Endowing networks with desired symmetries and modular behavior

P. Khanra,[1,*] S. Ghosh,[2,*] D. Aleja,[3,4] K. Alfaro-Bittner,[3] G. Contreras-Aso,[3] R. Criado,[3] M. Romance,[3]
S. Boccaletti,[3,5,6,7] P. Pal,[8] and C. Hens[2,†]

[1]*Department of Mathematics, State University of New York at Buffalo, Buffalo 14260, USA*
[2]*Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Gachibowli, Hyderabad 500032, India*
[3]*Universidad Rey Juan Carlos, Calle Tulipán s/n, 28933 Móstoles, Madrid, Spain*
[4]*Department of Internal Medicine, University of Michigan, Ann Arbor, Michigan 48109, USA*
[5]*CNR - Institute of Complex Systems, Via Madonna del Piano 10, I-50019 Sesto Fiorentino, Italy*
[6]*Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region, 141701, Russian Federation*
[7]*Complex Systems Lab, Department of Physics, Indian Institute of Technology, Indore - Simrol, Indore 453552, India*
[8]*Department of Mathematics, National Institute of Technology, Durgapur 713209, India*

(Received 3 January 2023; accepted 18 October 2023; published 17 November 2023)

Symmetries in a network regulate its organization into functional clustered states. Given a generic ensemble of nodes and a desirable cluster (or group of clusters), we exploit the direct connection between the elements of the eigenvector centrality and the graph symmetries to generate a network equipped with the desired cluster(s), with such a synthetical structure being furthermore perfectly reflected in the modular organization of the network's functioning. Our results solve a relevant problem of designing a desired set of clusters and are of generic application in all cases where a desired parallel functioning needs to be blueprinted.

DOI: 10.1103/PhysRevE.108.054309

I. INTRODUCTION

Synchronization of networked units is a behavior observed far and wide in natural and man-made systems: from brain dynamics and neuronal firing, to epidemics, or power grids, or financial networks [1–10]. It may either correspond to the setting of a state in which all units follow the same trajectory [11–13], or to the emergence of structured states where the ensemble splits into different subsets, each one evolving in unison. This latter case is known as cluster synchronization (CS) [14–30] and is the subject of many studies in both single-layer [20,21] and multilayer networks [31,32]. Swarms of animals, or synchrony (within subunits) in power grids or brain dynamics, are indeed relevant examples of CS.

Before discussing the details of CS, we should first introduce the problem of total or full synchronization in a network of dynamical units. This problem amounts to understanding the conditions required for the dynamical units to become synchronous ($x_i(t) = x_j(t)$, $\forall\, i, j = 1, ..., N$) after some transient time has elapsed.

The case of identical dynamical units is specially relevant, as many real coupled systems are described within this framework [18–20]. In this situation, the analytical characterization of the total synchronizability of a network of identical dynamical units was carried out in [33], where it was shown that the dynamical equation

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \lambda \sum_{i=1}^{N} \mathcal{A}_{ij}\, g(\mathbf{x}_j - \mathbf{x}_i), \qquad (1)$$

with $\mathbf{x}(t) \in \mathbb{R}^m$ being the dynamical state, $f : \mathbb{R}^m \to \mathbb{R}^m$ a chaotic map and the coupling is given by λ, the coupling strength, $\mathcal{A}$ the adjacency matrix of the network, and $g : \mathbb{R}^m \to \mathbb{R}^m$ the coupling function, admits a stable synchronized solution when the maximum Lyapunov exponents associated to each perturbation away from the synchronization manifold in the eigendirections of the graph's Laplacian are negative. This is the celebrated master stability function approach to synchronization, whose details we will not go into, instead referring the interested reader to [3,4,8,33].

For coupling strengths weaker than that required for total synchronization, the long-term behavior of the network may consist of subsets of nodes that are separately synchronized. These already stable synchronous subgroups are usually called "clusters," hence the term cluster synchronization. The underlying symmetries of a network are responsible for the way nodes split in functional clusters during CS. From a graph theoretic perspective, these clusters are the *orbits* of the graph and are the ingredients of the associated symmetry groups. A *symmetry* (or automorphism) in a graph G is a permutation σ of the nodes of G that preserves adjacency, i.e., $\sigma(G)$ is isomorphic to G. If a symmetry σ exists such that $\sigma(i) = j$ for some couple of nodes i, j, then the two nodes i and j are in the same synchronization's cluster during CS [20,21,34].

While the issue of identifying and computing symmetries in a given network has recently found very efficient solutions

[*] These authors contributed equally to this work.
[†] Corresponding author: chittaranjanhens@gmail.com

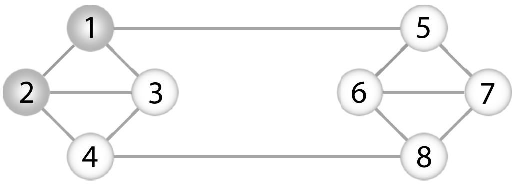

FIG. 1. Symmetries and eigenvector centrality. Schematic representation of an eight-node three-regular graph, where all nodes have the same eigenvector centrality but there is no symmetry σ such that $\sigma(1) = 2$.

[35], the converse problem is still open: can one design a network of arbitrary number of nodes (N) and links (L) endowed with an arbitrary set of orbital clusters? In other words, given a desired cluster of nodes (or a group of clusters), can one generate a graph with density $d = \frac{2L}{N(N-1)}$ endowed with those symmetries which would produce, during CS, exactly the prescribed functional cluster(s)? A first attempt to solve the problem was offered in Ref. [36], where the construction of a feasible quotient graph was proposed as a way to generate networks with prescribed symmetries, a process that implies a noticeable computational complexity and may even be unfeasible for large size networks.

By exploiting the direct connection between the elements of the eigenvector centrality (EVC) and the clusters of a network [37], we here introduce an effective method able to generate networks with desired sets of nodes, links, and clusters, where furthermore the graph structure is perfectly reflected in the modular network's functioning.

This article is structured as follows: In Sec. II we present the theoretical ingredients necessary for the general method we then present. This method is, however, computationally expensive; therefore in Sec. III we present an alternative version which takes advantage of the distinct types of edges which can be present in the network between nodes belonging to the same and/or different clusters. In Sec. IV we give numerical evidence of the claim that the prescribed nodes indeed form synchronization clusters. We end the discussion with some conclusions in Sec. V.

II. A GENERAL, SYMMETRY-BASED METHOD

We start by recalling that if a symmetry σ exists in G permutating nodes i and j, then all local invariants (such as the degree, the average distance, etc.) of i and j must be the same. In addition, Ref. [37] demonstrated that $c(i) = c(j)$ [where $c(i)$ and $c(j)$ are the eigenvector centrality of nodes i and j]. It should be remarked that the opposite [i.e., $c(i) = c(j)$] implying the existence of a symmetry σ such that $\sigma(i) = j$) is not always guaranteed. For instance, the example of Fig. 1 is a graph where all the nodes have degree 3, in which $c(i) = c(j) = 1/8$ for all pairs i, j, and where, however, there is no symmetry σ such that $\sigma(1) = 2$, because the average distances from node 1 and node 2 differ. At the same time, counterexamples like the one in Fig. 1 constitute *pathological cases* limited to regular graph structures, because the construction of symmetries is strongly related with the computation of the isomorphism between graphs [38]. In particular, it has been computationally tested that the EVC is indeed a proper indicator for spotting isomorphic graphs in the case of networks constructed through random processes, for which no realization was found to occur of a graph with two nodes with the same EVC and with no permutating symmetries [39]. Motivated by this evidence, we now move to discuss the methodology for the design of a connected graph G of size N with k prescribed nontrivial clusters.

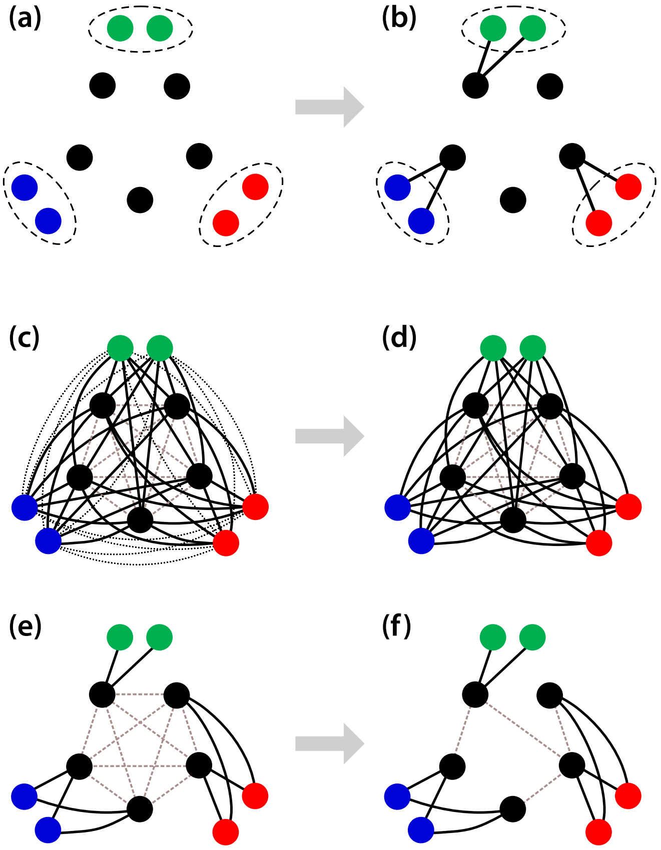

FIG. 2. Network construction. (a) Three clusters (green, blue, and red) are desired. Five black nodes are the trivial (or single-node) clusters. (b) Original motifs are formed by connecting nodes within clusters to black nodes: small star graphs are formed where black nodes are the hubs. (c) The original subnetworks are embedded (see text for the procedure). Dotted (dashed) lines denote CC (BB) links (see text for definition) as they bind clustered (unclustered) nodes. Solid lines denote CB links, which have an end in a clustered node and the other in a trivial cluster. (d) The first step of the specific method consists in removing all CC links. (e) Then a portion of CB links is judiciously removed to preserve an equal neighborhood for each element of each given cluster. After the removal, different clusters may contain nodes of different degree. (f) Finally, as many BB links as needed are removed to reach the desired network density.

For the sake of clarity, we illustrate our method with reference to a small set of $N = 11$ nodes, where the goal is to construct a connected network having three nontrivial, desired clusters of two nodes each [shown with filled green, blue, and red circles in Fig. 2(a), where the black circles represent instead the set of trivial clusters]. This is obtained by means of three consecutive steps.

(1) The first step is the creation of subnetworks, or motifs. Here, one considers all nodes of a desired cluster (say, for

instance, the green circles) and connects them with a randomly selected trivial cluster (one black circle). A star subnetwork is then formed with the black circle as the hub and all the nodes in the cluster as the leaves. Star subnetworks are made in the same way for all other desired clusters [red and blue circles in Fig. 2(a)]. The result is the intermediate disconnected network depicted in Fig. 2(b). By construction, the EVCs of the leaves of each such subnetwork will be at a same value. Notice that, if necessary, one may equally accomplish this first step by either forming rings or complete graphs with the nodes participating in each individual cluster (in both cases, indeed, the EVC elements corresponding to the nodes of the desired clusters will be the same). The adjacency matrices for the three star subnetworks [$\mathcal{A}_1$ (green-black), $\mathcal{A}_2$ (blue-black), and $\mathcal{A}_3$ (red-black)] are

$$\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{A}_3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{2}$$

At the same time, the permutation matrices for each of the star subnetworks are given by

$$\mathcal{P}_1 = \mathcal{P}_2 = \mathcal{P}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{3}$$

which indeed satisfy $P_iA_i = A_iP_i$ $(i = 1, 2, 3)$.

(2) The second step consists in embedding the matrices $\mathcal{P}_i$s and A_is, for constructing the permutation ($\mathcal{P}$) and adjacency ($\mathcal{A}$) matrices of the entire network, as

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_1 & 0 & 0 & 0 \\ 0 & \mathcal{P}_2 & 0 & 0 \\ 0 & 0 & \mathcal{P}_3 & 0 \\ 0 & 0 & 0 & \mathcal{I} \end{bmatrix} \quad \text{and} \quad \mathcal{A} = \begin{bmatrix} \mathcal{A}_1 & \mathcal{I} & \mathcal{I} & \mathcal{I} \\ \mathcal{I} & \mathcal{A}_2 & \mathcal{I} & \mathcal{I} \\ \mathcal{I} & \mathcal{I} & \mathcal{A}_3 & \mathcal{I} \\ \mathcal{I} & \mathcal{I} & \mathcal{I} & B \end{bmatrix}, \tag{4}$$

where $\mathcal{I}$'s are matrices of appropriate order with unit entries and B is the adjacency matrix corresponding to the trivial clusters. The connected network defined by $\mathcal{A}$ is depicted in Fig. 2(c) and endowed with the desired clusters.

In summary (and extending the illustration to generic network of size N with k desired clusters), the first two steps consist in constructing k subnetworks with corresponding adjacency matrices $\mathcal{A}_i$ $(i = 1, \dots, k)$. For each subnetwork ($\mathcal{A}_i$), one then considers the underlying permutation matrix $\mathcal{P}_i$ such that $\mathcal{P}_i\mathcal{A}_i = \mathcal{A}_i\mathcal{P}_i$ $(i = 1, \dots, k)$. Embedding such units, one ends up with the adjacency matrix

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_1 & \mathcal{I} & \cdots & \cdots & \mathcal{I} \\ \mathcal{I} & \mathcal{A}_2 & \mathcal{I} & \cdots & \mathcal{I} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \cdots & \cdots & \mathcal{A}_k & \mathcal{I} \\ \mathcal{I} & \mathcal{I} & \cdots & \mathcal{I} & B \end{bmatrix}_{N\times N} \tag{5}$$

and the permutation matrix

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_1 & 0 & \cdots & 0 & 0 \\ 0 & \mathcal{P}_2 & & \cdots & 0 \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \mathcal{P}_k & 0 \\ 0 & 0 & \cdots & 0 & I \end{bmatrix}_{N\times N}. \tag{6}$$

The resulting network is invariant under the action of $\mathcal{P}$, but the desired clusters may not be disjoint.

(3) The third step consists in obtaining a network with all desired clusters properly disjoint and with the desired density $d \equiv \frac{2L}{N(N-1)}$. Because of what is discussed for the counterexample of Fig. 1, this has to be achieved by a random process of edge removal. Starting from $\mathcal{A}$, the edge removal procedure is as follows:

(a) Select a given percentage l of edges at random that are not part of the subnetworks $\mathcal{A}_i$;

(b) Check the connectivity of the network corresponding to the adjacency matrix A_l that results from removing the selected edges from A;

(c) Check the invariance of the resulting network under the action of the permutation $\mathcal{P}$ i.e., $\mathcal{P}\mathcal{A}_l = \mathcal{A}_l\mathcal{P}$.

If either step (b) or (c) fails, the edges selected in step (a) are not removed, and a second set of l random links are chosen to again test the steps (b) and (c). Otherwise, the edges are removed, and steps (a,b,c) are repeated until the desired network density is obtained. Note that, in this process, one may actually get different networks of the same density with the desired clusters, i.e., the solution of the problem is not unique.

III. AN EFFICIENT ALGORITHM

It should be remarked that condition (3c) requires checking the permutation invariance ($\mathcal{P}\mathcal{A}_l = \mathcal{A}_l\mathcal{P}$) at each step, an operation which may become demanding as the size of the network increases. For networks of arbitrary size, we therefore introduce a more specific method which takes advantage of the fact that condition (3c) is always guaranteed when all the members of each given cluster have the same neighborhood of other network's nodes, so that any two elements of a cluster have the same adjacency. Looking at Fig. 2(c), one immediately sees that edges can be divided into three different groups. A first group connects members of different clusters. These links [depicted as dotted lines in Fig. 2(c)] will be called, from here on, color-color (or CC) links, since they bind nodes of different colors. The second group is made by black-black (or BB) links (dashed lines in the figure) that have both ends in trivial clusters. Finally, the third group is made by CB (or color-black) links [solid lines in Fig. 2(c)], which have an end in an element of a cluster and the other end in a trivial cluster. If N_k is the number of nodes of the kth cluster, m the number of distinct (nontrivial) clusters, and $N_m = \sum_{k=1}^{m} N_k$ the total number of clustered nodes, the initial number of CC, BB, and CB links is

$$
\begin{aligned}
N_{CC} &= \frac{1}{2}\sum_{k=1}^{m}\sum_{j\neq k} N_k N_j,\\
N_{BB} &= (N - N_m)(N - N_m - 1)/2,\\
N_{CB} &= \sum_{k=1}^{m} N_k (N - N_m).
\end{aligned}
\tag{7}
$$

With this in mind, we now turn to explain the three steps of the alternative, more efficient method.

(1) The first step of the specific method is to remove all the N_{CC} links [see Fig. 2(d)], which still preserves the feature of same adjacency for each node of each given cluster.

(2) The second step consists in removing judiciously a portion of CB links. While the links forming the original motifs [see Fig. 2(b)] cannot be removed, if a link is removed connecting a given node of a cluster to a black node, then all the other links connecting all the other nodes of the same cluster to the same black node have to be removed simultaneously to preserve an equal neighborhood. The result is illustrated in Fig. 2(e), and one has the additional freedom of imposing a desired degree to each of the clusters [in the example of Fig. 2(e), green and red nodes end up with having degree 1, while blue nodes have degree 2].

(3) Finally, the third step is removing randomly as many BB links as needed to reach the desired network density [see Fig. 2(f)]. Removing BB links does not affect the neighborhoods of clustered nodes, and therefore permutation is always warranted. The only care here is to check the connectedness of the resulting network. Notice that there is a lower bound for the desired density which is approximately given by $\bar{d} = \frac{2[N_k+(N-N_k-1)]}{N(N-1)} = \frac{2}{N}$ (the case where all clustered nodes have degree 1, and the $N - N_k$ trivial clusters form an open ring structure).

IV. NUMERICAL SIMULATIONS

Let us now move to show that the cluster organization provided by our method(s) allows the constructed network to behave collectively in the desired modular way. To this purpose, we first use the exact procedure of the method to design a network of size $N = 1000$, with two clusters of sizes 20 and 10, respectively, and a desired link density $d = 0.01$. Then we investigate CS with such a setup. We associate each node i to a three-dimensional state vector $\mathbf{x}_i \equiv (x_i, y_i, z_i)$ which obeys the Rössler oscillator equations [40]:

$$
\begin{aligned}
\dot{x}_i &= -y_i - z_i,\\
\dot{y}_i &= x_i + a y_i + \lambda \sum_{i=1}^{N} \mathcal{A}_{ij}(y_j - y_i),\\
\dot{z}_i &= b + z_i(x_i - c),
\end{aligned}
\tag{8}
$$

where dots denote temporal derivatives, the adjacency matrix $\mathcal{A}$ encodes the information of the constructed network, and λ is a real parameter quantifying the coupling strength. The used parameters are $a = 0.1$, $b = 0.1$, and $c = 18$, for which each Rössler oscillator develops a chaotic dynamics. Notice that the coupling term affects only the second variable of each oscillator, a circumstance which determines a class-II synchronization scenario (see the details in Chap. 5 of Ref. [4]), where complete synchronization is warranted above a certain threshold ($\bar{\lambda}$).

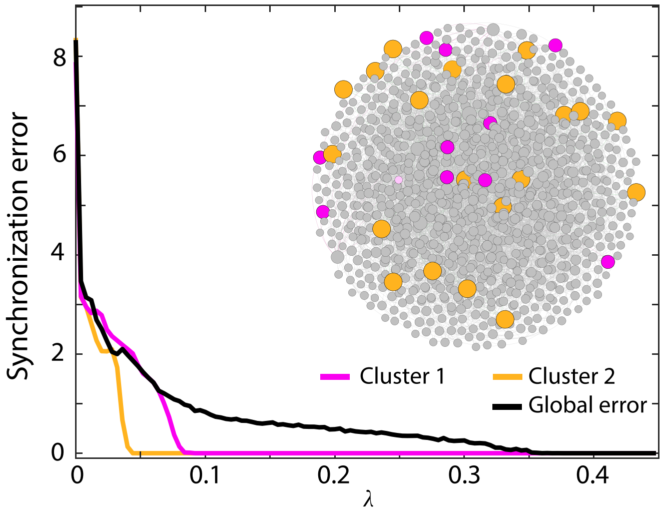

FIG. 3. Structure-induced modular functioning. Cluster and global synchronization errors (see text for definition and legend for color code) vs λ for the constructed network of size $N = 1000$ having two different nontrivial clusters of 20 (yellow nodes, degree 4) and 10 units (magenta nodes, degree 2). A pictorial sketch of the network is shown in the inset, where gray circles are used to depict all trivial clusters. Notice that nodes in the inset have different sizes only for a better visibility, with no connection with their topological properties.

In order to describe what happens for $\lambda < \bar{\lambda}$, one can monitor the behavior of the kth cluster synchronization errors S_k using the time-averaged root-mean-square deviation defined by

$$
S_k = \left\langle \left(\frac{1}{N_k} \sum_{i\in v_k} (y_i - \bar{y})^2 \right)^{1/2} \right\rangle_{\Delta T}, \tag{9}
$$

where v_k is the set of nodes contained in cluster k, $\bar{y}$ is the ensemble average of y within the kth cluster, and $\langle . \rangle_{\Delta T}$ denotes temporal average over a time window ΔT [41]. The synchronization errors for the two clusters, as well as the global synchronization error $S_{\text{glob}} = \langle (\frac{1}{N}\sum_i (y_i - \bar{y}_{\text{glob}})^2)^{1/2} \rangle_{\Delta T}$ (with $\bar{y}_{\text{glob}}$ being the ensemble average of the variable y over the entire network), are reported in Fig. 3. The inset in the figure shows a pictorial representation of the constructed network, prepared using the software GEPHI, with the two clusters drawn with different colors (magenta and yellow). Looking at Fig. 3, it is seen that S_{glob} decays to zero for a much stronger coupling strength than those for which the synchronization errors for the two clusters vanish. Our numerical results show that S_{glob} vanishes at $\lambda = \bar{\lambda} \sim 0.41$, while the two clusters reach synchronization at different values of $\lambda < \bar{\lambda}$, so that a large range of coupling strength exists ($0.09 < \lambda < \bar{\lambda}$) for which the network organizes in a CS state where the two clusters operate in parallel at different synchronized states.

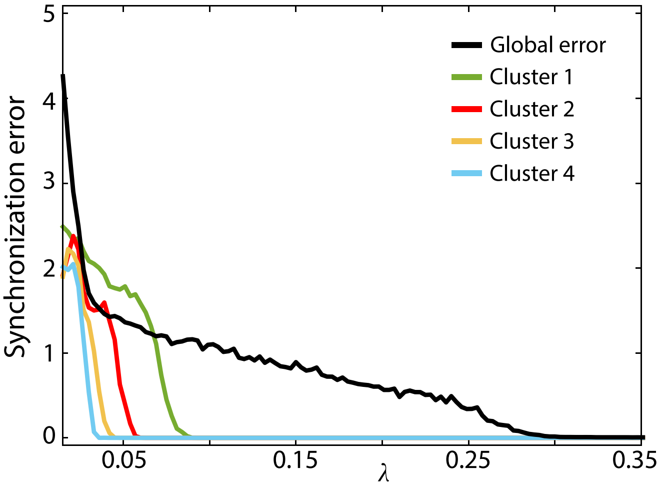

FIG. 4. Large size networks with differentiated clusters. Cluster and global synchronization errors (see text for definition) vs λ for a network of size $N = 10\,000$ where four (very well differentiated in size) clusters have been imprinted by our reduced (specific) method. Namely, clusters 1 to 4 (see color code in the legend) consist of 1000 (degree 2), 300 (degree 3), 100 (degree 4), and 30 (degree 5) nodes, respectively.

Finally, we show that our method is effective also when networks have a very large size, as well as when clusters are very well differentiated. For this purpose, we construct a network with $N = 10\,000$ nodes and density $d = 9.8 \times 10^{-4}$, and we use our reduced (specific) method to blueprint four clusters with sizes spanning more than an order of magnitude. Namely, clusters 1 to 4 are designed to contain, respectively, 1000, 300, 100, and 30 nodes. The degree of a node in each cluster is mentioned in the caption of Fig. 4. Once again, we associate to each node a vector obeying the Rössler oscillator equations [40] with the same parameters ($a = 0.1$, $b = 0.1$, and $c = 18$) used in Fig. 3. The synchronization errors for the four clusters, as well as $S_{\rm glob}$, are reported in Fig. 4. Also in this case, one easily sees that the imprinted cluster structure is perfectly reflected by the modular organization of the network's functioning during CS: $S_{\rm glob}$ vanishes at $\lambda = \bar{\lambda} \sim 0.30$, whereas the four different clusters reach synchronization at different values of the coupling strength in the range $0.03 < \lambda < 0.09$, and therefore a large range of λ exists ($0.09 < \lambda < \bar{\lambda}$) for which the collective network dynamics consists of a CS state with the desired four clusters at works in different synchronized states. Moreover, the results of Figs. 3 and 4 are beautifully fitting with the analytic predictions given by the master stability approach: For class-II systems (as it is the present case), Chap. 5 of Ref. [4] establishes that the threshold for complete synchronization is nothing but $\frac{\nu*}{\lambda_2}$, where $\nu*$ is the value at which the master stability function [33] crosses zero (which, whatever it is, it is the same for the two cases reported in Figs. 3 and 4), and λ_2 is the second smallest eigenvalue of the Laplacian matrix. Therefore, this implies a rigorous prediction that the ratio between the two values of $\bar{\lambda}$ at which the global errors vanish in Figs. 3 and 4 should be equal to the reciprocal of the ratio of the two values of λ_2, which is exactly what happens in our simulations.

V. CONCLUSIONS

In conclusion, we here introduced a method of generic application which allows for the generation of networks with arbitrary number of nodes and links endowed with an arbitrary (and desired) set of orbital clusters, in a way that the graph's parallel functioning occurs into exactly the preselected cluster(s). This has been accomplished by exploiting the direct connection between the elements of the eigenvector centrality and the clusters of a network. We then have shown that such a synthetically generated cluster structure is perfectly reflected in the parallel (modular) organization of the network's functioning during cluster synchronization, even for very large sized networks and for clusters well differentiated in size.

Asymmetry-induced cluster synchronization has recently become the subject of extensive study [28,42–45], in which "asymmetry" implies nonidentical oscillators (e.g., different frequencies in phase oscillators or chaotic oscillators with different parameters) or heterogeneities in the interactions. One may inquire whether these networks can still exhibit such symmetry-induced cluster synchronization in case a certain number of specific links are removed from the original graph, for example, subgraphs might exhibit cluster synchronization [42,46]. Thus, our method might help to construct stable clustered synchronization for a wide range of coupling strengths in a network of nonidentical oscillators.

As our results are of generic application, and therefore they are of value in a wealth of practical circumstances where networks have to be synthesized and/or generated with the scope of ensuring a predesired parallel functioning. For example, in social dynamics, it has been studied that a network of multiagent systems in which symmetric nodes show the same consensus values at steady states [47]. It could be then interesting to design a metapopulation network and study how rumor states of symmetric nodes behave before cluster synchronization, i.e., to look at the transient states of the spreading of rumors of those nodes. In this case, with our method we could design a network with coherent rumor or identical consensus states of target nodes [48–50].

On the other hand, there have been several experiments involving networks of optoelectronic devices and semiconductor lasers, where cluster synchronization is observed and theoretically described by the master stability function formalism [19,51–53]. Therefore, our method could be useful to create, from scratch, a network in which one could study the dynamical behavior of patterns (formed by the synchronized clusters). Despite most of these experiments being designed with the purpose of confirming theoretical predictions, synchronization and cluster synchronization in networks of optoelectronic devices and semiconductor lasers have been studied due to their potential applications in communications, reservoir computing, and sensing [54].

ACKNOWLEDGMENTS

R.C. and M.R. acknowledge funding from Projects No. PGC2018-101625-B-I00 (Spanish Ministry, AEI/FEDER, UE) and No. M1993 (URJC Grant). C.H. is financially supported by the INSPIRE-Faculty Grant (Code: IFA17-PH193). G.C.-A. acknowledges funding from URJC Grant PREDOC-21-026-2164.

[1] M. Breakspear, Dynamic models of large-scale brain activity, Nat. Neurosci. **20**, 340 (2017).

[2] S. Strogatz, *Sync: The Emerging Science of Spontaneous Order* (Penguin, UK, 2004).

[3] A. Pikovsky, M. Rosenblum, J. Kurths, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences* (Cambridge University Press, Cambridge, England, 2003), Vol. 12.

[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Complex networks: Structure and dynamics, Phys. Rep. **424**, 175 (2006).

[5] F. Dörfler and F. Bullo, Synchronization and transient stability in power networks and nonuniform Kuramoto oscillators, SIAM J. Control Optim. **50**, 1616 (2012).

[6] A. E. Motter, S. A. Myers, M. Anghel, and T. Nishikawa, Spontaneous synchrony in power-grid networks, Nat. Phys. **9**, 191 (2013).

[7] P. Ashwin, S. Coombes, and R. Nicks, Mathematical frameworks for oscillatory network dynamics in neuroscience, J. Math. Neurosci. **6**, 2 (2016).

[8] S. Boccaletti, A. N. Pisarchik, C. I. del Genio, and A. Amann, *Synchronization: From Coupled Systems to Complex Networks* (Cambridge University Press, Cambridge, England, 2018).

[9] F. A. Rodrigues, T. KD. M. Peron, Peng Ji, and J. Kurths, The Kuramoto model in complex networks, Phys. Rep. **610**, 1 (2016).

[10] J. Gómez-Gardeñes, Y. Moreno, and A. Arenas, Paths to synchronization on complex networks, Phys. Rev. Lett. **98**, 034101 (2007).

[11] P. Kundu, C. Hens, B. Barzel, and P. Pal, Perfect synchronization in networks of phase-frustrated oscillators, Europhys. Lett. **120**, 40002 (2017).

[12] P. Kundu, P. Khanra, C. Hens, and P. Pal, Optimizing synchronization in multiplex networks of phase oscillators, Europhys. Lett. **129**, 30004 (2020).

[13] M. Brede and A. C. Kalloniatis, Frustration tuning and perfect phase synchronization in the Kuramoto-Sakaguchi model, Phys. Rev. E **93**, 062315 (2016).

[14] T. Dahms, J. Lehnert, and E. Schöll, Cluster and group synchronization in delay-coupled networks, Phys. Rev. E **86**, 016202 (2012).

[15] P. S. Skardal, E. Ott, and J. G. Restrepo, Cluster synchrony in systems of coupled phase oscillators with higher-order coupling, Phys. Rev. E **84**, 036208 (2011).

[16] V. Nicosia, M. Valencia, M. Chavez, A. Díaz-Guilera, and V. Latora, Remote synchronization reveals network symmetries and functional modules, Phys. Rev. Lett. **110**, 174102 (2013).

[17] Peng Ji, T. KD. M. Peron, P. J. Menck, F. A. Rodrigues, and J. Kurths, Cluster explosive synchronization in complex networks, Phys. Rev. Lett. **110**, 218701 (2013).

[18] F. Sorrentino and E. Ott, Network synchronization of groups, Phys. Rev. E **76**, 056114 (2007).

[19] C. R. S. Williams, T. E. Murphy, R. Roy, F. Sorrentino, T. Dahms, and E. Schöll, Experimental observations of group synchrony in a system of chaotic optoelectronic oscillators, Phys. Rev. Lett. **110**, 064104 (2013).

[20] L. M. Pecora, F. Sorrentino, A. M. Hagerstrom, T. E. Murphy, and R. Roy, Cluster synchronization and isolated desynchronization in complex networks with symmetries, Nat. Commun. **5**, 4079 (2014).

[21] F. Sorrentino, L. M. Pecora, A. M. Hagerstrom, T. E. Murphy, and R. Roy, Complete characterization of the stability of cluster synchronization in complex dynamical networks, Sci. Adv. **2**, e1501737 (2016).

[22] M. Lodi, F. D. Rossa, F. Sorrentino, and M. Storace, Analyzing synchronized clusters in neuron networks, Sci. Rep. **10**, 16336 (2020).

[23] A. Bergner, M. Frasca, G. Sciuto, A. Buscarino, E. J. Ngamga, L. Fortuna, and J. Kurths, Remote synchronization in star networks, Phys. Rev. E **85**, 026208 (2012).

[24] F. Sorrentino and L. Pecora, Approximate cluster synchronization in networks with symmetries and parameter mismatches, Chaos **26**, 094823 (2016).

[25] L. V. Gambuzza and M. Frasca, A criterion for stability of cluster synchronization in networks with external equitable partitions, Automatica **100**, 212 (2019).

[26] A. B. Siddique, L. Pecora, J. D. Hart, and F. Sorrentino, Symmetry-and input-cluster synchronization in networks, Phys. Rev. E **97**, 042217 (2018).

[27] Y. S. Cho, T. Nishikawa, and A. E. Motter, Stable chimeras and independently synchronizable clusters, Phys. Rev. Lett. **119**, 084101 (2017).

[28] Y. Wang, L. Wang, H. Fan, and X. Wang, Cluster synchronization in networked nonidentical chaotic oscillators, Chaos **29**, 093118 (2019).

[29] B. Karakaya, L. Minati, L. V. Gambuzza, and M. Frasca, Fading of remote synchronization in tree networks of Stuart-Landau oscillators, Phys. Rev. E **99**, 052301 (2019).

[30] L. Zhang, A. E. Motter, and T. Nishikawa, Incoherence-mediated remote synchronization, Phys. Rev. Lett. **118**, 174102 (2017).

[31] F. D. Rossa, L. Pecora, K. Blaha, A. Shirin, I. Klickstein, and F. Sorrentino, Symmetries and cluster synchronization in multilayer networks, Nat. Commun. **11**, 3179 (2020).

[32] F. Sorrentino, L. M. Pecora, and L. Trajkovic, Group consensus in multilayer networks, IEEE Trans. Network Sci. Eng. **7**, 2016 (2020).

[33] L. M. Pecora and T. L. Carroll, Master stability functions for synchronized coupled systems, Phys. Rev. Lett. **80**, 2109 (1998).

[34] M. Golubitsky, I. Stewart, and D. G. Schaeffer, *Singularities and Groups in Bifurcation Theory: Volume II* (Springer Science & Business Media, New York, 2012), Vol. 69.

[35] R. J. Sánchez-García, Exploiting symmetry in network analysis, Commun. Phys. **3**, 87 (2020).

[36] I. Klickstein and F. Sorrentino, Generating graphs with symmetry, IEEE Trans. Network Sci. Eng. **6**, 836 (2018).

[37] P. Khanra, S. Ghosh, K. Alfaro-Bittner, P. Kundu, S. Boccaletti, C. Hens, and P. Pal, Identifying symmetries and predicting cluster synchronization in complex networks, Chaos, Solitons & Fractals **155**, 111703 (2022).

[38] R. Mathon, A note on the graph isomorphism counting problem, Inf. Process. Lett. **8**, 131 (1979).

[39] N. Meghanathan, Use of eigenvector centrality to detect graph isomorphism, arXiv:1511.06620.

[40] O. E. Rössler, An equation for continuous chaos, Phys. Lett. A **57**, 397 (1976).

[41] Integration method in simulations. Our simulations were performed with an adaptive Tsit integration algorithm implemented

in JULIA. Moreover, in each trial the network was simulated for a total period of 1500 time units, and synchronization errors were averaged over the last $\Delta T = 100$ time units.

[42] Y. Zhang, T. Nishikawa, and A. E. Motter, Asymmetry-induced synchronization in oscillator networks, Phys. Rev. E **95**, 062215 (2017).

[43] S. Panahi and F. Sorrentino, Group synchrony, parameter mismatches, and intragroup connections, Phys. Rev. E **104**, 054314 (2021).

[44] Y. Zhang, J. L. Ocampo-Espindola, I. Z. Kiss, and A. E. Motter, Random heterogeneity outperforms design in network synchronization, Proc. Natl. Acad. Sci. USA **118**, e2024299118 (2021).

[45] A. Nazerian, S. Panahi, I. Leifer, D. Phillips, H. A. Makse, and F. Sorrentino, Matryoshka and disjoint cluster synchronization of networks, Chaos **32**, 041101 (2022).

[46] J. D. Hart, Y. Zhang, R. Roy, and A. E. Motter, Topological control of synchronization patterns: Trading symmetry for stability, Phys. Rev. Lett. **122**, 058301 (2019).

[47] C. Tomaselli, L. V. Gambuzza, F. Sorrentino, and M. Frasca, Multiconsensus induced by network symmetries, Syst. Control Lett. **181**, 105629 (2023).

[48] J. T. Davis, N. Perra, Q. Zhang, Y. Moreno, and A. Vespignani, Phase transitions in information spreading on structured populations, Nat. Phys. **16**, 590 (2020).

[49] D. Maki and M. Thompson, *Mathematical Models and Applications* (Prentice-Hall, Englewood Cliffs, NJ, 1973).

[50] G. Ferraz de Arruda, L. G. S. Jeub, A. S. Mata, F. A. Rodrigues, and Y. Moreno, From subcritical behavior to a correlation-induced transition in rumor models, Nat. Commun. **13**, 3049 (2022).

[51] A. Argyris, M. Bourmpos, and D. Syvridis, Experimental synchrony of semiconductor lasers in coupled networks, Opt. Express **24**, 5600 (2016).

[52] Y. Han, S. Xiang, and L. Zhang, Cluster synchronization in mutually-coupled semiconductor laser networks with different topologies, Opt. Commun. **445**, 262 (2019).

[53] C. R. S. Williams, F. Sorrentino, T. E. Murphy, and R. Roy, Synchronization states and multistability in a ring of periodic oscillators: Experimentally variable coupling delays, Chaos **23**, 043117 (2013).

[54] Y. K. Chembo, D. Brunner, M. Jacquot, and L. Larger, Optoelectronic oscillators with time-delayed feedback, Rev. Mod. Phys. **91**, 035006 (2019).