsilon + A\right\| \\
&= \left\|P_i^\varepsilon (A_\varepsilon - A)(P_i^\varepsilon)^{-1} - (A_\varepsilon - A)\right\| \\
&\leqslant \left\|P_i^\varepsilon (A_\varepsilon - A)(P_i^\varepsilon)^{-1}\right\| + \|(A_\varepsilon - A)\| \\
&= 2\,\|A_\varepsilon - A\|.
\end{aligned} \tag{4}$$

The difference matrix $(A_\varepsilon - A)$ is the adjacency matrix corresponding to the graph that just has the repaired edges. Therefore, $\|[P_\varepsilon, A]\|$ is less than the number of repaired edges multiplied by 4 (2 because the graph is undirected and A is symmetric and another 2 because of the coefficient in front of the norm). Hence, a reasonable approach to repair G is to add the minimum number of edges required for the graph to have a symmetry.

J. Stat. Mech. (2022) 073403

To summarize this approach, finding pseudosymmetries in a graph using brute force is computationally intractable and not guaranteed to find an optimally repaired graph with minimal modifications. Instead, in this paper, we formulate the problem in terms of the balanced coloring problem. It is known that balanced coloring is a pre-processing step in finding the automorphisms of the network as it is used in the popular McKay's algorithm Nauty [12]. This is because all orbits are balanced colored (but not the opposite). Thus, finding first a minimal balanced coloring of the graph (which can be found in quasilinear time) leads to the orbits of the graphs, which can then be used to find the generators of the automorphism group. In section 3, we make this idea more precise.

Based on these ideas, rather than searching for pseudo automorphisms like in [11], here we search for pseudobalanced coloring [14, 15], which are more general symmetries than automorphisms. We use an integer linear program that adds the minimum number of edges to a graph so that we make the graph 'more symmetric' in terms of balanced coloring. Figure 1(a) shows the perfect balanced coloring (see definition in next section) found in the symmetric network and the comparison with the same perfect balanced coloring applied to the incomplete network in figure 1(b). We see how a single missing link can destroy not only the automorphism group but also the perfect balanced coloring. The goal of the formulation is to first find the pseudobalanced coloring of figure 1(a) in the graph in figure 1(b), and at the same time repair the missing link that transform the pseudobalanced coloring into a perfect balanced coloring for figure 1(b).

3. Definitions and problems

In this section, we describe both a general version of the pseudobalanced K-coloring problem and the specific case that we are interested in. We believe the general version contains many interesting variants that pose important challenges to be solved as the graph can be undirected or directed and weighted or unweighted. In addition, the perturbations on the graph in the most general version are permitted to have very general restrictions or lack thereof. The specific version of our problem is on an undirected, unweighted graph and the only perturbations allowed are the addition of edges.

Let $G = (V, E)$ be a given graph. We let $A \in \mathbb{R}^{|V|\times|V|}$ denote the weighted node-node adjacency matrix associated with G. We also define the *directed* complement of E to be $E^{\mathrm{C}} = \{(i, j) \in V \times V : ij \notin E, i \neq j\}$ and the undirected complement of E to be $E' = \{ij : i, j \in V, ij \notin E\}$. We require both E^{C} and E' for our formulation when the graph is undirected. When the graph is directed, $E^{\mathrm{C}} = E'$. To help emphasize the graph type, we use the convention that consecutive node indices, e.g. ij, represent an undirected edge. An ordered pair of node indices, e.g. (i, j), represent a directed edge. We also recall that a *partition* of a given set S, is a collection of pairwise disjoint subsets of S whose union is all of S, i.e. if $\mathcal{C}$ is a partition of S then

$$\bigcup_{C\in\mathcal{C}} C = S \quad \text{and, for all } C, D \in \mathcal{C},\ C \cap D = \emptyset. \tag{5}$$

J. Stat. Mech. (2022) 073403

Balanced coloring has been defined by several authors, e.g. see [12, 16, 43, 44] for a definition corresponding to the one we use where it is sometimes referred to as an equitable partition. A stricter definition of balanced coloring is given and used in [14, 18] although their definition is the same as ours for unweighted, i.e. binary graphs. There are different definitions including ones that are unrelated and more similar to traditional graph coloring, e.g. [45].

In our definition, we fix K, the number of colors, i.e. clusters, as the objectives of our eventual problem are to both determine the minimum number of links to add to a graph so that a balanced coloring exists, but also to determine the 'best' number of colors, i.e. the best K.

Definition 3.1. Balanced K-coloring. A *balanced K-coloring* of G is a partition, $\mathcal{C}$, of the node set V which satisfies the following:

- The cardinality of $\mathcal{C}$ is K.
- In the case that G is an undirected graph, the condition is as follows. For all $C \in \mathcal{C}$, all pairs of distinct nodes $p, q \in C$, and all $D \in \mathcal{C}$,

$$\sum_{j \in D\,:\,pj \in E} A_{pj} = \sum_{j \in D\,:\,qj \in E} A_{qj}. \tag{6}$$

- In the case that G is a directed graph, equation (6) corresponds to two separate conditions resulting in three kinds of balanced coloring. The two conditions are as follows. For all $C \in \mathcal{C}$, all pairs of distinct nodes $p, q \in C$, and all $D \in \mathcal{C}$,

$$\sum_{j \in D\,:(p,j) \in E} A_{pj} = \sum_{j \in D\,:(q,j) \in E} A_{qj} \tag{7}$$

and

$$\sum_{j \in D\,:(j,p) \in E} A_{jp} = \sum_{j \in D\,:(j,q) \in E} A_{jq}. \tag{8}$$

A *directed out-balanced coloring* is if only equation (7) is enforced and a *directed in-balanced coloring* is if only equation (8) is enforced. A fully directed balanced K-coloring has both conditions enforced.

We also define the *minimal balanced coloring* which, intuitively, corresponds to the minimum number of colors K to color the graph so that no nodes with different colors are balanced. Because the number of colors, K, to color a graph is unique [46], we omit K in this definition.

Definition 3.2. Minimal balanced coloring. A *minimal balanced coloring* is a balanced K-coloring, $\mathcal{C}$, where the following is also true. For every $p \in V$, let $C_p \in \mathcal{C}$ be the unique set that contains p. For every distinct $p, q \in V$ such that $q \notin C_p$ there exists $D \in \mathcal{C}$ such that equation (6) is violated for the undirected case or equations (8) or (7) is violated in the directed case.

For example, in the undirected case, the following must be true for some set $D \in \mathcal{C}$.

J. Stat. Mech. (2022) 073403

$$\sum_{j\in D:pj\in E} A_{pj} \neq \sum_{j\in D:qj\in E} A_{qj}. \tag{9}$$

In other words, in an in-balanced coloring partition, two nodes with the same color 'receive' the same colors from connected nodes. A minimal balanced coloring is a balanced coloring with the minimal number of colors. We call a color *trivial* if there is only one node that belongs to this color. A *non-trivial color* is a color that is not trivial. A coloring in which each color is trivial (corresponding to the discrete partition) is called discrete. We refer to a node as *trivial* if it possesses a trivial color and call a node *symmetric* if its color is non-trivial. Abusing this definition, we also say a node is symmetric to another if they are both the same color.

Relation to graph fibrations. In the framework of graph fibrations [14], a cluster of nodes with the same balanced color is equivalent to a *fiber* (see methods in [24]). Nodes in a fiber have isomorphic input trees. When an admissible set of dynamical equations is attached to the graph, nodes of the same balanced color (or fiber) are predicted to be synchronous (cluster synchronization). A symmetry fibration is a transformation that collapses nodes in a minimal balanced color cluster into a single representative node in the base of the graph. We note that, while nodes belonging to the same orbit are also synchronous, nodes of the same balanced color do not necessarily belong to the same orbit of the automorphism group. However, all nodes in each orbit do have the same balanced color [40, 47]. Thus, balanced coloring, fibers and symmetry fibrations represent more general symmetries than automorphisms, and reveal cluster synchronization in the graph that is not captured by orbital partitions, see [14] for more details.

Relation to graph automorphisms. The minimal balanced coloring and automorphisms are fundamentally related. In this case, the number of colors found is a lower bound on the number of automorphism orbits. Moreover, the nodes in each nontrivial color found corresponds to potential automorphisms of the graph.

To explain the connection between coloring and automorphisms further, we describe how graph automorphisms and isomorphisms are found. (Note that the two problems are equivalent [48].) The graph isomorphism and automorphism algorithms we are aware of [43, 49–52] all use a search tree to find isomorphisms/automorphisms. For a given graph G, each node of this search tree represents a balanced coloring of G. The root of the tree corresponds to the minimal balanced coloring. Each child in the search tree is found using the *individualization-refinement* process. In this process a child's coloring is obtained by *refining* its parent's coloring by choosing a node of a non-trivial color (in a parent), assigning it with a unique color and obtaining a new balanced coloring of the graph while keeping this node *individualized* (having a unique (trivial) color). This individualization-refinement process is repeated until the coloring is discrete, hence all leaves of the search tree correspond to the discrete coloring. In the last step, strings corresponding to all leaves are constructed by combining the rows of the adjacency matrix in the order defined by the coloring of each leaf. Due to the fact that the search tree is isomorphism-invariant, whenever strings corresponding to two leaves are the same, colored graphs in these two leaves are isomorphic and hence a color-preserving mapping between these two graphs is an automorphism of G. Readers interested in more details can refer to [44, 47, 51, 53].

J. Stat. Mech. (2022) 073403

The size of graph's search tree has a deep connection with the number of non-trivial colors (NNTC) in the minimal balanced coloring of the graph. Each non-trivial color requires refinement resulting in more children nodes and thereby increasing the search tree size. Therefore, repairing the graph to a version with similar topology but fewer nodes trivially colored so that more nodes have non-trivial colors than in the original graph. Repairing graphs in this way creates a 'more symmetric' version of the graph, that is, a version of the graph that has (or at least may have) more elements in the automorphism group. Any graph could be repaired to a complete graph, i.e. a graph with a link between every pair of nodes. The automorphism group of the complete graph is a symmetric group, but naturally is unlikely to be similar topologically to the original graph. In particular, we wish to use the minimum number of repairs necessary to find a more symmetric version of the graph so that the essential network topology of the original graph is maintained as much as possible. Thus, the repairing process must balance the trade-off between making the graph more symmetric (by increasing the NNTC) versus making minimal changes to the graph topology.

Relation to stochastic block models. It is worth mentioning a parallel to stochastic block models (SBMs) introduced in the context of social networks by Holland *et al* [54]. SBM is a graph generative model in which the graph is partitioned into groups and edges between nodes are placed with the probability defined by the clusters these nodes belong to. In other words, SBMs are the stochastic version of a graph partition (different from the equitable partition), where the analogous constraints of equations (6)–(8) hold only in *expectation*. We refer readers to [55] (section 2.1 and figure 4) for a more rigorous discussion of the connection between SBMS and equitable partitions (i.e. balanced coloring, fibers).

We now define the general version of this problem. In this version, existing edge weights are allowed to be perturbed, i.e. changed, and non-existent edges are allowed to be introduced with weights restricted in some manner (or not). We still wish for a minimum amount of perturbations so that the balanced conditions are enforced. We first formally define the set of *generalized pseudobalanced colorings* of a graph and then the corresponding optimization problem which occurs over this set. We also add the option that the colorings considered must adhere to some prior knowledge about the given graph. In particular, we allow that some nodes' colors are already known. Such an option corresponds to prior expert knowledge about the graph structure and our methods permit this restriction.

Definition 3.3. Generalized pseudobalanced K-coloring. Let $G = (V, E)$ denote a graph with edge weight matrix $A \in \mathbb{R}^{|V|\times|V|}$ and $\mathcal{R} \subseteq \mathbb{R}^{|E|+|E'|}$ given. Let $\mathcal{F}$ denote a collection of disjoint subsets of V. A generalized pseudobalanced coloring of G is an ordered triple $(\mathcal{C}, \Phi, \Omega)$ with the following true.

- $\mathcal{C}$, the coloring, is a partition of the node set V with $|\mathcal{C}| = K$, i.e. $\mathcal{C}$ satisfies equation (5).
- If $\mathcal{F} \neq \emptyset$ then for all pair of distinct sets $C, D \in \mathcal{F}$, there must exist a pair of distinct sets $S, T \in \mathcal{C}$ such that $C \subseteq S$ and $D \subseteq T$. If $\mathcal{C}$ satisfies this condition, then we say that $\mathcal{C}$ *respects* $\mathcal{F}$.
- $\Phi = (\phi_{ij}) \in \mathbb{R}^{|E|}$ is a vector of perturbations on the edge weights.

J. Stat. Mech. (2022) 073403

- $\Omega = (\omega_{ij}) \in \mathbb{R}^{|E'|}$ is a vector of new weights that do not exist in G.
- The perturbation vectors Φ and Ω are related and restricted through the set $\mathcal{R}$, i.e.

$$(\Phi, \Omega) \in \mathcal{R}. \tag{10}$$

- Appropriate balanced equations are satisfied, i.e. equation (6), equation (8), and/or equation (7) are satisfied by $(\mathcal{C}, \Phi, \Omega)$ depending on whether G is undirected or directed. If undirected, then for all $C \in \mathcal{C}$, all pairs of distinct nodes $p, q \in C$ and all $D \in \mathcal{C}$,

$$\sum_{j \in D\,:\,pj \in E} (A_{pj} + \varphi_{pj}) + \sum_{j \in D\,:\,pj \in E'} \omega_{pj} = \sum_{j \in D\,:\,qj \in E} (A_{qj} + \varphi_{qj}) + \sum_{j \in D\,:\,qj \in E'} \omega_{qj}. \tag{11}$$

If G is directed, then one or both of the following two conditions replaces equation (11). Recall that E and E' are ordered pairs of nodes in what follows. For all $C \in \mathcal{C}$, all pairs of distinct nodes $p, q \in C$ and all $D \in \mathcal{C}$,

$$\sum_{j \in D\,:(p,j) \in E} (A_{pj} + \varphi_{pj}) + \sum_{j \in D\,:(p,j) \in E'} \omega_{pj} = \sum_{j \in D\,:(q,j) \in E} (A_{qj} + \varphi_{qj}) + \sum_{j \in D\,:(q,j) \in E'} \omega_{qj} \tag{12}$$

$$\sum_{j \in D\,:(j,p) \in E} (A_{jp} + \varphi_{jp}) + \sum_{j \in D\,:(j,p) \in E'} \omega_{jp} = \sum_{j \in D\,:(j,q) \in E} (A_{jq} + \varphi_{jq}) + \sum_{j \in D\,:(j,q) \in E'} \omega_{jq} \tag{13}$$

For a given positive integer $K \in \{1, \ldots, |V|\}$, we let $\mathcal{G}_{G,K,\mathcal{F},\mathcal{R}}$ denote the set of all generalized pseudobalanced K-colorings on G with fixed nodes $\mathcal{F}$, i.e. $|\mathcal{C}| = K$, $\mathcal{C}$ respects the subsets of $\mathcal{F}$, and edge and non-edge perturbations restricted to $\mathcal{R}$. In particular,

$$\mathcal{G}_{G,K,\mathcal{F},\mathcal{R}} = \{(\mathcal{C}, \Phi, \Omega) : \mathcal{C}\ \text{respects}\ \mathcal{F}, |\mathcal{C}| = K, (5), (10),\ \text{and} (*)\ \text{are satisfied.}\}$$

Here, $(*)$ represents the appropriate choice(s) of (11), (12) and/or (13).

Definition 3.4. The optimal repair problem (ORP). The optimization problem of interest is then find the minimum amount of perturbations to the edges (addition/removal/weight changes) so that a balanced K-coloring exists on the perturbed graph:

$$\min \left\{ \sum_{ij \in E} |\omega_{ij}| + \sum_{ij \in E'} |\phi_{ij}| : (\mathcal{C}, \Phi, \Omega) \in \mathcal{G}_{K,G,\mathcal{F},\mathcal{R}} \right\}. \tag{14}$$

There are many interesting cases of ORP, equation (14), involving the particular restrictions to $\mathcal{R}$, e.g. $\mathcal{R}$ are box constrained by an appropriate small constant or $\mathcal{R}$ is an elliptical set. In this paper, we focus on the following restricted case: we consider undirected and unweighted graphs where links are permitted to be added but not removed from the graph. We note, however, that our methods can be extended to directed graphs and also weighted edges.

J. Stat. Mech. (2022) 073403

We call a *restricted pseudobalanced K-coloring* of a given graph as the case of the generalized pseudobalanced K-coloring where the graph is unweighted, link perturbations and removals are not allowed, and only new unweighted links are allowed to be added. We also allow for some of the nodes to have fixed colors in advance, e.g. by running a balanced coloring algorithm in advance of repairs or leveraging expert knowledge. Formally, we define this as follows and we call it *pseudobalanced K-coloring* for short:

Definition 3.5. Pseudobalanced K-coloring (PBC). Let $G = (V, E)$ be a given undirected graph with node-node adjacency matrix $A \in \{0,1\}^{|V|\times|V|}$, and $K \in \{1, \ldots, |V|\}$ and $\mathcal{F}$ to be a given collection of disjoint subsets of V. Denote the set of pseudobalanced K-colorings of G by $\mathcal{P}_{G,K}$ where

$$\mathcal{P}_{G,K,\mathcal{F}} = G_{G,K,\mathcal{F},\{0\}^{|E|}\times\{0,1\}^{|E'|}},$$

i.e. $\phi_{ij} = 0$ for all $ij \in E$ and $\omega_{ij} \in \{0,1\}$ for all $ij \in E'$. The optimization problem equation (14) reduces to

$$\min\left\{\sum_{i,j\in V, ij\notin E} \omega_{ij} : (\mathcal{C}, \Omega) \in \mathcal{P}_{K,G,\mathcal{F}}, \forall\, \omega_{ij} \in \Omega\right\}. \tag{15}$$

As we show in the subsequent section, solving the in-balanced ORP on directed graphs is unlikely to be efficiently solvable. We conjecture the same is true of the PBC.

4. Problem complexity

The first step of many algorithms for graph isomorphism is to find the minimal balanced coloring, which is polynomial time solvable [43, 49–52]. In our paper, we use the algorithm of Kamei and Cock [18]. The complexity of the variants we have described of the balanced coloring problem are, to the best of our knowledge, unknown. In this section, we prove that finding an in-balanced K-coloring on a weighted graph is weakly NP-Hard when $\mathcal{F} \neq \emptyset$. We note that the three conditions that the proof relies on are (1) the allowance of weighted edges, (2) that the graph is directed, and (3) the inclusion of fixed nodes, i.e. $\mathcal{F} \neq \emptyset$. As a corollary, the in-balanced ORP on directed weighted graphs is also weakly NP-Hard.

Our proof reduces the weighted, directed in-balanced K-coloring problem to the partition problem. In this problem, a list of positive integers is given and the decision problem is to determine if there is a way to partition the list into 2 sets elements so that the sums of each set is equal. Formally, this problem can be stated as follows.

J. Stat. Mech. (2022) 073403

Definition 4.1. Let $n \in \mathbb{Z}_+$, and a set of positive integers, denoted $X = \{x_1, \ldots, x_n\}$, be given whose sum is even. The decision problem is to determine if there is a partition of X into 2 sets, denoted by S and T so that

$$\sum_{x \in S} x = \sum_{x \in T} x. \tag{16}$$

The partition problem is weakly NP-Hard [48] and we can show that being able to solve the weighted directed in-balanced K-coloring problem solves the partition problem.

Theorem 4.2. *Finding a weighted directed in-balanced K-coloring with fixed nodes is weakly NP-Hard.*

Proof. Let n and $X = \{x_1, \ldots, x_n\}$ be a given instance of partition. The graph we construct consists of $K-3$ nodes in a cycle and two in-stars. The first in-star has three nodes and the other has $n+1$ nodes. Let ω be equal to the sum of the weights in W. The weights on the first in-star will all be identically $\omega/2$ and the weights on the second in-star will be $x_1, \ldots, x_n$. The weights on the edges in the cycle with $K-3$ nodes are $\omega, 2\omega, \ldots, (K-3)\omega$. If $K-3=1$, then add a loop on the node with weight ω. More precisely, let $G = (S_1 \cup S_2 \cup C, E_1 \cup E_2 \cup E_C)$ where $S_1 = \{u_0, u_1, u_2\}$ and $S_2 = \{v_0, v_1, \ldots, v_n$. Let $C = \{c_1, \ldots, c_{K-3}\}$. Let $E_1 = \{u_1u_0, u_2u_0\}$ and $E_2 = \{v_iv_0 : i = 1, \ldots, n\}$. Let $E_C = \{c_ic_j : i < j\}$. For $ij \in E_1 \cup E_2$, the weight on ij is $w_{u_1u_0} = w_{u_2u_0} = \omega/2$ and $w_{v_iv_0} = x_i$ for $i = 1, \ldots, n$. For $ij \in E_C$, the weight is $w_{ij} = \omega$. Set $\mathcal{F} = \{\{u_0, v_0\}, \{u_1\}, \{u_2\}\}$. Note that this forces any in-balanced coloring to have $\omega/2$ weight going from nodes in a partition with u_1 to v_0 as well as the same $\omega/2$ weight going from nodes in a partition with u_2 to v_0. Note that if $K = 3$, then both C and E_C are empty sets. If not then, then any balanced coloring must assign the nodes of C distinct colors from all others, i.e. the nodes of C are all isolated and use $K-3$ colors. Thus, the nodes $v_1, \ldots, v_n$ must be in a partition either with u_1 or u_2 as there are edges that enter $v_1, \ldots, v_n$. Thus, a partition exists if and only if there is a balanced coloring or corresponds exactly to the color $v_1, \ldots, v_n$ are assigned. □

We have an immediate corollary.

Corollary 4.3. *The in-balanced ORP is weakly NP-Hard.*

Proof. Note that if any instance of ORP is solved with a zero optimal objective, then the original graph and has a balanced K-coloring. □

We are unaware of any other complexity results involving variants of balanced coloring/equitable partition. Interesting conjectures and open questions include the following.

- We conjecture that finding an undirected weighted balanced coloring with fixed nodes is NP-Hard.
- Famously, the complexity of graph isomorphism is unknown although many variants are efficiently solvable, including when the input graph has bounded degree [44]. Given that finding a minimal balanced coloring is a preprocessing step for the graph isomorphism algorithms [12, 49–52], is there a connection between theorem 4.2 and graph isomorphism?

J. Stat. Mech. (2022) 073403

5. Linear integer program formulation

In this section, we formulate equation (15) as an integer linear program. We also note that our formulation extends to the directed case as well as to the addition of positive or negative weights. Modifying the formulation to account for edge perturbations is possible but would require the addition of a new family of variables. The latter changes could require the formulation to become a mixed-integer linear program although this would depend on whether the perturbation restriction was to a discrete set or not.

We consider a problem of form equation (15), i.e. we are given an undirected graph $G = (V, E)$ and a positive integer K which denotes the number of colors we require for our pseudobalanced coloring. We let $M = |V| - 1$.

We use three sets of decision variables indexed over V, E, and $\mathcal{K} := \{1, \ldots, K\}$. For every $p, q \in V$, $k \in \mathcal{K}$, and $(i, j) \in E^{\mathrm{C}}$, let

$$x_{pq} = \begin{cases} 1 & \text{if nodes } p \text{ and } q \text{ are the same color} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{pk} = \begin{cases} 1 & \text{if node } p \text{ is color } k \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ijk} = \begin{cases} 1 & \text{if the pseudoedge of } (i, j) \text{ exists and } j \text{ is color } k \\ 0 & \text{otherwise} \end{cases}$$

Note that z_{ijk} is an indicator of a pseudoedge that also models the color of the node j. See equation (23) below for how the color assignment of j controls this. Controlling the decision variable in this manner allows equation (24) to be linear. However, because each pseudoedge has both a forward and backwards direction, the sums of the variables over the possible color assignments must equal each other. See equation (25) below for this constraint.

The natural objective function is to minimize the sum of pseudoedges. However, there are edges that we wish to incentify more than others, so we minimize the weighted sum of pseudoedges.

$$\min \sum_{(i,j) \in E^{\mathrm{C}}} c_{ij} \sum_{k \in \mathcal{K}} z_{ijk}. \tag{17}$$

Setting $c_{ij} = 1$ minimizes the sum of pseudoedges. For the graphs we solved, we found setting

$$c_{ij} = \frac{1}{d_i d_j}, \tag{18}$$

where d_i is the degree of node i, for $(i, j) \in E^{\mathrm{C}}$ produced the best results. We discuss this objective in more detail below.

J. Stat. Mech. (2022) 073403

We must ensure that every color is used at least once.

$$\sum_{i\in V} y_{ic} \geqslant 1, \quad k \in \mathcal{K}. \tag{19}$$

We must ensure that every node is assigned a color.

$$\sum_{k\in\mathcal{K}} y_{ik} = 1, \quad i \in V. \tag{20}$$

We must ensure that a color cannot be assigned to two distinct nodes unless they are indeed the same color.

$$x_{pq} + 1 \geqslant y_{pk} + y_{qk}, \quad p, q \in V, p \neq q, k \in \mathcal{K}. \tag{21}$$

Note that equation (21) only prevents the same color from being assigned to two nodes that are *different* colors. Nodes p and q with $x_{pq} = 1$ could be assigned different colors. Thus, to prevent this, we also need the following constraints.

$$(1 - x_{pq}) \geqslant y_{pk} - y_{qk}, \quad (1 - x_{pq}) \geqslant y_{qk} - y_{pk}, \quad p, q \in V, p \neq q, k \in \mathcal{K}. \tag{22}$$

We must enforce that pseudoweights are only assigned if the color is permitted.

$$z_{ijk} \leqslant W y_{jk}, \quad (i, j) \in E^{\mathrm{C}}, k \in \mathcal{K}. \tag{23}$$

We must ensure that the edge and pseudoedge weights from any pair of nodes that are the same color agree to every other color.

$$\left(\sum_{j\in V\,:\,pj\in E} A_{pj} y_{jk} + \sum_{j\in V\,:\,(p,j)\in E^{\mathrm{C}}} z_{pjk}\right) - \left(\sum_{j\in V\,:\,qj\in E} A_{qj} y_{jk} + \sum_{j\in V\,:\,(q,j)\in E^{\mathrm{C}}} z_{qjk}\right) \leqslant M_{pq}(1 - x_{pq}), \quad p, q \in V, k \in \mathcal{K}, p \neq q. \tag{24}$$

Note that this inequality is not symmetric and must be posed twice for every pair of nodes.

We must ensure that the sum of pseudoweights of the forward and backwards edges agree.

$$\sum_{k\in\mathcal{K}} z_{pqk} = \sum_{k\in\mathcal{K}} z_{qpk}, \quad pq \in E'. \tag{25}$$

In order to improve the formulation, we add the following antisymmetry constraints [56].

$$\sum_{i\in V} y_{ik} \geqslant \sum_{i\in V} y_{i(k+1)}, \quad k = 1, \ldots, K - 1. \tag{26}$$

This imposes an arbitrary ascending order on the size of the color sets.

J. Stat. Mech. (2022) 073403

The complete formulation for the instances we solve is then as follows.

$$
\begin{aligned}
B^* := \min \quad & \sum_{(i,j)\in E^{\mathrm{C}}} c_{ij} \sum_{k\in\mathcal{K}} z_{ijk} && (17)\\
\text{s.t.} \quad & (19),\ (20),\ (21),\ (23),\ (24),\ (25),\ (26) && \\
& y_{ik}, x_{ij} \in \{0,1\} && i,j \in V, k \in \mathcal{K}\\
& z_{ijk} \in \{0,1\} && (i,j) \in E^{\mathrm{C}}, k \in \mathcal{K}.
\end{aligned}
$$

(PBCIP)

For some experimental samples, it is also possible to have advance knowledge about nodes that are of the same color. For instance, many biological networks studied in [14] contain clusters of nodes in perfect balanced coloring already. These perfect colors are found by the algorithm of Kamei and Cock [18]. In these cases, it is important to keep these perfect colorings, and search for repairs that will not break these balanced colors. In such a case, as a pre-processing step, for all nodes in perfect balanced coloring $u, v \in V$, we add the constraint:

$$x_{uv} = 1. \tag{27}$$

6. Computational results for repairing a graph

Solving the integer linear program (PBCIP) from section 5 finds the minimum number of edges to add to a given graph to ensure a balanced coloring for a fixed number of colors. This is an important step towards our goal of finding pseudosymmetries of a graph. Our overall repair method is as follows and takes a candidate input graph with n nodes.

(a) Pre-processing: use the algorithm of Monteiro *et al* [19] or Kamei and Cock [18] to find a minimal balanced coloring present in the graph and identify the initial non-trivial colors. For the colored nodes, set equation (27). Let C denote the NNTC and T the number of trivial colors in this initial coloring.

(b) For $k = C$ to $C + T$ do the following:

1. Solve (PBCIP) for k colors, with constraints of form equation (27) added to fix the nodes that were already one of the C non-trivial colors.
2. Construct the resulting graph with the added edges dictated by (PBCIP) and color the nodes using the algorithm of Monteiro *et al* [19] again.

(c) *Evaluate* the resulting T graphs to identify what is the number of colors of the *best* repaired graph employing *indices* characterizing graph topology and stability analysis, see below.

In this section, we investigate different heuristics using graph topology and stability to accomplish step (c). We are aware of the methods presented in [11], whose method uses

J. Stat. Mech. (2022) 073403

pseudosymetries (section 2) and applies repairs manually using expert knowledge, in [41], whose method uses quasifibrations and applies repairs based on the similarity between nodes' input trees in directed graphs, in [39], who devises a Monte-Carlo randomized method that minimizes equation (1), $\|[P, A]\|$, and [57], whose method applies repairs in order to create an idealized graph. In our approach we generate a series of potential solutions by repeatedly solving (PBCIP) as described in steps (a) and (b). In order to accomplish step (c), a repaired graph is selected from this series of solutions by analyzing functions that characterize the underlying graph topology and stability of the solution. We call these graph functions *indices*.

We consider the same networks analyzed in the pseudosymmetry analysis done by Morone *et al* [11] from the connectome of *C. elegans* [13] obtained from the curated graph studied in [5]. The graphs studied are induced subgraphs of the connectome composed of neurons that are known to be involved in the locomotion function of the worm. These neurons have been identified in many experiments [13, 31] and are classified into command interneurons (like AVBL, AVBR or AVAL, AVAR) and motor neurons (the series of neurons denoted by VA, DA, and VB, DB, for dorsal and ventral neurons). The connectome is composed of two types of connections, gap junctions and chemical synapse. Around 10% of the synapses are gap junctions. We test the algorithm only with undirected networks, so we use the connectome of gap-junctions connections between those forward and backward neurons. The connectome of chemical synapses is made of directed edges, and it is not used here, but in the test of the repaired algorithm of [41] which uses quasifibrations and directed graphs. We study two undirected gap junction networks referred to as the *forward circuit* and the *backward circuit*. These networks represent the neural circuitry responsible for forward and backward movement of the worm *C. elegans*.

Following [11], we obtained the data from [5]. The undirected edges of this connectome are weighted since, generally, there are more than one gap junction connection between a pair of neurons. Typically, a neuron has a long axon which touches another neuron in different places along axon-to-axon or axon-to-soma contacts. The weight of the gap junctions is the number of those contacts. Here we will ignore the weight of the gap junctions and consider a binary adjacency matrix of 0 and 1 representing the absence or the present of a link. The neural network of *C. elegans* is known to have a 25% variability between specimens [5, 13].

Morone *et al* [11] presented a repair of these networks done manually by completing the network with the fewest possible links chosen by taking into account biological knowledge on the functions of the neurons. Although this manually repair network is not optimal, nor the true one, we will refer to this network as the 'ground truth' graph (or 'manual') with the ideal symmetry, so it provides an opportunity to test our more automated repair method.

The original (unrepaired) backward and forward circuits obtained from [5] are shown in figures 2(a) and 3(a), respectively. The original backward circuit contains 17 colors: 11 trivial and 6 non-trivial and forward circuit contains 15 colors: 13 trivial and 2 non-trivial. These colors are obtained using Kamei and Cock algorithm [18] for perfect balanced coloring. Solving (PBCIP) results in 12 possible repaired graphs for the backward circuit with the total number of colors ranging from 6 to 17 and 14 possible

J. Stat. Mech. (2022) 073403

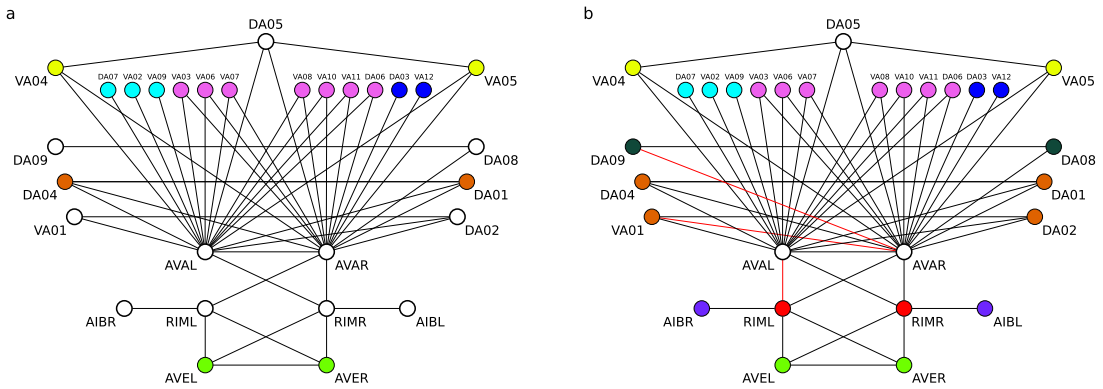

Figure 2. (a) The original backward circuit from [5] with 17 colors and (b) the repaired circuit resulting from our method with 12 colors. White nodes in both represent nodes with trivial, i.e. individual unique colors.

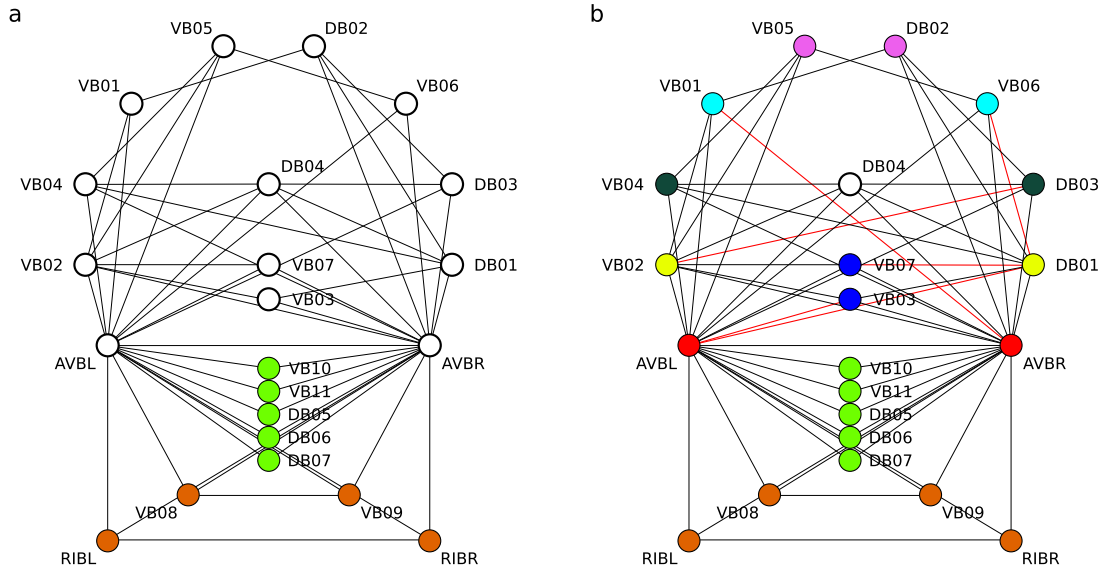

Figure 3. (a) The original forward circuit from [5] with 15 colors and (b) the repaired circuit resulting from our method with 9 colors. White nodes in the original represent nodes with trivial, i.e. individual unique colors.

repaired graphs for the forward circuit with the total number of colors ranging from 2 to 15. In section 6.1 we describe these graphs and in section 6.2 we justify our selection of the best repaired graphs with 12 colors and 9 colors for the backward and forward circuit, respectively. Our selected solution graphs are shown in figures 2(b) and 3(b).

6.1. Backward and forward circuit repairs

We first describe the repairs of the backward circuit. We observe that the addition of just a few edges can drastically change the coloring. For example, adding the edge

J. Stat. Mech. (2022) 073403

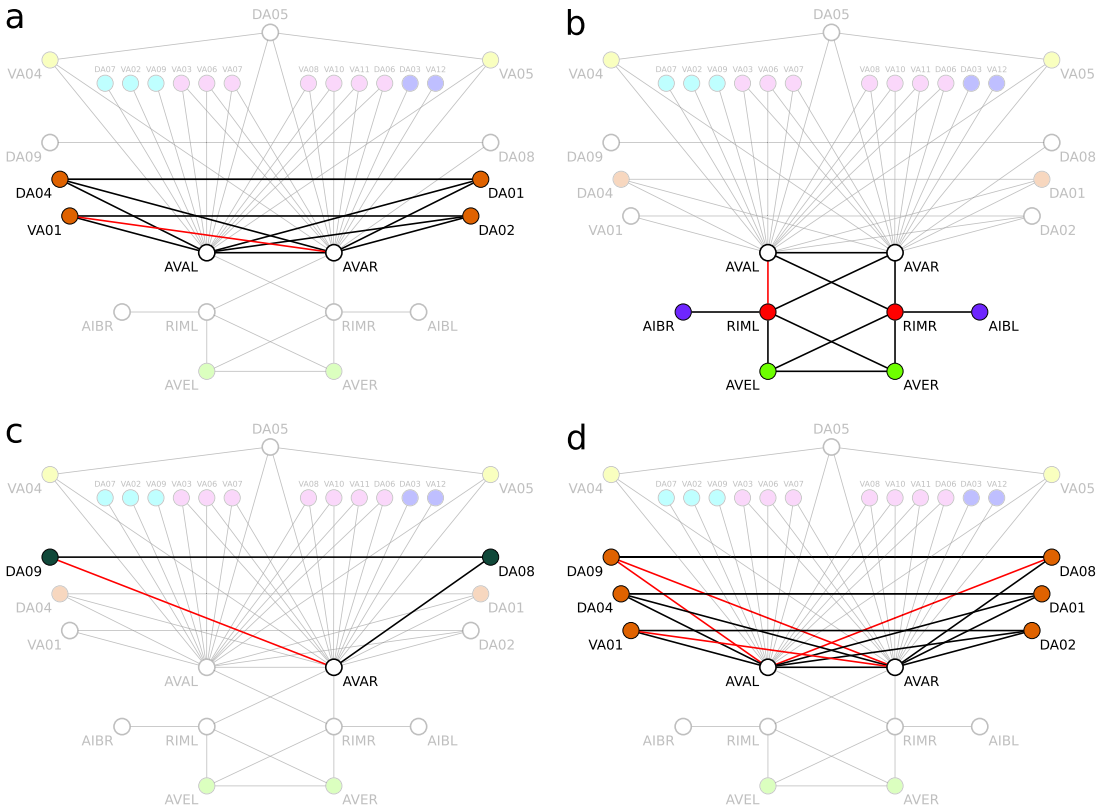

Figure 4. Repairs of the backward circuit I: (a) edge (AVAR, VA01) colors nodes VA01 and DA02 orange together with DA01 and DA04, (b) edge (AVAL, RIML) repairs the symmetry in the bottom part of the graph, (c) edge (AVAR, DA09) puts nodes DA08 and DA09 in the same color, (d) edges (AVAL, DA09), (AVAL, DA08), (AVAR, DA09), (AVAR, VA01) repair nodes VA01, DA01, DA02, DA04, DA08 and DA09 to the same color. This repair presents a slightly more complex version of repairs in (a) and (b).

(AVAL, RIML) (as shown in figure 4(b)) will make nodes RIML, RIMR, AIBR and AIBL symmetric. All repairs of the backward circuit can be decomposed into more simple repairs applied to the different part of the graph which are shown in figures 4 and 5. Using this simple decomposition of repairs we demonstrate the way our method combines them to obtain compound repairs in table 1. Note, repair figure 4(d) is the combination of repair figures 4(a) and (c), which are included in columns of table 1 corresponding to the number of colors between 11 and 9 for the simplicity of the interpretation. The repair with 9 colors shown in figure 5(c) is a combination of all the simple repairs and all the repairs with 17–9 colors can be visualized using figure 2(a) and table 1.

Thus far, we have described the repairs with the number of colors between 9 and 17. The eight-color graph is shown in figure 5(d). This solution combines all the repairs above additionally symmetrizing nodes DA08 and DA09 with the cluster DA01, DA02, DA04 and VA01 and nodes AIBL and AIBR with nodes DA03 and VA12 in order to symmetrize hubs AVAL and AVAR. Repairs with less than 8 colors have a very

J. Stat. Mech. (2022) 073403

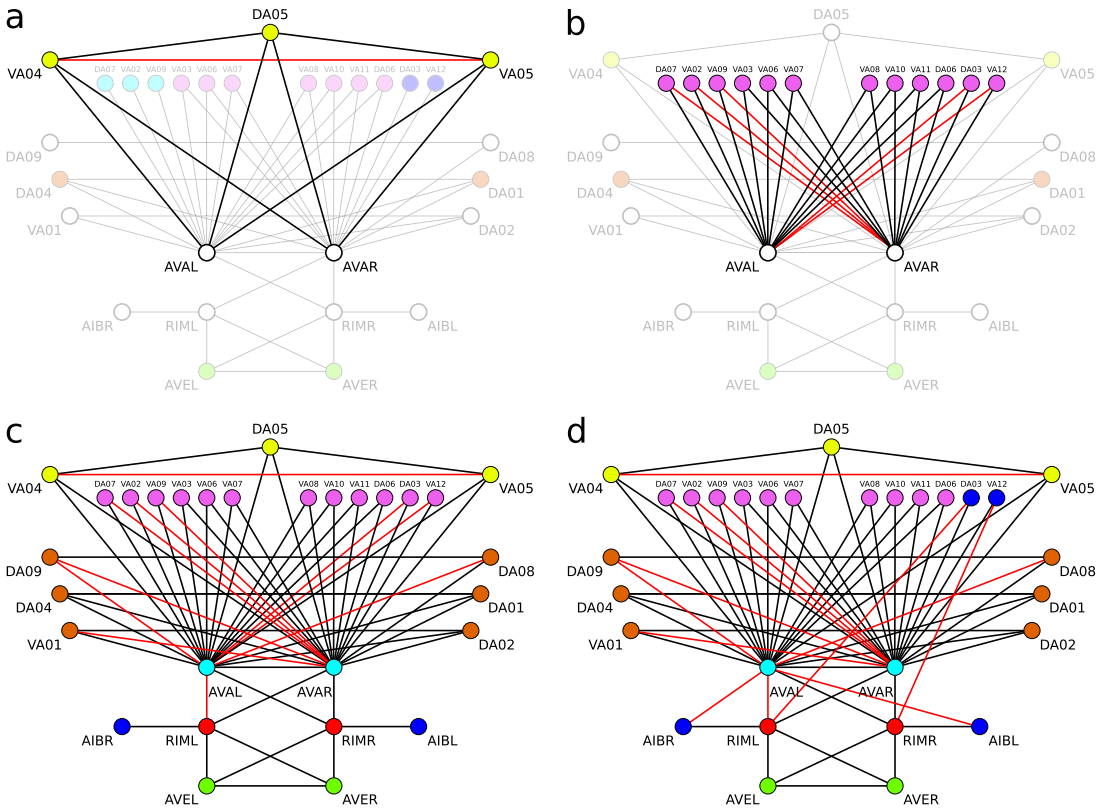

Figure 5. Repairs of the backward circuit II: (a) edge (VA04, VA05) adds node DA05 to the yellow color. (b) (AVAL, DA03), (AVAL, VA12), (AVAR, DA07), (AVAR, VA02), (AVAR, VA09). (c) Repair of the backward circuit with 9 colors. (d) Repair of the backward circuit with 8 colors.

Table 1. Incidence matrix of edge repairs 1–6 from figures 4 and 5 in the repairs of the graph corresponding to 17–9 colors. Each row represents the repair of the graph with different number of colors. As the number of colors decreases, the solution to (PBCIP) utilizes more compounded combinations of repairs.

Repair	17 colors	16 colors	15 colors	14 (=13) colors	12 colors	11 colors	10 colors	9 colors
Figure 4(a)		X		X	X	X	X	X
Figure 4(b)			X	X	X	X	X	X
Figure 4(c)					X	X	X	X
Figure 4(d)						X	X	X
Figure 5(a)							X	X
Figure 5(b)								X

complicated structure that is hard to visualize, and, as we will see later, the number of colors of the best solution lies far above this range, therefore we omit these repairs for simplicity.

J. Stat. Mech. (2022) 073403

Note that the 9 color repair in figure 5(c) only has 7 colors. Recall, as described in step (b)(1) of section 6, the formulation (PBCIP) has constraints of the form equation (27) to fix color of nodes that were known to be balanced as a result of preprocessing. Therefore, the colors of the fixed nodes DA07, VA02 and VA09 and nodes DA03 and VA12 cannot be changed or merged together with the pink color. To obtain the coloring in figure 5(c) in step (b)(2) our method obtains coloring using the minimal balanced coloring algorithm [19], which combines all these nodes.

Another thing to notice is that the solutions with 14 and 13 colors are the same. This happens due to the fact that the solution obtained from solving (PBCIP) with 14 colors assigns DA04 and DA01 a color that is different from the color of VA01 and DA02 even though the repair in figure 4(a) is applied. In the 13 color solution obtained by solving (PBCIP) all these nodes are assigned the same color as non-minimal balanced coloring is permitted and the minimum number of repaired edges required to obtain 13 colors is the same as that for 14 colors.

Next, we present the repairs of the forward circuit. The topology of the forward circuit is quite different from the topology of the backward circuit. As we saw above, the backward circuit can be decomposed into a few independent parts that can be repaired separately. The forward circuit, on the other hand, can only be decomposed into two parts: the bottom part, namely nodes DB05, DB06, DB07, VB10, VB11, VB08, VB09, RIBL and RIBR that are symmetric and are therefore fixed by the formulation and the top part consisting of the rest of the nodes. The top part has high connectivity and the optimal repairs obtain solutions with symmetries that act on most of the nodes. Instead of studying repairs as combinations as we did in the backward circuit, we examine all repairs separately. Figure 6 shows the obtained repairs starting from the original circuit with 15 colors and ending with 7 colors. Table 2 shows the incidence matrix of the edges used in these repairs. As before, we omit solutions with 2–6 colors since it will be shown that the best solution is not in this range.

Observing solutions in figure 6 and table 2 we come to a few conclusions. First, we observe that due to the use of the objective that incentivizes the addition of edges between the high degree nodes, hubs are the first nodes to be repaired in the 14-color repair and they stay repaired almost throughout the rest of the repairs. Second, as the number of colors increases, the number of non-trivially colored nodes increases until almost none of the nodes are colored trivially. Third, the high connectivity of the top part of the circuit leads to instability in the use of the repaired edges and in the clustering of nodes. That is, edges that are used in the repairs with K colors are often not used in the repairs with $K-1$ colors and nodes that have the same color for K colors may not have the same color for $K-1$ colors.

Morone *et al* [11] showed the importance of circulant structures in the locomotion networks. In particular, a circulant matrix in the adjacency matrix corresponds to *cycles* in the network. A cycle is a path in a network plus an edge from the last node to the first, e.g. in figure 3(b), VB02-DB03-DB02-VB01-VB02 or VB04-DB01-VB06-VB05-VB04. Cycles are important to generate oscillatory behavior [11, 15, 58] and, as was mentioned before, the *C. elegans* locomotion process follows oscillatory patterns. Therefore we look for circulant submatrices in the ideal solution for the forward circuit. In general, finding circulant submatrices in a given adjacency matrix is a problem on its own, and we

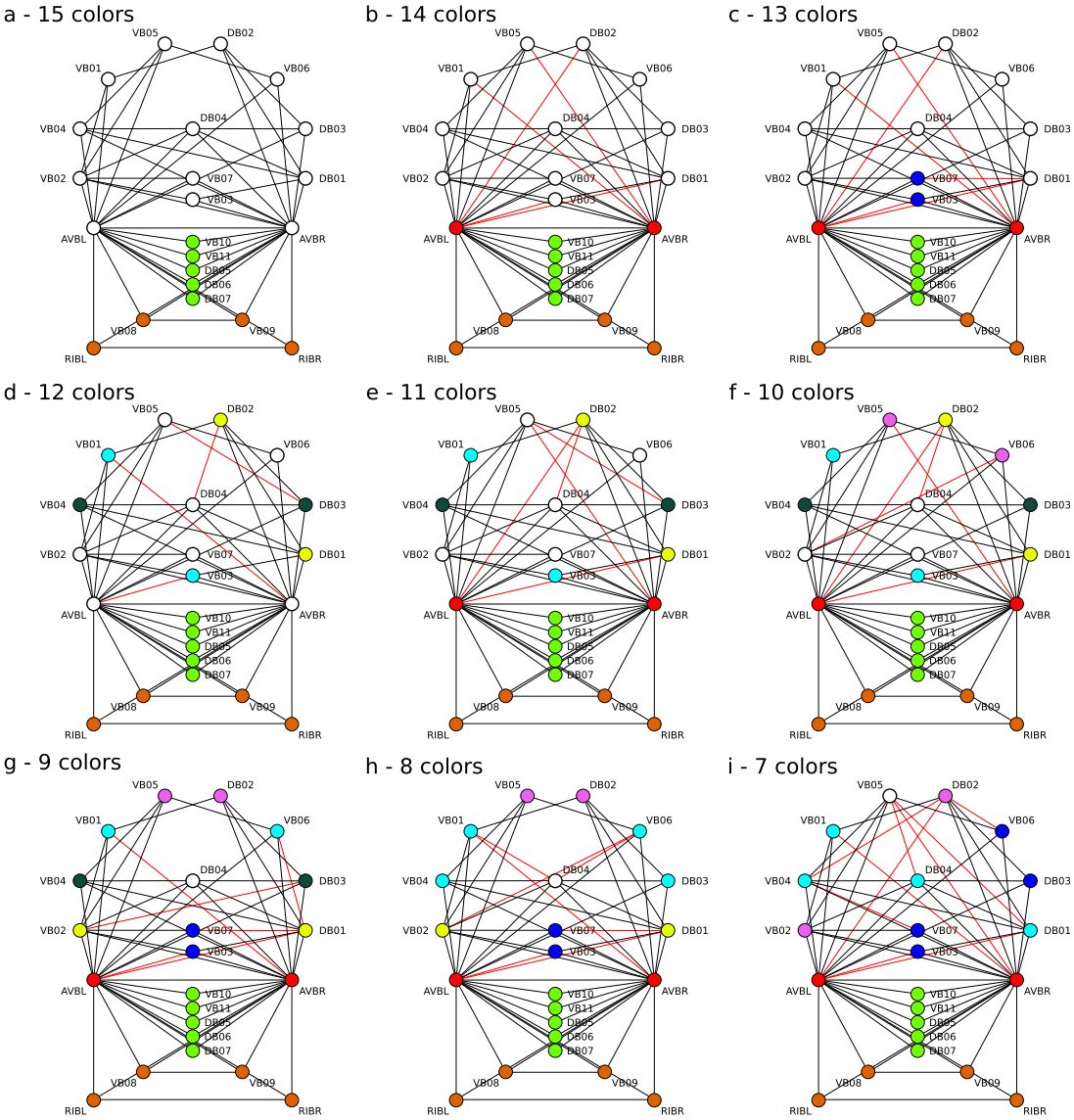

Figure 6. (a) The original forward circuit from [5], (b)–(i) repaired versions of the forward circuit with 14–7 colors. Colors correspond to the minimal balanced coloring. Red links are repaired edges, i.e. edges not present in the original graph.

are not aware of an algorithm that could do this automatically. For the small matrices considered here, we resort to find this circulant matrices by inspection, as done in [11]. Figure 7 shows the adjacency matrix of the optimal forward circuit in figures 6(g) and 11(a) found by the formulation. The first eleven rows and columns represent the top part of the forward circuit (excluding hubs AVBL and AVBR) comprising most of the motor neurons. We note that the 8 by 8 submatrix in the upper-left corner of this adjacency matrix is (transpose) block circulant. That is, it is composed of two matrices. A circulant matrix that represents a four-cycle permutation of VB02-DB03-DB02-VB01-VB02 and

Table 2. Incidence matrix of edge repairs in the forward circuit shown in figure 6. Each row represents a different edge and each column the repair of the graph with different number of colors. Some edges are omitted to improve the readability. The obtained solutions are inconsistent about which edges are repaired, i.e. the obtained solutions with $K-1$ colors utilize edges that are very different from the ones used in the solution with K colors. Note that some edges are omitted.

Edge	15 colors	14 colors	13 colors	12 colors	11 colors	10 colors	9 colors	8 colors	7 colors
VB02, VB06		X			X				
VB05, VB07		X							
AVBR, VB05			X	X					
DB01, VB05			X						X
DB02, VB06			X						
VB03, VB05				X		X			
VB01, VB05				X					X
DB03, VB04				X					
VB01, VB07					X		X		X
DB03, VB02					X			X	
DB01, DB03					X				
AVBL, DB02					X				
AVBL, DB01						X	X	X	
DB04, VB06						X		X	X
VB01, VB04						X			X
DB02, VB02						X			

also of VB04-DB01-VB06-VB05-VB04:

$$\mathcal{F} = \mathrm{circ}(0,1,0,1) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \tag{28}$$

and a matrix $\mathcal{B}$ in the off-diagonal, such that the full 8 by 8 matrix is represented as:

$$\mathcal{BC} = \begin{bmatrix} \mathcal{F} & \mathcal{B} \\ \mathcal{B}^T & \mathcal{F} \end{bmatrix}. \tag{29}$$

The transpose is needed due to the undirected character of the graph. The same circulant structure, with exact the same loops represented by $\mathcal{F}$, has been found in the manually curated solution in [11]. This result is encouraging since it implies that the formulation is able to find not only close to optimal balance colorings but also by-product structures like circulant cycles in the network, which, in the particular case of locomotion, are necessary for the oscillatory motion of the animal.

6.2. Indices on the graph

To identify the best repair over the number of K colors, we consider a set of indices that characterize the graph topology and the stability of the dynamical solution imposed on

J. Stat. Mech. (2022) 073403

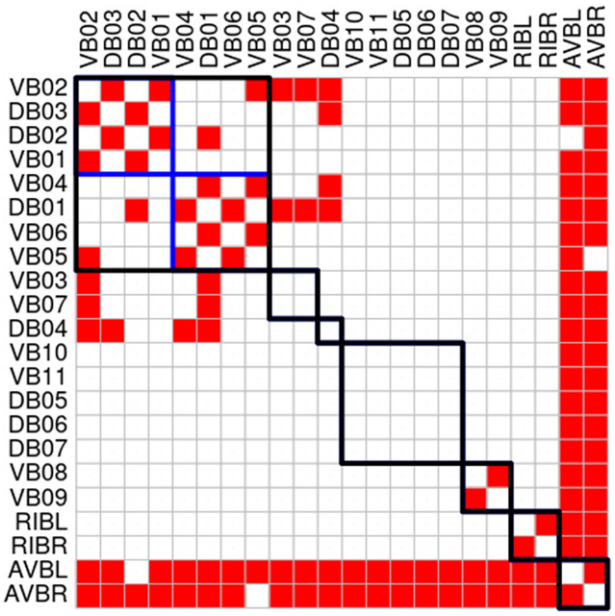

Figure 7. Circulant structure in the optimal repaired network found by the formulation in the forward circuit of figures 6(g) and 11(a). This circulant structure is the same as found in the manually crafted solution in [11].

the graph. We are looking for indices that have a qualitative change in their behavior that can indicate the most optimal solution of K colors.

First, we list a few indices characterizing the partition induced by the balanced coloring. We consider the *number of trivial colors* and the NNTC as a function of the K colors. Each of these count the numbers of colors that are trivial and non-trivial, respectively. We also consider the *number of nodes in non-trivial colors* which calculates the total number of nodes that belong to non-trivial colors.

Figure 8 (rows 1–3) shows the values of these indices obtained on the repaired graphs. First, we consider the NNTC. As per the intuition stated in section 3, a higher NNTC indicates the possible increase in the number of symmetries of the graph. Additionally, we can gain some insight by examining the limits of high number of colors and low number of colors. For a high number of colors, all non-fixed nodes will be in the separate classes, hence NNTC will be low. For a low number of colors a lot of non-fixed nodes will merge with fixed colors, thus NNTC will be low as well. We see in figure 8 that NNTC possesses two local minima for high and low numbers of colors and has a local maximum in between them. We use the solution corresponding to the local maximum as our best solution to pick the optimal number of colors which we call K_{opt}.

Using the NNTC index we choose the best solutions for the forward and backward circuit to be at $K_{\text{opt}} = 9$ and 12 colors, respectively. The best solution is shown in green and the manual solution obtained in [11] is shown in red. Index values for the manual solution are shown with red horizontal lines. The number of colors for the best solution of the forward circuit was the same as the manual solution and the repairs used are similar. The number of colors for the backward circuit is different from the manual solution. We

J. Stat. Mech. (2022) 073403

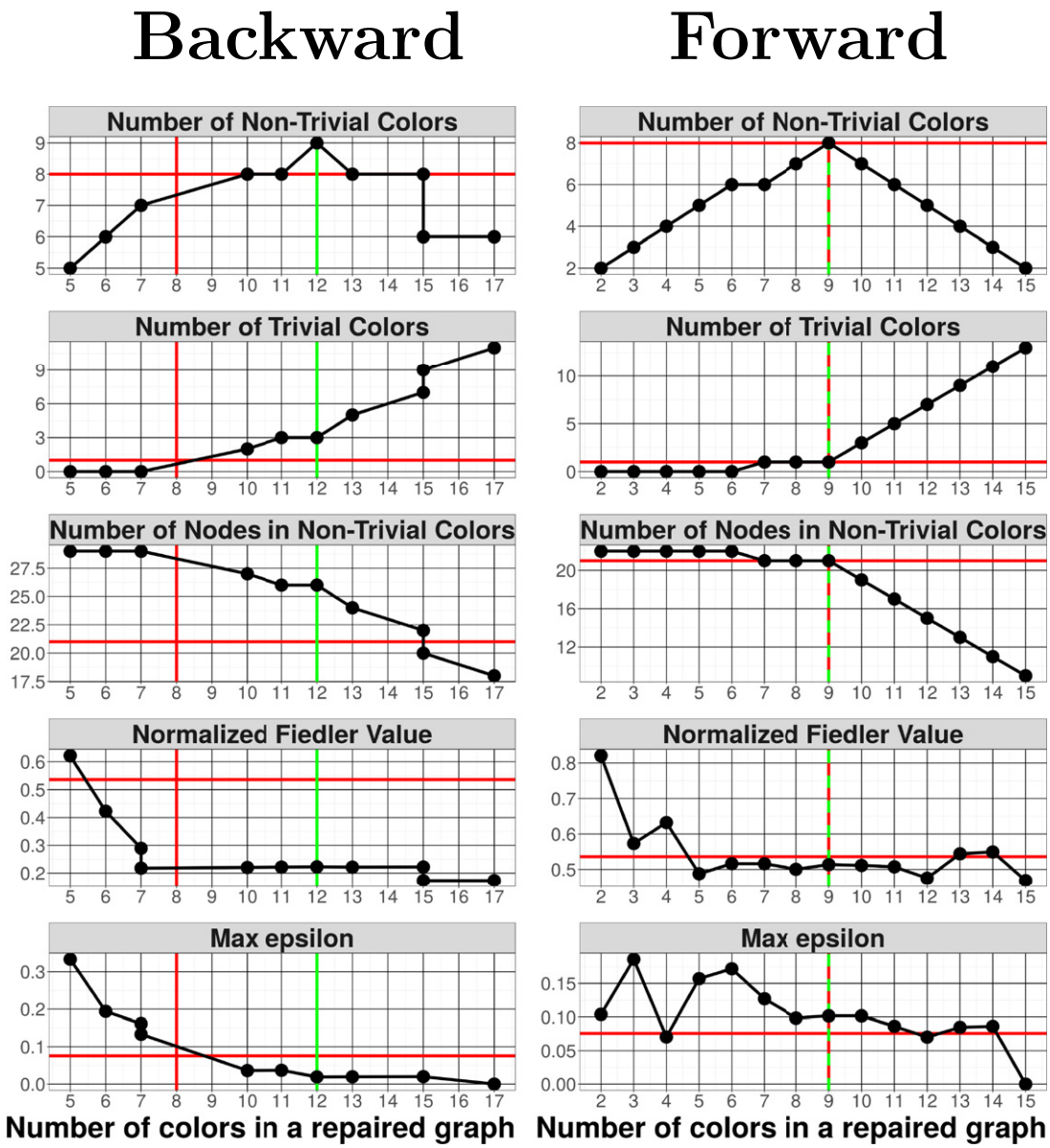

Figure 8. Color indices of the backward circuit (on the left) and forward circuit (on the right). Vertical red lines indicate the number of colors in a graph from Morone and Makse [11]. The horizontal red line indicates the value of the index for the same network. The green area on the left shows the optimal range of the number of colors for the backward circuit. The dashed green line on the right shows the optimal solution for the forward circuit.

believe this is because the 12 color solution does not restore the symmetry between hubs creating more colors as shown by the dark and light blue colors in figure 2(b). We will further discuss these differences in section 8. The number of trivial colors stays approximately constant below the best number of colors and starts increasing above it.

J. Stat. Mech. (2022) 073403

The number of nodes in non-trivial colors behaves similarly, but decreases instead of increasing above the chosen solution.

The graph indices above characterize the coloring of the graph, but we cannot limit our consideration to the topology of the graph since biological networks and neural networks model systems that perform functions vital for the organism's or ecosystem's survival. Therefore the dynamics of the network also needs to be accounted for. Next, we search for the optimal graph over K by looking for the repair that could provide the largest stability to a given dynamical solution on the graph. In particular, we look for a graph that could provide a larger synchronizability, as defined by a larger stability of the cluster synchronization solution predicted by the balanced coloring on the graph.

Consider the eigenspectrum of the graph Laplacian. For a graph, $G = (V, E)$ with node-node binary adjacency matrix A, the random walk graph Laplacian [59] is the matrix,

$$\begin{aligned} L_G^{\mathrm{R}} &= D^{-1}(D - A) = I - D^{-1}A = I - D^{-1/2}(D^{-1/2}AD^{-1/2})D^{1/2} \\ &= D^{-1/2}(I - D^{-1/2}AD^{-1/2})D^{1/2} = D^{-1/2}L_G^{\mathrm{N}}D^{1/2} \end{aligned} \tag{30}$$

where D is a $|V| \times |V|$ diagonal matrix with the D_{ii} equaling the degree of node i and L_G^{N} is the normalized graph Laplacian. The matrix L_G^R is symmetric, positive semidefinite, and singular so its eigenvalues are all nonnegative and real with at least one equaling zero. The eigenvalues of the graph Laplacian are well known to be associated with both the combinatorial and dynamical properties of the graph [60–62]. In particular we consider *the normalized Fiedler value* defined as the second smallest eigenvalue of the random walk Laplacian. This has been shown to provide a measure of graph complete synchronizability [63, 64] for networks in which the response of the individual nodes is adjusted based on the number of incoming connections, which is consistent with the case of neural networks [59]. Complete synchronization of a graph is the state in which for all $i, j \in V : x_i(t) = x_j(t)$. Complete synchronizability, then, refers to the ability of the graph to achieve stable complete synchronization.

Different clusters of a biological network, e.g. clusters in the backward and forward circuits, perform different biological functions through cluster synchronization from the balanced colorings. These clusters are able to perform independent synchronized functions, and still be integrated in the network. Thus, a functional biological network requires cluster synchronization. Complete synchronization, instead, is deleterious for the organism. Therefore repairs that increase complete synchronizability of the graph are undesirable.

Figure 8 (4th row) demonstrates the normalized Fiedler value for the obtained repairs. We observe that the normalized Fiedler value of the found best solution K_{opt} is in the range of the lowest values of all solutions. This behavior suggests that the best solution is one of the solutions that is less prone to the complete synchronizability, which is desirable for the dynamics of the system.

To summarize, we saw that the NNTC can be used as an indicator for the best solution K_{opt}. Therefore, in this paper we identify the best solution as the one with the highest NNTC. We also examined other indices such as the mean color size (the number of nodes divided by the number of colors), average clustering coefficient, average path

J. Stat. Mech. (2022) 073403

length, the Randic index, the number of repaired edges and normalized Fiedler value, but they exhibited erratic behavior that did not seem indicative of any critical point on the studied graphs to choose the optimal one based on these indices. However, the normalized Fiedler value presents a plateau with minimum near K_{opt} (figure 8) indicating that there is a range of graphs with colors around K_{opt} that minimize the stability of the complete synchronization state which is beneficial for the biological network.

We believe that further work needs to be done in order to identify the best index, in particular, we believe the ideal index would need to more accurately characterize the synchronizability and stability of the dynamics on the graph. For instance, it would be desirable to analyze the stability of the cluster synchronization solution, rather than the instability of the complete synchronization solution as captured by the Fiedler value. The theory of cluster synchronizability is being developed in [65] and could provide a good index candidate to choose the optimal coloring solution. Here we only consider two networks that are fairly symmetric from the start and, in order to avoid overfitting, we choose NNTC as the graph function to select our solutions. An interesting open question is then to find the best index or indices to use to select the best candidate solution.

7. Automorphism groups and pseudosymmetries of the repaired graphs

In this section, we describe how the factorization into normal subgroups of the automorphism group of a graph [66] performed in the *C. elegans* connectome in [11] partitions the graph into functional clusters, and how this partition can be obtained from (PBCIP). We emphasize that a repair of the graph guided by balanced coloring provides a less stringent condition on the number of repaired edges than a repair following the restoration of the full symmetry automorphism group as done in [11]. That is, repairing the network to achieve perfect balanced coloring will always require less (or at most equal) number of edges than repairing the network with automorphisms. This is because, automorphisms impose more strict conditions on the structure of the network than balanced colorings and fibrations. This is particularly true when the graph is directed and one is only interested in cluster synchronization. Cluster synchronization is exclusively determined by the information that a node receives through its in-degree from the entire network. This information is taken into account either by the input tree of the node as in the fibration formalism of [14] or by the analogous in-balanced coloring as considered here. The out-balanced coloring is irrelevant for cluster synchronization (see [14] for further details).

Instead, the symmetries imposed by automorphisms require invariance of the full adjacency, including the in and out-degree. Thus, automorphisms require more stringent conditions than what is required to achieve cluster synchronization. Nevertheless, Morone *et al* [11] showed that automorphisms exist in the connectome, in particular in the undirected gap junction connectome where the distinction between in and out balanced is meaningless. These symmetries pose the particular property of factorization of the symmetry group, and this factorization separates the neurons into known classes like interneurons, motor and touch neurons. Thus, this particular symmetry group

J. Stat. Mech. (2022) 073403

has a functional connotation. Therefore, next, we relate the pseudobalanced coloring formalism with the pseudosymmetry formalism of [11].

For a graph, $G = (V, E)$, let $\texttt{Aut}(G)$ denote the automorphism group of G. For any permutation $p \in \texttt{Aut}(G)$, the *support of* p [11, 12, 66], denoted by $\texttt{supp}(p)$ is defined as the nodes which do not have the same labels after the permutation p has operated on G. Formally, this can be written

$$\texttt{supp}(p) := \{i \in V | p(i) \neq i\}. \tag{31}$$

Two permutations, $p, q \in \texttt{Aut}(G)$, are said to be *support-disjoint* if the node labels they change are completely different, i.e. if $\texttt{supp}(p) \cap \texttt{supp}(q) = \emptyset$. Two sets of permutations $P, Q \subseteq \texttt{Aut}(G)$, are support-disjoint if every permutation in P is support disjoint with every permutation in Q. Note that any two support-disjoint permutations commute.

Let S denote the set of generators of an automorphism group $\texttt{Aut}(G)$. Partition S into n support-disjoint subsets $S = S_1 \cup S_2 \cup \cdots \cup S_n$ such that none of the S_i can be further decomposed into smaller support-disjoint subsets. Each S_i generates a group H_i that is a subgroup of $\texttt{Aut}(G)$. Since all S_i are support-disjoint, H_i commute with each other and, since S is a union of all S_i, $\texttt{Aut}(G)$ is a direct product of all H_i:

$$\texttt{Aut}(G) = H_1 \times H_2 \times \cdots \times H_n. \tag{32}$$

All H_i are normal subgroups, since they commute with the rest of the group. Therefore, partitioning the set of generators of an automorphism group into irreducible support-disjoint subsets creates a factorization of this group. The uniqueness and irreducibility of this representation has been shown in [66]. Note that H_i acts on the separate non-intersecting subsets of nodes defining a partition (also referred to as a factorization) on the nodes of a graph. We call the equivalence classes of nodes created by this partition *sectors* [11].

Therefore, we have a way to factorize the graph via the automorphism group associated with it. We describe how to classify the obtained factors. Each factor is vertex-transitive, but vertex-transitive graphs are too broad a class to admit a complete classification. Nevertheless, we apply a simple computational approach that allows us to classify most of the obtained subgroups. In order to classify each normal subgroup we determine whether it is isomorphic to a symmetry group from this list: a symmetric group, a dihedral group, a cyclic permutation group, or an alternating group of sizes 1 to n, where n is the number of nodes in the graph. If the normal subgroup is isomorphic to one of these classes, then we assign this class to it. The implementation uses NAUTY [67], Sympy [68] and Sage [69] and is available at https://github.com/makselab/PseudoBalancedColoring.

To illustrate the decomposition process, we apply the algorithm to the repaired backward circuit. The automorphism group is decomposed into

$$\texttt{Aut}(G) = H_1 \times H_2 \times H_3 \times H_4 \times H_5 \times H_6 \times H_7 \times H_8, \tag{33}$$

where $H_1 = D_4$ applied to nodes DA01, DA04, DA02 and VA01, $H_2 = S_2$ applied to nodes VA04 and VA05, $H_3 = S_2$ applied to nodes AVEL and AVER, $H_4 = S_7$ applied to nodes DA06, VA03, VA06, VA07, VA08, VA10 and VA11, $H_5 = S_2$ applied to nodes

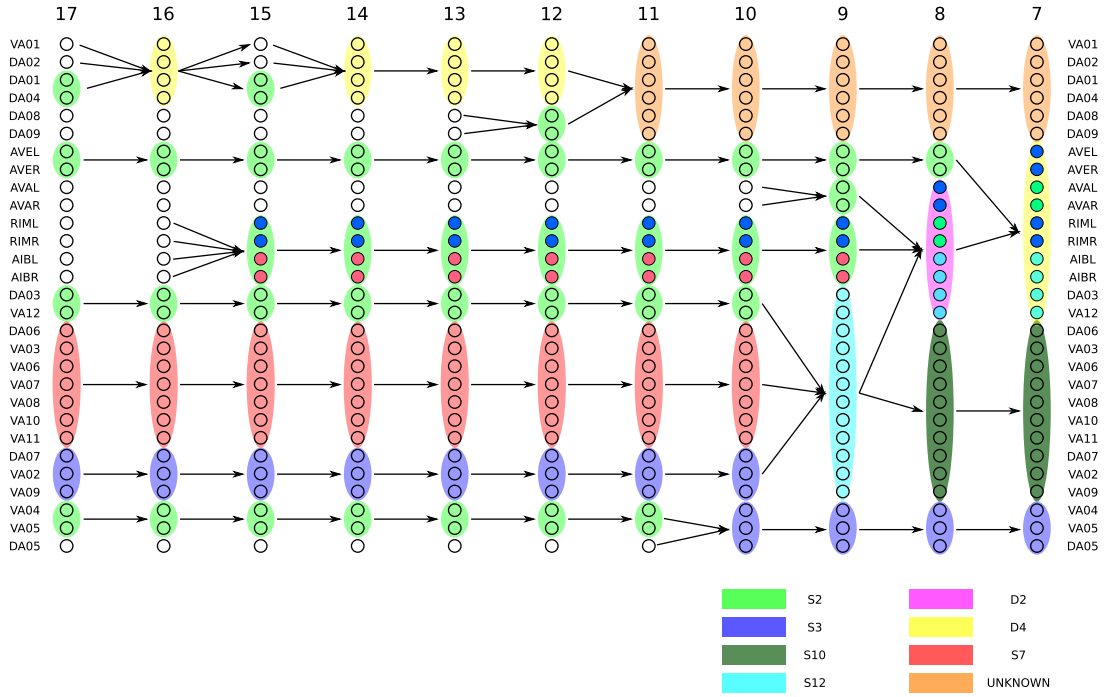

Figure 9. Step-by-step transformation of the backward circuit automorphism group. Each column corresponds to the repair with K colors. Each row corresponds to one node. Colored circles show 7 classes of the normal subgroups. Nodes are colored inside the sector according to the orbit they belong to. For example, group S_7 in 15 color repair has only one orbit, hence all the nodes are of the same color. Group of RIMR, RIML, AIBL, AIBR in 15 color repair has two orbits which are shown with red and blue colors.

DA03 and VA12, $H_6 = S_3$ applied to nodes DA07, VA02 and VA09, $H_7 = S_2$ applied to nodes AIBR, AIBL, RIML and RIMR, $H_8 = S_2$ applied to nodes DA08 and DA09.

We apply this algorithm to the repaired graphs obtained by our algorithm on the backward and forward circuits. The result of this analysis for the backward circuit is shown in figure 9. Similar to the earlier presented conclusions from table 1, we see here that between 17 and 9 colors, the formulation (PBCIP) combines different simple repairs to obtain a symmetrized version of the network. This is evident by the fact that almost all new subgroups for $K-1$ colors are created by merging some nodes (or subgroups) in K colors and nodes never move between subgroups. Note that our computational approach was able to classify all of the subgroups aside from the subgroup marked as 'UNKNOWN' in the backward circuit.

The results for the forward circuit are shown in figure 10. As observed before, the forward circuit is decomposed into two parts: a top part and a bottom part. As expected, the bottom part of the graph represented by the bottom 9 nodes of figure 10 stays unchanged. The top part, as seen before, does not have a consistent partition and an increase in the number of colors combines different nodes in normal subgroups. Two things can be observed: (1) as the number of colors decreases, the number of symmetric nodes increases, and (2) nodes are mostly combined to exhibit mirror symmetry (D_1).

J. Stat. Mech. (2022) 073403

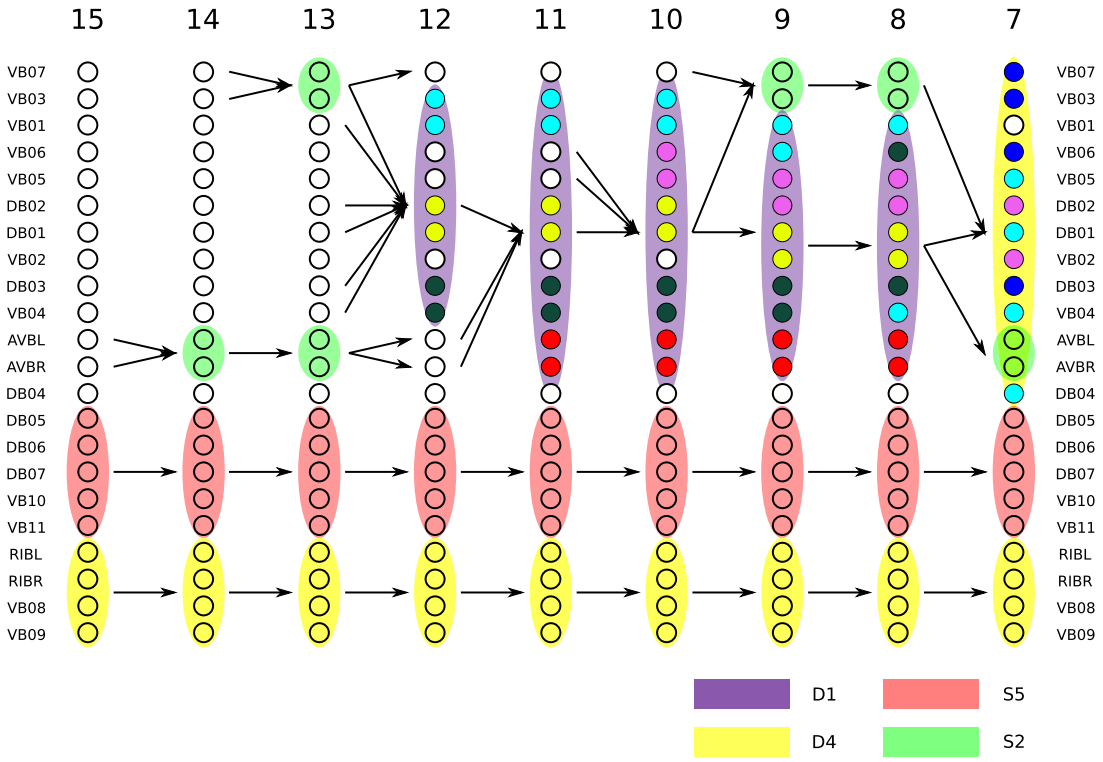

Figure 10. Step-by-step transformation of the forward circuit automorphism group. Each column corresponds to the repair with K colors. Each row corresponds to one node. Colored circles show three classes of the normal subgroups. Nodes are colored inside the sector according to the orbit they belong to. Colors of orbits correspond to node colors in figure 6. Nodes of the white color inside the sector do not belong to the sector. For example, group with nodes DB03, DB04, VB05 and VB06 in 13 color repair contains two orbits shown with yellow and blue colors and nodes DB02, VB01 and VB04 do not belong to this sector.

Note that groups D_1 and S_2 are isomorphic and we use them interchangeably, D_1 is more appropriate in a case of mirror symmetry and is therefore used here. For example, in figure 6(g) there is mirror symmetry in the top part between the nodes on the left and the nodes on the right.

To complete our analysis of the repaired networks, we find ε described in section 2. ε represents the cutoff of the norm in equation (2), i.e.

$$\|[P_\varepsilon, A]\| \leqslant \varepsilon \tag{34}$$

where P_ε represent all permutations in automorphism group of the repair of a graph G. We denote the repaired graph G_ε; ε then can be found using $\texttt{Aut}(G_\varepsilon)$ as:

$$\varepsilon = \max_{P_\varepsilon \in \texttt{Aut}(G_\varepsilon)} \|[P_\varepsilon, A]\| . \tag{35}$$

Due to the size of the automorphism group, finding ε using the entire automorphism group is computationally intractable. To give an approximation, we find ε on the set

of generators of the normal subgroups. That is, we first calculate the maximum ε on generators of each normal subgroup and then we take the resulting ε as their maximum. To simplify the interpretation of the obtained ε, we define

$$\epsilon = \frac{\varepsilon}{4M}, \tag{36}$$

where M is the total number of edges in the amended graph and 4 is the constant appearing due to the circumstances described in section 2. Then, ϵ is the maximum of the number of edges that needs to be added to the original graph in order to make the permutation P_ε a part of the automorphism group of the amended graph normalized by the total number of edges in the graph. Figure 8 (last row) shows the values of ϵ for the obtained repairs. We see that all obtained repair graphs are well below 0.25 (25%, except the backward repair with 5 colors) which is the value found in [5] to be associated with the natural variability in the number of links that are different across the five animals that have been used to build the connectome in [13]. Therefore, from the point of view of pseudosymmetries, each of the repaired networks can be a candidate for the 'blueprint' ideal symmetry network of *C. elegans*. Note that figure 8 shows the maximum values of ϵ as defined in equation (35).

8. Comparison with other link prediction methods

To complete our analysis, we compare the results obtained by the formulation (PBCIP) against the results of the manual repairs obtained in [11] and some of the traditional link prediction methods.

We begin by comparing our results with the manual repair 'ground truth' of [11]. Figure 11 shows the ideal (a) and manual (b) solutions for the forward circuit. The bottom part was left unchanged in both solutions. The top part of the circuit is very similar between solutions: coloring is exactly the same and the only difference is the addition of edges (AVBL, DB02) and (AVBR, VB05) in the manual solution. The similarity between these solutions can be ascribed to the fact that both repair methods have a similar objective. Morone *et al* [11] obtained their manual solution by repairing the symmetry between the hubs and adding edges to complete the symmetry between the rest of the nodes. Likewise, the formulation (PBCIP) with the weighted objective in equation (18) incentifies repairs between nodes with higher degrees (hubs).

The difference between the solutions can be interpreted by observing that the manual solution decomposes the automorphism group of the top part into S_2 (AVBL, AVBR) $\times D_1$ (VB02, VB04, VB01, VB05, DB02, VB06, DB03, DB01) $\times S_2$ (VB03, VB07), while our solution decomposes it into D_1 (AVBL, AVBR, VB02, VB04, VB01, VB05, DB02, VB06, DB03, DB01) $\times S_2$ (VB03, VB07). Colorings corresponding to these decompositions are the same and therefore, because the objective of (PBCIP) is to minimize the weighted sum of edges added, adding two extra edges is not optimal. However, the logical assumption that the function of the hubs is different from the rest of the nodes led [11] to dividing hubs into a separate subgroup. We conjecture that the solutions to

J. Stat. Mech. (2022) 073403

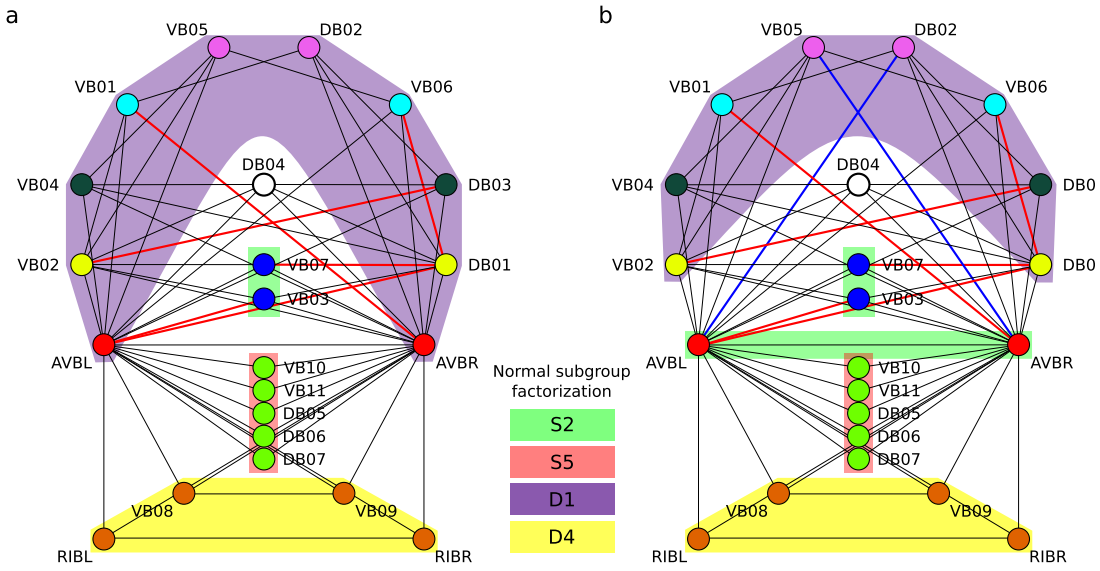

Figure 11. Comparison between the most optimal repair solution found by the formulation (a) and manually crafted 'ground truth' solution (b) from [11] for the forward circuit. Red edges are repaired in both solutions and blue edges are repaired only in the manual solution. Background colors show the sectors associated with the normal subgroup factorization. Notice the complete agreement between both repairs regarding the colors, and almost complete agreement regarding the pseudoedges and normal subgroup factorization. The pseudo edges are the same except for two extra blue edges in the manual solution. This does not affect the colors. AVBL-AVBR belongs to the motor group D_1 in the formulation but they are an independent sector and subgroup S_2 in the manual solution. The blue edges were added to the manual solution in [11] to factorize the command interneurons AVB from the motor sector, but the formulation found another more optimal solution.

(PBCIP) would have separated hubs in a bigger network in order to symmetrize other parts of the network connected to them.

Figure 12(a) shows the ideal solution obtained by our formulation and figure 12(b) shows the manual solution for the backward circuit. We can separately compare the repairs applied to different parts of the circuit. Firstly, repairs (AVAL, RIML) and (VA01, AVAR) were obtained in both solutions. Secondly, nodes DA08 and DA09 were repaired to be symmetric similar to the manual version, but links between these two nodes and AVAL needed to symmetrize them with the rest of orange nodes were not repaired. Lastly, hubs AVAL and AVAR along with the manually restored nodes with S_{12} symmetry in pink were not repaired. The solution with 9 colors in figure 5(c) does symmetrize AVAL-AVAR, all of the pink nodes and DA08-DA09, but creates an extra symmetry between nodes DA05, VA04 and VB05. Hence, this repair can be obtained, but it was not chosen due to the fact that the maximum of NNTC corresponds to the solution with 12 colors. This provides evidence that further analysis of indices is needed to improve the way to identify the best solution.

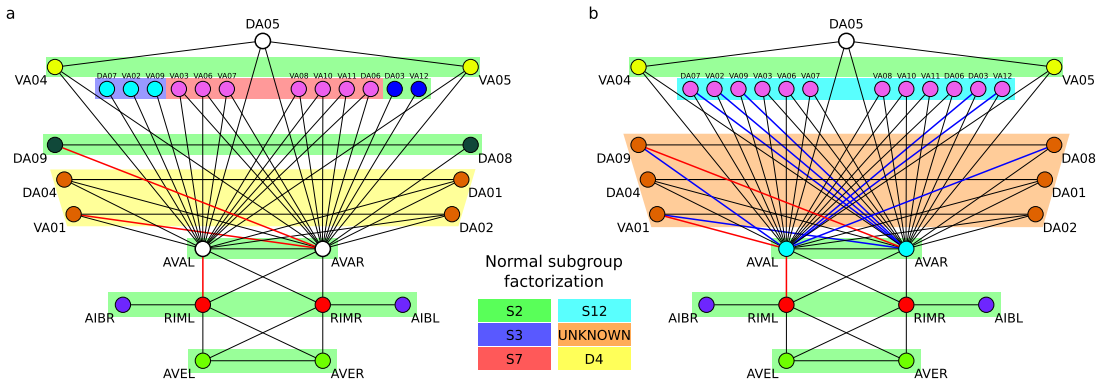

Figure 12. Comparison between the most optimal repair solution found by the formulation (a) and the manually crafted 'ground truth' solution (b) from [11] for the backward circuit. Red edges are repaired in both solutions and blue edges are repaired only in the manual solution. Background colors show the sectors associated with the normal subgroup factorization. The crucial AVAL-RIML link is found by both solutions. However the formulation finds that the S_2 normal subgroup of the manual solution can be repaired with fewer pseudoedges by breaking this subgroup in three sectors as shown. As in the forward circuit, figure 11(b), the manual solution attempts to factorize the interneurons AVAL-AVAR from the motor sector, at the expense of creating more pseudoedges than needed. In contrast, the formulation finds a better solution with less pseudoedges.

Now we take the manual repair as the ground truth and compare the performance of the formulation with some of the traditional link prediction methods. The link prediction problem is a problem of predicting the probability of the existence of the edge between two not connected nodes given the observed data. This probability is defined as the similarity between considered nodes obtained using a topological measure of the graph [32, 34, 36]. To repair a graph using a link prediction approach, an empirically chosen number of top ranked edges is predicted to exist [35]. Commonly used measures of similarity are divided into local, global and quasi-local classes. Further, we will define some of them following [32] and propose a way to estimate their performance.

First, we introduce a few indices characterizing the local topology of the graph. Let S_{xy} be the distance between nodes x and y, $\Gamma(x)$ be the neighborhood of node x (set of nodes connected to x), $|X|$ be the cardinality of set X and k_x be the degree of node x. *Preferential attachment index* calculates the similarity between two nodes based on their degree:

$$S_{xy}^{\mathrm{PA}} = k_x \times k_y. \tag{37}$$

This measure is widely used in scale-free networks and is generalized by the Randic index [70, 71]. In particular, $s(G) = \sum_{x,y\in V} S_{xy}$ is called a scale-free metric and is used as a measure of scale-freeness of the graph [72]. A degree based metric is also used while generating a random scale-free network with the Barabasi–Albert model [73]. In this model, preferential attachment implies that each new added node in a generated graph is connected to other nodes with the probability proportionate to their degree.

Common neighbors (CN) similarity metric is based on the assumption that nodes that have a lot of CN are likely to be neighbors themselves. This assumption has had success in social networks [74, 75], however, as will be discussed in more detail later, it does not increase the symmetry of the network. *CN* is defined as:

$$S_{xy}^{\mathrm{CN}} = |\Gamma(x) \cap \Gamma(y)|. \tag{38}$$

We also use two differently normalized versions of the CN metric: *Salton index* normalized by the degree of the nodes

$$S_{xy}^{\mathrm{Salton}} = \frac{S_{xy}^{\mathrm{CN}}}{\sqrt{k_x \times k_y}} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{k_x \times k_y}} \tag{39}$$

and *Jaccard similarity* normalized by the total size of the node neighborhood:

$$S_{xy}^{\mathrm{Jaccard}} = \frac{S_{xy}^{\mathrm{CN}}}{|\Gamma(x) \cup \Gamma(y)|} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}. \tag{40}$$

An example of the similarity based on the global topology of the graph is *Katz index*. This index counts the number of walks (paths that can visit nodes and edges more than once) between considered nodes of increasing length weighted by the coefficient $\beta < 1$. $(A)_{xy}^{k}$ is the number of walks of length k between nodes x and y [76]. Katz index is defined as:

$$S_{xy}^{\mathrm{Katz}} = \beta A_{xy} + \beta^2 (A)_{xy}^{2} + \beta^3 (A)_{xy}^{3} + \ldots. \tag{41}$$

This series converges for β less than the inverse of the largest eigenvalue of A to the expression:

$$S_{xy}^{\mathrm{Katz}} = ((I - \beta A)^{-1} - I)_{xy}, \tag{42}$$

where I is an identity matrix.

To analyze the accuracy of each method, we used the traditional hypothesis testing performance metrics. The confusion matrix in table 3 introduces four basic metrics: true positive (TP), false negative (FN), false positive (FP) and true negative (TN) that compare edges identified by the manual 'ground truth' solution (true/false) with the edges obtained by a method (positive/negative). Other important metrics include precision, recall, miss rate, accuracy and F-measure that are calculated as functions of

J. Stat. Mech. (2022) 073403

Table 3. The confusion matrix identifies four variables: TP, TN, FP and FN depending on the positive or negative outcome of the predicted result and the ground truth. True and false corresponds to the ground truth and positive and negative corresponds to the prediction.

		Prediction by a method	
		Predicted	Not predicted
Truth	Predicted	True positive (TP). Correct prediction. Edge repaired by a manual repair and a method	False negative (FN). Type II error. Edge repaired by a manual repair, but not repaired by a method
Ground	Not predicted	False positive (FP). Type I error edge repaired by a method, but not repaired manually	True negative (TN). Correct rejection. Edge not repaired by either manual repair or a method

TP, FN, FP and TN as:

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}}, \\ \text{Miss Rate} &= \frac{\text{FN}}{\text{FN}+\text{TP}}, \\ \text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}, \\ F-\text{measure} &= \frac{\text{TP}}{\text{TP}+\frac{1}{2}(\text{FP}+\text{FN})}. \end{aligned} \tag{43}$$

In this paper, we use two approaches to choose the number of top edges to be repaired. First, as shown in table 4, we choose it to be equal to the number of edges repaired by the formulation with the coefficient in the objective equal to $c_{ij} = \frac{1}{d_i d_j}$ (6 in the forward circuit and 3 in the backward). Second, we use the process described in section 6.2, apply it to a given reconstruction method and choose the number of edges corresponding to the maximum NNTC of the obtained graphs. Table 5 shows results obtained by using this approach. Since the adjacency matrix is fairly sparse, the TN value is very high for all the methods. Therefore the best metric for the overall performance assessment is the F-measure that is proportional to the number of edges guessed correctly and inversely proportional to the sum of missed and falsely identified edges.

The formulation (PBCIP) with sum and multiplication objectives, $c_{ij} = \frac{1}{d_i+d_j}$ and $c_{ij} = \frac{1}{d_i d_j}$, respectively, outperform all other objectives and link prediction methods from the literature on both graphs. The next best method is the preferential attachment. This likely happened due to the fact that the repaired networks are hub-driven. That is, both networks have two hubs that are connected to most of the other nodes and to each other, therefore edges that are repaired by the preferential attachment are those that connect

J. Stat. Mech. (2022) 073403

Table 4. Performance of the different link prediction methods I.

Link prediction metric	# of colors	TP	FP	FN	TN	Precision	Recall	Miss rate	Accuracy	F-measure
Forward										
PBCIP ($c_{ij}=1$)	9	4	2	4	350	0.67	0.5	0.5	0.98	0.57
PBCIP ($c_{ij}=\frac{1}{\max(d_i,d_j)}$)	10	4	2	4	350	0.67	0.5	0.5	0.98	0.57
PBCIP ($c_{ij}=\frac{1}{d_i+d_j}$)	9	6	0	2	354	1	0.75	0.25	0.99	0.86
PBCIP ($c_{ij}=\frac{1}{d_i d_j}$)	9	6	0	2	354	1	0.75	0.25	0.99	0.86
Preferential attachment	16	5	1	3	352	0.83	0.63	0.37	0.99	0.71
Common neighbors	16	3	3	5	348	0.5	0.38	0.62	0.98	0.43
Salton index	18	0	6	8	342	0	0	1	0.96	0
Jaccard similarity	19	0	6	8	342	0	0	1	0.96	0
Katz index	16	0	6	8	342	0	0	1	0.96	0
Backward										
PBCIP ($c_{ij}=1$)	11–12	3	0	7	707	1	0.3	0.7	0.99	0.46
PBCIP ($c_{ij}=\frac{1}{\max(d_i,d_j)}$)	12	3	0	7	707	1	0.3	0.7	0.99	0.46
PBCIP ($c_{ij}=\frac{1}{d_i+d_j}$)	12	3	0	7	707	1	0.3	0.7	0.99	0.46
PBCIP ($c_{ij}=\frac{1}{d_i d_j}$)	12	3	0	7	707	1	0.3	0.7	0.99	0.46
Preferential attachment	17	1	2	9	703	0.33	0.1	0.9	0.98	0.15
Common neighbors	20	0	3	10	701	0	0	1	0.98	0
Salton index	21	0	3	10	701	0	0	1	0.98	0
Jaccard similarity	20	0	3	10	701	0	0	1	0.98	0
Katz index	18	0	3	10	701	0	0	1	0.98	0

hubs to nodes with the highest degree. In bigger networks with more than two hubs, preferential attachment is likely to connect pairs of unrelated nodes with high degrees.

Methods based on the different normalizations of the CN metric performed fairly poorly. These methods are not likely to uncover hidden symmetries because of their underlying assumption: nodes that have a lot of CN are likely to be neighbors themselves. The highest ranking edge repaired by the Salton index in the forward circuit is (DB06, VB11). This edge connects two nodes of the same color and breaks the symmetry between them and the rest of the nodes of that color. Hence, this repair made the network less symmetric rather than more symmetric. Consider now two random nodes of the same color. Recall, nodes of the same color have exactly the same number of neighbors (in an unweighted network) of all the other colors. Very often it means having a lot of the same neighbors. Therefore, measures that are based on the number of CN will often rank nodes of the same color as likely to have a connection, which is likely to break some symmetry in the network rather than restoring it. The possible alternative symmetry-restoring assumption may be: nodes that have a lot of CN are likely to

J. Stat. Mech. (2022) 073403

Table 5. Performance of the different link prediction methods II.

Link prediction metric	# of colors	TP	FP	FN	TN	Precision	Recall	Miss rate	Accuracy	F-measure
				Forward						
PBCIP ($c_{ij} = \frac{1}{d_i d_j}$)	9	6	0	2	354	1	0.75	0.25	0.99	0.86
Preferential attachment	17	8	40	0	316	0.17	1	0	0.89	0.29
Common neighbors	18	8	151	0	205	0.05	1	0	0.59	0.1
Salton index	17	8	176	0	180	0.04	1	0	0.52	0.08
Jaccard similarity	17	8	191	0	165	0.04	1	0	0.48	0.08
Katz index	17	6	101	2	253	0.06	0.75	0.25	0.72	0.1
				Backward						
PBCIP ($c_{ij} = \frac{1}{d_i d_j}$)	12	3	0	7	707	1	0.3	0.7	0.99	0.46
Preferential attachment	19	10	77	0	637	0.11	1	0	0.89	0.21
Common neighbors	20	1	84	9	621	0.01	0.1	0.9	0.87	0.02
Salton index	20	0	5	10	699	0	0	1	0.98	0
Jaccard similarity	19	0	5	10	699	0	0	1	0.98	0
Katz index	20	0	34	10	670	0	0	1	0.94	0

have more CN, which will keep the symmetry between the nodes with a lot of CN and increase the symmetry in their neighborhood.

Ultimately, we hope that PBCIP leads to a method that can discover and restore missing links in an input graph. To provide evidence PBCIP recovers missing links, we compare the performance of the formulation with traditional link reconstruction methods on a subset of our inputs suitably perturbed. The conventional approach to this problem is to take a network, remove a few edges at random and compare the average performance of a few different algorithms in restoring these edges by repeating this process multiple times. However, a comparison of PBCIP with traditional algorithms is difficult. As seen above, backward and forward circuits contain imperfect symmetries from the start and PBCIP is already restoring the minimum number of network links leading to the best possible symmetry relative to the objective. Thus, using PBCIP on an input generated by removing some edges from the original circuits is likely to restore 'extra' links in addition to the randomly removed links along with providing a principled reason, i.e. the minimum number of links according to our weighted objective, for the number of links restored. In contrast, traditional methods provide a 'ranking' of all potential edges and the exact approach to choosing the number of highest ranking edges in this problem is non-trivial. A detailed comparison would require more and larger instances than what we present in this paper. With these caveats, we present (modest) evidence that our method outperforms other methods in restoring missing links.

We focus on the backward circuit alone due to the simplicity in its interpretability. The analysis is performed as follows.

(a) Start with the original backward circuit.

(b) Remove s edges at random while making sure that the reduced graph is connected. Denote the removed edges $E_{\sim}$.

J. Stat. Mech. (2022) 073403

Table 6. We compare the performance of different link prediction methods including PBCIP in restoring the original backward circuit with s edges removed at random. The last row contains the method denoted 'random edges' that chooses repaired edges at random presenting a benchmark showing that other methods perform better than random. Note that this analysis is based on only 10 networks for each value of s.

	$s=1$	$s=2$	$s=3$
PBCIP ($c_{ij}=\frac{1}{d_i d_j}$)	50%	25%	33%
Preferential attachment	10%	10%	20%
Common neighbors	20%	0%	0%
Salton index	0%	0%	0%
Jaccard similarity	10%	0%	0%
Katz index	0%	5%	0%
Random edges	0%	0%	0%

(c) Apply PBCIP and obtain the repaired edges, denoted by E_{PBCIP}.

(d) Apply traditional link prediction methods choosing the top $|E_{\text{PBCIP}}|$ edges and denote the obtained edges E_i.

(e) Calculate what percentage of the original removed edges ($E_\sim$) are predicted by PBCIP (E_{PBCIP}) and other reconstruction methods (E_i).

(f) Find the average performance of each method by repeating steps (b)–(e) 10 times for each $s \in \{1,2,3\}$.

The summary of this analysis is presented in table 6. We observe that, as before, PBCIP considerably outperformed other methods for all s. We also see that, as before, preferential attachment performed better than other traditional methods, which is likely due to the hub-rich structure of the graph.

9. Conclusion

In summary, we described a method that allows a user to repair a graph to a more symmetric version and compared our results with the manual repair obtained in [11] as well as showing better performance than any of the traditional link prediction methods. We conclude with observations and ideas for future work.

(a) The pseudobalanced coloring obtained by solving (PBCIP) formulation sometimes is not minimal as the objective minimizes the weighted sum of links added. So if a pseudobalanced K-coloring had the same optimal objective as a pseudobalanced $(K+1)$-coloring, the pseudobalanced K-coloring could be returned for the $K+1$ case with one nontrivial node made trivial. This occurred five times and these solutions were not chosen. Moreover, our problem had two objectives: (1) find the pseudobalanced K-coloring with the most non-trivial nodes, and (2) minimize the

J. Stat. Mech. (2022) 073403

weighted sum of links added. Thus, our results can be interpreted as determining the best solution to use along the pareto optimal tradeoff curve between the number of colors, K, and the optimal weighted sum of added links. Under this interpretation, the aforementioned balanced $(K+1)$-coloring would not be on the pareto optimal curve as the K-coloring would be considered more optimal on both objectives.

(b) During preprocessing some of the nodes in the network are assigned fixed colors using constraints of the form equation (27). As a result, sets of nodes assigned different non-trivial colors in the original partitioning cannot be merged together. Therefore, repairs like the pink S_{12} group in figure 5(c) are not obtainable by solving (PBCIP) and need to be obtained as an additional step. This can be solved by implementing a two-step process as described in steps (b)(1) and (b)(2) of section 6.

(c) Removal of edges or reduction of their is not allowed due to the fact that pseudoedges in definition 3.5 are constrained to non-negative weights on non-edges. This choice is appropriate when studying neural networks of gap junctions in *C. elegans*. The reconstruction of this network is done using images of cross-sectional areas of the worm obtained using electron microscopy [77]. Each connection is followed from the neuron through all the layers leading to another neuron, therefore it is much more likely to miss a link than to add a non-existent one. When working on a different biological network, the possibility of the removal of edges or edge weight reduction may be required.

(d) The two graphs studied in this paper have a high degree of symmetry. We deduced that, for these graphs, the NNTC is a good indicator of the best solution and found the best solution using this index. However further investigation on a greater variety of graphs, including those with directed and weighted edges and those of bigger size, is required to make final conclusions. Biological networks present real dynamical systems that need to be able to maintain certain stable synchronization patterns, therefore indices characterizing stability and synchronizability of the cluster synchronization solution predicted by the balanced coloring are likely to give a better indication of ideal repairs, at least in biological networks, which are expected to produce stable cluster synchronization of their units [15].

(e) There are two potential ways of how this problem can be formulated: 1—find the minimal number of edges to add to find a graph with a given number of colors, 2—find the repair with minimal number of colors given the amount of edges that can be added. Both formulations make sense and both formulations would produce a number of graphs that will need to be analyzed in order to choose the best solution. The number of colors in a graph is no more than n, the number of nodes, therefore the 1st formulation will generate no more than n graphs. The number of edges in a graph is no more than n^2, therefore the 2nd formulation can generate up to n^2 graphs. In the interest of decreasing the search space, we chose the 1st formulation and the 2nd formulation was not explored.

(f) Solutions to optimization problems can often be improved by using expert insights about the object of optimization. For example, in section 8 we saw that the result

J. Stat. Mech. (2022) 073403

obtained with $c_{ij} = 1$ can be improved by using an assumption that edges between nodes with higher degrees are easier to repair by setting $c_{ij} = \frac{1}{d_i d_j}$. Additional insight based on the biological considerations can provide further improvements to the result.

(g) SBMs have important applications to network reconstruction methods [78–80]. In particular, Guimerà and Sales-Pardo [78] describe how the reliability of each potentially missing or spurious link can be calculated using graph generative models. As such, a generative model promotes certain topological features in the repaired graph, i.e. if certain features appear more often in generated graphs, the corresponding links are considered more reliable. We speculate that if a generative model for random equitable graphs (as described for example in [81]) is chosen with the proper set of parameters, the reconstruction will promote the higher degree of symmetry in the reconstructed network. Parameters of such model can be identified from the large scale analysis of the equitable partitions (balanced colorings) in networks performed in [14] using the fibrationSymmetries library (available at https://github.com/makselab/fibrationSymmetries). Additionally, one of the fundamental problems in network reconstruction via SBMs, and also in general, is the choice of the number of edges to be added/removed. We believe that indices derived from synchronizability and stability as discussed in section 6.2 can provide a novel approach to this problem.

(h) Our method can be applied to directed graphs by modifying the constraints equation (24) in (PBCIP) suitably to account for the version of the directed balanced coloring problem that is being solved. In particular, the edge set E must be modified so that only direct edges are used and the number of constraints reduced to reflect whether both directions of the balancing is required. Future work will include a formulation to repair directed networks and its comparison with the quasifibration formalism developed in [41].

(i) In this paper we applied our method to the unweighted graphs as a simplest case, however this method can be applied to weighted graphs without further modification by relaxing the integrality of the z_{ij} variables. This would change (PBCIP) from an integer linear program to a mixed-integer linear program.

(j) Our method is reasonably fast for small graphs. In order to be applied to graphs of bigger size, a more computationally efficient methods such as the Bender's decomposition method described in appendix A.2 needs to be implemented.

Overall, we presented a formulation of the ORP following the PBC of the graph. Our analysis shows encouraging results for the case of binary unweighted undirected networks as compared to a manually curated graph and other methods for missing link prediction. More work needs to be done to extend the formulation for large scale networks with weighted and directed edges, including negative weights which are important in all biological networks which contain inhibitory interactions.

J. Stat. Mech. (2022) 073403

Acknowledgments

We thank F Operti, C Smith, I Stewart, A Nazerian and G Verret for many helpful discussions. Funding was provided by NIBIB and NIMH through the NIH BRAIN Initiative Grant R01 EB028157 and NIH Grant 1R21EB028489. We would like to thank the UNM Center for Advanced Research Computing, supported in part by the National Science Foundation, for providing the high performance computing resources used in this work. We would also like to thank the referees for their careful reading of our manuscript and especially acknowledge their assistance in improving the complexity results section.

Data availability statement

All code and data are available at https://osf.io/prt5g/ and https://github.com/MakseLab/PseudoBalancedColoring.

Appendix A

A.1. Additional complexity comments

Here, we show that balanced coloring is hard when the balanced condition is only enforced between nodes of the same color versus the original version where the balanced condition is enforced between nodes of different colors as well. Such a restriction implies the directed-version of the balanced K-coloring problem is NP-Hard when only a subset of the constraints are enforced. *This further means the balanced coloring problem is easier when additional constraints are added as the directed-version of the balanced K-coloring problem is polynomial time solvable*. We note that a simpler version of this proof could be accomplished by adding directed in-edges to the star graph. Formally, we define this variant as follows.

Definition A.1. Let K be a given positive integer and $G = (V, E)$ a given directed graph. The in-directed self-balanced K-coloring problem is to determine whether there exists a partition, $\mathcal{C}$, of V which satisfies the following. For all $C \in \mathcal{C}$, and all pairs of distinct nodes $p, q \in C$,

$$\sum_{j \in C:(j,p)\in E} A_{jp} = \sum_{j \in C:(j,q)\in E} A_{jq}. \tag{44}$$

Also, $|\mathcal{C}| = K$.

Note how equation (44) is a subset of equation (8) restricted to the sum of edge weights within the color set of p and q. We show that finding such a coloring is NP-Hard. We show this by reducing *traditional vertex K-coloring* to restricted directed in-balanced K-coloring. Traditional vertex K coloring is the problem of coloring a graph

so that no two vertices of the same color share an edge. Formally, we can define it as follows.

Definition A.2. Let K be a given positive integer and $G = (V, E)$ a given undirected graph with edge weights w_e for $e \in E$. The traditional vertex K-coloring problem is to determine whether there exists a partition, $\mathcal{C}$, of V such that $|\mathcal{C}| = K$ and for all $C \in \mathcal{C}$, if $u, v \in C$ then $uv \notin E$.

For $K \geqslant 3$, traditional vertex K-coloring is NP-complete as shown by Karp [82]. By augmenting the input graph to K-vertex coloring with the appropriate subgraph, we can perform the reduction.

Theorem A.3. *Let $G = (V, E)$ and $K \geqslant 3$ be a given instance of the K-vertex coloring problem. The restricted directed in-balanced K-coloring problem is NP-Hard.*

Proof. Let $G' = (V \cup W, E' \cup F)$ be a directed graph where V is the set of nodes from the K-vertex coloring instance, E' is the set of directed edges representing edges in both direction for every edge in E, i.e.

$$E' = \{ij, ji : ij \in E\},$$

$W = \{v_1, \ldots, v_k\}$ are K additional nodes, and

$$W = \{v_i v_j : i > j\}$$

are a set of edges so that v_1 has $K - 1$ edges from nodes $v_2, \ldots, v_k$, v_2 has $K - 2$ edges, etc. First, note that each of node in W must be assigned a different color or else the balanced condition would be violated. This implies that no node can have any incoming edge from a node of the same color. Thus, the coloring on V must be a K-vertex coloring of the original graph. □

A.2. A Bender's decomposition approach

For the size instances we report on, the runtime to solve the integer linear program was acceptable. However, we suspect runtimes will scale poorly on larger instances. Because of this, we describe an adaptation of a Bender's decomposition approach due to Codato and Fischetti [83] that accelerate solution times for integer programs with big-M constraints such as equation (24) for future implementation. In this approach, we define

$$L_{\mathcal{V}} = \{\mathbf{s} = (x, y) \in \{0,1\}^{|V|^2} \times \{0,1\}^{|V| \times K} : (19),\ (20),\ (21),\ (26), \mathcal{V}\} \tag{45}$$

where $\mathcal{V}$ denotes a set of constraints of form equation (52) as defined below. The leader problem is to find a solution $\mathbf{s}^* = (x^*, y^*) \in L_{\mathcal{V}}$. Given such an $\mathbf{s}^*$, we calculate

$$EK^* = \{(i, j, k) : y^*_{jk} = 1, (i, j) \in E^{\mathrm{C}}\} \text{ and } V_2^* = \{(p, q) : x^*_{pq} = 1\}. \tag{46}$$

Note that EK^* represent the set of all non-edges that the solution $\mathbf{s}^*$ permit to exist and that V_2^* is the set of node pairs which must be balanced based on the solution $\mathbf{s}^*$.

J. Stat. Mech. (2022) 073403

Then, given an upper bound U^* on B^*, the follower problem constraints are as follows. We have an objective bound constraint.

$$\sum_{(i,j,k)\in EK^*} z_{ijk} \leqslant U^* - \delta. \tag{47}$$

The constraints in equation (24) are modified as $\mathbf{s}^*$ indicates which nodes must be balanced.

$$\sum_{j:(p,j,k)\in EK^*} z_{pjk} + \sum_{j\in V:pj\in E} A_{pj} = \sum_{j:(q,j,k)\in EK^*} z_{qjk} + \sum_{j\in V:qj\in E} A_{qj}, (p,q)\in V_2^*, \quad k\in K. \tag{48}$$

Finally, equation (25) are modified to only include the pseudoedges used.

$$\sum_{k:(p,q,k)\in EK^*} z_{pqc} = \sum_{k:(q,p,k)\in EK^*} z_{qpk}, pq\in E'. \tag{49}$$

Then, we define

$$\mathcal{F}_{\mathbf{s}^*} = \{z\in\{0,1\}^{|V|^2\times K} : (47),\ (48),\ (49)\} \tag{50}$$

and the follower problem is to find a solution in $\mathcal{F}_{\mathbf{s}^*}$. If $\mathcal{F}_{\mathbf{s}^*}$ is empty then an *irreducible inconsistent subsystem (IIS)* is found [84]. An IIS is a subset of the constraints of $\mathcal{F}_{\mathbf{s}^*}$ such that

- The system defined by the subset of constraints are infeasible; and
- Removing any one of the constraints results in a feasible system.

Note that the constraints in the IIS are indexed by a union of a subset of $V_2^*\times K$ and a subset of E'. Using the IIS found, let VK_2^I and E_I' denote these subsets, respectively.

Note that each of the constraints in the IIS corresponds to several binary variables in equation (45). First, consider an arbitrary element $(p,q,c)\in VK_2^I$. In this case, the binary variable $x_{pq}^* = 1$ must be true for the equality constraint to be a part of the formulation. In addition, the constraint could be violated some z_{pjc} and z_{qjc} are not in the formulation as $y_{jc} = 0$, i.e. $(p,j,c)\notin EK^*$ or $(q,j,c)\notin EK^*$ and $pj\in E'$ or $qj\in E'$. Let $VK_C^* = \{(j,c) : y_{jc} = 0\}$. Then, the indices of the y_{jc} set to zero that could potentially cause the infeasiblity are described by the set $\{(j,c)\in VK_C^* : pj\in E' \text{ or } qj\in E'\}$.

Now consider an arbitrary $pq\in E_I'$. In this case, the cause of the violation must be due to one or more z_{pjc} or z_{qjc} being held to zero. So, again, the set of indices of y_{jc} that could cause the infeasibility is $\{(j,c)\in VK_C^* : pj\in E' \text{ or } qj\in E'\}$.

Indices of the binary variables that could be causing the infeasibility are then

$$\begin{aligned} I_x &= \left\{(p,q) : \exists c\in K, (p,q,c)\in VK_2^I\right\} \\ I_y &= \{(j,c)\in VK_C^* : (pj\in E' \text{ or } qj\in E') \text{ and } (\exists c\in K, (p,q,c) \\ &\quad \times\ \in VK_2^I \text{ or } pq\in E_I')\}. \end{aligned} \tag{51}$$

J. Stat. Mech. (2022) 073403

Algorithm 1. Bender's decomposition of (PBCIP).

Result: An optimal solution, $(\mathbf{s}^*, z^*) = (x^*, y^*, z^*)$, to (PBCIP)

Initialize $\mathcal{V}$ to the empty set;

Use Eq. (45) to form $L_{\mathcal{V}}$;

Set U^* to an upperbound, e.g., the maximum possible edge weight times $|V|^2$;

Initialize z^* to zero.;

while $L_{\mathcal{V}} \neq \emptyset$ **do**

  Find a feasible solution $\mathbf{s}^* \in L_{\mathcal{V}}$;

  Use $\mathbf{s}^*$ to find EK^* and V_2^* via Eq. (46);

  Use Eq. (50) to form $\mathcal{F}_{\mathbf{s}^*}$;

  if $\mathcal{F}_{\mathbf{s}^*} \neq \emptyset$ **then**

    Solve Eq. (53) and update U^* and z^*;

  else

    Find an IIS, calculate I_x and I_y via Eq. (51);

    Use I_x and I_y to add a valid inequality to $\mathcal{V}$ via Eq. (52);

  end

end

return $(\mathbf{s}^*, z^*) = (x^*, y^*, z^*)$

The valid inequality that should be added to equation (45) is

$$\sum_{(j,c)\in I_y} (1 - y_{jc}) + \sum_{(p,q)\in I_x} x_{pq} \leqslant |I_y| + |I_x| - 1. \tag{52}$$

Note that this inequality may not be the strongest possible as the cause of the infeasibility in any particular constraint could be from one or more z_{pqc} variables or, in the case of the constraints in VK_2^I, from the setting that x_{pq} is one (which causes the existence of the constraint). Thus, we must have a systematic way of adding the z_{pqc} variables back to equation (50).

If equation (50) is feasible, an improved upperbound can be found by solving the following optimization version of the follower problem.

$$U^* = \min \left\{ \sum_{ij\notin E, c\in K} z_{ijc} : (48), (49) \right\} \tag{53}$$

Denote the optimal solution to equation (53) by z^*. The objective value U^* is used to update the upper bound in used in equations (47) and (50) is then resolved.

The precise description of the method is as follows. We note that δ has not been specified and is dependent on the weights of the edges in the input graph. For an unweighted graph, we can use $\delta = 1$ (algorithm 1).

References

[1] Alon U 2019 *An Introduction to Systems Biology: Design Principles of Biological Circuits* (Boca Raton, FL: CRC Press)
[2] Buchanan M, Caldarelli G, De Los Rios P, Rao F and Vendruscolo M 2010 *Networks in Cell Biology* (Cambridge: Cambridge University Press)
[3] Klipp E, Liebermeister W, Wierling C and Kowald A 2016 *Systems Biology: A Textbook* (New York: Wiley)
[4] Stewart I 2004 *Nature* **427** 601
[5] Varshney L R, Chen B L, Paniagua E, Hall D H and Chklovskii D B 2011 *PLoS Comput. Biol.* **7** e1001066
[6] Bock D D *et al* 2011 *Nature* **471** 177
[7] Guelzim N, Bottani S, Bourgine P and Képès F 2002 *Nat. Genet.* **31** 60
[8] Liu Z-P, Wu C, Miao H and Wu H 2015 *Database* (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4589691/)
[9] Métris A *et al* 2017 *NPJ Syst. Biol. Appl.* **3** 1
[10] Santos G M d M, Aguiar C M L and Mello M A R 2010 *Apidologie* **41** 466
[11] Morone F and Makse H A 2019 *Nat. Commun.* **10** 1
[12] McKay B D 1981 *Congr. Numer.* **30** 45
[13] White J G, Southgate E, Thomson J N and Brenner S 1986 *Phil. Trans. R. Soc. London* B **314** 427
[14] Morone F, Leifer I and Makse H A 2020 *Proc. Natl Acad. Sci. USA* **117** 8306
[15] Leifer I, Morone F, Reis S D S, Andrade J S Jr, Sigman M and Makse H A 2020 *PLoS Comput. Biol.* **16** e1007776
[16] Belykh I and Hasler M 2011 *Chaos* **21** 016106
[17] Golubitsky M and Stewart I 2006 *Bull. Am. Math. Soc.* **43** 305
[18] Kamei H and Cock P J A 2013 *SIAM J. Appl. Dyn. Syst.* **12** 352
[19] Monteiro H S, Leifer I, Reis S D, Andrade J S Jr and Makse H A 2022 Fast algorithm to identify cluster synchrony through fibration symmetries in large information-processing networks *Chaos* **32** 033210
[20] DeVille L and Lerman E 2013 arXiv:1303.3907
[21] Pecora L M, Sorrentino F, Hagerstrom A M, Murphy T E and Roy R 2014 *Nat. Commun.* **5** 1
[22] Nijholt E, Rink B and Sanders J 2016 *J. Differ. Equ.* **261** 4861
[23] Aguiar M A, Dias A P and Ruan H 2021 *Physica* D **429** 133065
[24] Leifer I, Sánchez-Pérez M, Ishida C and Makse H A 2021 *BMC Bioinform.* **22** 363
[25] Strogatz S H 2018 *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering* (Boca Raton, FL: CRC Press)
[26] Arenas A, Díaz-Guilera A, Kurths J, Moreno Y and Zhou C 2008 *Phys. Rep.* **469** 93
[27] Rodrigues F A, Peron T K D, Ji P and Kurths J 2016 *Phys. Rep.* **610** 1
[28] Pikovsky A *et al* 2001 *Synchronization: A Universal Concept in Nonlinear Sciences* (Cambridge: Cambridge University Press)
[29] Gupta S, Campa A and Ruffo S 2014 *J. Stat. Mech.* R08001
[30] Monod J 1970 *Proc. 11th Nobel Symp.* (Södergarn, Lidingö, Sweden August 1968) ed A Engström and B Strandberg (New York: Wiley) Stockholm: Almqvist and Wiksell p 436 1969
[31] Kato S, Kaplan H S, Schrödel T, Skora S, Lindsay T H, Yemini E, Lockery S and Zimmer M 2015 *Cell* **163** 656
[32] Lu L and Zhou T 2011 *Physica* A **390** 1150
[33] Getoor L and Diehl C P 2005 *SIGKDD Explor. Newsl.* **7** 3
[34] Liben-Nowell D and Kleinberg J 2007 *J. Am. Soc. Inf. Sci.* **58** 1019
[35] Chen H, Li X and Huang Z 2005 *Proc. 5th ACM/IEEE-CS Joint Conf. Digital Libraries (JCDL'05)* (Piscataway, NJ: IEEE) pp 141–2
[36] Clauset A, Moore C and Newman M E J 2008 *Nature* **453** 98
[37] Harary F 1969 *Graph Theory* (Boca Raton, FL: CRC Press)
[38] Dixon J D and Mortimer B 1996 *Permutation Groups* (Berlin: Springer)
[39] Liu Y 2020 arXiv:2012.05129

J. Stat. Mech. (2022) 073403

[40] Kudose S 2009 *Acta Math. Sin.* **1** 1–9
[41] Boldi P, Leifer I and Makse H A 2021 Quasifibrations of graphs to find symmetries in biological networks (arXiv:2111.06999)
[42] Gurobi Optimization 2021 LLC, Gurobi optimizer reference manual (https://www.gurobi.com/)
[43] McKay B 1976 Backtrack programming and the graph isomorphism problem *Master's Thesis* University of Melbourne, Department of Mathematics, Melbourne, Australia
[44] McKay B D and Piperno A 2014 *J. Symb. Comput.* **60** 94
[45] Feige U and Kogan S 2010 *J. Graph Theory* **64** 277
[46] Stewart I 2007 *Mathematical Proceedings of the Cambridge Philosophical Society* vol 143 (Cambridge: Cambridge University Press) pp 165–83
[47] McKay B D and Piperno A 2021 Practical graph isomorphism. The individualization-refinement method (https://pallini.di.uniroma1.it/Introduction.html) (Accessed: 28 September 2021)
[48] Garey M R and Johnson D S 1978 *J. Assoc. Comput. Mach.* **25** 499
[49] Darga P T, Liffiton M H, Sakallah K A and Markov I L 2004 *Proc. 41st Annual Design Automation Conf.* pp 530–4
[50] Junttila T and Kaski P 2007 *2007 Proc. 9th Workshop on Algorithm Engineering and Experiments (ALENEX)* (Philadelphia, PA: SIAM) pp 135–49
[51] Piperno A 2008 arXiv:0804.4881
[52] López-Presa J L and Fernández Anta A 2009 *Int. Symp. Experimental Algorithms* (Berlin: Springer) pp 221–32
[53] Grohe M, Kersting K, Mladenov M, Schweitzer P, Van den Broeck G, Kersting K and Natarajan S 2017 *Color Refinement and Its Applications* (Boston, MA: MIT Press) (https://doi.org/10.7551/mitpress/10548.003.0023)
[54] Holland P W, Laskey K B and Leinhardt S 1983 *Soc. Netw.* **5** 109
[55] Schaub M T and Peel L 2020 arXiv:2009.07196
[56] Sherali H D and Smith J C 2001 *Manage. Sci.* **47** 1396
[57] Gambuzza L V, Frasca M, Sorrentino F, Pecora L M and Boccaletti S 2020 *IEEE Trans. Netw. Sci. Eng.* **8** 282
[58] Purcell O, Savery N J, Grierson C S and Di Bernardo M 2010 *J. R. Soc. Interface.* **7** 1503
[59] Belykh I, De Lange E and Hasler M 2005 *Phys. Rev. Lett.* **94** 188101
[60] Fiedler M 1973 *Czechoslovak Math. J.* **23** 298
[61] Chung F R K 1997 *Spectral Graph Theory* (*CBMS Regional Conference Series in Mathematics*) vol 92 (Providence, RI: American Mathematical Society)
[62] Ghosh A and Boyd S 2006 *Linear Alg. Appl.* **418** 693
[63] Wu C W and Chua L O 1995 *IEEE Trans. Circuits Syst.* I **42** 494
[64] Zhou C, Motter A E and Kurths J 2006 *Phys. Rev. Lett.* **96** 034101
[65] Nazerian A, Panahi S, Leifer I, Phillips D, Makse H A and Sorrentino F 2022 *Chaos* **32** 041101
[66] MacArthur B D, Sánchez-García R J and Anderson J W 2008 *Discrete Appl. Math.* **156** 3525
[67] McKay B D 2007 (Computer Science Dept, Australian National University) p 225
[68] Meurer A *et al* 2017 *Peer J. Comput. Sci.* **3** e103
[69] The Sage Developers 2021 SageMath, the sage mathematics software system (version 9.3) (https://sagemath.org)
[70] Randic M 1975 *J. Am. Chem. Soc.* **97** 6609
[71] Kincaid R K and Phillips D J 2011 *WIREs Comput. Stat.* **3** 557
[72] Li L, Alderson D, Doyle J C and Willinger W 2005 *Int. Math.* **2** 431
[73] Albert R and Barabási A-L 2002 *Rev. Mod. Phys.* **74** 47
[74] Newman M E 2001 *Phys. Rev.* E **64** 025102
[75] Kossinets G 2006 *Soc. Netw.* **28** 247
[76] Newman M 2018 *Networks* (Oxford: Oxford University Press)
[77] Durbin R M 1987 Studies on the development and organisation of the nervous system of caenorhabditis elegans (https://us-east-2-02850030-inspect.menlosecurity.com/safeview-fileserv/tc_download/0fa2141cdc3c8ceca8d31e6747a0862f25d8b28e1f0553b42a53fbff3310ff94/?&cid=NE0F0AE126DEA_&rid=c869aa57a2336b00c5d544d1c213e5da&file_url=https%3A%2F%2Fwww.wormatlas.org%2FDurbin%2Fdurbinthesis.pdf&type=original)
[78] Guimerà R and Sales-Pardo M 2009 *Proc. Natl Acad. Sci. USA* **106** 22073
[79] Peixoto T P 2018 *Phys. Rev.* X **8** 041011
[80] Ghasemian A, Hosseinmardi H, Galstyan A, Airoldi E M and Clauset A 2020 *Proc. Natl Acad. Sci. USA* **117** 23393
[81] Newman M and Martin T 2014 *Phys. Rev.* E **90** 052824

J. Stat. Mech. (2022) 073403

[82] Karp R M 1972 *Complexity of Computer Computations* (New York: Plenum) pp 85–103
[83] Codato G and Fischetti M 2006 *Oper. Res.* **54** 756
[84] Van Loon J N M 1981 *Eur. J. Oper. Res.* **8** 283

J. Stat. Mech. (2022) 073403