

# Synchrony and Elementary Operations on Coupled Cell Networks

M.A.D. Aguiar<sup>1</sup>, A.P.S. Dias<sup>2</sup> and H. Ruan<sup>3</sup>

<sup>1</sup> Centro de Matemática da Universidade do Porto\*,

Rua do Campo Alegre, 687, 4169-007 Porto, Portugal

Faculdade de Economia, Universidade do Porto,

Rua Dr Roberto Frias, 4200-464 Porto, Portugal

<sup>2</sup> Departamento de Matemática, Centro de Matemática, Universidade do Porto,

Rua do Campo Alegre, 687, 4169-007 Porto, Portugal

<sup>3</sup> Fachbereich Mathematik, Universität Hamburg,

Bundesstraße 55, 20146 Hamburg, Germany

E-mail: maguiar@fep.up.pt apdias@fc.up.pt haibo.ruan@math.uni-hamburg.de

July 9, 2013

## Abstract

On a finite graph (network), let every node (cell) represent an individual dynamics given by a system of ordinary differential equations, and every arrow (edge) encode the dynamical influence of the tail node to the head node. We then have defined a coupled cell system that is associated with the given network structure. Subspaces that are defined by equalities of cell coordinates and left invariant under every coupled cell system respecting the given network structure are called synchrony subspaces. These are completely determined by the network structure and form a complete lattice under set-inclusions. We analyze the transition of the lattice of synchrony subspaces of a network that is caused by structural changes in the network topology, such as deletion or/and addition of cells or/and edges, or/and rewirings of edges. We give necessary and sufficient conditions under which lattice elements persist or disappear. Our analysis is both algebraic and algorithmic.

AMS classification scheme numbers: 34C15 37C10 05C76 06B23

Keywords: Coupled cell network; graph operation; coupled cell system; synchrony subspace; lattice.

---

\*Research funded by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT - Fundação para a Ciência e a Tecnologia under the project PEst-C/MAT/UI0144/2011 and PTDC/MAT/100055/2008.

# 1 Introduction

In the theory of coupled cell networks developed by Stewart, Golubitsky and their co-workers in [18, 11, 10] or Field in [9], a *network* is a finite set of nodes (or cells) linked together by a finite number of arrows and dynamical systems that are consistent with this network structure are called the admissible *coupled cell systems*. More precisely, every cell of the network represents an individual dynamics given by a finite set of ordinary differential equations and each arrow describes the interaction between the two connected individuals. In analogue to other structures of dynamical systems such as symmetry or the Hamiltonian, network structure imposes strong restrictions on the dynamics of the associated coupled cell systems. One striking example of that is the existence of *synchrony subspaces*, which are spaces defined by equalities of some cell coordinates that are flow-invariant for all coupled cell systems associated with the given network structure. See, for example, the paper by Aguiar *et al.* [1] where coupled cell systems supporting heteroclinic behavior are analyzed and part of the crucial elements guaranteeing that kind of dynamics is the existence of synchrony subspaces for which the restricted systems have the desired dynamic properties. Those restricted systems are also coupled cell systems associated with smaller networks, called the *quotient networks*.

Synchrony subspaces can be determined solely by the underlying network structure. More precisely, Stewart, Golubitsky and their co-workers proved in [18, 11] that synchrony subspaces of coupled cell systems associated with a network structure are in one-to-one correspondence to certain equivalence relations defined on the set of cells of the network satisfying a combinatorial property. These are called *balanced equivalence relations* (cf. Definition 2.5). More specifically, denote by  $\approx$  an equivalence relation on the cell set of the network and by  $\Delta_{\approx}$  the (polydiagonal) subspace given by equalities of cell coordinates among  $\approx$ -equivalent cells. Then, it is shown in [18, 11] that a polydiagonal  $\Delta_{\approx}$  is a synchrony subspace if and only if it is left invariant under the network adjacency matrices (one for each edge type), which happens if and only if  $\approx$  is balanced.

By Stewart [17], the set of all synchrony subspaces of a network forms a complete lattice under the set-inclusion (see also Aldis [5]). Equivalently, the set of all balanced equivalence relations on the set of cells of a network  $\mathcal{G}$ , denoted by  $\Lambda_{\mathcal{G}}$ , is a complete lattice under the refinement of equivalence relations. Aguiar and Dias [2] showed how to obtain the lattice of synchrony subspaces for a general network and presented an algorithm that generates this lattice. Indeed, it is proved that this problem can be reduced to finding the lattice of synchrony subspaces for *regular networks*, that is, networks with only one cell type and one arrow type, and such that every cell receives the same number of input arrows. For a regular network, the lattice of synchrony subspaces can be obtained using the eigenvalue structure of the network adjacency matrix. This approach was motivated by the work of Kamei [13] on the class of regular networks where the adjacency matrix has only simple eigenvalues. See also Kamei and Cock [14] for a computer algorithm searching for all possible balanced equivalence relations using symbolic matrix computations.

In the realm of networks in nature and science, it is a common knowledge that

networks having different network topology support different patterns of dynamic behavior. A vivid example is the gene regulatory networks, which, under different conditions, exhibit different regulation patterns accompanied by different transcriptional network topologies (cf. Zhang *et al.* [19] and Luscombe *et al.* [16]). As one of the most investigated network-specific dynamic, network synchronizability shows to vary as the network structure varies (cf. Atay and Biyikoglu [6], Lu *et al.* [15] and Chen and Duan [7]). Furthermore, Hagberg and Schult [12] discussed how to engineer a diffusively coupled network using elementary edge operations to enhance the network synchronization.

As mentioned earlier, in the context of coupled cell systems, the connecting topology of a network dictates the lattice of balanced equivalence relations (or equivalently, the lattice of synchrony subspaces). Thus, it is natural and of interest to ask how these lattices evolve as the underlying topology of the network changes. In this perspective, Aguiar and Ruan [4] considered non-product binary operations on networks such as the join and the coalescence; Aguiar and Dias [3] addressed the same question for product operations on networks.

In this paper we evaluate the impact of structural changes in network topology on the lattice of balanced equivalence relations of the network, through elementary graph operations such as addition and/or deletion of cells and/or edges, and/or rewiring of edges. As one can expect, the lattice of balanced equivalence relations changes in general: some relations persist, some disappear and some emerge. In particular, we describe necessary and sufficient conditions for balanced equivalence relations to persist or disappear.

Let  $\mathcal{G}_1$  be a coupled cell network and  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by an elementary graph operation on edges or nodes. Since the sets of the adjacency matrices of  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are in general, not identical, they may not generate the same linear space. Thus, the coupled cell systems associated with  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are generally not ODE-equivalent (cf. Dias and Stewart [8]). In particular, some or all synchrony subspaces can be different for  $\mathcal{G}_1$  and  $\mathcal{G}_2$ .

More specifically, let  $\sim_{I_1}$  denote the *input equivalence relation* on the set of cells in  $\mathcal{G}_1$ , where two cells are said to be  $\sim_{I_1}$ -equivalent if and only if their input sets are isomorphic and admit a bijection preserving edge types. Here the *input set of a cell* is the multiset of the tail cells of its input edges (cf. Definition 2.2). In Subsection 3.1 we consider removing or adding an edge, whose head cell we denote by  $c_o$ . We show that a balanced relation  $\bowtie$  on  $\mathcal{G}_1$  is again balanced on  $\mathcal{G}_2$  if and only if  $c_o$  is not  $\bowtie$ -equivalent with any other cell (cf. Lemma 3.3) and vice versa if one interchange  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . It follows that a sufficient condition for  $\Lambda_{\mathcal{G}_2} \subseteq \Lambda_{\mathcal{G}_1}$  is that  $[c_o]_{\sim_{I_2}}$  contains only one element in  $\mathcal{G}_2$ . A special case is when  $\mathcal{G}_1$  is a *homogeneous network* (network with only one input type of cell), then we have  $\Lambda_{\mathcal{G}_2} \subseteq \Lambda_{\mathcal{G}_1}$ , where  $\Lambda_{\mathcal{G}_2}$  consists of those balanced relations  $\bowtie$  in  $\Lambda_{\mathcal{G}_1}$  such that  $[c_o]_{\bowtie} = \{c_o\}$ .

The rewiring case is addressed in Subsection 3.2. Here, by a *rewiring* of a network, we mean a graph operation on edges of the network under which the input equivalence relation is preserved. A rewiring of a network can be viewed as a composition of several *elementary rewirings*, where an elementary rewiring takes place in the input set of only one cell by replacing one input edge with another one of the same type. Our

goal of this subsection is then to recover  $\Lambda_{\mathcal{G}_2}$  from  $\Lambda_{\mathcal{G}_1}$ , where  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by an elementary rewiring. In principle, one can view an elementary rewiring as first removing and then adding an edge, thus apply the result from Subsection 3.1 consecutively. However, as we will see, it is more advantageous to consider the rewiring as a one-step operation, since it is, in contrast to the deletion or addition of an edge, an operation preserving the input equivalence relation.

It should be mentioned that some partial results on rewiring have been obtained by Field in [9], where he considered invariants of a network under repatching and explored the patch equivalence for balanced equivalence relations. More precisely, given a network  $\mathcal{G}_1$  and an equivalence relation  $\bowtie$  on its cell set  $C$ , a network  $\mathcal{G}_2$  is called a *repatching* of  $\mathcal{G}_1$  if the number of edges, per edge-type, from cells in  $[c]_{\bowtie}$  to every cell in  $[d]_{\bowtie}$  is the same on both  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , for all  $c, d \in C$ . The networks  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are called *patch  $\bowtie$ -equivalent* according to the terminology in Field [9]. It follows that for patch  $\bowtie$ -equivalent networks  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , the relation  $\bowtie$  is balanced on  $\mathcal{G}_2$  if and only if it is balanced on  $\mathcal{G}_1$  (cf. Field [9]).

In Subsection 3.2, we characterize those balanced equivalence relations from  $\Lambda_{\mathcal{G}_2}$  that are inherited from  $\Lambda_{\mathcal{G}_1}$  (cf. Lemma 3.10). In particular, we extend the results of Field [9] mentioned above. That is, suppose the rewiring operation replaces an input edge to  $c_o$  from  $d$  with one from  $a$ , then a balanced equivalence relation  $\bowtie$  on  $\mathcal{G}_1$  is again balanced on  $\mathcal{G}_2$  if and only if  $a \bowtie d$  or  $[c_o]_{\bowtie} = \{c_o\}$ .

In Section 4 we consider graph operations on nodes and start with the case when a node, together with all its edges, is removed. We show that  $\Lambda_{\mathcal{G}_2}$  of the resulting network can be completely recovered from  $\Lambda_{\mathcal{G}_1}$  of the initial network, if the input equivalence relation of  $\mathcal{G}_2$  refines the (projected) input equivalence relation of  $\mathcal{G}_1$ . In this case,  $\Lambda_{\mathcal{G}_2}$  is given by the projected balanced relations  $\bowtie$  on  $\mathcal{G}_1$  such that  $[c_o]_{\bowtie} = \{c_o\}$  for  $c_o$  being the removed node (cf. Propositions 4.10 and 4.14). As special cases, when there are no outgoing edges from  $c_o$  to other cells, or when  $\mathcal{G}_1$  is a homogeneous network, then  $\Lambda_{\mathcal{G}_2}$  is completely recoverable from  $\Lambda_{\mathcal{G}_1}$  (cf. Corollary 4.16).

The paper is organized as follows. In Section 2, we introduce notations and give a compact presentation about coupled cell networks, coupled cell systems, synchrony subspaces and balanced equivalence relations on networks. Our main results appear in Sections 3-4. All the results are accompanied with examples and algorithms. In all the algorithms, we refer to Algorithm A.1, which is an adaptation of Algorithm 6.3 in [2] (cf. Appendix A).

In Subsection 3.1 we analyze the effect of removing or adding an edge upon the lattice of balanced equivalence relations on the networks. The main results are Lemma 3.3 and Corollary 3.6. In Subsection 3.2 we consider elementary rewirings. The main result is given by Lemma 3.10. In Subsection 3.3, we present an algorithm which generates the new lattice  $\Lambda_{\mathcal{G}_2}$  based on the initial lattice  $\Lambda_{\mathcal{G}_1}$  and our theoretical results, in case of removing, adding or rewiring an edge (for networks having only one cell type and one edge type) (cf. Algorithm 3.15). We end the Section 3 with some computational examples in Subsection 3.4 and extending the results to removing and adding multiple edges including rewirings in Subsection 3.5.

In Section 4 we investigate the effect of operations on nodes. As for the case of

edge deletion or addition, there are balanced relations that can be recovered from the initial lattice (cf. Propositions 4.10, 4.14 and Corollary 4.16) and there can be new ones that emerge (cf. Proposition 4.17). Finally, we extend our results to the addition of a node in Subsection 4.4 and then to the addition and/or deletion of multiple nodes in Subsection 4.5.

## 2 Preliminaries

In this section we introduce notations and give a brief review on coupled cell networks, coupled cell systems, synchronous subspaces and balanced equivalence relations on networks. We follow the framework of Stewart, Golubitsky *et al.* [18, 11], where more details can be found. See also Golubitsky and Stewart [10] for a survey on the subject.

### 2.1 Coupled cell networks and coupled cell systems

A coupled cell network is a directed graph whose nodes represent the cells and edges describe couplings among the cells, where equivalence relations on nodes and edges are indicated by different shapes of nodes and edges in the graph. More precisely:

**Definition 2.1** A *coupled cell network* consists of a finite nonempty set  $C$  of *nodes* or *cells* and a finite nonempty set  $\mathcal{E} = \{(d, c) : d, c \in C\}$  of *edges* or *arrows* and two equivalence relations:  $\sim_C$  on  $C$  and  $\sim_E$  on  $\mathcal{E}$  such that the *consistency condition* is satisfied: if  $(d_1, c_1) \sim_E (d_2, c_2)$ , then  $d_1 \sim_C d_2$  and  $c_1 \sim_C c_2$ . We write  $\mathcal{G} = (C, \mathcal{E}, \sim_C, \sim_E)$ .  $\diamond$

Note that a coupled cell network may have multiple edges and loops.

By a *multiset*, we mean a generalized notion of set, in which elements are allowed to appear more than once. For a multiset  $A$  and  $x \in A$ , define the *multiplicity* of  $x$  as the number of copies of  $x$  contained in  $A$ , denoted by  $m(x, A)$ ; for a subset  $B \subset A$ , define the *multiplicity* of  $B$  as  $m(B, A) := \sum_{x \in B} m(x, A)$ .

In the theory of coupled cell networks, the concept of *input sets* plays a key role.

**Definition 2.2** Let  $c, d \in C$  be such that  $e := (d, c) \in \mathcal{E}$ . Then, the cell  $d$  is called the *tail cell* and  $c$  is called the *head cell* of  $e$ . The edge  $e$  is called an *input edge* of  $c$ . The set of all tail cells of input edges of  $c$ , which is a multiset, is called the *input set* of  $c$ , usually denoted by  $I(c)$ . For an edge type  $e$  of  $\mathcal{G}$ , denote by  $I^e(c) \subset I(c)$  the multiset of the tail cells of input edges of  $c$  that are of type  $e$ . Two cells  $c_1, c_2 \in C$  are called *input-equivalent*, denoted by  $c_1 \sim_I c_2$ , if  $\#I^e(c_1) = \#I^e(c_2)$ , for all edge-type  $e$ , where  $\#I^e(c_i)$  denotes the cardinality of the multiset  $I^e(c_i)$ ,  $i = 1, 2$ .  $\diamond$

It follows from the consistency condition that the input equivalence relation  $\sim_I$  refines the cell equivalence relation  $\sim_C$ .

The coupling structure of a coupled cell network with edge types  $e_1, e_2, \dots, e_s$ , can be represented by *adjacency matrices*  $A_1, A_2, \dots, A_s$ , one for each edge type, where  $A_l := (a_{ij}^{(l)})$  for  $a_{ij}^{(l)} = m(c_j, I^{e_l}(c_i))$ ,  $l \in \{1, \dots, s\}$ , where  $c_i$  denotes the  $i$ -th cell of the network.

**Definition 2.3** A coupled cell network is called *homogeneous*, if it has only one input-equivalence class. A *regular* network is a homogeneous network with only one edge-equivalence class.  $\diamond$

In a homogeneous network, all cells are of identical type and receive the same number of input edges per edge type. The number, which is the cardinality of the input set, is called the *valency* of the network.

**Definition 2.4** Given a coupled cell network  $\mathcal{G} = (C, \mathcal{E}, \sim_C, \sim_E)$ , associate to every cell  $c \in C$  a finite-dimensional real vector space  $P_c$ , called the *cell phase space*, such that  $\sim_C$ -equivalent cells have identical phase spaces. A *coupled cell system* is a system of ordinary differential equations whose structure is consistent with the network structure of  $\mathcal{G}$  and whose *total phase space*  $P$  is defined by the direct product of all the cell phase spaces. More precisely, if  $x = (x_c)_{c \in C}$  denotes the coordinate system on  $P$ , then it is of form  $\dot{x} = F(x)$ , where the system associated with cell  $j$  has the form

$$\dot{x}_j = f_j(x_j; x_{i_1}, \dots, x_{i_m}).$$

Here, the first argument  $x_j$  in  $f_j$  represents the internal dynamics of the  $j$ -th cell and each of the remaining variables  $x_{i_p}$  indicates an input edge from the  $i_p$ -th cell to the  $j$ -th cell. Input edges of the same type directed to the  $j$ -th cell correspond to the invariance of  $f_j$  under permutation of the corresponding variables. Systems associated with input equivalent cells are given by the same function  $f_j$ , but the permutation invariance under variables is determined by the input types of the cells. Vector fields  $F$  satisfying the above properties are called  *$\mathcal{G}$ -admissible*.  $\diamond$

As a special case, if  $\mathcal{G}$  is a homogeneous network, then we have  $f_j \equiv f$  for all cells. Furthermore, if  $\mathcal{G}$  is a regular network with valency  $v$ , then the coupled cell system has the form

$$\dot{x}_j = f(x_j; \overline{x_{i_1}, \dots, x_{i_v}}),$$

where the overbar in  $f$  indicates that  $f$  is invariant under any permutation of the cell coordinates  $x_{i_1}, \dots, x_{i_v}$ .

## 2.2 Synchrony subspaces and balanced equivalence relations

The concept of balanced equivalence relations on coupled cell networks is closely related to synchrony subspaces admitted by admissible coupled cell systems. The following definition follows [11].

**Definition 2.5** Given a network  $\mathcal{G} = (C, \mathcal{E}, \sim_C, \sim_E)$ , an equivalence relation  $\bowtie$  defined on  $C$  is called *balanced* if for every  $c, d \in C$  with  $c \bowtie d$ , there exists an isomorphism between their input sets  $I(c)$  and  $I(d)$ ,  $\beta : I(c) \rightarrow I(d)$ , preserving the equivalence relations  $\sim_E$  and  $\bowtie$ : for all  $i \in I(c)$  we have  $i \bowtie \beta(i)$  and  $(i, c) \sim_E (\beta(i), d)$ .  $\diamond$

**Remark 2.6** It follows from the definition that any balanced equivalence relation refines the input equivalence relation  $\sim_I$ .  $\diamond$

Using the multisets and multiplicity, we have the following equivalent definition as Definition 2.5 (cf. [4]).

**Definition 2.7** Given a network  $\mathcal{G} = (C, \mathcal{E}, \sim_C, \sim_E)$ , an equivalence relation  $\bowtie$  defined on  $C$  is called *balanced*, if for  $c, d \in C$  with  $c \bowtie d$ , we have

$$m([\alpha]_{\bowtie}, I^\epsilon(c)) = m([\alpha]_{\bowtie}, I^\epsilon(d)), \quad \forall \alpha \in C \quad (2.1)$$

for every edge type  $\epsilon$  in  $\mathcal{E}$ . ◇

To avoid lengthy notations, we omit the superscript “ $\epsilon$ ” in (2.1) by writing

$$m([\alpha]_{\bowtie}, I(c)) = m([\alpha]_{\bowtie}, I(d)), \quad \forall \alpha \in C.$$

**Definition 2.8** Let  $\mathcal{G} = (C, \mathcal{E}, \sim_C, \sim_E)$  be a network together with a choice of the total phase space  $P$ . Let  $\bowtie$  be an equivalence relation defined on  $C$  such that it refines  $\sim_C$ . Define the *polydiagonal subspace* associated with  $\bowtie$  by

$$\Delta_{\bowtie} = \{ \mathbf{x} \in P : x_c = x_d \text{ whenever } c \bowtie d, \forall c, d \in C \} .$$

The polydiagonal subspace  $\Delta_{\bowtie}$  of  $P$  is called a *synchrony subspace* if it is flow-invariant for all  $\mathcal{G}$ -admissible vector fields on  $P$ . ◇

The concept of synchrony subspace and that of balanced equivalence relation are related in the following way (cf. [11]).

**Theorem 2.9** (cf. [11]) *Given a coupled cell network  $\mathcal{G}$ , an equivalence relation  $\bowtie$  on the cell set  $C$  and a choice  $P$  of the total phase space, the polydiagonal subspace  $\Delta_{\bowtie}$  is a synchrony subspace if and only if  $\bowtie$  is balanced.*

Indeed, as stated in Corollary 2.10 of [2], it follows from Theorem 2.9 that for a coupled cell network  $\mathcal{G}$  and an equivalence relation  $\bowtie$  defined on its cell set  $C$ , the polydiagonal subspace  $\Delta_{\bowtie}$  is a synchrony subspace for any choice of the total phase space  $P$  if and only if it is flow-invariant for *all linear* admissible vector fields for all the cell phase spaces chosen to be  $\mathbf{R}$ .

For homogeneous networks, if every cell phase space is chosen to be  $\mathbf{R}$ , then the set of all linear admissible vector fields is generated by the linear maps given by the adjacency matrices (one for each edge type) together with the identity map on  $\mathbf{R}^n$ . It follows then that a polydiagonal subspace  $\Delta_{\bowtie}$  is a synchrony subspace for any choice of the total phase space  $P$  if and only if it is left invariant by all the adjacency matrices. A more general result is formulated in Theorem 4.2 of [2]: a polydiagonal subspace is a synchrony subspace for a general network if and only if it is left invariant by all the adjacency matrices (one for each edge type).

Given a network  $\mathcal{G}$  with the set  $C$  of cells, we denote by  $M_{\mathcal{G}}$  the set of all equivalence relations on  $C$  and by  $\Lambda_{\mathcal{G}}$  the set of all balanced equivalence relations on  $C$ . Both sets have a partially ordered structure, using the *relation of refinement*  $<$  defined as: for two equivalence relations  $\bowtie_i$  and  $\bowtie_j$ , we say  $\bowtie_i$  *refines*  $\bowtie_j$ , written as  $\bowtie_i < \bowtie_j$ , if

$$[c]_i \subseteq [c]_j \quad \forall c \in C.$$

Here,  $[c]_l$  denotes the  $\bowtie_l$ -equivalence class of cell  $c$ , for  $l = i, j$ .

As pointed out in [17], both sets of balanced equivalence relations and synchrony subspaces for a given network are complete lattices, with respect to the relation of refinement and the inclusion of subspaces, respectively. See [2, Section 3] for a compact discussion on this topic.

**Example 2.10** Figure 1 shows two examples of 4-cell networks. The network on the left has two edge types and one cell type. Moreover, cells 1, 2 are input isomorphic and cells 3, 4 are input isomorphic. The network on the right is regular of valency 2, since it has only one cell type and one edge type and all the cells are input isomorphic, all receiving 2 edges.

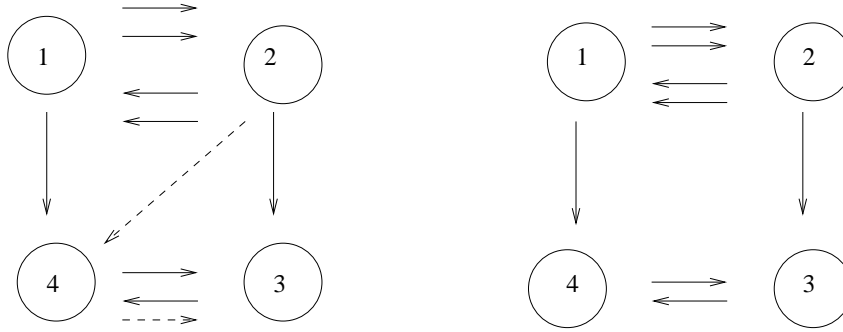


Figure 1: Two examples of 4-cell networks, where the network on the right is regular of valency 2.

The coupled cell systems associated to the network on the left of Figure 1 satisfy

$$\begin{aligned}\dot{x}_1 &= f(x_1, \overline{x_2, x_2}) \\ \dot{x}_2 &= f(x_2, \overline{x_1, x_1}) \\ \dot{x}_3 &= g(x_3, \overline{x_2, x_4, x_4}) \\ \dot{x}_4 &= g(x_4, \overline{x_1, x_3, x_2})\end{aligned}$$

where  $f : (\mathbf{R}^k)^3 \rightarrow \mathbf{R}^k$  and  $g : (\mathbf{R}^k)^4 \rightarrow \mathbf{R}^k$  are smooth and invariant under permutation of the two cell coordinates under the bars. Note that there are two types of cell systems,  $f$  and  $g$ , due to the fact that there are two input classes of cells.

The coupled cell systems associated to the network on the right of Figure 1 satisfy

$$\begin{aligned}\dot{x}_1 &= f(x_1, \overline{x_2, x_2}) \\ \dot{x}_2 &= f(x_2, \overline{x_1, x_1}) \\ \dot{x}_3 &= f(x_3, \overline{x_2, x_4}) \\ \dot{x}_4 &= f(x_4, \overline{x_1, x_3})\end{aligned}\tag{2.2}$$

where  $f : (\mathbf{R}^k)^3 \rightarrow \mathbf{R}^k$  is smooth and invariant under permutation of the last two cell coordinates. Here, the network is regular, so any coupled cell system with structure consistent with this network structure is determined by only one type of cell systems.



For the network on the left of Figure 1, the only nontrivial balanced equivalence relation on the set of cells  $C = \{1, 2, 3, 4\}$  is  $\{\{1, 2\}, \{3\}, \{4\}\}$ . For the 4-cell regular network on the right of Figure 1, we have three nontrivial balanced equivalence relations on  $C$ :

$$\begin{aligned}\bowtie_1 &= \{\{1, 2\}, \{3\}, \{4\}\} \\ \bowtie_2 &= \{\{1, 2\}, \{3, 4\}\} \quad . \\ \bowtie_3 &= \{\{1, 3\}, \{2, 4\}\}\end{aligned}$$

It follows then that the polydiagonals  $\Delta_{\bowtie_1}, \Delta_{\bowtie_2}, \Delta_{\bowtie_3}$  are synchrony subspaces. That is, they are flow-invariant under any coupled cell system of the type described in (2.2). For example, for  $\bowtie_2$ , we have

$$\Delta_{\bowtie_2} = \{\mathbf{x} \in (\mathbf{R}^k)^4 : x_1 = x_2, x_3 = x_4\}.$$

The equations (2.2) restricted to  $\Delta_{\bowtie_2}$  satisfy

$$\begin{aligned}\dot{x}_1 &= f(x_1, \overline{x_1, x_1}) \\ \dot{x}_3 &= f(x_3, \overline{x_1, x_3}) \quad .\end{aligned}$$

Note that we can interpret these equations as coupled cell systems associated with the 2-cell network of Figure 2. This network is called the *quotient* network of the network on the right of Figure 1 by  $\bowtie_2$ .

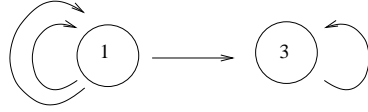


Figure 2: The 2-cell quotient network of the network on the right of Figure 1 by the balanced equivalence relation with classes  $\{1, 2\}$  and  $\{3, 4\}$ .

◇

Theorem 5.2 of [11] shows that for any synchrony subspace  $\Delta_{\bowtie}$  there is always a network, called the *quotient network*, such that the restrictions of admissible vector fields to  $\Delta_{\bowtie}$  are the admissible vector fields of the quotient network. More precisely, given a network  $\mathcal{G}$  and a balanced equivalence relation  $\bowtie$  on its cells, the quotient network  $\mathcal{G}/\bowtie$  is defined naturally as follows: the set of cells of  $\mathcal{G}/\bowtie$  is formed by one cell of each  $\bowtie$ -equivalence class and the edges in the quotient network are the projection of edges in the original network. Moreover, the cell and the edge relations are preserved. Specifically, cells in the quotient  $\mathcal{G}/\bowtie$  representing two distinct equivalence classes, say  $[c]_{\bowtie}$  and  $[c']_{\bowtie}$ , are cell-equivalent if and only if cells  $c$  and  $c'$  are cell-equivalent in  $\mathcal{G}$ . Moreover, given a cell in the quotient, representing the cells in a  $\bowtie$ -class  $[c]_{\bowtie}$ , the number of edges of type  $e$  directed from a cell representing the cells in a  $\bowtie$ -class  $[d]_{\bowtie}$  to that cell is  $m([d]_{\bowtie}, I^e(c))$ : in notation, we have the input edge  $\bar{e} = ([d]_{\bowtie}, [c]_{\bowtie})$  with multiplicity  $m([d]_{\bowtie}, I^e(c))$ .

### 3 Graph operations on edges

In this section, we consider graph operations on edges of a coupled cell network including deletion, addition and rewiring of edges. Our aim is to recover the lattice of balanced equivalence relations of the resulting network from that of the initial one.

We start by considering the case of deleting or adding a single edge in Subsection 3.1, where we discuss primarily the deletion case and leave the addition case as a complete analog (cf. Remark 3.8). In Subsection 3.2, we analyze the case of rewiring an edge. In Subsection 3.3 we present an algorithm that can handle the cases of deletion, addition or rewiring of an edge simultaneously. We give several computational examples in Subsection 3.4. We close the section by extending our results to the cases of deleting, adding and/or rewiring multiple edges in Subsection 3.5.

In what follows, let  $\mathcal{G}_1$  denote an initial coupled cell network and  $\mathcal{G}_2$  the resulting network from an edge operation. The input equivalence relation on  $\mathcal{G}_i$  is denoted by  $\sim_{I_i}$  and the input set on  $\mathcal{G}_i$  is denoted by  $I_i(\cdot)$ , for  $i = 1, 2$ .

**Definition 3.1** A balanced equivalence relation  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}$  is called *recoverable from*  $\Lambda_{\mathcal{G}_1}$ , if  $\bowtie_2 \in \Lambda_{\mathcal{G}_1}$ .  $\diamond$

In what follows, we denote by  $\Lambda_{\mathcal{G}_2}^R$  the set of all balanced equivalence relations in  $\Lambda_{\mathcal{G}_2}$  that are recoverable from  $\Lambda_{\mathcal{G}_1}$ . Then, we have  $\Lambda_{\mathcal{G}_2}^R = \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$ .

#### 3.1 Deletion and addition of an edge

Let  $\mathcal{G}_1 = (C, \mathcal{E}_1, \sim_C, \sim_E)$  be a coupled cell network and  $e_o = (d_o, c_o) \in \mathcal{E}_1$  be an edge directing from  $d_o$  to  $c_o$ . Let  $\mathcal{G}_2$  be the coupled cell network obtained from  $\mathcal{G}_1$  by removing  $e_o$ , then

$$\mathcal{G}_2 = (C, \mathcal{E}_1 \setminus \{e_o\}, \sim_C, \sim_E) := (C, \mathcal{E}, \sim_C, \sim_E). \quad (3.3)$$

Thus, the input sets of the two networks satisfy

$$I_2(x) = \begin{cases} I_1(x), & \text{if } x \neq c_o \\ I_1(x) \setminus \{d_o\}, & \text{if } x = c_o, \end{cases} \quad (3.4)$$

for  $x \in C$ . Here, notice that:

**Remark 3.2**  $[c_o]_{\sim_{I_1}} \cap [c_o]_{\sim_{I_2}} = \{c_o\}$ , for input equivalence relations  $\sim_{I_1}$  and  $\sim_{I_2}$ .  $\diamond$

**Lemma 3.3** Let  $\bowtie$  be an equivalence relation on  $C$ . Then, we have:

(i) If  $\#[c_o]_{\bowtie} = 1$ , then  $\bowtie$  is balanced on  $\mathcal{G}_1$  if and only if it is balanced on  $\mathcal{G}_2$ , i.e.,

$$\bowtie \in \Lambda_{\mathcal{G}_1} \Leftrightarrow \bowtie \in \Lambda_{\mathcal{G}_2}.$$

(ii) If  $\#[c_o]_{\bowtie} > 1$ , then  $\bowtie$  cannot be balanced on both  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , i.e.,  $\bowtie \notin \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$ .

**Proof** (i) By definition, an equivalence relation  $\bowtie$  is balanced on  $\mathcal{G}_2$  if and only if for all  $x, y \in C$  such that  $x \bowtie y$ , the following condition holds:

$$m([\alpha]_{\bowtie}, I_2(x)) = m([\alpha]_{\bowtie}, I_2(y))$$

for all  $\alpha \in C$ . Now, let  $x, y \in C$  such that  $x \bowtie y$  and  $x \neq y$ . As  $\#[c_o]_{\bowtie} = 1$ , it follows that  $x \neq c_o$  and  $y \neq c_o$ . By (3.4), the input sets of  $x$  and  $y$  remain the same under the deletion of  $e_o$  and so

$$m([\alpha]_{\bowtie}, I_2(x)) = m([\alpha]_{\bowtie}, I_1(x)) = m([\alpha]_{\bowtie}, I_1(y)) = m([\alpha]_{\bowtie}, I_2(y))$$

holds for all  $\alpha \in C$ . Thus  $\bowtie$  is balanced on  $\mathcal{G}_1$  if and only if it is balanced on  $\mathcal{G}_2$ .

(ii) Since every balanced equivalence relation of a network must refine the input equivalence relation on the network, the statement follows from Remark 3.2.  $\square$

An alternative proof of Lemma 3.3 can be given from an algebraic viewpoint. We explain it in the following remark.

**Remark 3.4** As mentioned in Subsection 2.2, see the discussion following Theorem 2.9, a synchrony subspace of an  $n$ -cell coupled cell network is precisely a polydiagonal subspace that is left invariant under all the adjacency matrices, one for each edge type. If  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by removing an edge  $e_o = (d_o, c_o)$ , then  $\mathcal{G}_2$  and  $\mathcal{G}_1$  have the same adjacency matrices for every edge type, apart from the type corresponding to the removed edge  $e_o$ . Denote by  $A_{\mathcal{G}_1}$  and  $A_{\mathcal{G}_2}$  the adjacency matrix for the edge type of  $e_o$  on  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , respectively. Then the relation between the lattices of synchrony subspaces of  $\mathcal{G}_1$  and  $\mathcal{G}_2$  is determined by the polydiagonals that are left invariant by both or just one of the adjacency matrices  $A_{\mathcal{G}_1}$  and  $A_{\mathcal{G}_2}$ . Let  $N = [n_{ij}]$  be the  $n \times n$  matrix such that  $n_{c_o, d_o} = -1$  and zero elsewhere. It follows that

$$A_{\mathcal{G}_2} = A_{\mathcal{G}_1} + N.$$

In the case  $d_o \neq c_o$ , the matrix  $N$  is a nilpotent matrix where the zero eigenvalue has algebraic multiplicity  $n$  and geometric multiplicity  $(n - 1)$  with eigenspace  $E_0 = \{\mathbf{x} : x_{d_o} = 0\}$ . Otherwise, if  $d_o = c_o$ , then  $N$  is a semi-simple matrix with eigenvalues 0 and  $-1$ , whose eigenspaces are given by  $E_0 = \{\mathbf{x} : x_{c_o} = 0\}$  and  $E_{-1} = \{\mathbf{x} : x_j = 0, \forall j \neq c_o\}$ . In both cases, a polydiagonal subspace is left invariant under  $N$  if and only if it does not include coordinates equalities of type  $x_{c_o} = x_j$  for some  $j \neq c_o$  in its definition. This fact coincides with the result of Lemma 3.3. More precisely, if a polydiagonal subspace does not include any equality of type  $x_{c_o} = x_j$ , then it is invariant under  $A_{\mathcal{G}_2}$  if and only if it is invariant under  $A_{\mathcal{G}_1}$ . If it does include some equality of that type, then it is not invariant under  $N$ , and so it is not simultaneously invariant under  $A_{\mathcal{G}_2}, A_{\mathcal{G}_1}$ .  $\diamond$

**Remark 3.5** Under the hypotheses of Lemma 3.3 (i), the quotient network  $\mathcal{G}_2 / \bowtie$  is obtained from  $\mathcal{G}_1 / \bowtie$  by removing the edge  $\bar{e}_o = ([d_o]_{\bowtie}, [c_o]_{\bowtie})$ , where  $[c_o]_{\bowtie} = \{c_o\}$ .  $\diamond$

By Definition 3.1 and Lemma 3.3, we have

$$\Lambda_{\mathcal{G}_2}^R = \{\bowtie \in \Lambda_{\mathcal{G}_1} : \#[c_o]_{\bowtie} = 1\}. \quad (3.5)$$

**Corollary 3.6** *The following holds:*

(i) *If  $\#[c_o]_{\sim_{I_2}} = \#[c_o]_{\sim_{I_1}} = 1$  then  $\Lambda_{\mathcal{G}_2} = \Lambda_{\mathcal{G}_2}^R = \Lambda_{\mathcal{G}_1}$ .*

(ii) *If  $\#[c_o]_{\sim_{I_2}} = 1$  and  $\#[c_o]_{\sim_{I_1}} > 1$  then  $\Lambda_{\mathcal{G}_2} = \Lambda_{\mathcal{G}_2}^R$ . In particular, this is the case when  $\mathcal{G}_1$  is a homogeneous network.*

(iii) *If  $\#[c_o]_{\sim_{I_2}} > 1$  and  $\bowtie \in \Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R$  then  $\#[c_o]_{\bowtie} > 1$ .*

**Proof** Observe that every balanced equivalence relation of a network must refine the input equivalence relation on the network. The statements then follow from Lemma 3.3.  $\square$

Thus, the problem of obtaining  $\Lambda_{\mathcal{G}_2}$  reduces to finding those  $\bowtie \in \Lambda_{\mathcal{G}_2}$  for which  $\#[c_o]_{\bowtie} > 1$ .

**Proposition 3.7** *Let  $\mathcal{G}_2$  be a network obtained from  $\mathcal{G}_1$  by removing an edge  $e_o = (d_o, c_o)$ . If*

$$\sim_{I_2} < \sim_{I_1}$$

*then, every balanced relation in  $\Lambda_{\mathcal{G}_2}$  is recoverable from  $\Lambda_{\mathcal{G}_1}$ . That is,*

$$\Lambda_{\mathcal{G}_2} = \Lambda_{\mathcal{G}_2}^R.$$

**Proof** If  $\sim_{I_2} < \sim_{I_1}$ , as  $[c_o]_{\sim_{I_2}} \cap [c_o]_{\sim_{I_1}} = \{c_o\}$ , we have  $\#[c_o]_{\sim_{I_2}} = 1$ . The result then follows from Corollary 3.6(i)(ii).  $\square$

**Remark 3.8** (i) The result of this subsection is completely applicable to the case of adding an edge. In particular, Lemma 3.3 is a symmetric statement with respect to  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , thus it still holds if  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by adding an input edge to  $c_o$ . As we will see in Subsection 3.3, the algorithm can be adapted similarly for both cases.

(ii) It should be mentioned that the case where an edge changes its edge type (for networks with more than one edge type) can be similarly treated, since it can be interpreted as a composition of first deleting an edge of one type and then adding an edge of another type. Note that this changes the relation of input equivalence among cells. Again, the balanced relations that are preserved under that edge operation are the ones where the head cell of the edge that changed its type forms a single class.  $\diamond$

## 3.2 Rewiring of an edge

Let  $\mathcal{G}_1$  be a coupled cell network and  $c_o, a, d$  be cells of  $\mathcal{G}_1$ . Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by the elementary rewiring where the edge  $(d, c_o)$  is replaced by the edge  $(a, c_o)$ , denoted by

$$\mathcal{G}_1 \xrightarrow{(c_o; d, a)} \mathcal{G}_2. \quad (3.6)$$

This operation can also be interpreted as first deleting the input edge  $(d, c_o)$  of  $c_o$  from  $d$  and then adding another input edge  $(a, c_o)$  from  $a$  of the same type. Alternatively,  $\mathcal{G}_1$  can be viewed as the network obtained from  $\mathcal{G}_2$  by the elementary rewiring where the edge  $(a, c_o)$  is replaced by the edge  $(d, c_o)$ , i.e.,

$$\mathcal{G}_2 \xrightarrow{(c_o; a, d)} \mathcal{G}_1.$$

The relation between the input sets of  $\mathcal{G}_1$  and  $\mathcal{G}_2$  is given by

$$I_2(x) = \begin{cases} I_1(x), & \text{if } x \neq c_o \\ (I_1(x) \setminus \{d\}) \cup \{a\}, & \text{if } x = c_o, \end{cases} \quad (3.7)$$

for  $x \in C$ .

**Remark 3.9** Note that  $[c_o]_{\sim_{I_1}} = [c_o]_{\sim_{I_2}}$ , in contrast to the deletion or addition of an edge (cf. Remark 3.2). In particular,  $\mathcal{G}_2$  is a regular (resp. homogeneous) network if and only if  $\mathcal{G}_1$  is a regular (resp. homogeneous) network.  $\diamond$

In the same spirit as Lemma 3.3, the following holds for the rewiring operation.

**Lemma 3.10** *Let  $\bowtie$  be an equivalence relation on  $C$  and  $\mathcal{G}_1, \mathcal{G}_2$  be related by (3.6). Then, we have:*

(i) *If  $d \in [a]_{\bowtie}$ , then  $\bowtie$  is balanced on  $\mathcal{G}_1$  if and only if it is balanced on  $\mathcal{G}_2$ , i.e.,*

$$\bowtie \in \Lambda_{\mathcal{G}_1} \Leftrightarrow \bowtie \in \Lambda_{\mathcal{G}_2}. \quad (3.8)$$

(ii) *If  $d \notin [a]_{\bowtie}$  and  $\#[c_o]_{\bowtie} = 1$ , then (3.8) holds.*

(iii) *If  $d \notin [a]_{\bowtie}$  and  $\#[c_o]_{\bowtie} > 1$ , then  $\bowtie \notin \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$ .*

**Proof** Let  $x \in C$ . If  $x \neq c_o$ , then  $I_1(x) = I_2(x)$  and consequently,

$$m([a]_{\bowtie}, I_1(x)) = m([a]_{\bowtie}, I_2(x)), \quad \forall a \in C.$$

Otherwise, if  $x = c_o$ , then  $I_2(x) = (I_1(x) \setminus \{d\}) \cup \{a\}$ .

(i) Since  $a \bowtie d$ , we have  $I_1(x)$  and  $I_2(x)$  are equal up to the  $\bowtie$ -equivalence, i.e.

$$m([a]_{\bowtie}, I_1(x)) = m([a]_{\bowtie}, I_2(x)), \quad \forall a \in C.$$

Thus,  $\bowtie \in \Lambda_{\mathcal{G}_1}$  if and only if  $\bowtie \in \Lambda_{\mathcal{G}_2}$ .

(ii) Note that  $\mathcal{G}_2$  can be viewed as a network obtained from  $\mathcal{G}_1$  by first deleting an input edge to  $c_o$  and then adding another input edge to  $c_o$ . Thus, (3.8) follows from Lemma 3.3 and Remark 3.8, due to the condition  $\#[c_o]_{\bowtie} = 1$ .

(iii) Assume to the contrary that  $\bowtie \in \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$ . Since  $\#[c_o]_{\bowtie} > 1$ , we take  $x \in C$  such that  $c_o \bowtie x$  and  $c_o \neq x$ . It follows from  $\bowtie \in \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$  that

$$m([\alpha]_{\bowtie}, I_1(c_o)) = m([\alpha]_{\bowtie}, I_1(x)), \quad \forall \alpha \in C$$

and

$$m([\alpha]_{\bowtie}, I_2(c_o)) = m([\alpha]_{\bowtie}, I_2(x)), \quad \forall \alpha \in C.$$

Combined with  $I_1(x) = I_2(x)$ , we have

$$m([\alpha]_{\bowtie}, I_1(c_o)) = m([\alpha]_{\bowtie}, I_2(c_o)), \quad \forall \alpha \in C. \quad (3.9)$$

On the other hand, since  $d \notin [a]_{\bowtie}$ , we have  $m([d]_{\bowtie}, I_2(c_o)) = m([d]_{\bowtie}, I_1(c_o)) - 1$ , which is a contradiction to (3.9). □

**Remark 3.11** The statement in Lemma 3.10 (i) agrees with Remark 14 (2) in Field [9]. ◇

**Remark 3.12** Under the hypothesis of Lemma 3.10, it follows from the definition of quotient network that for  $\bowtie$  balanced on  $\mathcal{G}_i$  with  $i \in \{1, 2\}$ , we have:

(i) If  $d \in [a]_{\bowtie}$  then  $\mathcal{G}_1 / \bowtie = \mathcal{G}_2 / \bowtie$ .

(ii) If  $d \notin [a]_{\bowtie}$  and  $\#[c_o]_{\bowtie} = 1$  then  $\mathcal{G}_2 / \bowtie$  is obtained from  $\mathcal{G}_1 / \bowtie$  by rewiring the input edge  $([d]_{\bowtie}, [c_o]_{\bowtie})$  to  $([a]_{\bowtie}, [c_o]_{\bowtie})$  of  $[c_o]_{\bowtie}$ . Thus, in general,  $\mathcal{G}_1 / \bowtie \neq \mathcal{G}_2 / \bowtie$ . ◇

It follows from Lemma 3.10 that

$$\Lambda_{\mathcal{G}_2}^R = \{\bowtie \in \Lambda_{\mathcal{G}_1} : (d \in [a]_{\bowtie}) \vee (\#[c_o]_{\bowtie} = 1)\}, \quad (3.10)$$

which is a parallel of (3.5). Then,

$$\Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R = \{\bowtie \in \Lambda_{\mathcal{G}_2} : d \notin [a]_{\bowtie} \wedge \#[c_o]_{\bowtie} > 1\}. \quad (3.11)$$

In fact, Corollary 3.6(i) and (iii) remain valid for  $\Lambda_{\mathcal{G}_2}^R, \Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R$  defined by (3.10)-(3.11). Thus, the problem of recovering  $\Lambda_{\mathcal{G}_2}$  then reduces to recovering those  $\bowtie \in \Lambda_{\mathcal{G}_2}$  for which  $d \notin [a]_{\bowtie}$  and  $\#[c_o]_{\bowtie} > 1$ .

### 3.3 Algorithm for edge operations

In this subsection we present an algorithm that generates the lattice of balanced equivalence relations of the new network  $\mathcal{G}_2$  based on that of the initial network  $\mathcal{G}_1$ , where  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by either removing, adding or rewiring an edge.

Without loss of generality, we can assume  $\mathcal{G}_1$  has only one cell type and one edge type, since as shown in [2], the calculation of the lattice of synchrony subspaces for a general coupled cell network reduces to this particular kind of networks.

The notations used here follow Sections 3.1 and 3.2. Recall that  $M_{\mathcal{G}_2}$  stands for the set of all equivalence relations on  $\mathcal{C}$ . We denote by

$$N = \{\bowtie \in M_{\mathcal{G}_2} : \bowtie < \sim_{I_2} \wedge \#[c_0]_{\bowtie} = 2 \wedge \#[x]_{\bowtie} = 1, \forall x \notin [c_0]_{\bowtie}\},$$

and for every  $\bowtie \in N$ , let

$$M_{\bowtie} = \{\bowtie \in \Lambda_{\mathcal{G}_2} : \bowtie < \bowtie\}, \quad \text{in case of edge deletion or addition,} \quad (3.12)$$

$$M_{\bowtie} = \{\bowtie \in \Lambda_{\mathcal{G}_2} : \bowtie < \bowtie \wedge d \notin [c]_{\bowtie}\}, \quad \text{in case of edge rewiring.} \quad (3.13)$$

Then, we have the following result.

**Proposition 3.13** *Let  $\mathcal{G}_2$  be a network obtained from  $\mathcal{G}_1$  by deletion, addition or rewiring of an edge. Let  $M_{\bowtie}$  be defined by (3.12) in case of deletion or addition of an edge, and by (3.13) in case of rewiring an edge. The set of balanced relations in  $\Lambda_{\mathcal{G}_2}$  that are not recoverable from  $\Lambda_{\mathcal{G}_1}$  is given by*

$$\Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R = \bigcup_{\bowtie \in N} M_{\bowtie}.$$

**Proof** It follows from the definition of  $\Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R$  and  $M_{\bowtie}$ .  $\square$

Consequently, the problem of recovering  $\Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R$  reduces to recovering  $M_{\bowtie}$  for every  $\bowtie \in N$ . Let  $\bowtie \in N$ . Then, the only non-trivial  $\bowtie$ -equivalence class is  $[c_0]_{\bowtie} = \{c_0, c\}$  for some  $c \neq c_0$  such that  $c \sim_{I_2} c_0$ , and the corresponding polydiagonal is given by

$$P = \{\mathbf{x} : x_{c_0} = x_c\}. \quad (3.14)$$

Using Algorithm A.1, which is an adaptation of Algorithm 6.3 in [2] to our setting, one can find the set of all synchrony subspaces contained in  $P$ .

More precisely, let  $A$  be the adjacency matrix of the  $n$ -cell network  $\mathcal{G}_2$ . Let  $\lambda_i$  for  $i = 1, \dots, t$  with  $t \leq n$ , be the eigenvalues of  $A$ , whose algebraic and geometric multiplicities are denoted by  $m_i^a$  and  $m_i^g$ , respectively. Let  $G_{\lambda_i}$  be the generalized eigenspaces of  $A$  for  $i = 1, \dots, t$ . Then a subspace  $S$  is invariant under  $A$  if and only if  $S$  can be written as a direct sum

$$S = (G_{\lambda_1} \cap S) \oplus \dots \oplus (G_{\lambda_t} \cap S).$$

For every polydiagonal  $P$  of the form (3.14), the Algorithm 3.15 starts by constructing the largest  $A$ -invariant subspace of  $\mathbb{R}^n$  contained in  $P$  and then uses the Algorithm A.1, which is an adaptation of Algorithm 6.3 from [2], to find all synchrony subspaces contained in that invariant subspace.

To express the largest  $A$ -invariant subspace of  $\mathbb{R}^n$  contained in  $P$ , define recursively the following subspaces of  $G_{\lambda_i}$  for every  $i \in \{1, 2, \dots, t\}$  by

$$\begin{cases} J_{\lambda_i}^1 &= (A - \lambda_i \text{Id}_n)^{-1}(0) \cap P, \\ J_{\lambda_i}^k &= (A - \lambda_i \text{Id}_n)^{-1}(J_{\lambda_i}^{k-1}) \cap P, \quad \text{for } 2 \leq k \leq p_i, \end{cases} \quad (3.15)$$

where  $p_i \geq 1$  is the minimal integer such that  $\ker(A - \lambda_i \text{Id}_n)^j = \ker(A - \lambda_i \text{Id}_n)^{p_i}$ , for all  $j > p_i$ . Note that

$$J_{\lambda_i}^k \subseteq \ker(A - \lambda_i \text{Id}_n)^k$$

and

$$J_{\lambda_i}^{k-1} \subseteq J_{\lambda_i}^k,$$

for  $2 \leq k \leq p_i$ . By definition of  $J_{\lambda_i}^k$ , we have the following observation.

**Proposition 3.14** *Let  $A$  be an adjacency matrix of an  $n$ -cell network and  $P$  be the polydiagonal subspace given by (3.14). For  $J_{\lambda_i}^k$  defined by (3.15) with  $i = 1, \dots, t$  and  $k = 1, \dots, p_i$ , we have:*

(i) *The largest  $A$ -invariant subspace of  $\mathbb{R}^n$  contained in the polydiagonal  $P$  is given by*

$$S = J_{\lambda_1}^{p_1} \oplus \dots \oplus J_{\lambda_t}^{p_t}.$$

(ii) *Any synchrony subspace  $\Delta$  of the network that contains equality  $x_{c_0} = x_c$  in its definition, is a polydiagonal subspace of  $S$  that is left invariant under  $A$ . We also call such  $\Delta$  a synchrony subspace of  $A$  restricted to  $S$ .*

**Algorithm 3.15** Let  $\mathcal{G}_1$  be an  $n$ -cell network having only one cell type and one edge type. Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by either removing, adding or rewiring an edge. Denote by  $A$  the  $n \times n$  adjacency matrix of the network  $\mathcal{G}_2$  whose eigenvalues  $\lambda_i$  with  $i = 1, \dots, t$  have algebraic and geometric multiplicities  $m_i^a$  and  $m_i^g$ , respectively.

- 1 In the deletion or addition case, let  $\Lambda_{\mathcal{G}_2} := \{\bowtie \in \Lambda_{\mathcal{G}_1} : \#[c_0]_{\bowtie} = 1\}$ . Otherwise, in the rewiring case, let  $\Lambda_{\mathcal{G}_2} := \{\bowtie \in \Lambda_{\mathcal{G}_1} : (d \in [a]_{\bowtie}) \vee (\#[c_0]_{\bowtie} = 1)\}$ .
- 2 <sup>1</sup>If  $\#[c_0]_{\sim_{l_2}} = 1$  then return  $\Lambda_{\mathcal{G}_2}$  and exit the algorithm. In the deletion or addition case, if  $\mathcal{G}_2$  is regular then let  $\Lambda_{\mathcal{G}_2} := \Lambda_{\mathcal{G}_2} \cup \{\{1, \dots, n\}\}$ .
- 3 <sup>2</sup>For each  $k \in [c_0]_{\sim_{l_2}} \setminus \{c_0\}$ 
  - 3.1 Consider the polydiagonal  $P := \{\mathbf{x} : x_{c_0} = x_k\}$ .
  - 3.2 For each  $i = 1, \dots, t$ , consider the subspace  $J_{\lambda_i}^1 := E_{\lambda_i} \cap P$ .
  - 3.3 If for all  $i = 1, \dots, t$ ,  $J_{\lambda_i}^1$  is the zero subspace then go to step 3.
  - 3.4 Consider only the nonzero subspaces  $J_{\lambda_{i_j}}^1$ , say for  $j = 1, \dots, s$ .
  - 3.5 Take  $J_{\lambda_{i_j}}^{p_{i_j}}$  according to (3.15), for  $j = 1, \dots, s$ .
  - 3.6 Let  $\tilde{V}$  be the set of synchrony subspaces returned by Algorithm A.1 executed on  $A$  restricted to

$$J_{\lambda_{i_1}}^{p_{i_1}} \oplus \dots \oplus J_{\lambda_{i_s}}^{p_{i_s}}.$$

---

<sup>1</sup>See Corollary 3.6 (i) and (ii).

<sup>2</sup>See Proposition 3.13-3.14.



3.7 Let  $\tilde{\Lambda}$  be the set of balanced equivalence relations corresponding to  $\tilde{V}$ . In case of rewiring, we discard those  $\bowtie$  satisfying  $d \in [c]_{\bowtie}$  from  $\tilde{\Lambda}^3$ .

3.8 Let  $\Lambda_{\mathcal{G}_2} := \Lambda_{\mathcal{G}_2} \cup \tilde{\Lambda}$ .

4 Return  $\Lambda_{\mathcal{G}_2}$ .

◇

### 3.4 Examples

In this subsection, we present two examples of networks, for which the algorithm is applied step by step to obtain the lattice of all balanced equivalence relations of the new network. The first example treats a series of edge deletions and the second one calculates for an edge rewiring.

**Example 3.16** Consider the networks  $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$  and  $\mathcal{G}_4$  given by Figure 3, where  $\mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4$  are obtained by successively removing edges  $(1, 3)$ ,  $(4, 1)$  and  $(2, 4)$  from  $\mathcal{G}_1$ . We start by generating the lattice  $\Lambda_{\mathcal{G}_1}$  of balanced equivalence relations on  $\mathcal{G}_1$  using Algorithm 6.3 in [2]. Then, we obtain successively the lattice  $\Lambda_{\mathcal{G}_i}$  based on  $\Lambda_{\mathcal{G}_{i-1}}$  using Algorithm 3.15, for  $i = 2, 3, 4$ . The result is summarized in Table 1.

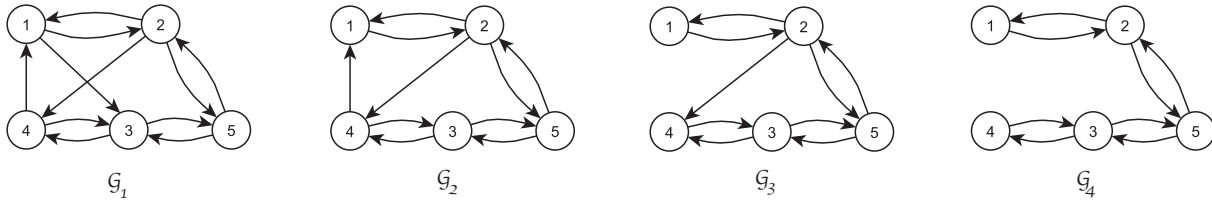


Figure 3: Four networks  $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4$ , where  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by removing the edge  $(1, 3)$ ,  $\mathcal{G}_3$  is obtained from  $\mathcal{G}_2$  by removing the edge  $(4, 1)$  and  $\mathcal{G}_4$  is obtained from  $\mathcal{G}_3$  by removing the edge  $(2, 4)$ .

More precisely, consider the network  $\mathcal{G}_2$  as obtained from  $\mathcal{G}_1$  by removing the edge  $(1, 3)$ . Since  $[3]_{\sim_2} = \{1, 2, 3, 4, 5\}$ , there can be additional balanced equivalence relations on  $\mathcal{G}_2$ , besides those from  $\Lambda_{\mathcal{G}_1}$  satisfying  $\#[3]_{\bowtie} = 1$  (cf. Corollary 3.6 (iii)). At step 1 of Algorithm 3.15,  $\Lambda_{\mathcal{G}_2}$  is set to be  $\{\bowtie_0^2, \bowtie_1^2, \bowtie_5^2\} := \{\bowtie_0^1, \bowtie_1^1, \bowtie_3^1\} = \Lambda_{\mathcal{G}_1}$ . Since  $\mathcal{G}_2$  is regular, at step 2 we add the balanced relation  $\bowtie_6^2 = \{\{1, 2, 3, 4, 5\}\}$  to  $\Lambda_{\mathcal{G}_2}$ . At step 3, every  $k \in \{1, 2, 4, 5\}$  is considered. Note that the eigenspaces of the adjacency matrix of  $\mathcal{G}_2$  are given by

$$E_{\frac{-1+\sqrt{5}}{2}} = \langle \left(1, 1, 2\frac{3-\sqrt{5}}{1-\sqrt{5}}, \frac{\sqrt{5}-3}{2}, \frac{\sqrt{5}-3}{2}\right) \rangle, \quad E_{\frac{-1-\sqrt{5}}{2}} = \langle \left(1, 1, 2\frac{3+\sqrt{5}}{1+\sqrt{5}}, -\frac{\sqrt{5}+3}{2}, -\frac{\sqrt{5}+3}{2}\right) \rangle,$$

$$E_2 = \langle (1, 1, 1, 1, 1) \rangle, \quad E_0 = \langle (-1, 1, -1, -1, 1) \rangle, \quad \text{and} \quad E_{-1} = \langle (2, -1, 2, -1, -1) \rangle$$

<sup>3</sup>This discard is not essential to obtain  $\Lambda_{\mathcal{G}_2}$ , but only to be consistent with Proposition 3.13.

| $\Lambda_{\mathcal{G}_1}$                                  | $\Lambda_{\mathcal{G}_2}$   | $\Lambda_{\mathcal{G}_3}$   | $\Lambda_{\mathcal{G}_4}$   |
|--|---|---|---|
| $\mathfrak{A}_0^1 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$ | $\mathfrak{A}_0^2 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\} = \mathfrak{A}_0^1$ | $\mathfrak{A}_0^3 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\} = \mathfrak{A}_0^1$ | $\mathfrak{A}_0^4 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\} = \mathfrak{A}_0^1$ |
| $\mathfrak{A}_1^1 = \{\{1, 2\}, \{3\}, \{4, 5\}\}$         | $\mathfrak{A}_1^2 = \{\{1, 2\}, \{3\}, \{4, 5\}\} = \mathfrak{A}_1^1$         | $\mathfrak{A}_1^3 = \{\{1\}, \{2\}, \{3\}, \{4, 5\}\} = \mathfrak{A}_2^1$     | $\mathfrak{A}_1^4 = \{\{1, 4\}, \{2, 3\}, \{5\}\}$                            |
| $\mathfrak{A}_2^1 = \{\{1\}, \{2\}, \{3\}, \{4, 5\}\}$     | $\mathfrak{A}_2^2 = \{\{1, 3\}, \{2, 5\}, \{4\}\}$                            |   |   |
|  | $\mathfrak{A}_3^2 = \{\{1, 3, 4\}, \{2, 5\}\}$                                |   |   |
|  | $\mathfrak{A}_4^2 = \{\{1, 3\}, \{2, 4, 5\}\}$                                |   |   |
|  | $\mathfrak{A}_5^2 = \{\{1\}, \{2\}, \{3\}, \{4, 5\}\} = \mathfrak{A}_2^1$     |   |   |
|  | $\mathfrak{A}_6^2 = \{\{1, 2, 3, 4, 5\}\}$                                    |   |   |

Table 1: The lattices of balanced equivalence relations for the networks  $\mathcal{G}_i$ ,  $i = 1, \dots, 4$  in Figure 3.

Thus,  $A$  is semisimple. For  $k = 1$ , consider the polydiagonal

$$P_1 := \{\mathbf{x} : x_3 = x_1\}.$$

Then, the only nonzero  $J_{\lambda_i}^1$ 's at step 3.4 are

$$J_{\lambda_i}^1 = E_{\lambda_i} \cap P_1 = E_{\lambda_i}, \text{ for } \lambda_i = 2, 0, -1.$$

Executing Algorithm A.1 at step 3.6 with  $A$  restricted to  $J_2^1 \oplus J_0^1 \oplus J_{-1}^1$ , we obtain at step 3.7 the set of synchrony subspaces corresponding to

$$\tilde{\Lambda}_1 = \{\mathfrak{A}_2^2, \mathfrak{A}_3^2, \mathfrak{A}_4^2\}.$$

Thus, we set  $\Lambda_{\mathcal{G}_2}$  to be  $\{\mathfrak{A}_0^2, \mathfrak{A}_1^2, \mathfrak{A}_5^2, \mathfrak{A}_6^2\} \cup \tilde{\Lambda}_1 = \{\mathfrak{A}_0^2, \mathfrak{A}_1^2, \mathfrak{A}_2^2, \mathfrak{A}_3^2, \mathfrak{A}_4^2, \mathfrak{A}_5^2, \mathfrak{A}_6^2\}$ . Analogously, for  $k = 2, 4, 5$ , consider the polydiagonals

$$P_2 := \{\mathbf{x} : x_3 = x_2\}, \quad P_3 := \{\mathbf{x} : x_3 = x_4\} \quad \text{and} \quad P_4 := \{\mathbf{x} : x_3 = x_5\},$$

and we obtain

$$\tilde{\Lambda}_2 = \{\}, \quad \tilde{\Lambda}_3 = \{\mathfrak{A}_3^2\} \quad \text{and} \quad \tilde{\Lambda}_4 = \{\}.$$

Therefore, at the end of Algorithm 3.15, we have

$$\Lambda_{\mathcal{G}_2} = \{\mathfrak{A}_0^2, \mathfrak{A}_1^2, \mathfrak{A}_5^2, \mathfrak{A}_6^2\} \cup \tilde{\Lambda}_1 \cup \tilde{\Lambda}_3 = \{\mathfrak{A}_0^2, \mathfrak{A}_1^2, \mathfrak{A}_2^2, \mathfrak{A}_3^2, \mathfrak{A}_4^2, \mathfrak{A}_5^2, \mathfrak{A}_6^2\}.$$

Next, consider the network  $\mathcal{G}_3$ , which is obtained from  $\mathcal{G}_2$  by removing the edge  $(4, 1)$ . In this case, we have  $\#[1]_{\sim_2} = 1$  (cf. Corollary 3.6 (ii)). Thus, Algorithm 3.15 exits at step 2 and the only balanced equivalence relations on  $\mathcal{G}_3$  are those balanced equivalence relations on  $\mathcal{G}_2$  such that  $\#[1]_{\mathfrak{A}} = 1$ . Therefore,  $\Lambda_{\mathcal{G}_3} = \{\mathfrak{A}_0^2, \mathfrak{A}_5^2\} := \{\mathfrak{A}_0^3, \mathfrak{A}_1^3\}$ .

In the end, consider the network  $\mathcal{G}_4$ , which is obtained from  $\mathcal{G}_3$  by removing the edge  $(2, 4)$ . At step 1 of Algorithm 3.15, we set  $\Lambda_{\mathcal{G}_4} = \{\mathfrak{A}_0^3\} := \{\mathfrak{A}_0^4\}$ . Since  $\#[4]_{\sim_1} = \{1, 4\}$ ,

we execute step 3 only for  $k = 1$  and consider the polydiagonal  $P := \{x : x_4 = x_1\}$ . The eigenspaces of the adjacency matrix of  $\mathcal{G}_4$  are given by

$$E_{-\sqrt{3}} = \langle (1, -\sqrt{3}, -\sqrt{3}, 1, 2) \rangle, \quad E_{\sqrt{3}} = \langle (1, \sqrt{3}, \sqrt{3}, 1, 2) \rangle,$$

$$E_0 = \langle (1, 0, 0, 1, -1) \rangle, \quad E_{-1} = \langle (-1, 1, -1, 1, 0) \rangle, \quad \text{and} \quad E_1 = \langle (1, 1, -1, -1, 0) \rangle.$$

Then, we have the following nonzero  $J_{\lambda_i}^1$ 's at step 3.4

$$J_{\lambda_i}^1 = E_{\lambda_i} \cap P_1 = E_{\lambda_i}, \text{ for } \lambda_i = 0, -\sqrt{3}, \sqrt{3}.$$

Executing Algorithm A.1 at step 3.6 with  $A$  restricted to  $J_{-\sqrt{3}}^1 \oplus J_0^1 \oplus J_{\sqrt{3}}^1$ , we have the set of synchrony subspaces corresponding to  $\tilde{\Lambda} = \{\mathfrak{A}_1^4\}$ . Therefore, at the end of Algorithm 3.15, we obtain  $\Lambda_{\mathcal{G}_4} = \{\mathfrak{A}_0^4\} \cup \tilde{\Lambda} = \{\mathfrak{A}_0^4, \mathfrak{A}_1^4\}$ .  $\diamond$

**Example 3.17** Consider the networks  $\mathcal{G}_1$  and  $\mathcal{G}_2$  given by Figure 4, where  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by rewiring the edge  $(3, 2)$  to  $(4, 2)$ . We first obtain the lattice  $\Lambda_{\mathcal{G}_1}$  of balanced equivalence relations on  $\mathcal{G}_1$  by Algorithm 6.3 in [2], and then obtain  $\Lambda_{\mathcal{G}_2}$  using Algorithm 3.15. The results are presented in Tables 2 and 3.

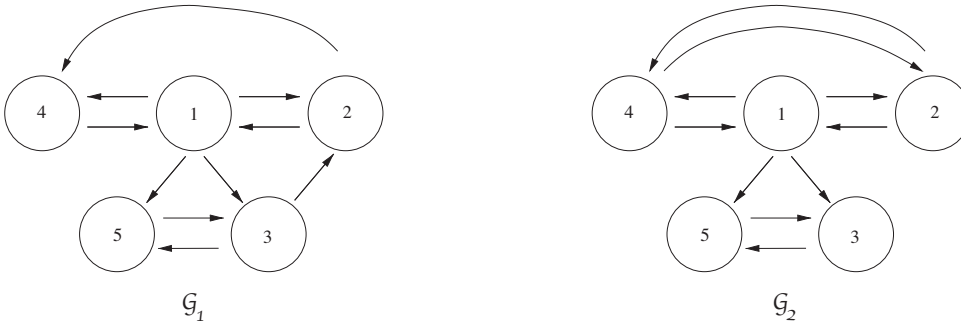


Figure 4: Network  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by the rewiring  $(2; 3, 4)$ .

|  |  |   |
|--|--|---|
| $\mathfrak{A}_0^1 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$ | $\mathfrak{A}_4^1 = \{\{1\}, \{2, 3, 5\}, \{4\}\}$ | $\mathfrak{A}_8^1 = \{\{1, 4\}, \{2, 3, 5\}\}$    |
| $\mathfrak{A}_1^1 = \{\{1, 4\}, \{2\}, \{3\}, \{5\}\}$     | $\mathfrak{A}_5^1 = \{\{1, 4\}, \{2, 5\}, \{3\}\}$ | $\mathfrak{A}_9^1 = \{\{1, 2, 5\}, \{3, 4\}\}$    |
| $\mathfrak{A}_2^1 = \{\{1\}, \{2, 5\}, \{3\}, \{4\}\}$     | $\mathfrak{A}_6^1 = \{\{1, 4\}, \{3, 5\}, \{2\}\}$ | $\mathfrak{A}_{10}^1 = \{\{1, 3, 4\}, \{2, 5\}\}$ |
| $\mathfrak{A}_3^1 = \{\{1\}, \{2\}, \{3, 5\}, \{4\}\}$     | $\mathfrak{A}_7^1 = \{\{1\}, \{2, 5\}, \{3, 4\}\}$ | $\mathfrak{A}_{11}^1 = \{\{1\}, \{2, 3, 4, 5\}\}$ |
|  |  | $\mathfrak{A}_{12}^1 = \{\{1, 2, 3, 4, 5\}\}$     |

Table 2: The balanced equivalence relations for the network  $\mathcal{G}_1$  of Figure 4.

|   |  |   |
|---|--|---|
| $\bowtie_0^2 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\} = \bowtie_0^1$ | $\bowtie_5^2 = \{\{1, 2, 4\}, \{3\}, \{5\}\}$                  | $\bowtie_{11}^2 = \{\{1, 2, 3\}, \{4, 5\}\}$                  |
| $\bowtie_1^2 = \{\{1, 2\}, \{3\}, \{4\}, \{5\}\}$                   | $\bowtie_6^2 = \{\{1\}, \{2, 3\}, \{4, 5\}\}$                  | $\bowtie_{12}^2 = \{\{1, 2, 4\}, \{3, 5\}\}$                  |
| $\bowtie_2^2 = \{\{1, 4\}, \{2\}, \{3\}, \{5\}\} = \bowtie_1^1$     | $\bowtie_7^2 = \{\{1, 2\}, \{3, 5\}, \{4\}\}$                  | $\bowtie_{13}^2 = \{\{1, 2, 5\}, \{3, 4\}\} = \bowtie_9^1$    |
| $\bowtie_3^2 = \{\{1\}, \{2, 4\}, \{3\}, \{5\}\}$                   | $\bowtie_8^2 = \{\{1\}, \{2, 4\}, \{3, 5\}\}$                  | $\bowtie_{14}^2 = \{\{1, 3, 4\}, \{2, 5\}\} = \bowtie_{10}^1$ |
| $\bowtie_4^2 = \{\{1\}, \{2\}, \{3, 5\}, \{4\}\} = \bowtie_3^1$     | $\bowtie_9^2 = \{\{1\}, \{2, 5\}, \{3, 4\}\} = \bowtie_7^1$    | $\bowtie_{15}^2 = \{\{1, 4, 5\}, \{2, 3\}\}$                  |
|   | $\bowtie_{10}^2 = \{\{1, 4\}, \{2\}, \{3, 5\}\} = \bowtie_6^1$ | $\bowtie_{16}^2 = \{\{1\}, \{2, 3, 4, 5\}\} = \bowtie_{11}^1$ |
|   |  | $\bowtie_{17}^2 = \{\{1, 2, 3, 4, 5\}\} = \bowtie_{12}^1$     |

Table 3: The balanced equivalence relations for the network  $\mathcal{G}_2$  of Figure 4.

More precisely, since the network  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by the rewiring (2; 3, 4), we apply Algorithm 3.15 for the rewiring case to obtain  $\Lambda_{\mathcal{G}_2}$ . Observe that the two networks are both regular, thus all cells 1, 2, 3, 4, 5 are input equivalent.

At step 1 of Algorithm 3.15,  $\Lambda_{\mathcal{G}_2}$  is composed of those balanced equivalence relations  $\bowtie$  on  $\mathcal{G}_1$  such that  $3 \bowtie 4$  or  $\#[2]_{\bowtie} = 1$ . By Table 2, the balanced equivalence relations such that  $3 \bowtie 4$  are  $\bowtie_7^1, \bowtie_9^1, \bowtie_{10}^1, \bowtie_{11}^1, \bowtie_{12}^1$ , and those satisfying  $\#[2]_{\bowtie} = 1$  are  $\bowtie_0^1, \bowtie_1^1, \bowtie_3^1$  and  $\bowtie_6^1$ . Thus, at step 1 of the algorithm, we set  $\Lambda_{\mathcal{G}_2} = \{\bowtie_0^2, \bowtie_2^2, \bowtie_4^2, \bowtie_9^2, \bowtie_{10}^2, \bowtie_{13}^2, \bowtie_{14}^2, \bowtie_{16}^2, \bowtie_{17}^2\}$  as shown in Table 3. Since  $\#[2]_{\sim_2} = 5$ , step 2 is skipped and step 3 is executed for every  $k \in \{1, 3, 4, 5\}$ . The eigenspaces of the adjacency matrix  $A$  of  $\mathcal{G}_2$  are given by

$$E_{-1} = \langle (1, -1, -1, 0, 0), (1, 0, 0, -1, -1), (1, 0, -1, -1, 0) \rangle,$$

$$E_2 = \langle (1, 1, 1, 1, 1) \rangle \quad \text{and} \quad E_1 = \langle (0, 0, 1, 0, 1) \rangle .$$

For  $k = 1$ , consider the polydiagonal

$$P_1 := \{\mathbf{x} : x_2 = x_1\} .$$

The only nonzero  $J_{\lambda_i}^1$ 's at step 3.4 are

$$J_{\lambda_i}^1 = E_{\lambda_i} \cap P_1 = E_{\lambda_i}, \text{ for } \lambda_i = 2, 1$$

and

$$J_{-1}^1 = E_{-1} \cap P_1 = \langle (-1, -1, 1, 2, 0), (0, 0, 1, 0, -1) \rangle .$$

Executing Algorithm 3.15 with  $A$  restricted to  $J_2^1 \oplus J_1^1 \oplus J_{-1}^1$  at step 3.6, we obtain the set of synchrony subspaces corresponding to

$$\tilde{\Lambda}_1 = \{\bowtie_1^2, \bowtie_5^2, \bowtie_7^2, \bowtie_{11}^2, \bowtie_{12}^2\} .$$

Thus, we set  $\Lambda_{\mathcal{G}_2}$  as  $\{\bowtie_0^2, \bowtie_2^2, \bowtie_4^2, \bowtie_9^2, \bowtie_{10}^2, \bowtie_{13}^2, \bowtie_{14}^2, \bowtie_{16}^2, \bowtie_{17}^2\} \cup \tilde{\Lambda}_1$  at step 3.8. Further, consider for  $k = 3$ , the polydiagonal

$$P_2 := \{\mathbf{x} : x_2 = x_3\} .$$

Then, we have the following nonzero  $J_{\lambda_i}^1$ 's

$$J_2^1 = E_2 \cap P_2 = E_2, \quad \text{and} \quad J_{-1}^1 = E_{-1} \cap P_2 = \langle (1, -1, -1, 0, 0), (1, 0, 0, -1, -1) \rangle.$$

Executing Algorithm 3.15 with  $A$  restricted to  $J_2^1 \oplus J_{-1}^1$ , we obtain the set of synchrony subspaces corresponding to

$$\tilde{\Lambda}_2 = \{\bowtie_6^2, \bowtie_{11}^2, \bowtie_{15}^2\}.$$

Similarly, for  $k = 4$ , consider the polydiagonal

$$P_3 := \{\mathbf{x} : x_2 = x_4\},$$

together with the nonzero subspaces  $J_{\lambda_i}^1$ 's given by

$$J_{\lambda_i}^1 = E_{\lambda_i} \cap P_3 = E_{\lambda_i}, \quad \text{for } \lambda_i = 2, 1$$

and

$$J_{-1}^1 = E_{-1} \cap P_3 = \langle (2, -1, -2, -1, 0), (0, 0, 1, 0, -1) \rangle.$$

Executing Algorithm 3.15 with  $A$  restricted to  $J_2^1 \oplus J_1^1 \oplus J_{-1}^1$ , we have the following set of balanced equivalence relations

$$\tilde{\Lambda}_3 = \{\bowtie_3^2, \bowtie_5^2, \bowtie_8^2, \bowtie_{12}^2\}.$$

In the end, consider for  $k = 5$ , the polydiagonal

$$P_4 := \{\mathbf{x} : x_2 = x_5\},$$

with the nonzero subspaces  $J_{\lambda_i}^1$ 's given by

$$J_2^1 = E_2 \cap P_2 = E_2, \quad \text{and} \quad J_{-1}^1 = E_{-1} \cap P_2 = \langle (2, -1, -1, -1, -1), (1, 0, -1, -1, 0) \rangle.$$

Executing Algorithm 3.15 with  $A$  restricted to  $J_2^1 \oplus J_{-1}^1$ , we have the following set of balanced equivalence relations

$$\tilde{\Lambda}_4 = \{\}.$$

Therefore, at the end of Algorithm 3.15, we have

$$\Lambda_{\mathcal{G}_2} = \{\bowtie_0^2, \bowtie_2^2, \bowtie_4^2, \bowtie_9^2, \bowtie_{10}^2, \bowtie_{13}^2, \bowtie_{14}^2, \bowtie_{16}^2, \bowtie_{17}^2\} \cup \bigcup_{i=1, \dots, 4} \tilde{\Lambda}_i$$

as listed in Table 3. ◇

### 3.5 Graph operations on multiple edges

We extend our results of Subsections 3.1 and 3.2 to edge operations involving multiple edges. More precisely, we consider the graph operations on edges of a coupled cell network including the deletion, addition and/or rewiring of multiple edges.

### 3.5.1 Deleting or adding multiple edges

Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by removing the multiple edges  $(d_i, c_i)$ , for  $i = 1, 2, \dots, s$ . Analogously,  $\mathcal{G}_1$  is obtained from  $\mathcal{G}_2$  by adding those edges. Let  $\bowtie$  be an equivalence relation on  $C$ . A necessary condition for  $\bowtie$  to be recoverable is that  $\bowtie$  refines both input relations  $\sim_{I_1}$  and  $\sim_{I_2}$ . More precisely, if we consider the equivalence relation  $\sim_{I_{1,2}}$  on  $C$  where

$$[c]_{\sim_{I_{1,2}}} = [c]_{\sim_{I_1}} \cap [c]_{\sim_{I_2}}, \quad \forall c \in C,$$

then  $\bowtie < \sim_{I_{1,2}}$ . However, as the example below shows, this is not a sufficient condition for  $\bowtie$  to be recoverable.

**Example 3.18** Consider the three 5-cell networks in Figure 5, where the network  $\mathcal{G}_2$  on the middle is obtained from the network  $\mathcal{G}_1$  on the left by deletion of the two edges  $(5, 1)$  and  $(4, 3)$ . Consider the 5-cell network  $\mathcal{G}_E$  on the right with the same set of cells of the two networks  $\mathcal{G}_i$  and these two edges. In this example we have:

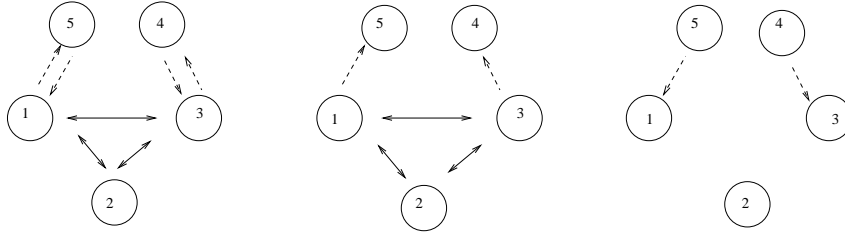


Figure 5: Three examples of 5-cell networks. The network  $\mathcal{G}_2$  on the middle is obtained from  $\mathcal{G}_1$  on the left by deletion of the two edges directed to cells 1 and 3 appearing at the network  $\mathcal{G}_E = \mathcal{G}_1 - \mathcal{G}_2$  on the right.

$$\sim_{I_1} = \{\{1, 3\}, \{2\}, \{4, 5\}\}, \quad \sim_{I_2} = \{\{1, 2, 3\}, \{4, 5\}\} \quad \text{and} \quad \sim_{I_{1,2}} = \{\{1, 3\}, \{2\}, \{4, 5\}\}.$$

Taking

$$\bowtie = \{\{1, 3\}, \{2\}, \{4\}, \{5\}\} < \sim_{I_{1,2}},$$

we have that  $\bowtie$  is balanced for  $\mathcal{G}_2$  but it is not balanced for  $\mathcal{G}_1$ . Note that  $\bowtie$  is not balanced for the network  $\mathcal{G}_E = \mathcal{G}_1 - \mathcal{G}_2$  on the right of Figure 5.  $\diamond$

In what follows, denote by  $\mathcal{G}_E = \mathcal{G}_1 - \mathcal{G}_2$  obtained by removing all edges of  $\mathcal{G}_2$  from  $\mathcal{G}_1$ . Using  $\mathcal{G}_E$ , we obtain the following result as an extension of Lemma 3.3 and Remark 3.8 (i).

**Lemma 3.19** *Let  $\bowtie$  be an equivalence relation on the set of cells  $C$  of the networks  $\mathcal{G}_1, \mathcal{G}_2$  and  $\mathcal{G}_E$ . Assume that  $\bowtie$  is balanced for  $\mathcal{G}_i$ , for some  $i \in \{1, 2\}$ . We have that,  $\bowtie$  is balanced on  $\mathcal{G}_j$ ,  $j \in \{1, 2\}$ ,  $j \neq i$ , if and only if it is balanced on  $\mathcal{G}_E$ , i.e.,*

$$\bowtie \in \Lambda_{\mathcal{G}_j} \Leftrightarrow \bowtie \in \Lambda_{\mathcal{G}_E}.$$

**Proof** Let  $\bowtie$  be an equivalence relation on the set of cells  $C$  of the networks  $\mathcal{G}_1, \mathcal{G}_2$  and  $\mathcal{G}_E$ . Assume that  $\bowtie \in \Lambda_{\mathcal{G}_i}$  for some  $i \in \{1, 2\}$ . We start by observing that, as  $\bowtie \prec \sim_{I_i}$  and

$$I_1(x) = I_2(x) + I_E(x), \quad \forall x \in C,$$

we have that  $\bowtie$  refines  $\sim_{I_j}$  for  $j \in \{1, 2\}, j \neq i$ , if and only if  $\bowtie$  refines the input relation for the network  $\mathcal{G}_E$ . Moreover, we have

$$m([\alpha]_{\bowtie}, I_1(x)) = m([\alpha]_{\bowtie}, I_2(x)) + m([\alpha]_{\bowtie}, I_E(x)), \quad \forall \alpha, x \in C. \quad (3.16)$$

As  $\bowtie \in \Lambda_{\mathcal{G}_i}$  then, for  $x \bowtie y$ , we have

$$m([\alpha]_{\bowtie}, I_i(x)) = m([\alpha]_{\bowtie}, I_i(y)), \quad \forall \alpha \in C.$$

It follows then, from (3.16), that

$$m([\alpha]_{\bowtie}, I_j(x)) = m([\alpha]_{\bowtie}, I_j(y)) \Leftrightarrow m([\alpha]_{\bowtie}, I_E(x)) = m([\alpha]_{\bowtie}, I_E(y)).$$

□

### 3.5.2 Rewiring multiple edges

Let  $\mathcal{G}_2$  be the network obtained by rewiring multiple edges of  $\mathcal{G}_1$ . If the rewiring takes place in the input set of a single cell  $c_0$  by replacing  $m$  edges with another  $m$  edges of the same type, such that

$$I_2(c_0) = I_1(c_0) \setminus \{d_1, \dots, d_m\} \cup \{a_1, \dots, a_m\}, \quad (3.17)$$

then Lemma 3.10 extends to the following result.

**Lemma 3.20** *Let  $\bowtie$  be an equivalence relation on  $C$ . Then, we have:*

(i) *If  $\#[c_0]_{\bowtie} = 1$  or  $\#(\{a_1, \dots, a_m\} \cap [\alpha]_{\bowtie}) = \#(\{d_1, \dots, d_m\} \cap [\alpha]_{\bowtie})$  for all  $\alpha \in C$ , then*

$$\bowtie \in \Lambda_{\mathcal{G}_1} \quad \Leftrightarrow \quad \bowtie \in \Lambda_{\mathcal{G}_2}.$$

(ii) *Otherwise,  $\bowtie \notin \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$ .*

In case of rewirings concerning multiple nodes  $c_0^1, \dots, c_0^n$  such that

$$I_2(c_0^i) = I_1(c_0^i) \setminus \{d_1^i, \dots, d_{m_i}^i\} \cup \{a_1^i, \dots, a_{m_i}^i\}, \quad i = 1, \dots, n, \quad (3.18)$$

we have the following lemma.

**Lemma 3.21** *Let  $\bowtie$  be an equivalence relation on  $C$ . Then, we have:*

(i) *If for each  $i \in \{1, 2, \dots, n\}$ , either*

(a)  $\#[c_0^i]_{\bowtie} = 1$ ; *or*

(b)  $[c_0^i]_{\bowtie} \subseteq \{c_0^1, \dots, c_0^n\}$  *and for all  $j \neq i$  with  $c_0^i \bowtie c_0^j$ , we have*

$$\begin{aligned} & \#(\{a_1^i, \dots, a_{m_i}^i\} \cap [\alpha]_{\bowtie}) - \#(\{d_1^i, \dots, d_{m_i}^i\} \cap [\alpha]_{\bowtie}) \\ &= \#(\{a_1^j, \dots, a_{m_j}^j\} \cap [\alpha]_{\bowtie}) - \#(\{d_1^j, \dots, d_{m_j}^j\} \cap [\alpha]_{\bowtie}), \quad \forall \alpha \in C; \end{aligned}$$

*or else*

(c)  $[c_0^i]_{\bowtie} \not\subseteq \{c_0^1, \dots, c_0^n\}$  *and  $\#(\{a_1^i, \dots, a_{m_i}^i\} \cap [\alpha]_{\bowtie}) = \#(\{d_1^i, \dots, d_{m_i}^i\} \cap [\alpha]_{\bowtie})$ ,  $\forall \alpha \in C$ ,*

*then we have*

$$\bowtie \in \Lambda_{\mathcal{G}_1} \Leftrightarrow \bowtie \in \Lambda_{\mathcal{G}_2}.$$

(ii) *Otherwise,  $\bowtie \notin \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$ .*

### 3.5.3 Deleting and adding multiple edges without rewirings

Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by both removing and adding multiple edges to  $\mathcal{G}_1$  such that no rewiring occurs. Assume the removed edges are

$$(d_1, c_1), (d_2, c_2), \dots, (d_s, c_s) \tag{3.19}$$

and the added edges are

$$(b_1, a_1), (b_2, a_2), \dots, (b_r, a_r), \tag{3.20}$$

where  $(d_i, c_i) \neq (b_j, a_j)$ , for all  $1 \leq i \leq s$  and  $1 \leq j \leq r$ . Moreover, as we are assuming no rewiring, for every two edges  $(d_i, c_i)$  and  $(b_j, a_j)$  of the same type, we have  $c_i \neq a_j$ , i.e. no pair of edges can be put together as a rewiring.

Denote by  $\mathcal{G}$  the network obtained from  $\mathcal{G}_1$  by removing the edges (3.19). Then,  $\mathcal{G}_2$  is obtained from  $\mathcal{G}$  by adding the edges (3.20). Let  $\mathcal{G}_E = \mathcal{G}_1 - \mathcal{G}$  and  $\mathcal{G}_F = \mathcal{G}_2 - \mathcal{G}$ .

**Lemma 3.22** *Let  $\mathcal{G}_2$  be obtained from  $\mathcal{G}_1$  by both deleting the multiple edges (3.19) and adding the multiple edges (3.20), such that there are no edges  $(d_i, c_i)$  and  $(b_j, a_j)$  of the same type with  $c_i = a_j$ . Let  $\bowtie \in \Lambda_{\mathcal{G}_i}$  be a balanced equivalence relation on  $\mathcal{G}_i$  for some  $i \in \{1, 2\}$ . Then, we have that  $\bowtie$  is balanced on  $\mathcal{G}_j$  for  $j \in \{1, 2\}$ ,  $j \neq i$ , if and only if it is balanced on  $\mathcal{G}_E$  and  $\mathcal{G}_F$ , i.e.,*

$$\bowtie \in \Lambda_{\mathcal{G}_j} \Leftrightarrow \bowtie \in \Lambda_{\mathcal{G}_E} \cap \Lambda_{\mathcal{G}_F}.$$

**Proof** Let  $\bowtie$  be an equivalence relation on the set of cells  $C$  of the networks  $\mathcal{G}_1, \mathcal{G}_2$  and  $\mathcal{G}_E, \mathcal{G}_F$ . Assume that  $\bowtie \in \Lambda_{\mathcal{G}_i}$  for some  $i \in \{1, 2\}$ . First note that, as  $\bowtie$  refines  $\sim_{I_i}$  and

$$I_1(x) = I_2(x) + I_E(x) - I_F(x), \quad \forall x \in C,$$



we have that if  $\succcurlyeq$  refines the input relations for  $\mathcal{G}_E$  and  $\mathcal{G}_F$  then it also refines  $\sim_{I_j}$  for  $j \in \{1, 2\}, j \neq i$ . Moreover, we have

$$m([\alpha]_{\succcurlyeq}, I_1(x)) = m([\alpha]_{\succcurlyeq}, I_2(x)) + m([\alpha]_{\succcurlyeq}, I_E(x)) - m([\alpha]_{\succcurlyeq}, I_F(x)), \quad \forall \alpha \in \mathcal{C}. \quad (3.21)$$

As  $\succcurlyeq \in \Lambda_{\mathcal{G}_i}$  is a balanced equivalence relation on  $\mathcal{G}_i$  then, for  $x \succcurlyeq y$ , we have

$$m([\alpha]_{\succcurlyeq}, I_i(x)) = m([\alpha]_{\succcurlyeq}, I_i(y)), \quad \forall \alpha \in \mathcal{C}.$$

Since there is no rewiring, we have

$$m([\alpha]_{\succcurlyeq}, I_E(x)) = 0 \quad \text{or} \quad m([\alpha]_{\succcurlyeq}, I_F(x)) = 0, \quad \forall \alpha, x \in \mathcal{C}.$$

Therefore, we can conclude from (3.21), that

$$m([\alpha]_{\succcurlyeq}, I_j(x)) = m([\alpha]_{\succcurlyeq}, I_j(y))$$

if and only if

$$m([\alpha]_{\succcurlyeq}, I_E(x)) = m([\alpha]_{\succcurlyeq}, I_E(y)) \quad \text{and} \quad m([\alpha]_{\succcurlyeq}, I_F(x)) = m([\alpha]_{\succcurlyeq}, I_F(y)).$$

That is,  $\succcurlyeq$  is balanced for  $\mathcal{G}_j$  if and only if it is balanced for both  $\mathcal{G}_E$  and  $\mathcal{G}_F$ .  $\square$

### 3.5.4 Deleting and adding multiple edges including rewirings

We discuss now the cases where  $\mathcal{G}_2$  is a network obtained from a network  $\mathcal{G}_1$  by removing and adding multiple edges to  $\mathcal{G}_1$ , but where some of edge deletions and additions involve rewirings. That is, following the notations (3.19)-(3.20), we have edges  $(d_i, c_i)$  and  $(b_j, a_j)$  of the same type and such that  $c_i = a_j$ . One of the directions of Lemma 3.22 still holds. More precisely, we have:

**Lemma 3.23** *Let  $\succcurlyeq$  be an equivalence relation on the set of cells  $\mathcal{C}$  of the networks  $\mathcal{G}_1, \mathcal{G}_2$  and  $\mathcal{G}_E, \mathcal{G}_F$ . Assume  $\succcurlyeq$  is a balanced equivalence relation on  $\mathcal{G}_i$ , for some  $i \in \{1, 2\}$ . If  $\succcurlyeq$  is balanced on both  $\mathcal{G}_E$  and  $\mathcal{G}_F$ , then it is balanced on  $\mathcal{G}_j$ . I.e.,*

$$\succcurlyeq \in \Lambda_{\mathcal{G}_E} \cap \Lambda_{\mathcal{G}_F} \quad \Rightarrow \quad \succcurlyeq \in \Lambda_{\mathcal{G}_j}. \quad (3.22)$$

**Proof** Consider the proof of Lemma 3.22 till equation (3.21) which holds. Again, as we are assuming that  $\succcurlyeq$  is a balanced equivalence relation on  $\mathcal{G}_i$ , for  $x, y \in \mathcal{C}$  such that  $x \succcurlyeq y$ , we have

$$m([\alpha]_{\succcurlyeq}, I_i(x)) = m([\alpha]_{\succcurlyeq}, I_i(y)), \quad \forall \alpha \in \mathcal{C}.$$

It follows then, from (3.21), that if

$$m([\alpha]_{\succcurlyeq}, I_E(x)) = m([\alpha]_{\succcurlyeq}, I_E(y)) \quad \text{and} \quad m([\alpha]_{\succcurlyeq}, I_F(x)) = m([\alpha]_{\succcurlyeq}, I_F(y)),$$

then

$$m([\alpha]_{\succcurlyeq}, I_j(x)) = m([\alpha]_{\succcurlyeq}, I_j(y))$$

and so  $\succcurlyeq$  is balanced for  $\mathcal{G}_j$ .  $\square$

The reverse direction of (3.22) does not hold in general. A simple example is to consider a network  $\mathcal{G}_1$  of 2 cells with  $I_1(1) = \{2\}$  and  $I_1(2) = \{1\}$  and a network  $\mathcal{G}_2$  with the same set of cells and such that  $I_1(1) = \{1\}$  and  $I_1(2) = \{1\}$ . The network  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by removing the edge  $(2, 1)$  and adding the edge  $(1, 1)$ , that is, by a rewiring. We have that  $\mathcal{G}_E$  is the network of the 2 cells with  $I_E(1) = \{2\}$  and  $I_E(2) = \{\}$  and  $\mathcal{G}_F$  is the network of the 2 cells with  $I_F(1) = \{1\}$  and  $I_E(2) = \{\}$ . Then,  $\bowtie = \{\{1, 2\}\} \in \Lambda_{\mathcal{G}_1} = \Lambda_{\mathcal{G}_2}$ , despite of the fact that  $\bowtie \notin \Lambda_{\mathcal{G}_E} = \Lambda_{\mathcal{G}_F}$ . Note that in this case, although  $\bowtie = \{\{1, 2\}\} \equiv_{\sim_{I_1} = \sim_{I_2}}$ , it does not refine the input relations for  $\mathcal{G}_E$  and  $\mathcal{G}_F$ .

One way to go around this is to decompose the graph operation into two operations: one concerns the rewiring part while the other forms a rewiring-free deletion/addition edge operation. More precisely, assume some of the deleted/added edges in (3.19)-(3.20) can be paired up as rewiring. Then, by successively applying all these rewirings on  $\mathcal{G}_1$ , one obtain a network  $\mathcal{G}$ , from which  $\mathcal{G}_2$  can then be obtained by a rewiring-free deletion/addition edge operation. Let

$$(\bar{d}_1, \bar{c}_1), (\bar{d}_2, \bar{c}_2), \dots, (\bar{d}_{\bar{s}}, \bar{c}_{\bar{s}}), \quad (3.23)$$

be the edges to be deleted from  $\mathcal{G}$  and

$$(\bar{b}_1, \bar{a}_1), (\bar{b}_2, \bar{a}_2), \dots, (\bar{b}_{\bar{r}}, \bar{a}_{\bar{r}}), \quad (3.24)$$

be the edges to be added to  $\mathcal{G}$  to obtain  $\mathcal{G}_2$ .

Recall that  $\Lambda_{\mathcal{G}}^R$  stands for the set of recoverable balanced equivalence relations on  $\mathcal{G}$  from  $\mathcal{G}_1$ . We say that a balanced equivalence relation  $\bowtie$  for  $\mathcal{G}_2$  is *recoverable from  $\mathcal{G}_1$  through the rewiring network  $\mathcal{G}$*  if  $\bowtie \in \Lambda_{\mathcal{G}}^R$ . Denote by  $\Lambda_{\mathcal{G}_2}^{R, \mathcal{G}}$  the set of all such  $\bowtie$ 's. Then,

$$\Lambda_{\mathcal{G}_2}^{R, \mathcal{G}} := \Lambda_{\mathcal{G}}^R \cap \Lambda_{\mathcal{G}_2} = \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}} \cap \Lambda_{\mathcal{G}_2},$$

where the balanced relations in  $\Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}}$  can be determined by Lemma 3.21 and the balanced relations in  $\Lambda_{\mathcal{G}} \cap \Lambda_{\mathcal{G}_2}$  can be determined by Lemma 3.22.

**Remark 3.24** In general, the choice of  $\mathcal{G}$  is not unique, for given  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . Thus,  $\Lambda_{\mathcal{G}_2}^{R, \mathcal{G}}$  will differ for different  $\mathcal{G}$ . An example is given in Figure 6, where there are two choices of  $\mathcal{G}$  given by  $\mathcal{G}_{R_1}$  and  $\mathcal{G}_{R_2}$ . More precisely, there are two ways to realize the addition of the self loop  $(1, 1)$ : it can be thought as a rewiring from the edge  $(2, 1)$  of  $\mathcal{G}_1$ , which results in  $\mathcal{G}_{R_1}$ , or it is a rewiring from the edge  $(3, 1)$  of  $\mathcal{G}_1$ , which gives  $\mathcal{G}_{R_2}$ . Consider  $\bowtie = \{\{1, 3\}, \{2\}\}$ , which is balanced on both  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . However, it is balanced for the rewiring network  $\mathcal{G}_{R_2}$  but not for  $\mathcal{G}_{R_1}$ .  $\diamond$

As shown in Remark 3.24, some recoverable equivalence relations may not be recoverable through all rewiring networks  $\mathcal{G}$ . However, there is always an appropriate choice of  $\mathcal{G}$  for a given recoverable equivalence relation. More precisely, let  $\bowtie \in \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$  be recoverable. One can choose rewirings that preserve the  $\bowtie$ -classes in the

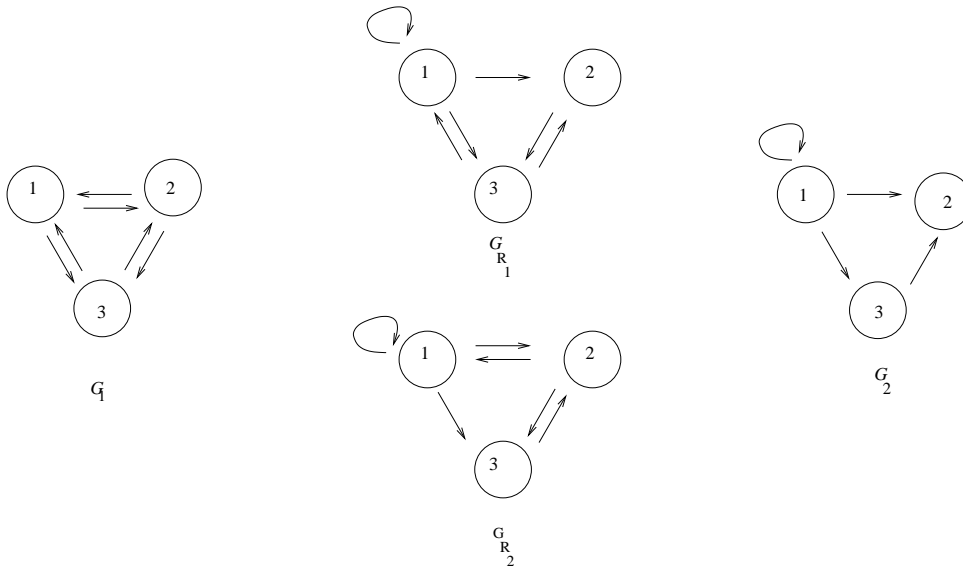


Figure 6: The network  $\mathcal{G}_2$  on the right can be obtained from the network on the left  $\mathcal{G}_1$  by deletion of two edges and two types of rewirings of one edge directed to cell 1.

following way. For every cell  $c$  with  $\#[c]_{\succ} > 1$ , assuming  $d_1, d_2, \dots, d_s$  are to be deleted from  $I(c)$  and  $a_1, a_2, \dots, a_r$  are to be added to  $I(c)$ , we take rewiring of form  $(c; d_i, a_j)$  for  $d_i \succ a_j$  until such form of rewiring is not possible any more. Then, consider every two  $\succ$ -equivalent cells  $c, c'$ . If  $d'_1, d'_2, \dots, d'_s$  and  $a'_1, a'_2, \dots, a'_r$  denote the elements to be deleted from and to be added to  $I(c')$ , respectively, then we form two rewirings of form  $(c; d, a)$  and  $(c'; d', a')$  simultaneously whenever  $d \succ d'$  and  $a \succ a'$  until choices are exhausted. For  $\succ$ -equivalent cells, after these two rounds of rewirings, they are left only with cells to be deleted from their input sets (or only with cells to be added). The rewiring network  $\mathcal{G}$  that is obtained by these two rounds of rewirings is a “preferred” choice for the given  $\succ$ . That is,  $\succ$  is recoverable through  $\mathcal{G}$ . For the example of Figure 6, this would lead to  $\mathcal{G}_{R_2}$ . In short, for every  $\succ \in \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}_2}$ , there is a rewiring network  $\mathcal{G}$  such that

- (i)  $\mathcal{G}$  is obtained from  $\mathcal{G}_1$  by successive rewirings;
- (ii)  $\mathcal{G}_2$  is obtained from  $\mathcal{G}$  by rewiring-free deletions and additions of edges; and
- (iii)  $\succ \in \Lambda_{\mathcal{G}_1} \cap \Lambda_{\mathcal{G}} \cap \Lambda_{\mathcal{G}_2}$ .

## 4 Graph operations on nodes

In this section, we analyze the change in lattices of balanced equivalence relations when nodes are removed or/and added to networks (together with their edges). In Subsection 4.1, we consider the case of removing from a network a single node, together with all its edges. The accompanied algorithm is presented in Subsection 4.2 and a computational

example is included in Subsection 4.3. In Subsection 4.4, we extend our results to graph operations involving multiple nodes.

## 4.1 Deletion of a node

Recall that every balanced equivalence relation is a refinement of the input equivalence relation on the set of cells of the network. Thus, if there is no directed edge from the removed cell to the other cells, removing the cell with all its edges will not change the input relation between the other cells. We show that, in this case and in the case where the network is homogeneous, the lattice of the new network can be completely recovered from the initial one.

In what follows, we denote by “ $-$ ” the multiset subtraction; that is, given two multisets  $A, B$ , by “ $A - B$ ” we mean the multiset composed of elements  $x$  such that  $x \in A$  and  $x \notin B$  with multiplicity  $m(x, A - B) = m(x, A)$ . For the usual set subtraction, we continue to use “ $\setminus$ ”.

Let  $\mathcal{G}_1 = (C_1, \mathcal{E}_1, \sim_{C_1}, \sim_{E_1})$  be a coupled cell network and  $c_o \in C_1$ . Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by removing the node  $c_o$  together with all its edges. That is,  $\mathcal{G}_2 = (C_2, \mathcal{E}_2, \sim_{C_2}, \sim_{E_2})$ , where  $C_2 = C_1 \setminus \{c_o\}$  and  $\mathcal{E}_2 = \mathcal{E}_1 - R$  for  $R = \{(a, b) \in \mathcal{E}_1 : a = c_o \text{ or } b = c_o\}$ . For  $x \in C_2$ , we have

$$I_2(x) = \begin{cases} I_1(x), & \text{if } c_o \notin I_1(x), \\ I_1(x) - \{c_o\}, & \text{otherwise.} \end{cases} \quad (4.25)$$

**Definition 4.1** Let  $M_{\mathcal{G}_i}$  denote the set of all equivalence relations on  $C_i$ . Define two maps between the sets  $M_{\mathcal{G}_1}$  and  $M_{\mathcal{G}_2}$ :

$$\begin{array}{ll} \text{Proj} : M_{\mathcal{G}_1} \rightarrow M_{\mathcal{G}_2} & \text{Lift} : M_{\mathcal{G}_2} \rightarrow M_{\mathcal{G}_1} \\ \varkappa \mapsto \varkappa' & \varkappa' \mapsto \overline{\varkappa'} \end{array}$$

where  $\varkappa'$  is obtained from  $\varkappa$  in the following way,

$$[c]_{\varkappa'} = \begin{cases} [c]_{\varkappa} & , \text{ if } c_o \notin [c]_{\varkappa} \\ [c]_{\varkappa} \setminus \{c_o\} & , \text{ if } c_o \in [c]_{\varkappa} \end{cases} ,$$

and  $\overline{\varkappa'}$  is defined by

$$[c]_{\overline{\varkappa'}} = \begin{cases} [c]_{\varkappa'} & , \text{ if } c \neq c_o \\ \{c_o\} & , \text{ if } c = c_o \end{cases} .$$

◇

**Example 4.2** Let  $C_1 = \{1, 2, 3, 4\}$ ,  $c_o = 1$  and the equivalence relations  $\varkappa_1 = \{\{1, 2\}, \{3, 4\}\}$  and  $\varkappa_2 = \{\{2\}, \{1, 3, 4\}\}$  on  $C_1$ . Then  $\varkappa'_1 := \text{Proj}(\varkappa_1) = \text{Proj}(\varkappa_2) := \varkappa'_2 = \{\{2\}, \{3, 4\}\}$  and the lifting of  $\varkappa'_1$  (and so of  $\varkappa'_2$ ) is  $\{\{1\}, \{2\}, \{3, 4\}\}$ . ◇

**Remark 4.3** (i) Note that, while the lifting is an injective map, the projection is not, in general, injective.

(ii) If  $\bowtie \in M_{\mathcal{G}_1}$  then  $\overline{\bowtie} < \bowtie$ . Moreover, if  $[c_0]_{\bowtie} = \{c_0\}$  then  $\overline{\bowtie} = \bowtie$ .  $\diamond$

**Lemma 4.4** If  $\bowtie \in \Lambda_{\mathcal{G}_1}$  is such that  $\#[c_0]_{\bowtie} = 1$  then its projection  $\bowtie' := \text{Proj}(\bowtie) \in \Lambda_{\mathcal{G}_2}$ .

**Proof** Let  $\bowtie \in \Lambda_{\mathcal{G}_1}$  be such that  $\#[c_0]_{\bowtie} = 1$ . We need to show

$$m([\alpha]_{\bowtie'}, I_2(x)) = m([\alpha]_{\bowtie'}, I_2(y)), \quad \forall \alpha \in C_2,$$

whenever  $x, y \in C_2$  are such that  $x \bowtie' y$  and  $x \neq y$ . Let  $x, y \in C_2$  be such that  $x \bowtie' y$  and  $x \neq y$ . Then, by definition of projection, we have  $x \bowtie y$ . As  $\bowtie$  is balanced, we have

$$m([\alpha]_{\bowtie}, I_1(x)) = m([\alpha]_{\bowtie}, I_1(y)), \quad \forall \alpha \in C_1.$$

Due to  $\#[c_0]_{\bowtie} = 1$ , i.e.,  $[c_0]_{\bowtie} = \{c_0\}$ , it follows from the definition of projection that  $[\alpha]_{\bowtie'} = [\alpha]_{\bowtie}$  and  $c_0 \notin [\alpha]_{\bowtie}$  for all  $\alpha \in C_2$ . Consequently,

$$\begin{aligned} m([\alpha]_{\bowtie'}, I_2(x)) &= m([\alpha]_{\bowtie}, I_1(x) - \{c_0\}) = m([\alpha]_{\bowtie}, I_1(x)) = m([\alpha]_{\bowtie}, I_1(y)) \\ &= m([\alpha]_{\bowtie}, I_1(y) - \{c_0\}) = m([\alpha]_{\bowtie'}, I_2(y)), \quad \forall \alpha \in C_2. \end{aligned}$$

Therefore,  $\bowtie' \in \Lambda_{\mathcal{G}_2}$ .  $\square$

**Remark 4.5** For any  $\bowtie \in M_{\mathcal{G}_1}$ , we have  $\#[c_0]_{\overline{\bowtie}} = 1$ . Thus, if  $\overline{\bowtie} \in \Lambda_{\mathcal{G}_1}$ , then by Lemma 4.4, we have  $\text{Proj}(\overline{\bowtie}) = \text{Proj}(\bowtie) = \bowtie' \in \Lambda_{\mathcal{G}_2}$ .  $\diamond$

Recall that  $\sim_{I_1}$  and  $\sim_{I_2}$  denote the input equivalence relations on  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , respectively. Consider the equivalence relation  $\sim'_{I_1} := \text{Proj}(\sim_{I_1})$  on  $C_2$ . Thus:

$$[c]_{\sim'_{I_1}} = [c]_{\sim_{I_1}} \cap C_2, \quad \forall c \in C_2.$$

**Lemma 4.6** If  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}$  is such that  $\bowtie_2 < \sim'_{I_1}$  then its lift  $\overline{\bowtie_2} := \text{Lift}(\bowtie_2) \in \Lambda_{\mathcal{G}_1}$ .

**Proof** Let  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}$  be such that  $\bowtie_2 < \sim'_{I_1}$ . Let  $x, y \in C_2$  be such that  $x \neq y$  and  $x \overline{\bowtie_2} y$ . In particular,  $x, y \neq c_0$ . We show that

$$m([\alpha]_{\overline{\bowtie_2}}, I_1(x)) = m([\alpha]_{\overline{\bowtie_2}}, I_1(y)), \quad \forall \alpha \in C_1.$$

Note that, since  $x \overline{\bowtie_2} y$  and  $\overline{\bowtie_2}$  is balanced, we have

$$m([\alpha]_{\bowtie_2}, I_2(x)) = m([\alpha]_{\bowtie_2}, I_2(y)), \quad \forall \alpha \in C_2.$$

In the case  $\alpha \neq c_0$ , we have  $[\alpha]_{\overline{\bowtie_2}} = [\alpha]_{\bowtie_2}$ . Thus,

$$\begin{aligned} m([\alpha]_{\overline{\bowtie_2}}, I_1(x)) &= m([\alpha]_{\bowtie_2}, I_1(x) - \{c_0\}) = m([\alpha]_{\bowtie_2}, I_2(x)) = m([\alpha]_{\bowtie_2}, I_2(y)) \\ &= m([\alpha]_{\bowtie_2}, I_1(y) - \{c_0\}) = m([\alpha]_{\overline{\bowtie_2}}, I_1(y)). \end{aligned}$$

Otherwise,  $\alpha = c_0$ . Then,  $[\alpha]_{\overline{\bowtie_2}} = \{c_0\}$ . Thus,

$$m([c_0]_{\overline{\bowtie_2}}, I_1(x)) = m(\{c_0\}, I_1(x)) = m(\{c_0\}, I_1(x) - I_2(x))$$

and

$$m([c_o]_{\overline{\varkappa}_2}, I_1(y)) = m(\{c_o\}, I_1(y)) = m(\{c_o\}, I_1(y) - I_2(y)).$$

Note that  $\varkappa_2 < \sim_{I_2}$ , since every balanced equivalence relation on  $\mathcal{G}_2$  refines  $\sim_{I_2}$ . As  $\varkappa_2 < \sim'_{I_1}$  and so  $\overline{\varkappa}_2 < \sim_{I_1}$ , it follows that  $x \sim_{I_1} y$  and  $x \sim_{I_2} y$ . Thus, there is a bijection preserving edge equivalence relation between  $I_1(x) - I_2(x)$  and  $I_1(y) - I_2(y)$ . Therefore,

$$m(\{c_o\}, I_1(x) - I_2(x)) = m(\{c_o\}, I_1(y) - I_2(y))$$

and so

$$m([c_o]_{\overline{\varkappa}_2}, I_1(x)) = m([c_o]_{\overline{\varkappa}_2}, I_1(y)).$$

□

**Lemma 4.7** Let  $\varkappa \in \Lambda_{\mathcal{G}_1}$ . For its projection  $\varkappa' := \text{Proj}(\varkappa)$  and the corresponding lifting  $\overline{\varkappa'} := \text{Lift}(\varkappa')$ , we have

$$\varkappa' \in \Lambda_{\mathcal{G}_2} \Rightarrow \overline{\varkappa'} \in \Lambda_{\mathcal{G}_1}.$$

**Proof** Since  $\varkappa \in \Lambda_{\mathcal{G}_1}$ , we have  $\varkappa < \sim_{I_1}$ . It follows from the definition of projection that  $\varkappa' = \text{Proj}(\varkappa) < \sim'_{I_1} = \text{Proj}(\sim_{I_1})$ . Thus, by Lemma 4.6, we have  $\overline{\varkappa'} = \text{Lift}(\varkappa') \in \Lambda_{\mathcal{G}_1}$  if  $\varkappa' \in \Lambda_{\mathcal{G}_2}$ . □

**Example 4.8** Figure 7 shows a 4-cell network  $\mathcal{G}_1$  and the 3-cell network  $\mathcal{G}_2$  obtained from  $\mathcal{G}_1$  by removing cell 1. Note that  $\varkappa = \{\{1, 3\}, \{2, 4\}\} \in \Lambda_{\mathcal{G}_1}$  and  $\varkappa' = \{\{3\}, \{2, 4\}\} \in \Lambda_{\mathcal{G}_2}$ . Lemma 4.7 states that  $\overline{\varkappa'} = \{\{1\}, \{3\}, \{2, 4\}\} \in \Lambda_{\mathcal{G}_1}$ .

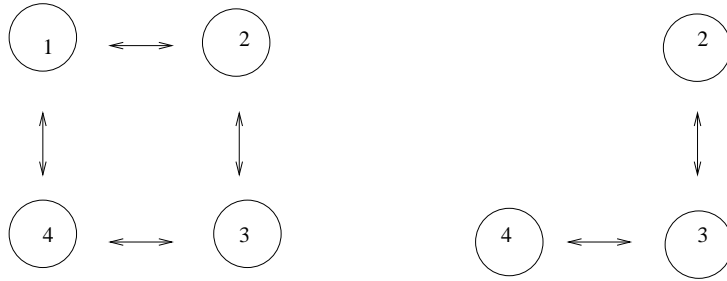


Figure 7: (Left) A 4-cell network  $\mathcal{G}_1$ . (Right) The 3-cell network  $\mathcal{G}_2$  is obtained from the network  $\mathcal{G}_1$  by removing cell 1.

◇

**Definition 4.9** A balanced equivalence relation  $\varkappa_2 \in \Lambda_{\mathcal{G}_2}$  is called *recoverable from*  $\Lambda_{\mathcal{G}_1}$ , if there exists  $\varkappa \in \Lambda_{\mathcal{G}_1}$  such that  $\text{Proj}(\varkappa) = \varkappa_2$ .

In what follows, we denote by  $\Lambda_{\mathcal{G}_2}^R$  the set of all balanced equivalence relations in  $\Lambda_{\mathcal{G}_2}$  that are recoverable from  $\Lambda_{\mathcal{G}_1}$ .

Returning to the networks of Example 4.8, we have that the balanced equivalence relation  $\varkappa' = \{\{3\}, \{2, 4\}\}$  of  $\mathcal{G}_2$  is recoverable from  $\Lambda_{\mathcal{G}_1}$ , since it is the projection of

the balanced equivalence relation  $\overline{\bowtie} = \{\{1\}, \{3\}, \{2, 4\}\} \in \Lambda_{\mathcal{G}_1}$ . We show in the next proposition that every recoverable equivalence relation of  $\mathcal{G}_2$  is the projection of a balanced equivalence relation  $\bowtie$  of  $\mathcal{G}_1$  with  $\#[c_o]_{\bowtie} = 1$ .

**Proposition 4.10** *Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by removing the cell  $c_o \in C_1$ . We have*

$$\Lambda_{\mathcal{G}_2}^R = \{ \text{Proj}(\bowtie) : \bowtie \in \Lambda_{\mathcal{G}_1} \text{ with } \#[c_o]_{\bowtie} = 1 \}.$$

**Proof** First note that, by Lemma 4.4, if  $\bowtie$  in  $\Lambda_{\mathcal{G}_1}$  with  $\#[c_o]_{\bowtie} = 1$ , then the projection  $\bowtie' = \text{Proj}(\bowtie)$  is in  $\Lambda_{\mathcal{G}_2}$  and so in  $\Lambda_{\mathcal{G}_2}^R$ .

Let  $\bowtie'$  be a recoverable balanced relation in  $\Lambda_{\mathcal{G}_2}^R$ . Since  $\bowtie'$  is recoverable, there exists  $\bowtie \in \Lambda_{\mathcal{G}_1}$  such that  $\text{Proj}(\bowtie) = \bowtie'$ . Thus, it follows from Lemma 4.7 that,  $\overline{\bowtie}$  is a balanced relation in  $\Lambda_{\mathcal{G}_1}$  with  $\#[c_o]_{\overline{\bowtie}} = 1$  and such that  $\text{Proj}(\overline{\bowtie}) = \bowtie'$ .  $\square$

**Remark 4.11** An alternative proof of Proposition 4.10 can be given more from an algebraic approach. We begin by recalling that a synchrony subspace for a general network is a polydiagonal space that is left invariant under all the network adjacency matrices, one for each edge type. See the discussion after Theorem 2.9. Without loss of generality, we can assume that  $\mathcal{G}_1$  and  $\mathcal{G}_2$  have only one edge type and that  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by deleting the first node. Let  $A_{\mathcal{G}_1}$  be the adjacency matrix of  $\mathcal{G}_1$  and  $\overline{A}_{\mathcal{G}_2}$  the matrix obtained from  $A_{\mathcal{G}_1}$  by making all the entries in the first row and in the first column equal to zero. Thus, if  $A_{\mathcal{G}_1} = [a_{ij}]$  then

$$A_{\mathcal{G}_1} = \overline{A}_{\mathcal{G}_2} + B$$

where  $B$  is the  $n \times n$  matrix

$$B = \left[ \begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline a_{21} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & 0 & \cdots & 0 \end{array} \right],$$

and  $\overline{A}_{\mathcal{G}_2}$  is given by (as a block matrix)

$$\overline{A}_{\mathcal{G}_2} = \left[ \begin{array}{c|ccc} 0 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & A_{\mathcal{G}_2} & \\ 0 & & & \end{array} \right],$$

where  $A_{\mathcal{G}_2}$  is the adjacency matrix of  $\mathcal{G}_2$ .

We start by observing that if two cells  $c, d \in C_2 = C_1 \setminus \{1\} = \{2, \dots, n\}$  are such that  $c \sim_{I_1} d$  and  $c \sim_{I_2} d$  then  $a_{c1} = a_{d1}$ . This follows from the definition of input relation applied to one edge type: the  $c, d$  row sums for both adjacency matrices  $A_{\mathcal{G}_1}$  and  $A_{\mathcal{G}_2}$  have to coincide.

Let  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}^R$ . We have that  $\bowtie_2$  is a balanced relation for  $\mathcal{G}_2$  such that  $\bowtie_2 = \text{Proj}(\bowtie_1)$  for a balanced relation  $\bowtie_1$  for  $\mathcal{G}_1$ . We have to show that in fact  $\bowtie_2 = \text{Proj}(\bowtie)$  for a balanced relation  $\bowtie$  for  $\mathcal{G}_1$  where  $[1]_{\bowtie} = \{1\}$ . Take the polydiagonal  $\Delta_{\bowtie_2} \subseteq \mathbb{R}^{n-1}$  associated with  $\bowtie_2$ . We know that it is of synchrony for  $\mathcal{G}_2$ . That is, it is  $A_{\mathcal{G}_2}$ -invariant. Take now the polydiagonal subspace of  $\mathbb{R}^n$  defined by

$$\Delta := \{(x_1, \dots, x_n) \in \mathbb{R}^n : (x_2, \dots, x_n) \in \Delta_{\bowtie_2}\}.$$

Trivially,  $\Delta$  is  $\bar{A}_{\mathcal{G}_2}$ -invariant. If we show now that  $\Delta$  is also  $B$ -invariant, it follows then that it is also  $A_{\mathcal{G}_1}$ -invariant, as  $A_{\mathcal{G}_1} = \bar{A}_{\mathcal{G}_2} + B$ . We conclude then that  $\Delta$  is a synchrony subspace for  $\mathcal{G}_1$  where, by construction of  $\Delta$ , we have  $\Delta = \Delta_{\bowtie}$  for the balanced relation  $\bowtie$  on  $C = \{1, \dots, n\}$  defined by:  $[1]_{\bowtie} = \{1\}$  and  $[c]_{\bowtie} = [c]_{\bowtie_2}$  for  $c \neq 1$ . Thus  $\bowtie_2 = \text{Proj}(\bowtie)$ .

It remains to show that  $\Delta$  is  $B$ -invariant. Let  $\mathbf{x} = (x_1, \dots, x_n) \in \Delta$ . If  $c \bowtie_2 d$  for  $c, d \neq 1$ , then  $x_c = x_d$ . By hypothesis  $\bowtie_2 = \text{Proj}(\bowtie_1)$  for a balanced relation  $\bowtie_1$  for  $\mathcal{G}_1$ . Thus  $c \bowtie_1 d$ . In particular, we have  $c \sim_{I_1} d$  and  $c \sim_{I_2} d$ , and as remarked above, we conclude that  $a_{c1} = a_{d1}$ . Now  $[B\mathbf{x}]_c = a_{c1}x_1$  and  $[B\mathbf{x}]_d = a_{d1}x_1$ , and so  $[B\mathbf{x}]_c = [B\mathbf{x}]_d$ .  $\diamond$

**Remark 4.12** For every  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}^R$ , the quotient network  $\mathcal{G}_2 / \bowtie_2$  is obtained from the quotient network  $\mathcal{G}_1 / \bowtie_1$ , with  $\bowtie_1 = \text{Lift}(\bowtie_2) \in \Lambda_{\mathcal{G}_1}$ , by removing the node  $[c_0]_{\bowtie_1}$  together with all its edges.

Obviously, in general, there can be balanced equivalence relations in  $\Lambda_{\mathcal{G}_2}$  that are not recoverable from the ones in  $\Lambda_{\mathcal{G}_1}$ .

**Example 4.13** Figure 8 shows an example of a 5-cell network  $\mathcal{G}_1$  and the 4-cell network  $\mathcal{G}_2$  obtained from  $\mathcal{G}_1$  by removing cell 1. In this case, as  $\sim_{I_1}$  is the trivial equivalence relation on  $C_1 = \{1, 2, 3, 4, 5\}$  with classes  $\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$ , we have that  $\Lambda_{\mathcal{G}_1} = \{\sim_{I_1}\}$ . Now,  $\mathcal{G}_2$  is homogeneous (regular) and every equivalence relation on  $C_2 = \{2, 3, 4, 5\}$  is balanced. Note that  $\mathcal{G}_2$  is  $\mathbf{S}_4$ -symmetric. Apart from the trivial relation on  $C_2$ , all the other relations in  $\Lambda_{\mathcal{G}_2}$  are not recoverable from the ones in  $\Lambda_{\mathcal{G}_1}$ .  $\diamond$

We note that in the previous example, where  $\Lambda_{\mathcal{G}_2} \neq \Lambda_{\mathcal{G}_2}^R$ , we have  $\sim_{I_2} \not\prec \sim'_{I_1}$ . Indeed, we have:

**Proposition 4.14** Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by removing the cell  $c_0 \in C_1$ . If

$$\sim_{I_2} \prec \sim'_{I_1}$$

then, every balanced relation in  $\Lambda_{\mathcal{G}_2}$  is recoverable from  $\Lambda_{\mathcal{G}_1}$ . That is,

$$\Lambda_{\mathcal{G}_2} = \Lambda_{\mathcal{G}_2}^R.$$



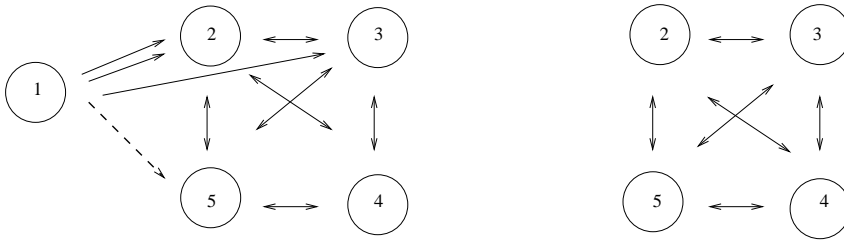


Figure 8: (Left) A 5-cell network  $\mathcal{G}_1$ . (Right) The 4-cell network  $\mathcal{G}_2$  is obtained from the network  $\mathcal{G}_1$  by removing cell 1 and it is  $\mathbf{S}_4$ -symmetric.

To show Proposition 4.14, we first prove a preliminary result.

**Proposition 4.15** *Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by removing the cell  $c_0 \in C_1$ . If  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}$  is such that  $\bowtie_2 < \sim'_{I_1}$  then  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}^R$ .*

**Proof** Let  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}$  be such that  $\bowtie_2 < \sim'_{I_1}$  and let  $\bowtie := \text{Lift}(\bowtie_2)$ . From the definition of lifting, we have  $\#[c_0]_{\bowtie} = 1$ . By Proposition 4.10, it suffices to prove that  $\bowtie \in \Lambda_{\mathcal{G}_1}$ . Since  $\bowtie_2$  is balanced on  $\mathcal{G}_2$  it follows, from Lemma 4.6, that  $\bowtie$  is balanced on  $\mathcal{G}_1$ . We have then  $\bowtie \in \Lambda_{\mathcal{G}_1}$ , with  $\#[c_0]_{\bowtie} = 1$  and  $\bowtie_2 = \text{Proj}(\bowtie)$ . Thus  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}^R$ .  $\square$

**Proof of Proposition 4.14** Let  $\bowtie' \in \Lambda_{\mathcal{G}_2}$ . Then,  $\bowtie' < \sim_{I_2}$ . Given the transitivity of the relation of refinement, as  $\sim_{I_2} < \sim'_{I_1}$ , we also have  $\bowtie' < \sim'_{I_1}$ . From Proposition 4.15, it follows that  $\bowtie' \in \Lambda_{\mathcal{G}_2}^R$ .  $\square$

**Corollary 4.16** *If  $\mathcal{G}_1$  is a homogeneous network, or there is no directed edge in  $\mathcal{G}_1$  from the removed cell  $c_0$  to the other cells in  $C_1$ , then  $\Lambda_{\mathcal{G}_2} = \Lambda_{\mathcal{G}_2}^R$ .*

**Proof** If  $\mathcal{G}_1$  is a homogeneous network, then  $\sim'_{I_1}$  is the equality relation on  $C_2$ , and so  $\sim_{I_2}$  refines  $\sim'_{I_1}$ . If there is no directed edge from the removed cell  $c_0$  in  $\mathcal{G}_1$  to the other cells, then  $\sim'_{I_1} = \sim_{I_2}$ . In both cases, by Proposition 4.14, the result follows.  $\square$

In the next proposition, we describe those balanced equivalence relations  $\bowtie' \in \Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R$ . Note that  $\bowtie' \not\prec \sim'_{I_1}$  if and only if

$$\exists a, b \in C_2 \text{ with } a \bowtie' b \quad \text{s.t.} \quad m(c_0, I_1^e(a)) \neq m(c_0, I_1^e(b)) \quad (4.26)$$

for some input type  $e$ .

**Proposition 4.17** *Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by removing the cell  $c_0 \in C_1$ . If*

$$\sim_{I_2} \not\prec \sim'_{I_1}$$

*then, the set of balanced relations in  $\Lambda_{\mathcal{G}_2}$  that are not recoverable from  $\Lambda_{\mathcal{G}_1}$  is given by*

$$\Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R = \left\{ \bowtie' \in \Lambda_{\mathcal{G}_2} : \bowtie' \not\prec \sim'_{I_1} \right\} = \left\{ \bowtie' \in \Lambda_{\mathcal{G}_2} : (4.26) \text{ is satisfied} \right\}.$$

**Proof** First suppose that  $\succ' \in \Lambda_{\mathcal{G}_2}$  is such that  $m(c_o, I_1^c(a)) = m(c_o, I_1^c(b))$  for all  $a, b \in C_2$  with  $a \succ' b$  and every input type  $e$ . Then,  $\succ' < \sim'_{I_1}$  and from Proposition 4.15 it follows that  $\succ'$  is in  $\Lambda_{\mathcal{G}_2}^R$ .

Now, if the equivalence relation  $\succ' \in \Lambda_{\mathcal{G}_2}$  is such that (4.26) holds for some input type  $e$ , then there can not exist  $\succ \in \Lambda_{\mathcal{G}_1}$  with  $\succ' := \text{Proj}(\succ)$  such that  $a \succ b$ , since otherwise, we would have  $\succ \not\prec \sim'_{I_1}$ , which is a contradiction. Thus  $\succ' \notin \Lambda_{\mathcal{G}_2}^R$ .  $\square$

## 4.2 Algorithm for deletion of a node

In the following, we present an algorithm that generates the lattice of balanced equivalence relations of  $\mathcal{G}_2$  based on that of  $\mathcal{G}_1$ , where  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by removing the cell  $c_o$ . As before, without loss of generality, we can assume  $\mathcal{G}_1$  has only one cell type and one edge type, since as shown in [2], the calculation of the lattice of synchrony subspaces for a general coupled cell network reduces to this particular kind of networks.

In the case  $\sim_{I_2} \not\prec \sim'_{I_1}$ , we define the following subset of  $C_2 \times C_2$ :

$$CI2 = \{(a, b) \in C_2 \times C_2 : a \sim_{I_2} b, a < b, m(c_o, I_1(a)) \neq m(c_o, I_1(b))\}.$$

For every  $(a, b) \in CI2$ , consider  $\succ_{(a,b)} \in M_{\mathcal{G}_2}$  defined by:

$$[a]_{\succ_{(a,b)}} = \{a, b\} \text{ and } \#[x]_{\succ_{(a,b)}} = 1, \forall x \neq a, b.$$

Take then

$$M_{(a,b)} = \{\succ \in \Lambda_{\mathcal{G}_2} : \succ_{(a,b)} < \succ\}.$$

It follows then from Proposition 4.17 that, if  $\sim_{I_2} \not\prec \sim'_{I_1}$ , then

$$\Lambda_{\mathcal{G}_2} \setminus \Lambda_{\mathcal{G}_2}^R = \bigcup_{(a,b) \in CI2} M_{(a,b)}.$$

**Algorithm 4.18** Let  $\mathcal{G}_1$  be a network having only one cell type and one edge type. Let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by deleting a node  $c_o \in C_1$  together with all its edges. Denote by  $A$  the adjacency matrix of the network  $\mathcal{G}_2$  whose eigenvalues  $\lambda_i$  with  $i = 1, \dots, t$  have algebraic and geometric multiplicities  $m_i^a$  and  $m_i^g$ , respectively.

- 1 Let  $\Lambda_{\mathcal{G}_2} := \{\text{Proj}(\succ) : \succ \in \Lambda_{\mathcal{G}_1} \text{ s.t. } \#[c_o]_{\succ} = 1\}$ .
- 2 If  $\sim_{I_2} < \sim'_{I_1}$ <sup>4</sup> then return  $\Lambda_{\mathcal{G}_2}$  and exit the algorithm. If  $\sim_{I_2} \not\prec \sim'_{I_1}$  and  $\mathcal{G}_2$  is regular then let  $\Lambda_{\mathcal{G}_2} := \Lambda_{\mathcal{G}_2} \cup \{C_2\}$ .<sup>5</sup>
- 3 Consider the subset  $CI2$  of  $C_2 \times C_2$ .

<sup>4</sup>The cases where  $\mathcal{G}_1$  is regular, or there is no directed edge in  $\mathcal{G}_1$  from the removed cell  $c_o$  to the other cells in  $C$  satisfy this condition. See Corollary 4.16.

<sup>5</sup>The equivalence relation with only the class  $C_2$  corresponds to the full synchronous subspace of  $\mathcal{G}_2$ .

4 For each  $(a, b) \in CI_2$

4.1 Consider the polydiagonal  $P := \{\mathbf{x} \in \mathbb{R}^{n-1} : x_a = x_b\}$ .

4.2 For each  $i = 1, \dots, t$ , consider the subspace  $J_{\lambda_i}^1 := E_{\lambda_i} \cap P$ .

4.3 If for all  $i = 1, \dots, t$ ,  $J_{\lambda_i}^1$  is the zero subspace then go to step 4.

4.4 Consider only the nonzero subspaces  $J_{\lambda_{i_j}}^1$ , say for  $j = 1, \dots, s$ .

4.5 Take  $J_{\lambda_{i_j}}^{p_{i_j}}$  according to (3.15), for  $j = 1, \dots, s$ .

4.6 Let  $\tilde{V}$  be the set of synchrony subspaces returned by Algorithm A.1 executed on  $A$  restricted to

$$J_{\lambda_{i_1}}^{p_{i_1}} \oplus \dots \oplus J_{\lambda_{i_s}}^{p_{i_s}}.$$

4.7 Let  $\tilde{\Lambda}$  be the set of balanced equivalence relations corresponding to  $\tilde{V}$ .

4.8 Let  $\Lambda_{\mathcal{G}_2} := \Lambda_{\mathcal{G}_2} \cup \tilde{\Lambda}$ .

5 Return  $\Lambda_{\mathcal{G}_2}$ .

◇

### 4.3 Example

**Example 4.19** Consider the networks  $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$  and  $\mathcal{G}_4$  given by Figure 9, where  $\mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4$  are obtained by successively removing nodes 6, 5 and 3 from  $\mathcal{G}_1$ . We start by

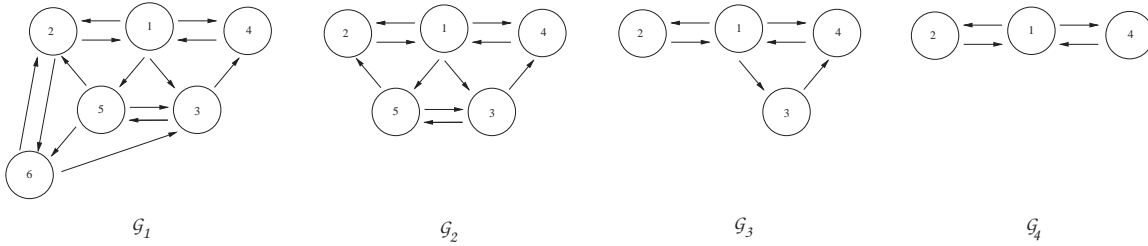


Figure 9: The network  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by removing the cell 6; the network  $\mathcal{G}_3$  is obtained from  $\mathcal{G}_2$  by removing the cell 5; and the network  $\mathcal{G}_4$  is obtained from  $\mathcal{G}_3$  by removing the cell 3.

generating the lattice  $\Lambda_{\mathcal{G}_1}$  using Algorithm 6.3 in [2]. Then, we obtain successively the lattice  $\Lambda_{\mathcal{G}_i}$  based on  $\Lambda_{\mathcal{G}_{i-1}}$  using Algorithm 4.18, for  $i = 2, 3, 4$ . The result is summarized in Tables 4– 6.

More precisely, the input equivalence relations of  $\mathcal{G}_i$ 's are listed below:

$$\sim_{I_1} = \{\{1, 4, 5, 6\}, \{2, 3\}\}, \quad \sim_{I_2} = \{\{1, 2, 3, 4, 5\}\}, \quad \sim_{I_3} = \{\{1, 4\}, \{2, 3\}\}, \quad \sim_{I_4} = \{\{1\}, \{2, 4\}\}.$$

|   |   |   |
|---|---|---|
| $\mathfrak{A}_0^1 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$ | $\mathfrak{A}_3^1 = \{\{1, 4\}, \{2, 3\}, \{5\}, \{6\}\}$ | $\mathfrak{A}_6^1 = \{\{1, 4, 5\}, \{2, 3\}, \{6\}\}$ |
| $\mathfrak{A}_1^1 = \{\{1\}, \{2, 3\}, \{4\}, \{5\}, \{6\}\}$     | $\mathfrak{A}_4^1 = \{\{1\}, \{2, 3\}, \{4, 5\}, \{6\}\}$ | $\mathfrak{A}_7^1 = \{\{1, 6\}, \{2, 3\}, \{4, 5\}\}$ |
| $\mathfrak{A}_2^1 = \{\{1\}, \{2\}, \{3\}, \{4, 5\}, \{6\}\}$     | $\mathfrak{A}_5^1 = \{\{1, 6\}, \{2\}, \{3\}, \{4, 5\}\}$ | $\mathfrak{A}_8^1 = \{\{1, 4, 5, 6\}, \{2, 3\}\}$     |

Table 4: The lattice of balanced relations for the network  $\mathcal{G}_1$  of Figure 9.

|  |  |  |   |
|--|--|--|---|
| $\mathfrak{A}_0^2 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$ | $\mathfrak{A}_4^2 = \{\{1\}, \{2, 3, 5\}, \{4\}\}$ | $\mathfrak{A}_8^2 = \{\{1\}, \{2, 3\}, \{4, 5\}\}$ | $\mathfrak{A}_{12}^2 = \{\{1, 2, 3\}, \{4, 5\}\}$ |
| $\mathfrak{A}_1^2 = \{\{1\}, \{2, 3\}, \{4\}, \{5\}\}$     | $\mathfrak{A}_5^2 = \{\{1\}, \{2\}, \{3, 4, 5\}\}$ | $\mathfrak{A}_9^2 = \{\{1\}, \{2, 4\}, \{3, 5\}\}$ | $\mathfrak{A}_{13}^2 = \{\{1, 4, 5\}, \{2, 3\}\}$ |
| $\mathfrak{A}_2^2 = \{\{1\}, \{2\}, \{3, 5\}, \{4\}\}$     | $\mathfrak{A}_6^2 = \{\{1, 2\}, \{3\}, \{4, 5\}\}$ | $\mathfrak{A}_{10}^2 = \{\{1, 2\}, \{3, 4, 5\}\}$  | $\mathfrak{A}_{14}^2 = \{\{1\}, \{2, 3, 4, 5\}\}$ |
| $\mathfrak{A}_3^2 = \{\{1\}, \{2\}, \{3\}, \{4, 5\}\}$     | $\mathfrak{A}_7^2 = \{\{1, 4\}, \{2, 3\}, \{5\}\}$ | $\mathfrak{A}_{11}^2 = \{\{1, 4\}, \{2, 3, 5\}\}$  | $\mathfrak{A}_{15}^2 = \{\{1, 2, 3, 4, 5\}\}$     |

Table 5: The lattice of balanced relations for the network  $\mathcal{G}_2$  of Figure 9.

|  |  |
|--|--|
| $\Lambda_{\mathcal{G}_3} = \{\mathfrak{A}_0^3, \mathfrak{A}_1^3, \mathfrak{A}_2^3\}$ | $\Lambda_{\mathcal{G}_4} = \{\mathfrak{A}_0^4, \mathfrak{A}_1^4\}$ |
| $\mathfrak{A}_0^3 = \{\{1\}, \{2\}, \{3\}, \{4\}\}$                                  | $\mathfrak{A}_0^4 = \{\{1\}, \{2\}, \{4\}\}$                       |
| $\mathfrak{A}_1^3 = \{\{1\}, \{2, 3\}, \{4\}\}$                                      | $\mathfrak{A}_1^4 = \{\{1\}, \{2, 4\}\}$                           |
| $\mathfrak{A}_2^3 = \{\{1, 4\}, \{2, 3\}\}$  |  |

Table 6: The lattices of balanced relations for the networks  $\mathcal{G}_3$  and  $\mathcal{G}_4$  of Figure 9.

and their corresponding projections are given by

$$\sim'_{I_1} = \{\{1, 4, 5\}, \{2, 3\}\}, \quad \sim'_{I_2} = \{\{1, 2, 3, 4\}\}, \quad \sim'_{I_3} = \{\{1, 4\}, \{2\}\}.$$

Note that we have

$$\sim_{I_2} \not\sim \sim'_{I_1}, \quad \sim_{I_3} < \sim'_{I_2}, \quad \text{and} \quad \sim_{I_4} \not\sim \sim'_{I_3}.$$

Consider the network  $\mathcal{G}_2$  as obtained from  $\mathcal{G}_1$  by removing the node 6. Since  $\sim_{I_2} \not\sim \sim'_{I_1}$ , there can be balanced equivalence relations on  $\mathcal{G}_2$  that are not recoverable from  $\mathcal{G}_1$  (cf.

Proposition 4.17). At step 1 of Algorithm 4.18,  $\Lambda_{\mathcal{G}_2}$  is set to be the set of recoverable equivalence relations on  $\mathcal{G}_2$  given by (cf. Proposition 4.10)

$$\begin{aligned}\Lambda_{\mathcal{G}_2}^R &= \{\varkappa_0^2, \varkappa_1^2, \varkappa_3^2, \varkappa_7^2, \varkappa_8^2, \varkappa_{13}^2\} \\ &= \left\{ \text{Proj}(\varkappa_0^1), \text{Proj}(\varkappa_1^1), \text{Proj}(\varkappa_2^1), \text{Proj}(\varkappa_3^1), \text{Proj}(\varkappa_4^1), \text{Proj}(\varkappa_6^1) \right\}.\end{aligned}$$

At step 2, since  $\sim_{I_2} \not\prec \sim'_{I_1}$  and  $\mathcal{G}_2$  is regular, let  $\Lambda_{\mathcal{G}_2} := \Lambda_{\mathcal{G}_2} \cup \{\varkappa_{15}^2\}$ . At step 3, the following set

$$CI2 = \{(1, 2), (1, 3), (2, 4), (2, 5), (3, 4), (3, 5)\},$$

is considered. For every  $\varkappa_{(a,b)} \in M_{(a,b)}$  with  $(a, b) \in CI2$ , the step 4 then finds all balanced equivalence relations  $\varkappa \in \Lambda_{\mathcal{G}_2}$  such that  $\varkappa_{(a,b)} \prec \varkappa$ . Altogether we obtain Table 5.

Next, consider the network  $\mathcal{G}_3$  which is obtained from  $\mathcal{G}_2$  by removing the node 5. Since  $\sim_{I_3} \prec \sim'_{I_2}$ , we have (cf. Proposition 4.14)

$$\Lambda_{\mathcal{G}_3} = \Lambda_{\mathcal{G}_3}^R = \{\varkappa_0^3, \varkappa_1^3, \varkappa_2^3\} = \left\{ \text{Proj}(\varkappa_0^2), \text{Proj}(\varkappa_1^2), \text{Proj}(\varkappa_7^2) \right\}$$

and Algorithm 4.18 exits at step 2. For the network  $\mathcal{G}_4$  which is obtained from  $\mathcal{G}_3$  by removing the node 3, the recoverable equivalence relations are given by

$$\Lambda_{\mathcal{G}_4}^R = \{\varkappa_0^4\} = \left\{ \text{Proj}(\varkappa_0^3) \right\}.$$

Since  $\sim_{I_4} \not\prec \sim'_{I_3}$ , Algorithm 4.18 goes on to step 3 considering

$$CI2 = \{(2, 4)\}.$$

Using  $\varkappa_{(2,4)} = \{\{1\}, \{2, 4\}\}$ , step 4 then finds an additional balanced equivalence relation  $\{\{1\}, \{2, 4\}\}$  on  $\mathcal{G}_4$  (cf. Table 6).  $\diamond$

## 4.4 Adding a node

The approach of Subsection 4.1 is completely applicable to the case of adding a node in the following sense. As before, consider that the network  $\mathcal{G}_1$  is obtained from the network  $\mathcal{G}_2$  by adding a node  $c_0$  together with some edges from/to  $c_0$ . Then, a balanced equivalence relation  $\varkappa \in \Lambda_{\mathcal{G}_1}$  is called *recoverable from  $\Lambda_{\mathcal{G}_2}$* , if there exists  $\varkappa_2 \in \Lambda_{\mathcal{G}_2}$  such that  $\varkappa = \text{Lift}(\varkappa_2)$ . Denote by  $\Lambda_{\mathcal{G}_1}^R$  the set of all balanced equivalence relations in  $\Lambda_{\mathcal{G}_1}$  that are recoverable from  $\Lambda_{\mathcal{G}_2}$ . By Lemma 4.6, if  $\varkappa_2 \in \Lambda_{\mathcal{G}_2}$  is such that  $\varkappa_2 \prec \sim'_{I_1}$  then its lift  $\overline{\varkappa_2} := \text{Lift}(\varkappa_2) \in \Lambda_{\mathcal{G}_1}$ . On the other hand, if  $\varkappa_2 \in \Lambda_{\mathcal{G}_2}$  is such that  $\varkappa_2 \not\prec \sim'_{I_1}$ , then  $\text{Lift}(\varkappa_2) \not\prec \text{Lift}(\sim'_{I_1})$ . Note that any refinement  $\sim$  of  $\sim'_{I_1}$  such that  $\#[c_0]_{\sim} = 1$  must refine  $\text{Lift}(\sim'_{I_1})$ , we conclude that  $\text{Lift}(\varkappa_2)$  cannot be a balanced relation for  $\mathcal{G}_1$ . It follows then from this discussion that

$$\Lambda_{\mathcal{G}_1}^R = \left\{ \text{Lift}(\varkappa_2) : \varkappa_2 \in \Lambda_{\mathcal{G}_2} \wedge \varkappa_2 \prec \sim'_{I_1} \right\}, \quad (4.27)$$

which is a parallel of Proposition 4.10. In fact,  $\Lambda_{\mathcal{G}_2}^R$  (in case of deleting a node) can be written as

$$\Lambda_{\mathcal{G}_2}^R = \{ \text{Proj}(\bowtie) : \bowtie \in \Lambda_{\mathcal{G}_1} \wedge \bowtie \prec \text{Lift}(\sim_{I_2}) \} \quad (4.28)$$

Moreover, in analogue to Proposition 4.14, we have

$$\sim_{I_1} \prec \text{Lift}(\sim_{I_2}) \quad \Rightarrow \quad \Lambda_{\mathcal{G}_1} = \Lambda_{\mathcal{G}_1}^R. \quad (4.29)$$

Indeed, for any  $\bowtie_1 \in \Lambda_{\mathcal{G}_1}$ , we have  $\bowtie_1 \prec \sim_{I_1} \prec \text{Lift}(\sim_{I_2})$  and thus  $\#[c_o]_{\bowtie_1} = 1$ . In particular, we have  $\bowtie_1 = \text{Lift}(\bowtie_2)$  for  $\bowtie_2 := \text{Proj}(\bowtie_1)$ , where  $\bowtie_2$  is balanced by Lemma 4.4. Also since  $\bowtie_1 \prec \sim_{I_1}$ , we have  $\bowtie_2 = \text{Proj}(\bowtie_1) \prec \sim'_{I_1}$ . Therefore,  $\bowtie_1 \in \Lambda_{\mathcal{G}_1}^R$ .

## 4.5 Graph operations on multiple nodes

We extend our results now to the cases of deletion or addition of several nodes, and both the addition and deletion of several nodes.

### 4.5.1 Deleting multiple nodes

One can show that Propositions 4.10 and 4.14 remain valid for deletion of multiple nodes. To this end, one only needs to adjust the definition of projection and use the same definition of recoverable relations. For a network  $\mathcal{G}_2$  obtained from  $\mathcal{G}_1$  by removing nodes  $\{c_1, c_2, \dots, c_s\}$  together with all their edges, the recoverable relations are precisely

$$\Lambda_{\mathcal{G}_2}^R = \{ \text{Proj}(\bowtie) : \bowtie \in \Lambda_{\mathcal{G}_1} \text{ s.t. } [c_i]_{\bowtie} \subset \{c_1, c_2, \dots, c_s\} \forall i = 1, 2, \dots, s \}.$$

**Example 4.20** In Example 4.19, the network  $\mathcal{G}_4$  can be viewed as obtained from  $\mathcal{G}_2$  by removing the nodes 3, 5 and we can recover  $\Lambda_{\mathcal{G}_4}$  using  $\Lambda_{\mathcal{G}_2}$ . If  $\sim''_{I_2}$  denotes the projection of  $\sim_{I_2}$  to  $\mathcal{G}_4$ , then  $\sim_{I_4} \prec \sim''_{I_2}$ . In this case of  $\mathcal{G}_4$ , all balanced equivalence relations are recoverable from  $\mathcal{G}_2$  by

$$\Lambda_{\mathcal{G}_4} = \{ \text{Proj}(\bowtie) : \bowtie \in \Lambda_{\mathcal{G}_2} \text{ s.t. } [3]_{\bowtie} \subset \{3, 5\} \wedge [5]_{\bowtie} \in \{3, 5\} \}.$$

◇

### 4.5.2 Adding multiple nodes

We can extend the discussion at Subsection 4.4 to the case of addition of multiple nodes in the following way. Let  $\mathcal{G}_1$  be a network obtained from  $\mathcal{G}_2$  by adding nodes  $\{c_1, c_2, \dots, c_s\}$  together with some edges from/to  $c_i$ 's. Define the *lift* of  $\bowtie_2 \in M_{\mathcal{G}_2}$ , denoted by  $\overline{\bowtie}_2$ , where  $[c_i]_{\overline{\bowtie}_2} = \{c_i\}$ , for  $i = 1, \dots, s$  and  $[x]_{\overline{\bowtie}_2} = [x]_{\bowtie_2}$  for  $x \in C_2$ . The *projection* can be canonically extended. The definition of *recoverable* relations then follows. It can be directly verified that Lemma 4.6 and thus (4.27)-(4.29) remain valid for the above defined lift and projection.

### 4.5.3 Deleting a node and adding another one

Let  $\mathcal{G}_1, \mathcal{G}, \mathcal{G}_2$  be related by

$$\mathcal{G}_1 \xrightarrow{\text{del. } c_0} \mathcal{G} \xrightarrow{\text{add. } a_0} \mathcal{G}_2,$$

i.e.,  $\mathcal{G}_2$  is obtained from  $\mathcal{G}_1$  by first deleting a node  $c_0$  and then adding a node  $a_0$ . Denote by  $\text{Proj}_{c_0}$  (resp.  $\text{Proj}_{a_0}$ ) and  $\text{Lift}_{c_0}$  (resp.  $\text{Lift}_{a_0}$ ) the projection and lifting maps related to the deletion (resp. the addition) operation. A balanced equivalence relation  $\bowtie_2 \in \Lambda_{\mathcal{G}_2}$  is called *recoverable from  $\mathcal{G}_1$* , if there exists  $\bowtie \in \Lambda_{\mathcal{G}}^R$  such that  $\bowtie_2 = \text{Lift}_{a_0}(\bowtie)$ . Recall that  $\Lambda_{\mathcal{G}}^R$  denotes the set of balanced relations on  $\mathcal{G}$  recoverable from  $\mathcal{G}_1$ , and  $\Lambda_{\mathcal{G}_2}^R$  denotes the set of balanced relations on  $\mathcal{G}_2$  recoverable from  $\mathcal{G}$ . We denote by  $\Lambda_{\mathcal{G}_2}^{R, \mathcal{G}_1}$  the set of balanced relations on  $\mathcal{G}_2$  that are recoverable from  $\mathcal{G}_1$ . Using our previous results (4.27)-(4.28), we have

$$\Lambda_{\mathcal{G}}^R = \{ \text{Proj}_{c_0}(\bowtie_1) : \bowtie_1 \in \Lambda_{\mathcal{G}_1} \wedge \bowtie_1 < \text{Lift}_{c_0}(\sim_I) \}$$

and

$$\Lambda_{\mathcal{G}_2}^R = \{ \text{Lift}_{a_0}(\bowtie) : \bowtie \in \Lambda_{\mathcal{G}} \wedge \bowtie < \text{Proj}_{a_0}(\sim_{I_2}) \}.$$

Consequently, we have

$$\begin{aligned} \Lambda_{\mathcal{G}_2}^{R, \mathcal{G}_1} &= \{ \text{Lift}_{a_0}(\bowtie) : \bowtie \in \Lambda_{\mathcal{G}}^R \wedge \bowtie < \text{Proj}_{a_0}(\sim_{I_2}) \} \\ &= \{ \text{Lift}_{a_0}(\text{Proj}_{c_0}(\bowtie_1)) : \bowtie_1 \in \Lambda_{\mathcal{G}_1} \wedge \bowtie_1 < \text{Lift}_{c_0}(\sim_I) \wedge \text{Proj}_{c_0}(\bowtie_1) < \text{Proj}_{a_0}(\sim_{I_2}) \}. \end{aligned} \quad (4.30)$$

### 4.5.4 Deleting and adding multiple nodes

The general case of multiple deletion and addition of nodes can be analyzed using the above discussion. More precisely, let  $\mathcal{G}_2$  be the network obtained from  $\mathcal{G}_1$  by deleting nodes  $\{c_1, c_2, \dots, c_s\}$  and adding nodes  $\{a_1, a_2, \dots, a_r\}$  (together with some edges from or to  $a_i$ 's). If  $\text{Proj}_c$  and  $\text{Lift}_c$  (resp.  $\text{Proj}_a$  and  $\text{Lift}_a$ ) denote the projection and lift related to the deletion (resp. addition) operation, then we have (similar to (4.30))

$$\Lambda_{\mathcal{G}_2}^{R, \mathcal{G}_1} = \{ \text{Lift}_a(\text{Proj}_c(\bowtie_1)) : \bowtie_1 \in \Lambda_{\mathcal{G}_1} \wedge \bowtie_1 < \text{Lift}_c(\sim_I) \wedge \text{Proj}_c(\bowtie_1) < \text{Proj}_a(\sim_{I_2}) \}.$$

## Acknowledgements

HR thanks the University of Porto for its hospitality and acknowledges additional support from the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT - Fundação para a Ciência e a Tecnologia under the project PTDC/MAT/100055/2008.

## References

- [1] M. Aguiar, P. Ashwin, A. Dias, and M. Field. Dynamics of coupled cell networks: synchrony, heteroclinic cycles and inflation. *J. Nonlinear Sci.* **21** (2) (2011) 271–323.
- [2] M.A.D. Aguiar and A.P.S. Dias. Synchrony in Coupled Cell Networks. *Preprint* 2012.
- [3] M.A.D. Aguiar and A.P.S. Dias. Synchrony in Product Coupled Cell Networks. *In preparation* 2013.
- [4] M.A.D. Aguiar and H. Ruan. Evolution of Synchrony under Combination of Coupled Cell Networks. *Nonlinearity* **25** (2012) 3155–3187.
- [5] J. W. Aldis. A polynomial time algorithm to determine maximal balanced equivalence relations. *J. Bifur. Chaos Appl. Sci. Engrg.* **18** (2) (2008) 407–427.
- [6] F. M. Atay and T. Biyikoglu. Graph operations and synchronization of complex networks. *Physical Review E* **72** (2005) 016217.
- [7] G. Chen and Z. Duan. Network synchronizability analysis: A graph-theoretic approach. *Chaos* **18** (2008) 037102.
- [8] A. P. S. Dias and I. Stewart. Linear Equivalence and ODE-equivalence for Coupled Cell Networks. *Nonlinearity* **18** (2005) 1003–1020.
- [9] M. Field. Combinatorial dynamics. *Dyn. Syst.* **19** (3) (2004) 217–243.
- [10] M. Golubitsky and I. Stewart. Nonlinear dynamics of networks: the groupoid formalism. *Bull. Amer. Math. Soc.* **43** (3) (2006) 305–364.
- [11] M. Golubitsky, I. Stewart and A. Török. Patterns of Synchrony in Coupled Cell Networks with Multiple Arrows. *SIAM J. Appl. Dynam. Sys.* **4** (1) (2005) 78–100.
- [12] A. Hagberg and D.A. Schult. Rewiring networks for synchronization. *Chaos* **18** 037105 (2008) 1–7.
- [13] H. Kamei. Construction of lattices of balanced equivalence relations for regular homogeneous networks using lattice generators and lattices indices. *Int. J. Bifur. Chaos Appl. Sci. Engrg.* **19** (11) (2009) 3691–3705.
- [14] H. Kamei and P. J. A. Cock. Computation of Balanced Equivalence Relations and Their Lattice for a Coupled Cell Network. *SIAM J. Appl. Dynam. Sys.* **12** (1) (2013) 352–382.
- [15] W. Lu, F. M. Atay and J. Jost. Synchronization of Discrete-Time Dynamical Networks with Time-Varying Couplings. *SIAM J. Math. Anal.* **39** (4) (2007) 1231–1259.
- [16] N. M. Luscombe, M. M. Babu, H. Yu, M. Snyder, S. A. Teichmann, M. Gerstein. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature* **431** (2004) 308–312. doi:10.1038/nature02782



- [17] I. Stewart. The lattice of balanced equivalence relations of a coupled cell network. *Math. Proc. Cambridge Philos. Soc.* **143** (1) (2007) 165–183.
- [18] I. Stewart, M. Golubitsky and M. Pivato. Symmetry groupoids and patterns of synchrony in coupled cell networks. *SIAM J. Appl. Dynam. Sys.* **2** (4) (2003) 609–646.
- [19] B. Zhang, H. Li, R. B. Riggins, M. Zhan, J. Xuan, Z. Zhang, E. P. Hoffman, R. Clarke and Y. Wang. Differential dependency network analysis to identify condition-specific topological changes in biological networks. *Bioinformatics* **25** (4) (2009) 526–532. doi: 10.1093/bioinformatics/btn660

## A Adaptation of Algorithm 6.3 in [2]

Algorithm 6.3 in [2] obtains the lattice of nontrivial synchrony subspaces of a regular  $n$ -cell network  $\mathcal{G}$ . It has as input the adjacency matrix  $A$  of a regular  $n$ -cell network  $\mathcal{G}$ , with valency  $v$ . If  $\lambda_1$  is the eigenvalue  $v$  of  $A$  with algebraic multiplicity 1, then it proceeds considering the eigenspaces and generalized eigenspaces of the other eigenvalues, as it is known in advance that the eigenspace corresponding to the valency  $v$  corresponds to the trivial full synchronous polydiagonal space. (If the adjacency matrix  $A$  of  $\mathcal{G}$  has complex eigenvalues, the calculations are done on  $\mathbf{C}^n$ , that is, interpreting  $A$  as  $A_c : \mathbf{C}^n \rightarrow \mathbf{C}^n$ .)

We give now the adaptation of this algorithm to our setup. Let  $A$  be a  $n \times n$  square matrix with real entries, obtained from an adjacency matrix of a not necessarily regular  $n$ -cell network with set of cells  $C$ . For  $c, c_0 \in C$ , where  $c \neq c_0$  and  $c$  belongs to the  $\sim_I$ -class of  $c_0$ , consider the polydiagonal  $P = \{\mathbf{x} : x_{c_0} = x_c\}$ . We are interested in executing the adaptation of Algorithm 6.3 [2], considering  $A$  restricted to the largest  $A$ -invariant subspace of  $\mathbb{R}^n$  that has nonzero intersection with  $P$ , finding then the polydiagonal subspaces that are left invariant under this restricted linear map. The changes we have so to do in Algorithm 6.3 [2] are to guarantee that at each step, all the sets of conditions that can give rise to a synchrony subspace include the condition  $x_{c_0} = x_c$ . This is done to avoid considering artificial possibilities that we know in advance will not give rise to a synchrony subspace.

**Algorithm A.1** Let  $A$  be an  $n \times n$  matrix with real entries. Let  $\lambda_i$  with  $i = 1, \dots, t$  and  $t \leq n$  be the eigenvalues of  $A$ , with  $m_i^a$  and  $m_i^g$ , respectively, the algebraic and geometric multiplicities. Let  $P := \{\mathbf{x} : x_{c_0} = x_c\}$ . If  $(1, \dots, 1)$  is an eigenvector of  $A$  corresponding to an eigenvalue of  $A$  (the valency  $v$  of the network) with algebraic multiplicity 1, then don't consider at the above set of eigenvalues that eigenvalue.

1 [Find polydiagonals] For each eigenvalue  $\lambda_i, i = 1, \dots, t$  of  $A$ :

- 1.1 Let  $(\mathbf{v}_1, \dots, \mathbf{v}_{m_i^g})$  be a basis of  $E_{\lambda_i}$ . Consider the matrix  $M$  whose columns correspond to the eigenvectors  $\mathbf{v}_1, \dots, \mathbf{v}_{m_i^g}$ .
- 1.2 Let  $C = \emptyset$ . For every pair of rows  $l_j, l_k$  of  $M$ :

- 1.2.1 If  $l_j = l_k$  then  $C = C \cup \{x_j = x_k\}$  and eliminate row  $l_k$  of  $M$ .<sup>6</sup>
- 1.3 Construct a four-column table for  $E_{\lambda_i}$  with a row containing: in the first entry the set of the equality conditions  $C$  found in step 1.2; in the second entry the corresponding polydiagonal dimension; in the third entry the basis of eigenspace  $E_{\lambda_i}$ ; in the fourth entry the number of vectors of the basis,  $m_i^g$ .
- 1.4 Let  $s$  be the number of remaining rows in  $M$ . Construct a new matrix  $\overline{M}$  with rows given by  $r_j - r_k$  for  $j = 1, \dots, s$  and  $k = j + 1, \dots, s$  where  $r_j, r_k$  are rows in  $M$ . Thus  $\overline{M}$  has  $\overline{s} = s(s - 1)/2$  rows, each corresponding to an equality  $x_j = x_k$  with  $j, k \in \{1, \dots, n\}$ .
- 1.5 Let  $S$  be the set of all the submatrices of  $\overline{M}$  with  $s - 2$  rows obtained from  $\overline{M}$  by elimination of rows.<sup>7</sup>
- 1.6 While  $S \neq \emptyset$ ,
- 1.6.1 Let  $N$  be a submatrix in  $S$  and  $S = S \setminus \{N\}$ .
- 1.6.2 Let  $r$  be the rank of  $N$ ;
- 1.6.3 If  $r < m_i^g$  then:
- 1.6.3.1 Let  $C_N$  be the set of equalities given by the rows of  $N$  and  $C$  be the set of equalities obtained in step 1.2. If there is no row in the table of  $E_{\lambda_i}$  corresponding to set of equalities  $C \cup C_N$  then add a new row to the table containing: in the first entry  $C \cup C_N$ ; in the second entry the corresponding polydiagonal dimension; in the third entry a basis of the subspace of the eigenvectors in  $E_{\lambda_i}$  that satisfy the set of equality conditions (obtained from the solution set of the homogeneous system with the coefficient matrix  $N$ ), and in the fourth entry the number of vectors of the basis,  $m_i^g - r$ .
- 1.6.4 Otherwise,  $r = m_i^g$ :
- 1.6.4.1 Consider the set  $S_N$  of all the submatrices of  $N$  obtained by eliminating one row of  $N$ .
- 1.6.4.2 Let  $S = S \cup S_N$ .
- 1.7 If  $m_i^g < m_i^a$  then:
- 1.7.1 Compute a basis of  $\text{Im}(A - \lambda_i \text{Id}_n)$ .
- 1.7.2 For each row in the table for  $E_{\lambda_i}$ ,
- 1.7.2.1 If the intersection of the subspace corresponding to the basis in that row with  $\text{Im}(A - \lambda_i \text{Id}_n)$  is a nonzero subspace then:  
Let  $B_1$  be a basis of that intersection;  
Let  $C$  be the first entry of the row (the set of equality conditions);

<sup>6</sup>If rows  $j$  and  $k$  of  $M$  are equal that means that  $x_j = x_k$  for all vectors  $\mathbf{x} = (x_1, \dots, x_n) \in E_{\lambda_i}$  and we can eliminate one of those rows. In particular, rows  $c_0$  and  $c$  are equal and so  $x_{c_0} = x_c \in C$ .

<sup>7</sup>Each row  $r_j - r_k$  of the matrix  $\overline{M}$  in step 1.4 corresponds to a set of coordinate equalities of the form  $x_j = x_k$  for  $j \neq k$ . So, each submatrix of  $\overline{M}$  in step 1.5 corresponds to a system of coordinate equalities. The reason to consider only submatrices of  $\overline{M}$  with rows up to  $s - 2$  is that the minimum dimension for any nontrivial polysynchronous subspace is 2 and there are  $s$  independent coordinate variables.

2 [Find sum-dense set] <sup>8</sup> Consider the empty set  $S$ . For each table, for each row of the table:

2.1 Let  $C$  be the set of equality conditions in that row and  $d$  the dimension of the polydiagonal subspace  $\Delta_{\bowtie}$  given by those conditions.

2.2 If the number of vectors in that row of the table equals  $d$ , or equals  $d - 1$  and  $(1, \dots, 1)$  is an eigenvector of  $A$ , <sup>9</sup> then there is an eigenvector basis of  $\Delta_{\bowtie}$  and thus  $\Delta_{\bowtie}$  is a synchrony subspace. Let  $S = S \cup \{\Delta_{\bowtie}\}$ .

2.3 If the number of vectors in that row of the table is lower than  $d - 1$ , or it is  $d - 1$  and  $(1, \dots, 1)$  is not an eigenvector of  $A$ , and there are more tables, then look at the other tables to find all the rows whose equality conditions include the set  $C$  of equality conditions.

2.3.1 If the total sum of the number of vectors equals  $d$ , or equals  $d - 1$  and  $(1, \dots, 1)$  is an eigenvector of  $A$ , then there is an eigenvector basis of  $\Delta_{\bowtie}$  and thus  $\Delta_{\bowtie}$  is a synchrony subspace. Let  $S = S \cup \{\Delta_{\bowtie}\}$ .

2.3.2 If the total sum of the number of vectors is still less than  $d - 1$ , or it is  $d - 1$  and  $(1, \dots, 1)$  is not an eigenvector of  $A$  then:

2.3.2.1 Eliminate that row of the table.

2.3.2.2 Let  $c = \#C$ . For each subset of  $c - 1$  conditions containing the condition  $x_{c_0} = x_c$  of the initial set  $C$  of  $c$  conditions:

If there is no row at the table with that set of  $c - 1$  conditions then add a new row to the end of the table differing from the deleted row only at the first and second entries: the first entry contains the set of the  $c - 1$  conditions and the second entry is  $n - c + 1$ , the dimension of the corresponding polydiagonal.

Otherwise, change the corresponding row: replacing the third entry by a basis of the subspace generated by the union of the bases in this row and the one in the deleted row; changing the fourth entry by the number of vectors of that basis. Move that row to the end of the table.

3 [Find the irreducible sum-dense set] Decompose  $S$  into the disjoint union  $\cup_{i=1}^r S_{j_i}$ , where each set  $S_{j_i}$  contains the synchrony subspaces in  $S$  of dimension  $j_i$ , with  $j_{i-1} < j_i$ , for  $i = 2, \dots, r$ . Let  $\mathcal{I}_{\mathcal{G}} = S_{j_1}$ .

3.1 For  $i = 2$  to  $r$ :

3.1.1 For each subspace  $E$  in  $S_{j_i}$ , if it is not a sum of subspaces in  $\mathcal{I}_{\mathcal{G}}$ , then let

$$\mathcal{I}_{\mathcal{G}} = \mathcal{I}_{\mathcal{G}} \cup E.$$

---

<sup>8</sup>This step finds a sum-dense set containing the minimal synchrony set for the lattice of synchrony subspaces. Recall [2, Theorem 5.11] where it is proved that the minimal synchrony set forms a sum-dense set for the lattice of synchrony subspaces.

<sup>9</sup> $(1, \dots, 1)$  is an eigenvector of  $A$  if the network is regular.

4 [Find the lattice] Let  $V_{\mathcal{G}} = \text{Sum}(\mathcal{I}_{\mathcal{G}})$ . Return( $\mathcal{I}_{\mathcal{G}}, V_{\mathcal{G}}$ )

◇

**Algorithm A.2** [JordanChain( $B_{k-1}, C, k$ )]

- 1 Let  $V_{k-1}$  be the subspace generated by the basis  $B_{k-1}$ .
- 2 Let  $V_k$  be the subspace of vectors  $v_k$  that satisfy  $(A - \lambda_i \text{Id}_n)v_k = v_{k-1}$  for some  $v_{k-1} \in V_{k-1}$ .<sup>10</sup>
- 3 Let  $B_C$  be the basis at the third entry in the table for  $E_{\lambda_i}$  corresponding to the set of equality conditions  $C$ .
- 4 If  $B_C$  is a basis of  $V_k$ , then exit the JordanChain routine.
- 5 Complete the basis  $B_C$  with a set  $\overline{B}_k$  of vectors forming a basis of  $V_k$ .<sup>11</sup>
- 6 Consider the matrix  $M$  whose columns are the vectors of the basis  $\overline{B}_k$ .
- 7 Construct a new matrix  $\overline{M}$  with rows given by  $r_j - r_k$ , with  $r_j$  and  $r_k$  rows in  $M$ , whenever  $x_j = x_k$  is in  $C$ .<sup>12</sup>
- 8 Let  $S$  be the set of all submatrices of  $\overline{M}$  of rank less than  $\#\overline{B}_k$  obtained from  $\overline{M}$  by elimination of rows but not eliminating the row corresponding to  $x_{c_0} = x_c$ .
- 9 Decompose  $S$  into disjoint union  $\bigcup_{i=0}^{\#\overline{B}_k-1} S_i$ , where each  $S_i$  is the set of all matrices in  $S$  with rank  $i$ . For each  $S_i$  remove any matrix  $N$  that is a submatrix of a matrix in  $S_i$  different from  $N$ .
- 10 If  $S \neq \emptyset$  then, for  $i = 0$  to  $\#\overline{B}_k - 1$ :
  - 10.1 While  $S_i \neq \emptyset$  do:
    - 10.1.1 Let  $N \in S_i$  and  $S_i = S_i \setminus \{N\}$ .
    - 10.1.2 Let  $\overline{\overline{B}}_k$  be a basis of the subspace of  $\langle \overline{B}_k \rangle$  obtained from the solution set of the homogeneous system with the coefficient matrix  $N$ .
    - 10.1.3 Let  $C_N$  be the set of equality conditions corresponding to the rows of  $N$ .<sup>13</sup>
    - 10.1.4 If  $C_N = C$ , then change the row corresponding to the set  $C$ : replacing the third entry by the basis  $B = B_C \cup \overline{\overline{B}}_k$  and the fourth entry by  $\#B$ .

<sup>10</sup> $V_k$  is a subspace of  $\ker(A - \lambda_i \text{Id}_n)^k$ .

<sup>11</sup> $\langle B_C \rangle \subseteq \ker(A - \lambda_i \text{Id}_n)^{k-1} \subseteq V_k$ .

<sup>12</sup>If row  $r_j - r_k$  of  $\overline{M}$  is zero, that means that  $x_j = x_k$  for all vectors in  $V_k$ .

<sup>13</sup>Equivalently,  $C_N$  is the set of equality conditions satisfied by the vectors in  $\langle \overline{\overline{B}}_k \rangle$ .

Otherwise, if there is no row at the table with the set of conditions  $C_N$ , then add a new row at the top of the table containing: in the first entry  $C_N$ ; in the second entry the corresponding polydiagonal dimension; in the third entry the basis  $B = B_C \cup \overline{\overline{B}}_k$ ; in the fourth entry  $\#B$ .

Else, go to step 10.1.

10.1.5 If the intersection of the subspace corresponding to the basis  $\overline{\overline{B}}_k$  with  $\text{Im}(A - \lambda_i \text{Id}_n)$  is a nonzero subspace then:

10.1.5.1 Let  $B_k$  be a basis of the intersection  $\langle B \rangle \cap \text{Im}(A - \lambda_i \text{Id}_n)$ .

10.1.5.2  $\text{JordanChain}(B_k, C_N, k + 1)$ .

◇

**Algorithm A.3** [Sum( $\mathcal{I}_{\mathcal{G}}$ )]

The set  $\mathcal{I}_{\mathcal{G}}$  contains the irreducible sum-dense set of the lattice  $V_{\mathcal{G}}$ .

1 Let  $V_{\mathcal{G}} = \mathcal{I}_{\mathcal{G}}$ .

2 Let  $s = \#\mathcal{I}_{\mathcal{G}}$ .

3 For  $i = 2$  to  $s$ ,

3.1 For every (possible) subset  $\{\Delta_{\llcorner j_1}, \dots, \Delta_{\llcorner j_i}\}$ , with  $j_k \neq j_l$ , of  $i$  synchrony subspaces in  $\mathcal{I}_{\mathcal{G}}$ ,

3.1.1 Let  $\Delta_{\llcorner} = \Delta_{\llcorner j_1} + \dots + \Delta_{\llcorner j_i}$ ,

3.1.2 If  $\Delta_{\llcorner}$  is a polydiagonal subspace then let  $V_{\mathcal{G}} = V_{\mathcal{G}} \cup \{\Delta_{\llcorner}\}$ .

4 If  $(1, \dots, 1)$  is an eigenvector of  $A$  then return  $V_{\mathcal{G}} \cup \{\Delta_0\}$  where  $\Delta_0$  is the full synchronous polydiagonal space. Otherwise, return  $V_{\mathcal{G}}$ .

◇