Generating Graphs with Symmetry

Isaac Klickstein and Francesco Sorrentino

Abstract—In the field of complex networks and graph theory, new results are typically tested on graphs generated by a variety of algorithms such as the Erdős-Rényi model or the Barabási-Albert model. Unfortunately, most graph generating algorithms do not typically create graphs with symmetries, which have been shown to have an important role on the network dynamics. Here, we present an algorithm to generate graphs with prescribed symmetries. The algorithm can also be used to generate graphs with a prescribed equitable partition but possibly without any symmetry. We also use our graph generator to examine the recently raised question about the relation between the orbits of the automorphism group and a graph’s minimal equitable partition.

Index Terms—Networks, Symmetry, Automorphism Group, Random Graphs

1 INTRODUCTION

Due to the interest in the field of studies on complex networks, a number of network generating algorithms have been proposed. Among these are the Erdős-Rényi random graph model [1], the Watts-Strogatz small world model [2], the Barabási-Albert model [3], which generates scale free networks, the static model [4] and the configuration model [5] (together with its uncorrelated version [6]) which have been used to reproduce scale free networks with given power-law degree distribution exponents, and a number of models that generate networks with assigned degree distribution and degree correlation [7], [8]. However, currently available network generating algorithms very rarely reproduce symmetries. It has been shown that symmetries are present and play an important role in dynamical systems with topology described by a graph [9], [10], [11]. It is thus important to introduce a simple network generating algorithm that can create networks with a desired number of symmetries.

Here, we present an algorithm that can generate graphs with prescribed symmetries. Additionally, the algorithm is extended to the case one wants to generate random graphs with a prescribed equitable partition [12]. Using the fact we are able to generate graphs with prescribed equitable partitions but possibly without symmetries, we can further investigate the recently raised question [13] concerning when the minimal equitable partition and the graph symmetries align, and when they do not.

In section 2, we present definitions of equitable partitions and the orbits of the automorphism group, as well as how we may compress a graph with an equitable partition to its quotient graph. In section 3 we derive the algorithm and present an example. Finally, in section 4 we use the algorithm to compare the minimal equitable partition with the partition induced by the orbits of the automorphism group and make concluding remarks in section 5.

2 PRELIMINARIES

Let $G = (\mathcal{V}, \mathcal{E})$ denote a simple, undirected, unweighted graph with the set of $n$ vertices $\mathcal{V}$ and the set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. As $G$ is undirected, each edge is an unordered pair of vertices $(v_i, v_j) = (v_j, v_i)$. Also, as $G$ is simple, we do not allow for any self-loops, $(v_i, v_i) \notin \mathcal{E}$, and no multi-edges, i.e., an edge $(v_i, v_j) \in \mathcal{E}$ may only appear once. A graph $G$ can be represented as an $n \times n$ adjacency matrix $G$ where $G_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and $G_{ij} = 0$ otherwise. We assume $G$ is undirected, the adjacency matrix $G$ is symmetric. A partition $\mathcal{C}$ of the vertices $\mathcal{V}$ satisfies the properties,

$$\mathcal{C} = \left\{ C_i \subset \mathcal{V} \middle| \bigcap \mathcal{C}_i = \emptyset, \quad |C_i| = n_i, \quad \sum_{i=1}^{P} n_i = n \right\}$$  (1)

where we call $C_i$ the $i$th cluster, $i = 1, \ldots, p$, $n_i$ is the number of nodes in cluster $C_i$, and $n$ is the total number of nodes in the graph. An equitable partition (or balanced coloring [14]) $C$ of a graph $G$ is a partition with the additional property that,

$$\sum_{v_k \in C_k} G_{ia} = \sum_{v_k \in C_k} G_{ja}, \quad \forall v_i, v_j \in C_\ell, \quad \forall C_k, C_\ell \in \mathcal{C}$$  (2)

The relation in Eq. (2) states that if two vertices are in the same cluster, $v_i, v_j \in C_\ell$, then they must be adjacent to the same number of vertices in each of the clusters. There are two balanced colorings we are particularly interested in, the minimal balanced coloring (MBC) and the orbits of the automorphism group (OAG) of the graph $G$. The MBC [14] is the balanced coloring $C$ of a graph $G$ that solves the optimization problem,

$$\min_p \quad (3)$$

subject to $|C| = p$ and $C$ is a balanced coloring.

The OAG is best defined using the symmetry group of permutations of the graph $G$. A permutation of the vertices $\mathcal{V}$ is a bijection $\pi : \mathcal{V} \mapsto \mathcal{V}$ which can be thought of as a shuffling of the vertices, i.e., no vertices are removed or created. Each permutation can be represented by an $n \times n$ permutation matrix $P$ where $P_{ij} = 1$ if $\pi(v_j) = v_i$ and $P_{ij} = 0$ otherwise. Theorem 1.1 states that the orbits of the automorphism group $\mathcal{A}(G)$ of $G$ are the sets of vertices that are indistinguishable under the action of any permutation in $\mathcal{A}(G)$.

I. Klickstein and F. Sorrentino are with the Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM, 87131. E-mail: iklick@unm.edu
The set of these symmetric permutations, or simply symmetries, is a group under composition called the automorphism group, Aut(G) \[13\]. The automorphism group induces an equitable partition of the nodes in the graph called the orbits of the automorphism group (OAG). In general, the equitable partitions OAG and MBC are not equal \[13,16,17\]. Coupled dynamical systems whose underlying graph has a non-trivial OAG partition can exhibit complex behavior where the nodes in the same cluster may behave similarly even if they are not directly connected \[18,19\].

The following theorem states the requirement for \( C = \mathcal{O} \).

**Theorem 1.** Let \( \phi_{ij} \) denote the operation of swapping two nodes \( v_i \) and \( v_j \) such that \( v_i, v_j \in C_k \). Also, let \( \mathcal{O} \) denote the partition of the nodes of graph \( G = (V, E) \) induced by its automorphism group Aut(G) and let \( C \) denote the partition of the nodes of the same graph induced by its MBC. Then \( C = \mathcal{O} \) if and only if for every \( \phi_{ij} \), one can construct a permutation

\[
\pi = \phi_{ij} \phi_{k_1 k_1'} \phi_{k_2 k_2'} \cdots \phi_{k_s k_s'}
\]

such that \( k_1 \neq i, j \) and \( k_s' \neq i, j \) for \( 1 \leq \ell \leq s \) and \( \pi \in \text{Aut}(G) \).

The theorem holds by the definition of the automorphism group, and is useful as a tool to check whether or not one should expect the minimal balanced coloring and the orbits of the automorphism group to coincide.

Typically, large random graphs generated with the Erdős-Rényi model, the Watts-Strogatz model, the Barabási-Albert model, and most others will not have non-trivial equitable partitions, that is, \( |C| = |V| \). If one generates a random network using any of these methods one will not see the effect that equitable partitions can have on the system dynamics. However, real networks are often characterized by a large number of symmetries \[9\]. This prompts us to study in this paper a procedure to generate large graphs with an assigned number of symmetries.

An equitable partition \( C \) of a graph \( G \) can be represented as a quotient graph \( Q = (C, D, F, W) \). An example of a quotient graph with a discussion of its four components is shown in Fig. 1. Each vertex in the quotient graph \( C_k \in C, k = 1, \ldots, p \) represents the set of vertices in the same cluster in the original graph \( G \). The self-loop magnitudes \( D : C \rightarrow \mathbb{N} \) is the number of edges each node \( v_i \in C_k \) receives from the other nodes in \( C_k \).

\[
D(C_k) = \sum_{v_i \in C_k} D_{ij}, \quad v_i \in C_k
\]

Note that by the definition of an equitable partition, the particular choice of \( v_i \in C_k \) does not affect the value of \( D(C_k) \). The edges \( F \subset C \times C \) represent those pairs of clusters with edges passing between them. Each edge has two weights \( W : F \times \{0, 1\} \rightarrow \mathbb{N}^+ \)

\[
W((C_k, C_l), 0) = \text{weight from } k \text{ to } l, \quad k < l
W((C_k, C_l), 1) = \text{weight from } k \text{ to } l, \quad k < l
\]

In words, for each edge \( F_i = (C_k, C_l) \), \( W(F_i, 0) \) is the weight to the lower indexed vertex from the higher indexed vertex and \( W(F_i, 1) \) is the weight to the higher indexed vertex from the lower indexed vertex in the quotient graph.

The quotient graph \( Q \) can be represented as a \( p \times p \) matrix \( Q \) with entries,

\[
Q_{ij} = \begin{cases} D(C_i), & i = j \\ W((C_i, C_j), 0), & (C_i, C_j) \in F, \quad i < j \\ W((C_i, C_j), 1), & (C_i, C_j) \in F, \quad i > j \\ 0, & \text{otherwise} \end{cases}
\]

One should read \( Q_{ij} \) as ‘vertices in cluster \( C_i \) receive \( Q_{ij} \) edges from vertices in cluster \( C_j \)’ and \( Q_{ii} \) as ‘each vertex in cluster \( C_i \) receives \( Q_{ii} \) edges from other vertices in \( C_i \)’. We remark that in this paper, we always distinguish between self-loops \( Q_{ii} \) (connections from a vertex to itself in the quotient graph) and edges \( Q_{ij} \) (connections between two different vertices in the quotient graph), as they will be treated very differently in the forthcoming derivations.

The approach of this paper is to generate a full network given knowledge of its quotient graph. As we will see, there are certain prerequisites a quotient graph must satisfy in order to be feasible, i.e., for the existence of a transformation mapping the quotient graph to a corresponding full graph. The quotient graph \( Q \) is unique for a given graph \( G \) with equitable partition \( C \). On the other hand, a single, feasible, quotient graph represents an infinite number of original graphs. Thus, reconstructing a graph \( G \) from a feasible quotient graph \( Q \) is not unique and we must select from this infinite set.

Our procedure illustrated in what follows is based on three steps: (i) we select a quotient graph and assess whether it is feasible, (ii) from a feasible quotient graph, we determine an equitable partition of the network nodes and (iii), we wire the edges of the network so as to ensure that the MBC and OAG coincide. Each one of these three steps is presented in detail in sections \[3.1,3.2\] and \[3.3\] respectively.
3 Results

Before defining the algorithm, we present some useful results from the graph theory literature.

Theorem 2 (Erdős-Gallai [20]). Let a = (a1, a2, . . . , an) be a non-increasing sequence of non-negative integers. The sequence a is realizable as the degree sequence of an undirected simple graph (i.e., one with no self-loops or multi-edges) if and only if,

1) \( \sum_{i=1}^{n} a_i \) is even, and
2) for 1 ≤ k ≤ n,
   \[
   \sum_{i=1}^{k} a_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{a_i, k\} \quad (9)
   \]

Corollary 1. If the degree sequence consists of a constant, a = (r, r, . . . , r), then the two conditions in Thm. 2 can be written in terms of the length of the sequence n.

1) If r is even, then n may be even or odd. If r is odd, then n must be even.
2) From the case r ≤ k in the second condition, it can be shown n ≥ r + 1. For the case r > k, the second condition is trivially satisfied.

Theorem 3 (Gale-Ryser [21]). Let a = (a1, a2, . . . , an) and b = (b1, b2, . . . , bn) be two non-increasing sequences of non-negative integers. The sequences a and b can be realized as the degree sequences of a simple bipartite graph if and only if,

\[
\sum_{i=1}^{k} a_i \leq \sum_{i=1}^{n} \min\{b_i, k\}, \quad 1 \leq k \leq n_1 \quad (10)
\]

or, equivalently,

\[
\sum_{i=1}^{k} b_i \leq \sum_{i=1}^{n} \min\{a_i, k\}, \quad 1 \leq k \leq n_2 \quad (11)
\]

Corollary 2. If a = (r1, r1, . . . , r1) and b = (r2, r2, . . . , r2) are two sequences of a constant integers r1 and r2 of lengths n1 and n2, respectively, then the conditions in Thm. 3 can be rewritten as,

\[
r_1 \leq n_2, \quad r_2 \leq n_1 \quad (12)
\]

and

\[
r_1 n_1 = r_2 n_2 \quad (13)
\]

Also useful will be the following result.

Lemma 1. Let A ∈ \( \mathbb{R}^{m \times n} \) be a matrix such that each row consists of a single positive entry, a single negative entry, and the remaining entries are all zero, and no columns of A consist entirely of zeros. If \( \mathcal{N}(A) \neq \emptyset \), there always exists a positive vector x, that is a vector with all strictly positive entries, such that x ∈ \( \mathcal{N}(A) \).

Proof. By assumption, let there exist at least one solution (besides x = 0) of \( Ax = 0 \) and let \( a_{ij} \in \mathbb{R}^{n} \) be the ith row of A with \( a_{i,j} > 0 \) and \( a_{i,k} < 0 \) by the construction of A. First, assume the entries of x are \( x_j > 0 \) and \( x_k < 0 \). Then, \( a_{i}^{T} x = a_{i,j} x_j + a_{i,k} x_k > 0. \) On the other hand, if \( x_j < 0 \) and \( x_k > 0 \), then \( a_{i}^{T} x < 0. \) By contradiction, if x ∈ \( \mathcal{N}(A) \), then either \( x_j > 0 \) and \( x_k > 0 \) or \( x_j < 0 \) and \( x_k < 0. \) If we find a negative vector x, that is a vector with all strictly negative entries, such that x ∈ \( \mathcal{N}(A) \), then obviously \( -x \in \mathcal{N}(A) \) as well, and \( -x \) is a positive vector.

Lemma 2. Let A be defined as in Lemma 1 with the additional constraint that all non-zero entries in A are rational numbers. If \( \mathcal{N}(A) \neq \emptyset \), then there exists a positive integer solution x ∈ \( \mathbb{Z}^{n} \) such that x ∈ \( \mathcal{N}(A) \).

Proof. By assumption, there exists some vector y ∈ \( \mathbb{R}^{n} \) such that Ay = 0. We can solve for a positive vector y by Gaussian elimination which involves only elementary operations so that each entry in y is rational number, i.e., \( y_i = \frac{p_i}{q_i} \) where \( p_i \) and \( q_i \) are integers. Define \( k = \sum_{i=1}^{n} a_i \), so that ky is an integer and thus x = ky is a positive integer vector. Clearly then, \( kAy = A(ky) = Ax = 0 \) so that x ∈ \( \mathcal{N}(A) \).

As we will see next, given a quotient graph Q from which we are to construct a symmetric unweighted graph \( \mathcal{G} \) we must perform three tasks; (i) verify that Q is a feasible quotient graph, and if it is (ii) determine the cardinality of each |\( C_i \)| = ni, i = 1, . . . , p, after which, finally, (iii) we wire the edges according to \( \mathcal{F} \) and \( \mathcal{W} \).

3.1 Feasibility of the Quotient Graph

Using Corollaries 1 and 2 we can construct the set of requirements in terms of the cardinalities ni for Q to be a feasible quotient graph.

\[
n_i \geq \max \left\{ \max_{(C_i, C_j) \in \mathcal{F}} Q_{ji}, Q_{ii} + 1 \right\} \quad (14a)
\]

\[
\text{mod}(Q_{ii}n_i, 2) = 0 \quad (14b)
\]

\[
Q_{ij}n_i = Q_{ji}n_j, \quad (C_i, C_j) \in \mathcal{F} \quad (14c)
\]

To enforce the constraint in Eq. (14b), we define a new set of variables \( x_i \) for each cluster \( C_i \).

\[
x_i = n_i, \quad Q_{ii} x_i \mod 2 = 0
\]

\[
2x_i = n_i, \quad Q_{ii} x_i \mod 2 = 1
\]

The definition of \( x_i \) modifies the constraint in Eq. (14a) slightly.

\[
x_i \geq \left( 1 - \frac{Q_{ii} x_i \mod 2}{2} \right) \max \left\{ \max_{(C_i, C_j) \in \mathcal{F}} Q_{ji}, Q_{ii} + 1 \right\} \quad (16)
\]

For notational ease, let the lower bound of \( x_i \) in Eq. (16) be defined as \( x_i^L \). The set of constraints in Eq. (14c) can be combined into a system of linear equations \( Ax = 0 \) where A is a \( |\mathcal{F}| \times |\mathcal{C}| \) matrix with each row corresponding to an edge \( \mathcal{F}_i = (C_i, C_k) \) with entries

\[
A_{ij} = \begin{cases} Q_{jk}, & \text{if } j < k \text{ and } Q_{jj} \mod 2 = 0 \\ 2Q_{jk}, & \text{if } j < k \text{ and } Q_{jj} \mod 2 = 1 \\ -Q_{kj}, & \text{if } j > k \text{ and } Q_{jj} \mod 2 = 0 \\ -2Q_{kj}, & \text{if } j > k \text{ and } Q_{jj} \mod 2 = 1 \\ 0, & \text{otherwise} \end{cases}
\]

The matrix A defined in Eq. (17) is of the form presented in Lemma 1. By Lemmas 1 and 2 if \( \mathcal{N}(A) \neq \emptyset \), then there exists an integer solution x such that \( Ax = 0, x \geq x^L \) and we can then reconstruct n using Eq. (15). Note that if we find one solution x, then any integer multiple sx, s ≥ 1, is also a solution. On the other hand, if \( \mathcal{N}(A) = \emptyset \), then there is no solution n, which leads to our first main result.

Theorem 4. The quotient graph Q = (C, D, F, W) is a feasible quotient graph if and only if the matrix A defined in Eq. (17) has a non-trivial null space.
This result provides a framework with which one may construct feasible quotient graphs. One first freely chooses \( p = |C| \), the number of clusters, and arbitrarily assigns \( Q_{ii} \), \( i = 1, \ldots, p \). One may also freely add \( p - 1 \) edges \( \mathcal{F}_j = (\mathcal{C}_j, \mathcal{C}_k) \) with weights \( W(\mathcal{F}_j, 0) \) and \( W(\mathcal{F}_j, 1) \). At this point, the maximum possible rank of \( A \) is \((p-1)\), and so the dimension of the null space is at least one. Any additional edges may be added only if they are linear combinations of the first \((p-1)\) rows of \( A \) and satisfy the condition that each row of \( A \) has one positive integer and one negative integer with all other entries zero.

### 3.2 The Equitable Partition of the Nodes

After verifying that \( Q \) is a feasible quotient graph simply by checking that \( N(A) \neq \emptyset \), the question remains how one may find \( n \), the cardinalities of each cluster \( n_1, n_2, \ldots, n_p \). One option is to compute a basis for the null space of \( A \) and then proceed as in the proof of Lemma 2. This procedure may lead to very large \( n_i \) during the scaling of the rational solution \( y \) to an integer solution. We propose an alternative approach by formulating an integer linear program (ILP) that can yield a minimal realization by which we mean the resulting graph \( \mathcal{G} \) has the fewest number of nodes,

\[
\begin{align*}
\min \quad & c^T x \\
\text{s.t.} \quad & Ax = 0 \\
& x \geq x^L
\end{align*}
\]

(18)

The vector \( c \) is any strictly positive vector whose particular choice will select different integer vectors \( x \) in the null space of \( A \).

By ensuring \( Q \) is a feasible quotient graph, we can be certain the dimension of the null space of \( A \) is at least one, and Eq. (18) has a solution. To solve instances of Eq. (18), we use the branch and cut algorithm implemented in COIN-OR’s package CBC [22]. Once we have solved Eq. (18), \( x^* \), we apply Eq. (15) to compute \( n \), and we may scale it larger by any integer multiple \( s \) if one is interested in a graph with a larger number of vertices.

### 3.3 Wiring the Edges

Wiring the edges must be done in such a way to ensure that the desired number of edges between clusters dictated by the quotient graph is satisfied and the maximum number of symmetries is present so the OAG and the MBC coincide. To simplify the notation in what follows, for each node \( v_j \in \mathcal{V} \), \( j = 1, \ldots, n \), in the graph, we assign a cluster specific label, \( u_i \in \mathcal{C}_k \), \( i = 0, \ldots, n_k - 1 \). Each self-loop and each edge in the quotient graph can be handled separately. First we will discuss the wiring to satisfy each self-loop, and second we will discuss the wiring to satisfy each edge in the quotient graph.

#### 3.3.1 Intra-Cluster Edges

Let \( u_i \in \mathcal{C}_k \), \( i = 0, \ldots, n_k - 1 \) be a node in cluster \( k \) and assume \( Q_{kk} > 0 \) (otherwise if \( Q_{kk} = 0 \) there are no intra-cluster edges to add for \( \mathcal{C}_k \)). In what follows, we must distinguish between the two cases that \( Q_{kk} \) is even or odd. If \( Q_{kk} \) is even, we add the edges,

\[
\begin{align*}
\left\{ \begin{array}{l}
(u_i, u_{(i+j) \mod n_k}), \\
(u_i, u_{(i+n_k-j) \mod n_k})
\end{array} \right. & \quad j = 1, \ldots, Q_{kk}/2
\end{align*}
\]

(19)

From the formulation of the ILP, \( n_k > Q_{kk} \) so that each node \( u_i \) is connected to \( Q_{kk} \) other nodes in the same cluster. We also show that the set of edges in Eq. (19) is symmetric, that is, if \( (u_i, u_{i'}) \) is an edge then \( (u_{i'}, u_i) \) is an edge as well. From the first line of Eq. (19) we see that \( i' = (i+j) \mod n_k \). Substituting this expression for \( i \) into the second line of Eq. (19), we must show that,

\[
\begin{align*}
& [(i + j) \mod n_k + n_k - j] \mod n_k \\
= & [i + j - qn_k + n_k - j] \mod n_k \\
= & [i + (1 - q)n_k] \mod n_k = i \mod n_k = i
\end{align*}
\]

(20)

In a similar fashion one can show that if \( i' = (i+n_k-j) \mod n_k \) from the second line of Eq. (19), we can substitute \( i' \) for \( i \) in the first line of Eq. (19). Thus, for \( 0 \leq i \leq n_k - 1 \), there are \( n_kQ_{kk}/2 \) unique edges created by Eq. (19) as every pair appears twice. If \( Q_{kk} \) is odd, we add the following edges,

\[
\begin{align*}
\left\{ \begin{array}{l}
(u_i, u_{(i+j) \mod n_k}), \\
(u_i, u_{(i+n_k-j) \mod n_k})
\end{array} \right. & \quad j = 1, \ldots, (Q_{kk} - 1)/2 \\
\left\{ \begin{array}{l}
(u_i, u_{(i+n_k/2) \mod n_k})
\end{array} \right. & \quad j = 1, \ldots, (Q_{kk} - 1)/2
\end{align*}
\]

(21)

By the formulation of the ILP, if \( Q_{kk} \) is odd, \( n_k \) is even so \( n_k/2 \) is an integer. Showing that the set of edges in Eq. (21) leads to each node \( u_i \in \mathcal{C}_k \) having intra-cluster degree \( Q_{kk} \) and that the set of edges is symmetric is very similar to the proof of Eq. (19) and thus we do not include it here.

#### 3.3.2 Inter-Cluster Edges

Let \( u_i \in \mathcal{C}_k \), \( i = 0, \ldots, n_k - 1 \) and \( w_j \in \mathcal{C}_\ell \), \( j = 0, \ldots, n_\ell - 1 \) be nodes in clusters \( k \) and \( \ell \), respectively. From the ILP formulation, we have the following constraints,

\[
\begin{align*}
n_kQ_{kl} = n_kQ_{lk}, & \quad n_k \geq Q_{\ell k}, \quad n_\ell \geq Q_{k\ell}
\end{align*}
\]

(22)

for each edge in the quotient graph. Let \( h = \gcd(n_k, n_\ell) \) so that \( n_k = d_kh \) and \( n_\ell = d_\ell h \), where \( \gcd(a, b) \) is the greatest common divisor of two integers \( a \) and \( b \). Also, find \( m \) such that,

\[
\frac{m}{c} = \frac{Q_{\ell k}}{n_k} = \frac{Q_{k\ell}}{n_\ell}
\]

(23)

Pick an ordered sequence of integers \( b = \{b_1, b_2, \ldots, b_m\} \) such that,

\[
\sum_{j=1}^{m} b_j = h, \quad b_j \geq 1
\]

(24)

For each node \( u_i \in \mathcal{C}_k \), \( i = 0, \ldots, n_k - 1 \), we create edges \( (u_i, w_j(i_r, r_2)) \) for \( r_1 = 0, \ldots, d_k - 1 \) and \( r_2 = 1, \ldots, m \), where \( w_j \in \mathcal{C}_\ell \), \( j = 0, \ldots, n_\ell - 1 \). We use the function \( f(i, r_1, r_2) = (i + r_1c + \sum_{j=1}^{r_2} b_j) \mod n_\ell \)

(25)

Note that each triplet \((i, r_1, r_2)\) yields a unique edge so that there are in total \( n_kd_km = n_kQ_{kl} \) edges, the required number of edges. Also, each vertex \( u_i \in \mathcal{C}_k \) is connected with \( Q_{kl} \) vertices in \( \mathcal{C}_\ell \). Alternatively, we can generate the
Fig. 2. An example of the process from quotient graph to full graph. (A) A graphical depiction of the quotient graph $Q$ with three vertices $C_1$, $C_2$, and $C_3$. Vertices $C_1$ and $C_2$ have self-loops $D(C_1) = 1$ and $D(C_2) = 2$. There are two edges, $F_1 = (C_1, C_2)$ and $F_2 = (C_1, C_3)$. The first edge has weights $W(F_1, 0) = 2$ and $W(F_1, 1) = 1$ while the second edge has weights $W(F_2, 0) = 1$ and $W(F_2, 1) = 2$. Also shown is the weighted quotient adjacency matrix $Q$ and the matrix $A$ that appears as the linear constraints in the ILP in Eq. (13). Note that the dimension of the null space of $A$ is one so that there exists a solution to $Ax = 0$. (B) The ILP that must be solved to find $x_1$, $x_2$, and $x_3$. Note that we will scale the population by $s = 2$. As $D(C_1)$ is odd, we must set $2x_1 = n_1$ while the other populations are $sx_2 = n_2$ and $sx_3 = n_3$. The consistent populations found are $n_1 = 4$, $n_2 = 8$, and $n_3 = 3$. (C) The sets of edges corresponding to each quotient graph self-loop and edge are computed. The edges prescribed by the self-loop in $C_1$ are found using Eq. (21). The edges prescribed in $C_2$ are found using Eq. (19). The edges in the full graph prescribed by the two edges in the quotient graph $F_1$ and $F_2$ are found using Eq. (25). (D) A diagram of the resulting graph consisting of $n = 14$ vertices. The vertices are shaded according to their cluster and the edges are textured according to their corresponding edge in the quotient graph shown in the diagram in (A).

same set of edges with respect to the vertices in $C_\ell$. For each vertex $w_i \in C_\ell$, $i = 0, \ldots, n_\ell - 1$, we create edges $(w_i, u_g(i, r_3, r_4))$ for $r_3 = 0, \ldots, d_\ell - 1$ and $r_4 = 1, \ldots, m$,

$$g(i, r_3, r_4) = \left( i + r_3 h + \sum_{j=1}^{r_4} b_{m-j+1} \right) \mod n_\ell $$  (26)

Once again, note that each triplet $(i, r_3, r_4)$ yields a unique edge so that there is in total $n_\ell d_\ell m = n_\ell Q_{\ell k}$ edges, the same number as in Eq. (25) by the constraint in the ILP restated in Eq. (22). Also, each vertex $w_i \in C_\ell$ is connected with $Q_{\ell k}$ vertices in $C_{\ell k}$. As both Eq. (25) and Eq. (26) create the same number of edges, we must prove the statement that the two sets of edges are equal, that is, for every edge $(u_i, w_f(i, r_1, r_2))$ created by Eq. (25), there is a corresponding edge $(w_f(i, r_1, r_2), u_i)$ created by Eq. (26). In other words, for every pair $(r_1, r_2)$, we must find a corresponding pair $(r_3, r_4)$ such that $g(f(i, r_1, r_2), r_3, r_4) = i$, or written with the definitions of $f$ and $g$,

$$\left[ \left( i + r_1 c + \sum_{j=1}^{r_2} b_j \right) \mod n_\ell + r_3 c \right] \mod n_\ell = i $$  (27)

Using the definition of the modulo, there exists some integer $q$ such that we may rewrite the term modulo $n_\ell$ as,

$$\left( i + r_1 h + \sum_{j=1}^{r_2} b_j \right) \mod n_\ell = i + r_1 h + \sum_{j=1}^{r_2} b_j - q n_\ell $$  (28)

There are two cases we must examine; if $r_2 < m$ and if $r_2 = m$.

Case 1: $r_2 < m$. Set $r_4 = m - r_2$ so that the second summation becomes,

$$\sum_{j=1}^{m-r_2} b_{m-j+1} = \sum_{j=r_2+1}^{m} b_j$$  (29)

Thus the total summation becomes $\sum_{j=1}^{m} b_j = c$. Substituting the results of Eqs. (28) and (29) into Eq. (27),

$$\left[ i + (r_1 + r_3 + 1) h - q n_\ell \right] \mod n_\ell $$  (30)

From Eq. (25), we know $0 \leq i \leq n_\ell - 1$ so that $i$ mod $n_\ell = i$, thus, we are left to find $r_3$ such that for some integer $t$,

$$(r_1 + r_3 + 1) h - q n_\ell = t n_\ell \Rightarrow r_3 = t d_\ell + q d_\ell - r_1 - 1 $$  (31)

From Eq. (26), we know $0 \leq r_3 \leq d_\ell - 1$. Applying the bounds and moving everything not dependent on $t$ to the expressions for the bounds,

$$r_1 + 1 - q d_\ell \leq t d_\ell \leq r_1 + q d_\ell $$  (32)

For there to certainly exist some integer $t$ that satisfies Eq. (32), the number of integers between the bounds, inclusive must be at least equal to $d_\ell$.

$$(d_\ell + 1 - q d_\ell) - (r_1 + 1 - q d_\ell) + 1 = d_\ell $$  (33)

Thus there exists precisely one value of $t$ which satisfies Eq. (31).

Case 2: $r_2 = m$. For this case, set $r_4 = m$ so both summations are over the total sequence $b$, each of which
weights to all be one. We choose to scale the solution by $s$ such that, $$(r_1 + r_3 + 2)h - qn_\ell = tn_k$$ (35)
Following the same procedure as in case 1, using the bounds $0 \leq r_3 \leq d_k - 1$, we see that $r_3$ must satisfy,
$$r_3 = td_k + qd_\ell - r_1 - 2$$
where the gap between the bounds, inclusive, is,
$$(d_k + r_1 + 1 - qd_\ell) - (r_1 + 2 - qd_\ell) + 1 = d_k$$ (37)
In summary, we have shown that Eqs. (25) and (26) each generate $n_i Q_{i\ell} = n_i Q_{i\ell}$ unique edges, and that for each edge in Eq. (25), the same edge also appears in Eq. (26), thus the sets of edges are equal.

### 3.4 An Example

A complete example of the process outlined in the previous sections 3.1, 3.2, and 3.3 is shown in Fig. 2. The diagram of the quotient graph is shown in Fig. 2(A) which consists of three vertices labeled $C_1$, $C_2$, and $C_3$. Vertices $C_1$ and $C_2$ have self-loops, $D(C_1) = 1$ and $D(C_2) = 2$. There are two edges $F_1 = (C_1, C_2)$ and $F_2 = (C_1, C_3)$. The first edge has weights $W(F_1, 0) = 2$ and $W(F_1, 1) = 1$ and the second edge has weights $W(F_2, 0) = 1$ and $W(F_2, 1) = 2$. The description of the quotient graph is summarized in the quotient adjacency matrix $Q$ also shown in Fig. 2(A). Additionally, the matrix $A$ as described in Section 3.2 is shown which has a null space of dimension one. The full ILP is shown in Fig. 2(B) where the weights in the cost function, $c = (c_1, c_2, c_3)$, are chosen to all be one. We choose to scale the solution by $s = 2$. Note that as $D(C_1)$ is odd, that $2sx_1 = n_1$ while $sx_2 = n_2$ and $sx_3 = n_3$. The lower bounds are found by using Eq. (16), $x_1^\ell = \frac{1}{2} \max \{2, 2, 1\}$, $x_2^\ell = \max \{3, 2\}$, and $x_3^\ell = \max \{1, 1\}$. The solution to the ILP is, $x_1^\ell = 1$, $x_2^\ell = 4$, and $x_3^\ell = 1$, which can be converted to the cluster cardinalities $n_1 = 4$, $n_2 = 8$, and $n_3 = 2$. In Fig. 2(C) the resulting edges for each self-loop and edge in the original quotient graph are listed. The edges created as described by $D(C_1)$ are generated using Eq. (21) while the edges created as described by $D(C_2)$ are generated using Eq. (19). Each edge created is sorted by the texture of its originating self-loop or edge in the original quotient graph. Finally, a diagram of the resulting graph with $n = 14$ vertices is shown in Fig. 2(D) where the nodes are shaded and the edges are textured according to their originating feature in the quotient graph in Fig. 2(A).

### 3.5 Random Graphs with Non-Trivial MBC

In principle there may be more than one way to choose the sequence $b$ in Eq. (24). After choosing one such sequence, the wiring procedure described in Eqs. (19), (21), (23), and (24) is deterministic so for each quotient graph $Q$ with cluster cardinalities $n_i$, $i = 1, \ldots, p$, the process will create one realization. The graph created so far has the property that the partition of the nodes induced by the MBC $C$ will be equal to the OAG, $O$, due to the particular wiring of the edges discussed in section 3.3.

The procedure laid out in the previous subsections can be extended to the case that one is interested instead in generating a random graph with non-trivial MBCs (and not necessarily the automorphism group of the graph). This can be done by randomly rewiring the edges of the network obtained in the procedure in sections 3.1, 3.2, and 3.3 in such a way that the quotient graph is preserved, as described next.

For each set of intra-cluster edges in cluster $C_k$, randomly choose 2 edges, $(u_i, u_j^\prime)$ and $(u_j, u_j^\prime')$. If $i \neq j^\prime$ and $j \neq i^\prime$, then remove these edges and add two new edges $(u_i, u_j^\prime)$ and $(u_j, u_j^\prime)$. Remove this process a suitable number of times.

For each set of inter-cluster edges between cluster $C_k$ and $C_{i^\prime}$, randomly choose 2 edges $(u_i, w_j)$ and $(u_j, w_j^\prime)$. Remove these edges and add two new edges $(u_i, w_j^\prime)$ and $(u_j, w_j^\prime)$. Repeat this process a suitable number of times.

### Algorithm 1 Random Graph with Non-Trivial Balanced Coloring

**Require:** $Q$ is a quotient graph

**Require:** $c$ is a positive vector of length $p$.

1. Construct $A$ from $Q$ according to Eq. (17).
2. Solve Eq. (18) for $x$.
3. If desired, scale $x_i \leftarrow sx_i$ for integer $s > 1$ for $i = 1, \ldots, p$.
4. $n_i = 2x_i$ if $Q_{ii}$ is odd, and $n_i = x_i$ otherwise, $i = 1, \ldots, p$.
5. for $i = 1, \ldots, p$ do
6. if $Q_{ii} > 0$ then
7. Use either Eq. (19) or Eq. (21) if $Q_{ii}$ is even or odd, respectively, to wire the intra-cluster edges.
8. if desired, randomize the edges via swapping.
9. end if
10. end for
11. for All edges $F_i = (C_k, C_{i^\prime})$ do
12. Use Eq. (23) to wire the inter-cluster edges.
13. if desired, randomize the edges via swapping.
14. end for

### 4 Comparing the MBC and the OAG

It has been shown the OAG and MBC may not align [13, 16, 17]. An example of this type of graph is presented in Fig. 3 where we show a network for which the equitable partition consists of two clusters, but the number of orbital partitions consists of ten clusters, the number of vertices, each consisting of a single vertex. We use the framework developed in this paper to numerically examine when to expect the MBC and the OAG to align and when they will not. In all cases, we generate graphs $G$ with the randomization procedure of section 3.5 so that the MBC and OAG may or may not align. We first examine how the size of the graph can affect $|O|$ where we adjust the size by scaling the cluster cardinalities $n$ by a positive integer $s$. We define the following metric,
$$f(O) = \frac{n - |O|}{n - |C|}$$ (38)
so that \(|O| = |C|\), \(f(O) = 1\), and if every orbit consists of a
single vertex, \(f(O) = 0\). Note that, by design, \(|C| < N\) so
that, while the numerator may go to zero, the denominator
does not change for a given quotient graph. We choose a
single quotient graph with only two vertices, \(C = \{C_1, C_2\}\),
self-loops \(D(C_1) = 0\) and \(D(C_2) = 1\), and one edge \(F = \{F_1\}\)
where \(F_1 = (C_1, C_2)\) with weights \(C(F_1, 0) = 2\) and
\(C(F_1, 1) = 3\), as shown in the inset of Fig. 4. After showing
that this quotient graph is feasible and solving Eq. (18) for
the cluster cardinalities, \(n_1 = 3\) and \(n_2 = 2\), we are then free
to scale \(n_1\) and \(n_2\) by any positive integer \(s\) before wiring
the graph. For each value of \(s\), we generate 1000 graphs,
randomly rewiring the edges as described in section 3.5.

We see in Fig. 4 that if \(s = 1\), i.e., each graph \(G\) is
selected from the smallest graphs that can be represented
by the quotient graph shown in the inset (following the
procedure presented in section 3), then the MBC and OAG
almost always align. As \(s\) is increased, we see that \(f(O)\)
decreases rapidly, indicating that the MBC and OAG almost
surely never align. For \(s > 6\), almost every graph generated
has almost no non-trivial orbits of the automorphism group,
that is \(|O| = n\). While Fig. 4 shows results that are specific
to the particular quotient graph in the inset, qualitatively
similar behavior is seen for all quotient graphs examined.

5 Conclusion

Symmetries in complex networks and graphs have been
shown to play an important role on the network dynamics,
e.g., in the context of network synchronization [11, 13],
[16, 18], and time averaged network dynamics [23]. However,
to the best of our knowledge, no algorithms have been
proposed that generate large networks with an assigned
number of symmetries. In this paper, we address this gap
in the literature and propose a generating algorithm that
is guaranteed to produce a network with an assigned number
of symmetries from knowledge of a feasible quotient graph.
We also show how this algorithm can be extended to gen-
erate a graph with an assigned minimal balanced coloring
(MBC), which may or may not coincide with the OAG of the
graph.

An analysis of anecdotal cases of networks has shown
that the OAG and MBC of a graph may not always align
[13, 17]. However, the question has remained unanswered of
how common it is for the OAG and MBC of a graph to align.
Here we take advantage of our graph generating
algorithm and show that when mapping a quotient network
to a larger network with either a desired MBC or OAG, these
two are never seen to align for large enough network size.
Our results indicate that the property that the MBC and the
OAG of a graph may not align is indeed a generic feature of
large graphs and networks.

Acknowledgments

We would like to thank Lou Pecora, David Phillips, and
Fabio Della Rossa for insightful conversations. This work
is supported by the National Science Foundation through NSF
grant CMMI-1400193, NSF grant CRISP-1541148, and ONR
Award No. N00014-16-1-2637 as well as HDTRA1-13-1-0020.

References

[1] P. Erdős and A. Rényi, “On random graphs i,” Publ. Math. Debre-
cen, vol. 6, pp. 290–297, 1959.
[2] D. J. Watts and S. H. Strogatz, “Collective dynamics of
’small-world’ networks,” nature, vol. 393, no. 6684, p. 440, 1998.
[Online]. Available: https://www.nature.com/articles/30918
[3] A.-L. Barabási and K. Albert, “Emergence of scaling in random
networks,” Science, vol. 286, no. 5439, pp. 509–512, 1999. [Online].
Available: http://science.sciencemag.org/content/286/5439/509

Fig. 3. A network generated with two clusters in its MBC. In (A), the
nodes are colored according to the MBC partition, where one can see
that each red node is connected to three other red nodes and a blue
node, while each blue node is connected to four red nodes and the other
blue node. In (B), the nodes are colored according to the OAG partition.
For this graph, each \(O_i \in O\) consists of a single node.

Fig. 4. Comparing the cardinality of the OAG \(O\) to the cardinality of
the MBC \(C\). Each graph is constructed using the quotient graph shown in
the inset, so that \(|C| = 2\). The edge weights and self-loop weights are shown
as well. The value of \(s\) is the factor with which we scale the solution to
Eq. (18). For each value of \(s\) we generate 1000 graphs. The edges are
randomly rewired preserving the MBC, and then the number of orbits of
the automorphism group is determined. The black line is the mean with
error bars representing one standard deviation.
[4] K.-I. Goh, B. Kahng, and D. Kim, “Universal behavior of load distribution in scale-free networks,” Physical Review Letters, vol. 87, no. 27, p. 278701, 2001. [Online]. Available: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.87.278701

[5] M. Molloy and B. Reed, “A critical point for random graphs with a given degree sequence,” Random structures & algorithms, vol. 6, no. 2-3, pp. 161–180, 1995. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.3240060204

[6] M. Catanzaro, M. Boguñá, and R. Pastor-Satorras, “Generation of uncorrelated random scale-free networks,” Physical Review E, vol. 71, no. 2, p. 027103, 2005. [Online]. Available: https://journals.aps.org/pre/abstract/10.1103/PhysRevE.71.027103

[7] M. E. J. Newman, “Assortative mixing in networks,” Physical review letters, vol. 89, no. 20, p. 208701, 2002. [Online]. Available: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.89.208701

[8] ——, “Mixing patterns in networks,” Physical Review E, vol. 67, no. 2, p. 026126, 2003. [Online]. Available: https://journals.aps.org/pre/abstract/10.1103/PhysRevE.67.026126

[9] B. D. MacArthur, R. J. Sánchez-García, and J. W. Anderson, “Symmetry in complex networks,” Discrete Applied Mathematics, vol. 156, no. 18, pp. 3525–3531, 2008.

[10] V. N. Belykh, I. V. Belykh, and E. Moskilde, “Cluster synchronization modes in an ensemble of coupled chaotic oscillators,” Physical Review E, vol. 63, no. 3, p. 036216, 2001.

[11] 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,” Science advances, vol. 2, no. 4, p. e1501737, 2016. [Online]. Available: http://advances.sciencemag.org/content/2/4/e1501737

[12] V. N. Belykh, G. V. Osipov, V. S. Petrov, J. A. K. Suykens, and J. Vandewalle, “Cluster synchronization in oscillatory networks,” Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 18, no. 3, p. 037106, 2008. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.2956986

[13] A. B. Siddique, L. Pecora, J. D. Hart, and F. Sorrentino, “Symmetry-and input-cluster synchronization in networks,” Physical Review E, vol. 97, no. 4, p. 042217, 2018. [Online]. Available: https://journals.aps.org/pre/abstract/10.1103/PhysRevE.97.042217

[14] I. Belykh and M. Hasler, “Mesoscale and clusters of synchrony in networks of bursting neurons,” Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 21, no. 1, p. 016106, 2011. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.3563588

[15] J. Lauri and K. Scapellato, Topics in graph automorphisms and reconstruction. Cambridge University Press, 2016, vol. 432.

[16] M. T. Schaub, N. O’Clery, Y. N. Billeh, J.-C. Delvenne, R. Lambiotte, and M. Barahona, “Graph partitions and cluster synchronization in networks of oscillators,” Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 26, no. 9, p. 094521, 2016.

[17] S. Kudose, “Equitable partitions and orbit partitions,” Acta Mathematica Sinica, pp. 1–9, 2009.

[18] V. Nicosia, M. Valencia, M. Chavez, A. Diaz-Guilera, and V. Latora, “Remote synchronization reveals network symmetries and functional modules,” Physical review letters, vol. 110, no. 17, p. 174102, 2013.

[19] L. M. Pecora, F. Sorrentino, A. M. Hagerstrom, T. E. Murphy, and R. Roy, “Cluster synchronization and isolated desynchronization in complex networks with symmetries,” Nature communications, vol. 5, p. 4079, 2014. [Online]. Available: https://www.nature.com/articles/ncomms5079

[20] S. Choudum, “A simple proof of the erdoes-gallai theorem on graph sequences,” Bulletin of the Australian Mathematical Society, vol. 33, no. 1, pp. 67–70, 1986.

[21] D. Gale, “A theorem on flows in networks,” Pacific J. Math, vol. 7, no. 2, pp. 1073–1082, 1957.

[22] R. Lougee-Heimer, “The common optimization interface for operations research: Promoting open-source software in the operations research community,” IBM Journal of Research and Development, vol. 47, no. 1, pp. 57–66, 2003. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5388980/

[23] A. B. Siddique, L. M. Pecora, and F. Sorrentino, “Symmetries in the time-averaged dynamics of networks: reducing unnecessary complexity through minimal network models,” arXiv preprint arXiv:1710.05257, 2017.