IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Undirected graphs: Is the shift-enabled condition trivial or necessary?

LIYAN CHEN[1,2], SAMUEL CHENG[3], (Senior Member, IEEE), KANGHANG HE[4], VLADIMIR STANKOVIC[4], (Senior Member, IEEE), and LINA STANKOVIC[4], (Senior Member, IEEE)

[1]Department of Computer Science and Technology, Tongji University, Shanghai, 201804, China (e-mail: chenliyan@tongji.edu.cn)
[2]Key Laboratory of Oceanographic Big Data Mining & Application of Zhejiang Province, Zhoushan Ocean University , Zhejiang 316022, PR China
[3]School of Electrical and Computer Engineering, University of Oklahoma, OK 74105, USA (e-mail: samuel.cheng@ou.edu)
[4]Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW U.K.(e-mail: kanghang.he@strath.ac.uk; vladimir.stankovic@strath.ac.uk; lina.stankovic@strath.ac.uk)

Corresponding author: Samuel Cheng (e-mail: samuel.cheng@ou.edu).

This research was supported by the European Union‘s Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement number 734331, and general scientific research projects of Zhejiang Education Department in 2020 under grant Y202044676.

ABSTRACT With the growing application of undirected graphs for signal/image processing on graphs and distributed machine learning, we demonstrate that the shift-enabled condition is as necessary for undirected graphs as it is for directed graphs.It has recently been shown that, contrary to the widespread belief that a shift-enabled condition (necessary for any shift-invariant filter to be representable by a graph shift matrix) can be ignored because any non-shift-enabled matrix can be converted to a shift-enabled matrix, such a conversion in general may not hold for a directed graph with non-symmetric shift matrix. This paper extends this prior work, focusing on undirected graphs where the shift matrix is generally symmetric. We show that while, in this case, the shift matrix can be converted to satisfy the original shift-enabled condition, the converted matrix is not associated with the original graph, that is, it does not capture anymore the structure of the graph signal. We show via examples, that a non-shift-enabled matrix cannot be converted to a shift-enabled one and still maintain the topological structure of the underlying graph, which is necessary to facilitate localized signal processing.

INDEX TERMS graph signal processing, shift-enabled graphs, shift-invariant filter, undirected graph

I. INTRODUCTION

GRAPH signal processing (GSP) extends classical digital signal processing (DSP) to signals on graphs, and provides a prospective solution to numerous real-world problems that involve signals defined on topologically complicated domains, such as social networks, point clouds, biological networks, environmental and condition monitoring sensor networks [1]. However, there are several challenges in extending classical DSP to signals on graphs, particularly related to the design and application of graph filters.

In classical, one-dimensional DSP, any linear, time-invariant, or shift-invariant, a filter that commutes with time-shift operator z^{-1} can be represented as a polynomial of z^{-1} leading to Z-transform of the filter. Conversely, if a linear filter can be represented as a polynomial of z^{-1}, the filter is linear and shift-invariant. Unfortunately, this concept does not simply generalize to GSP, partly because the definition of a "shift" for a graph is not obvious [2]. Commonly, in the GSP literature, a graph is uniquely described by a "shift matrix" or a "shift operator"[1], S [3]–[5], which has been extensively used for time/vertex (or spatial)-domain filter design (see [1], [2] and references therein for frequency-domain and time/vertex-domain filtering). For example, adjacency matrices, for general graphs, and (normalized) Laplacian matrices, for undirected graphs, are some popular choices for the shift matrix. Though the GSP framework has been developed for both directed graphs [6] and undirected graphs, undirected graphs have found wider applications for numerous image/signal processing tasks [1], [2] and distributed machine learning tasks [7], [8]. A graph is called an undirected graph if all edges between the vertices are bidirectional. This means that the adjacency matrix is symmetric, which implies that the combinatorial Laplacian matrix is symmetric and positive semi-definite. Then a frequency representation of graph signals can be developed based on real and non-negative eigenvalues of graph Laplacian.

Recently, various methods have been developed that apply computationally efficient graph filtering on large graphs including frequency-domain filtering via Fast Graph Fourier Transforms [9], Wavelet Transform [10], and filter deterministic approximations for empirical spectral statistics [11]. For example, Fast Graph Fourier Transform (FGFT) is used in [9] to implement a graph filter via sparse shift matrix decomposition. While the sparsity constraint in [9] helps reducing the implementation complexity, it does not impose any constraint on whether the approximately decomposed shift matrix maintains the structure of the original graph.

In practice, for a large data set, distributed computation, rather than centralized is needed [5]. A graph node may only have access to information collected by nodes near it, and in the extreme case, only to signals in its immediate neighbourhood. Even when the architecture is not completely distributed, for a large graph with millions or billions of vertices, a naive implementation of the graph filter through direct matrix multiplication will be computationally intractable [6], [12].

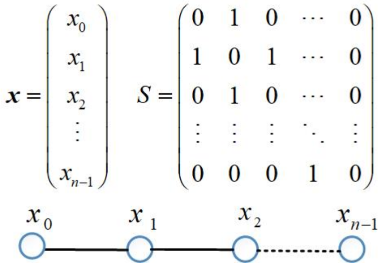

$x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{pmatrix}$
$S = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$
x_0
x_1
x_2
x_{n-1}

FIGURE 1: $\boldsymbol{x}$: graph signal; S: adjacency matrix of path graph which is shift-enabled.

In order to make graph filtering feasible, even for very large graphs, it is necessary to perform the filtering locally [13], [14]. For example, consider a sensor network represented by a graph, where the edges and edge weights of the graph depend on the distance between the sensors. In this case, efficient filtering boils down to merely mixing the signals acquired by a sensor with those of the nearest sensors. Otherwise, if the filter output at any graph vertex is a linear combination of inputs at *all* vertices, filtering will be practically infeasible for "big data" graphs [5]. Therefore, in practice, we expect that a node can only impose direct influence on an adjacent node through the shift operator. For practical design purposes, it is advantageous to be able to decompose filters in a form of a polynomial of such a shift matrix. An example is shown in Fig.1, where $\boldsymbol{x}$ is a graph signal and S is a sparse

[1] The term "shift" comes from the analogy with z^{-1} operator in $Z-$transform of classical DSP.

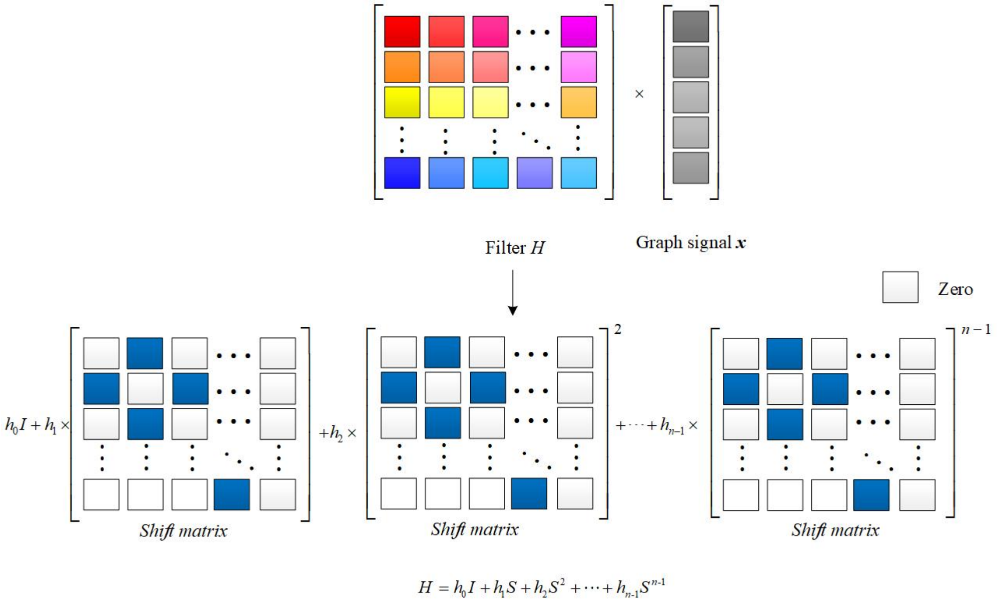


FIGURE 2: H is a polynomial in S. Different colors represent different values in matrices.

matrix accounting for the local structure of the path graph [15]. Just as shown in Fig.2, if the filter H can be represented as the polynomial of S, then the filter H can be computed locally at the nodes of the graph. The importance of this polynomial representation has been reiterated in a recent survey paper (see Section II.F of [1]).

Although a nice, but loose, analogy between S and z^{-1} can be established [1], unlike classical DSP, if a graph filter is shift-invariant (the shift matrix commutes with the target filter), this does not automatically imply that a polynomial representation of the filter exists [16]. Ref. [3] argues that, for any shift matrix S, there exists a converted shift matrix $\tilde{S}$ such that graph filter H is a polynomial in $\tilde{S}$. However, it is not sufficient just to have H to be represented as a polynomial of any arbitrary $\tilde{S}$. One should also ensure that $\tilde{S}$ indeed describes the same graph as S (see details in Definition 2), that is, the converted graph shift should keep the same topological structure as the original one. It was shown in [3] that any filter commuting with shift matrix S can be represented as a polynomial in S provided that the characteristic and minimal polynomial of the shift matrix are equal (in the rest of this paper, as in [17], we will refer to this condition as *shift-enabled condition*, see also Definition 1). However, in [3], this condition was immediately disregarded, surmising that one may convert any shift matrix that does not satisfy the shift-enabled condition into one that does. Based on this conclusion, most researchers currently assume that the shift-enabled condition simply holds or ignore the condition completely. However, it was proved in [17], through a counterexample, that such a conversion may not hold for a directed graph with an asymmetric shift matrix.

A. CONTRIBUTION

Extending our previous work that looked at directed graphs [17], in this paper, we focus on undirected graphs, and illustrate with examples that when the symmetric shift matrix of an undirected graph is non-shift-enabled, the conversion suggested in [3] could lead to a very different graph that does not necessarily capture the structure of the original graph signal. Namely, though the conversion would provide a shift-enabled graph that facilitates the polynomial representation of the shift-invariant filters, the newly designed graph might no longer capture the structure of the graph signal it was originally designed to model[2] However, [16] did not explicitly investigate the implication of the shift matrix conversion as proposed in [3], and does not facilitate localised filtering.

Referring to our wireless sensor network example in the introduction, in the original graph the output of the filtering at each vertex only involves inputs of the vertex's immediate neighbourhoods. However, in the converted graph, sensors that are far apart might be strongly connected, that is, each output at a vertex could be a linear combination of inputs at almost all vertices, thus filtering in such converted graph will be computationally unaffordable for "big data" graphs in practice which further emphasizes the importance of the shift-enabled condition [17].

In this manuscript we focus on the case when the topology of the graph is given (for example, by communication constraints due to sensor placement and communication protocols applied). In some other applications, mainly related to information processing via graphs, there is freedom to define the topology of the graph (or learn it from the data), in which case the shift matrix (and hence graph topology) can be optimised [12].

The outline of the paper is as follows. Section II describes the basic concepts and key properties of a shift-enabled graph. Section III provides examples to prove that the shift-enabled condition is essential for the symmetric graph. Section IV concludes the paper.

II. BASIC CONCEPTS AND PROPERTIES OF SHIFT-ENABLED GRAPHS

In this section, we briefly review the concepts of shift-enabled graphs and their properties relevant to this paper. For more details, see [2]–[5].

Let $\mathcal{G} = (V, A)$ be a graph, where $V = \{v_0, v_1, \cdots, v_{n-1}\}$ is a set of vertices and $A \in \mathbb{C}^{n\times n}$ is the adjacency matrix of the graph. Let $\boldsymbol{x} = (x_0, x_1, \cdots, x_{n-1})^T$ be a graph signal, where each sample $x_i \in \boldsymbol{x}$ corresponds to a vertex $v_i \in V$.

In particular, if $\mathcal{G}$ is a directed circular graph, then the corresponding adjacency matrix is given by: $A = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$. Then $A\boldsymbol{x} = (x_{n-1}, x_0, \cdots, x_{n-2})^T$, that is, multiplication by A shifts each signal sample to the next vertex. Thus, A is often called shift operator or shift matrix, which is similar to time-shift operator z^{-1} in DSP. In practice, other matrices, which reflect the structure of the graph, can be used to define a graph shift, such as the Laplacian matrix and the normalized Laplacian matrix for undirected graphs, and the probability transition matrix. Here, we use S to denote the general shift matrix, whether it is A, (normalized) Laplacian matrix, or the probability transition matrix.

In classical 1-D DSP, a shift-invariant filter F has a Z-transform (polynomial representation in z^{-1}), defined as

$$F(z^{-1}) = \sum_{k=-\infty}^{+\infty} f_k z^{-k},$$

where f_k is a polynomial coefficient. Moreover, from the shift-invariance property, it follows that the filtered output of a shifted input is equal to the shifted filtered output of the original input. In other words, the shift operation and the filter commute. That is, $Fz^{-1} = z^{-1}F$, which directly follows from the above polynomial representation (see, e.g., [16]).

Extending this concept to GSP, we also define a shift-invariant filter H as the one that com-

[2] Ref. [16] also reiterated the relationship among polynomial representation, shift-invariant, and alias-free filter.

mutes with the shift matrix, i.e., $HS = SH$. However, unlike in the classical DSP case, a shift-invariant filter does not necessarily have a polynomial representation in terms of the shift operator S. Yet, H can be represented as a polynomial in S if the shift matrix S satisfies the following condition.

Definition 1 (Shift-enabled graph [17]): A graph $\mathcal{G}$ is shift-enabled if its corresponding shift matrix S satisfies $p_S(\lambda) = m_S(\lambda)$, where $p_S(\lambda)$ and $m_S(\lambda)$ are the minimum polynomial and the characteristic polynomials of S, respectively. We also say that S is shift-enabled when the above condition is satisfied. Otherwise, S and the corresponding graph, are non-shift-enabled.

For shift-enabled graphs, the following theorem is the basis of linear, shift-invariant filter design.

Theorem 1: The shift matrix S is shift-enabled if and only if every matrix H commuting with S is a polynomial in S [3].

This theorem implies that as long as the shift matrix S does not satisfy the shift-enabled condition (i.e., $m_S(\lambda) \neq p_S(\lambda)$), there will always be some shift-invariant filters (and thus some filters) that cannot be represented as a polynomial of S. Ref. [3] de-emphasized the shift-enabled condition by suggesting that we may work around it with the following theorem.

Theorem 2 (Theorem 2 in [3]): For any shift matrix S, there exists a converted matrix $\tilde{S}$ and matrix polynomial $r(\cdot)$, such that $S = r(\tilde{S})$ and $m_{\tilde{S}}(\lambda) = p_{\tilde{S}}(\lambda)$.

While the above theorem is correct, it does not take into account that the target filter H may not be shift-invariant with respect to the converted shift matrix. In particular, for a directed graph, in general, S is not symmetric, and thus not jointly diagonalized with H. Consequently, one can show that generally there exists no converted shift-enabled $\tilde{S}$ that can maintain shift-invariance with the target filter when the graph is directed and S is asymmetric [17].

However, the conversion method suggested in [3] does hold for undirected graphs when H can be jointly diagonalized with S. Yet, as we will show in the following, the converted $\tilde{S}$ may not describe the same graph as the original S. This makes the whole conversion process moot. Hence, the shift-enabled condition is important regardless of whether the graph is directed or not (i.e., the shift matrix is asymmetric or not).

III. THE NECESSITY OF SHIFT-ENABLED CONDITION FOR UNDIRECTED GRAPHS

In this section, we demonstrate the necessity of shift-enabled condition for undirected graphs by examples. Naturally, we focus on cases when a shift-invariant filter H cannot be written as a polynomial in S, which is not shift-enabled. We follow the procedure of [3] to construct a shift-enabled graph $\tilde{\mathcal{G}}$ with shift matrix $\tilde{S}$, and investigate if this “conversion" facilitates local processing.

Before giving a concrete example, let us first review the conversion process described in [3]. As mentioned earlier, even though the conversion process does not hold for arbitrary shift matrices, it can be applied to symmetric shift matrices.

According to Lemma 2 in Appendix A, two symmetric and commuting matrices S and H are simultaneously diagonalizable. Thus, there exists an invertible matrix T such that $S = T\Lambda_S T^{-1}$ and $H = T\Lambda_H T^{-1}$, where Λ_S and Λ_H are composed of the eigenvalues of S and H, respectively. Then, a new matrix $\Lambda_{perturb}$ with distinct diagonal elements can be generated by slightly perturbing the values of Λ_S. The new shift matrix is calculated as $\tilde{S} = T\Lambda_{perturb}T^{-1}$. According to Lemma 1 and Lemma 2, the restructured shift matrix $\tilde{S}$ satisfies $p_{\tilde{S}}(\lambda) = m_{\tilde{S}}(\lambda)$ and $H\tilde{S} = \tilde{S}H$. Hence, from Theorem 1, H is a polynomial in $\tilde{S}$.

This simply demonstrates that, as advocated in [3], for undirected graphs, it is always possible to find a shift-enabled matrix $\tilde{S}$ such that $H = h(\tilde{S})$ exists. However, it is not sufficient to have H represented as a polynomial of any arbitrary $\tilde{S}$.

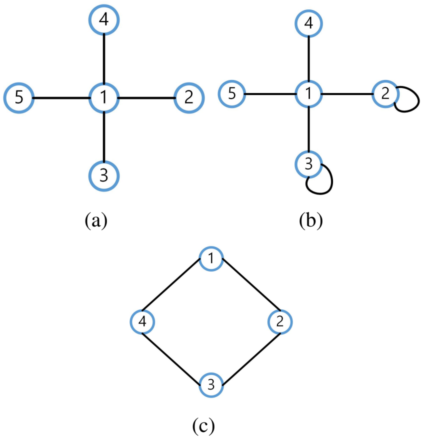
A natural and basic constraint is that the converted $\tilde{S}$ should facilitate “local processing", that is, it should describe topologically the same graph, which is essential in virtually all GSP applications, such as filter design [18], sampling [19], denoising [20], and classification [21], otherwise, the conversion will dramatically increase the calculation complexity. To ensure that the converted graph facilitates "local processing", that is, an implementation of an L-th order polynomial filter requires L data exchanges between neighbouring nodes [16], we introduce Definition 2. In fact, the definition of a matrix describing a graph (see details in Definition 2) is not new in spectral graph theory. In particular, the matrix of "loose description" is widely used in the context of an inverse eigenvalue problem and zero-forcing problem [22], [23]. We introduce "strict description" since we would like to accommodate graphs with self-loops. In a nutshell, two shift matrices describe the same graph if the conversion from one to another preserves the graph topological structure, implying that filtering under the converted graph can be performed locally. The precise definition is specified as follows.

Definition 2 ([22]–[24]): Shift matrices S and $\tilde{S}$ strictly describe the same graph if 1) $S_{i,j} \neq 0$ if and only if $\tilde{S}_{i,j} \neq 0$ for any i and j, and 2) $\tilde{S}$ is symmetric if and only if S is symmetric. We will say S and $\tilde{S}$ loosely describe the same graph if the first condition is relaxed to 1') $S_{i,j} \neq 0$ if and only if $\tilde{S}_{i,j} \neq 0$ only for $i \neq j$. That is, we allow some i where only $S_{i,i}$ or $\tilde{S}_{i,i}$ equal to 0.

We have the following theorem that shows that $\tilde{S}$ and S cannot describe the same graph structure.

Theorem 3: Following the conversion procedure of [3], it is impossible to guarantee the following three conditions to be satisfied simultaneously:

- C1. $\tilde{S}$ is shift-enabled (i.e., $p_{\tilde{S}}(\lambda) = m_{\tilde{S}}(\lambda)$).
- C2. H is shift-invariant on $\tilde{S}$ (i.e., $H\tilde{S} = \tilde{S}H$).
- C3. $\tilde{S}$ and S loosely[3] describe the same graph.



(a) (b)

(c)

FIGURE 3: Graph topology used in the examples. (a) Original graph with shift matrix S. (b) Converted shift matrix $\tilde{S}$ which loosely describes the same graph as S. (c) Cycle graph with shift matrix S'.

We prove this theorem by two examples (Sections III-A and III-B)

A. EXAMPLE 1 THAT $\tilde{S}$ CAN LOOSELY BUT NOT STRICTLY DESCRIBE THE ORIGINAL GRAPH

For Example 1, original shift matrix S is shift-invariant with respect to the filter matrix H (i.e., C2 hold). As C1 does not hold (i.e., $p_S(\lambda) \neq m_S(\lambda)$), the transformed new $\tilde{S}$ strictly describes the same graph as S is non-shift-enabled. However, if we relax the constraint on $\tilde{S}$, the shift-enabled matrix $\tilde{S}$ that loosely describes the graph can be found.

Let us start with a non-shift-enabled graph as shown in Fig.3(a) which is a star graph [4]. The

[3] Hence, it is also impossible for $\tilde{S}$ and S to strictly describe the same graph if C3 is not satisfied.

[4] The star graph of order n is a tree on n nodes with one node having vertex degree $n-1$ and the other $n-1$ having vertex degree 1 [25] .

shift matrix [5] of the undirected graph is

$$S = \begin{pmatrix} 0&1&1&1&1\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0 \end{pmatrix}. \tag{1}$$

It is clear that $p_S(\lambda) = \lambda^3(\lambda-2)(\lambda+2) \neq \lambda(\lambda-2)(\lambda+2) = m_S(\lambda)$ and hence S is non-shift-enabled.

Since shift-enabled condition is not just sufficient but also necessary [17], there must exist a shift-invariant filter not representable as a polynomial of S. Indeed, one example for such a filter is $H = \begin{pmatrix} 0&0&0&0&0\\ 0&1&-1&0&0\\ 0&-1&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{pmatrix}$. It can be readily verified that $HS = 0 = SH$ and thus the filter is shift-invariant, and it is impossible to find a polynomial representation of H in terms of S. Note that $S^n_{2,3} = S^n_{2,4}$ [6] for all $n \in \mathbb{N}$. Thus for any polynomial $h(S)$, we must have $h(S)_{2,3} = h(S)_{2,4}$. But since $H_{2,3} = -1 \neq 0 = H_{2,4}$, $H \neq h(S)$ for any polynomial function $h(\cdot)$.

1) Extension of H to a class of filters

We can extend H to the following class of filters that all cannot be represented as polynomials of S:

$$\mathbb{H} = \{\alpha H + q(S) | \alpha \in \mathbb{R}\},$$

where $q(S)$ is a polynomial of S.

Since apparently $q(S)S = Sq(S)$ for any polynomial $q(S)$ and $HS = SH$ as discussed above, any filter $\alpha H + q(S) \in \mathbb{H}$ commutes with S as well. Thus any filter in $\mathbb{H}$ is shift-invariant. However, since H is not representable as a polynomial of S, as discussed above, so does $\alpha H + q(S)$.

From the examples presented above, we note that when the shift-enabled condition is violated, we may find an infinite number of shift-invariant filters that are not representable as polynomials of S.

2) Shift-enabled $\tilde{S}$ that strictly describes the original graph does not exist

First, let us restrict the converted shift matrix $\tilde{S}$ to strictly describe the same graph as S. Thus $\tilde{S}$ could be written as

$$\tilde{S} = \begin{pmatrix} 0 & \tilde{S}_{1,2} & \tilde{S}_{1,3} & \tilde{S}_{1,4} & \tilde{S}_{1,5}\\ \tilde{S}_{1,2} & 0&0&0&0\\ \tilde{S}_{1,3} & 0&0&0&0\\ \tilde{S}_{1,4} & 0&0&0&0\\ \tilde{S}_{1,5} & 0&0&0&0 \end{pmatrix} \tag{2}$$

with non-zeros $\tilde{S}_{1,2}, \tilde{S}_{1,3}, \tilde{S}_{1,4}$, and $\tilde{S}_{1,5}$. We can readily verify that the characteristic polynomial is $p_{\tilde{S}}(\lambda) = \lambda^3(\lambda^2 - \tilde{S}^2_{12} - \tilde{S}^2_{13} - \tilde{S}^2_{14} - \tilde{S}^2_{15})$ and 0 is the triple eigenvalue of $\tilde{S}$. According to Lemma 1, a shift-enabled real symmetric shift matrix has to have unique eigenvalues and thus $\tilde{S}$ is not shift-enabled. Therefore, all graphs which have the same structure as Fig.3(a) are non-shift-enabled.

3) Shift-enabled $\tilde{S}$ that loosely describes the original graph exists

Next, let us relax $\tilde{S}$ so that it may just loosely describe the original graph. In other words, we allow the diagonal elements to be non-zero which maintains most of the topological structure of the original graph. In applications where diffusion or state transition matrices are treated as shift matrices, the diagonal elements can be interpreted as the returning probabilities of the current state to itself. Thus, the converted shift matrix $\tilde{S}$ can be written as

$$\tilde{S} = \begin{pmatrix} \tilde{S}_{1,1} & \tilde{S}_{1,2} & \tilde{S}_{1,3} & \tilde{S}_{1,4} & \tilde{S}_{1,5}\\ \tilde{S}_{1,2} & \tilde{S}_{2,2} & 0&0&0\\ \tilde{S}_{1,3} & 0 & \tilde{S}_{3,3} & 0&0\\ \tilde{S}_{1,4} & 0&0&\tilde{S}_{4,4}&0\\ \tilde{S}_{1,5} & 0&0&0&\tilde{S}_{5,5} \end{pmatrix}. \tag{3}$$

Many solutions that satisfy shift-enabled and shift-invariant conditions can be found. For instance,

$$\tilde{S} = \begin{pmatrix} 0&1&1&1&1\\ 1&1&0&0&0\\ 1&0&1&0&0\\ 1&0&0&0&0\\ 1&0&0&0&0 \end{pmatrix} \tag{4}$$

is one such solution, where the original graph structure is only slightly modified as shown in Fig.3(b). One can verify that the eigenvalues

[5] Without loss of generality, we choose the adjacency matrix as the shift matrix in the following examples.

[6] $S^k_{i,j}$ denotes the (i,j)-element of matrix S^k.

$(-1.8136, 0, 0.4707, 1, 2.3429)$ of $\tilde{S}$ are distinct and thus $\tilde{S}$ is shift-enabled. Moreover, one can also readily verify that $H\tilde{S} = \tilde{S}H$. By Theorem 1, the above two conditions ensure that H is a polynomial in $\tilde{S}$.

B. EXAMPLE 2 WHEN THE CONVERTED SHIFT MATRIX CAN NEITHER STRICTLY NOR LOOSELY DESCRIBE THE ORIGINAL GRAPH (PROOF OF THEOREM 3)

For Example 2, the original shift matrix S is shift-invariant with respect to the filter matrix H (i.e., $HS = SH$), but C1 does not hold (i.e., $p_S(\lambda) \neq m_S(\lambda)$). The transformed new $\tilde{S}$ cannot satisfy C1 and C3 simultaneously, that is, any matrix that either strictly or loosely describes the original graph is non-shift-enabled. This further emphasizes the importance of the shift-enabled condition.

There are situations where no shift-enabled $\tilde{S}$ exists even after we relax the graph structure constraint as in the earlier example.

Consider shift matrix

$$S' = \begin{pmatrix} 0&1&0&1 \\ 1&0&1&0 \\ 0&1&0&1 \\ 1&0&1&0 \end{pmatrix} \tag{5}$$

as shown in Fig.3(c).

It can easily be seen that the eigenvalues of S', $(0, 0, 2, -2)$, are not unique. Thus S' is non-shift-enabled according to Lemma 1. So we do expect that there exists shift-invariant filter not representable by S'. Indeed, we can easily show that filter $H' = \begin{pmatrix} 0&0&-1&1 \\ 0&-1&1&0 \\ -1&1&0&0 \\ 1&0&0&-1 \end{pmatrix}$ is such a filter.

First, $H'S' = S'H'$ and thus H' is shift-invariant under S'. Furthermore, $(S')^n_{1,2} = (S')^n_{1,4}$ for all $n \in \mathbb{N}$, and so $h(S')_{1,2} = h(S')_{1,4}$ for any polynomial $h(S')$. But since $H'_{1,2} = 0 \neq 1 = H'_{1,4}$, $H' \neq h(S')$ for any polynomial function $h(\cdot)$.

Let us prove that it is impossible to find a converted shift matrix $\tilde{S}'$ which is shift-enabled and commutes with H' by only changing the weights of nonzero and diagonal elements.

Consider a general symmetric matrix

$$\tilde{S}' = \begin{pmatrix} \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & 0 & \tilde{S}'_{1,4} \\ \tilde{S}'_{1,2} & \tilde{S}'_{2,2} & \tilde{S}'_{2,3} & 0 \\ 0 & \tilde{S}'_{2,3} & \tilde{S}'_{3,3} & \tilde{S}'_{3,4} \\ \tilde{S}'_{1,4} & 0 & \tilde{S}'_{3,4} & \tilde{S}'_{4,4} \end{pmatrix} \tag{6}$$

which has arbitrary weights on nonzero and diagonal elements. That is, $\tilde{S}'$ loosely describes the same graph as S'.

$H' = h(\tilde{S}')$ clearly implies that H' commutes with $\tilde{S}'$, namely, $H'\tilde{S}' = \tilde{S}'H'$ is a necessary condition for $H' = h(\tilde{S}')$. It follows from $H'\tilde{S}' = \tilde{S}'H'$ that $\tilde{S}'_{1,1} = \tilde{S}'_{2,2} = \tilde{S}'_{3,3} = \tilde{S}'_{4,4}$ and $\tilde{S}'_{1,2} = \tilde{S}'_{1,4} = \tilde{S}'_{2,3} = \tilde{S}'_{3,4}$, i.e.,

$$\tilde{S}' = \begin{pmatrix} \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & 0 & \tilde{S}'_{1,2} \\ \tilde{S}'_{1,2} & \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & 0 \\ 0 & \tilde{S}'_{1,2} & \tilde{S}'_{1,1} & \tilde{S}'_{1,2} \\ \tilde{S}'_{1,2} & 0 & \tilde{S}'_{1,2} & \tilde{S}'_{1,1} \end{pmatrix}. \tag{7}$$

Following Cayley-Hamilton Theorem [26], if H' is a polynomial in $\tilde{S}'$, then $H' = h(\tilde{S}') = h_0 I + h_1 \tilde{S}' + h_2 \tilde{S}'^2 + h_3 \tilde{S}'^3$, where I as the identity matrix. In fact, it is easy to prove that $(\tilde{S}')^k_{1,2} = (\tilde{S}')^k_{1,4}$, for $k = 0, 1, 2, 3$. Hence, $h(\tilde{S}')_{1,2} = h(\tilde{S}')_{1,4}$ which contradicts with $H'_{1,2} \neq H'_{1,4}$. Thus, for this example, the filter H' cannot be represented as a polynomial in the converted shift matrix $\tilde{S}'$ which even just loosely describes the original graph.

C. DISCUSSION

1) Extend the conclusion to weighted graph

Equation (1) and (5) in this section can be easily extended to symmetric weighted graphs. Assume their shift matrices are $S_{weight} = \begin{pmatrix} 0 & S_{1,2} & S_{1,3} & S_{1,4} & S_{1,5} \\ S_{1,2} & 0 & 0 & 0 & 0 \\ S_{1,3} & 0 & 0 & 0 & 0 \\ S_{1,4} & 0 & 0 & 0 & 0 \\ S_{1,5} & 0 & 0 & 0 & 0 \end{pmatrix}$ and $S'_{weight} = \begin{pmatrix} 0 & S'_{1,2} & 0 & S'_{1,4} \\ S'_{1,2} & 0 & S'_{2,3} & 0 \\ 0 & S'_{2,3} & 0 & S'_{3,4} \\ S'_{1,4} & 0 & S'_{3,4} & 0 \end{pmatrix}$, respectively. In fact, S_{weight} coincides with (2), and S'_{weight} is a special form of (6). According to the previous discussion, conclusions in Section III still hold for weighted graphs. That is, the shift-enabled condition is also necessary for weight graph.

2) Conversion from non-shift-enabled to shift-enabled graph

When the filter matrix H is not a polynomial of the shift matrix S, authors in [12] address an optimized approximation of H for a fixed S in Section III.B of [12]. This appears to be a promising solution. But in practice, we are of course more concerned about H than S. So we should consider how we can adjust S (but try to maintain the topological structure of S as much as possible) so that we only need a minimum change in H. Note that not changing H at all in general is not always possible as shown in Section III-B but at the same time the example also does not rule out the possibility that modifying S alone is sufficient. Authors in [12] also suggest a way to design S for a fixed H in Section III.C of [12]. But the method is only limited to rank-1 H. So while the idea from Section III.C of [12] is interesting, it is quite limiting in practice as well.

Consequently, when the shift-enabled condition for S and $HS = SH$ are not simultaneously satisfied, finding an appropriate representation for H is really a rather complex and difficult problem and we will leave it to future studies.

IV. CONCLUSIONS AND FUTURE WORK

For a non-shift-enabled graph, even if we can easily "transform" the symmetric shift matrix S into one that satisfies the shift-enabled condition, the new $\tilde{S}$ may be irrelevant since it describes a very different graph from S. That is, the operator $\tilde{S}$ on a graph signal may involve mixing inputs far beyond its neighbourhood and become impractical for huge graphs. Combined with the necessity of the shift-enabled condition for directed graph [17], we demonstrated in this paper that the shift-enabled condition is essential for any graph structure. Good future directions to this work are to explore a shift that "approximately" describes the original graph as the conversion is quite different and analyze when it is worth applying the new shift-enabled operator according to additional communications costs or constraints. In particular, if it is already known that one such shift does not exist, one possible direction to explore shift "approximately" describes the original graph instead (some nonzero off-diagonal element may not correspond to an actual edge). But we will leave this to future study.

Note that even though we consider the adjacency matrix as the shift matrix in our examples, the conclusion applies to other shift matrices. In particular, one can readily verify that the conclusion still holds if we use the Laplacian matrix as the shift matrix in the example in Section III-B.

V. APPENDIX

A. LEMMAS

It is easily determined whether a graph is shift-enabled by the following lemmas.

Lemma 1: If shift matrix S is a real symmetric matrix, then S is shift-enabled, if and only if all eigenvalues of S are distinct [1].

Lemma 1 indicates that an undirected graph is shift-enabled if and only if its eigenvalues are all distinct.

As both shift matrix S and filter matrix H are symmetric, we can obtain the following lemma.

Lemma 2: If shift matrix S and filter matrix H are diagonalizable (this condition always holds for symmetric matrix) then S and H are simultaneously diagonalizable (by an invertible matrix) if and only if $HS = SH$ (see Theorem 1.3.12 in [27]).

B. EXTENSION OF STAR GRAPH IN SECTION III-A

The example from Section III-A can be extended to a star graph with more than five vertices. In this case, the shift matrix of a star graph with N vertices is $S_N = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}$. Let a filter be $H_N = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$ and according to (2), shift operator that strictly describes the original graph satisfies $\tilde{S}_{N(strict)} = \begin{pmatrix} 0 & \tilde{S}_{1,2} & \tilde{S}_{1,3} & \cdots & \tilde{S}_{1,N} \\ \tilde{S}_{1,2} & 0 & 0 & \cdots & 0 \\ \tilde{S}_{1,3} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{S}_{1,N} & 0 & 0 & \cdots & 0 \end{pmatrix}$.

Repeating the similar argument as before, we can readily verify that the following five conclusions still hold simultaneously: (i) S_N is not shift-enabled for $p_{S_N}(\lambda) = \lambda^{N-2}(\lambda^2 - (N-1))$. (ii) H_N is shift-invariant, i.e., $H_N S_N = 0 = S_N H_N$. (iii) $H_N \neq h(S_N)$ for any polynomial function $h_N(\cdot)$. (iv) H_N can be extended to the following class of filters that none can be represented as polynomials of S_N:

$$\mathbb{H}_{\mathbb{N}} = \{\alpha H_N + q(S_N) | \alpha \in \mathbb{R}\},$$

where $q(S_N)$ is a polynomial of S_N. (v) Shift-enabled $\tilde{S}_{N(strict)}$ can not strictly describes the original graph.

ACKNOWLEDGMENT

The authors thank B. Zhao for helpful discussions.

REFERENCES

[1] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” Proceedings of the IEEE, vol. 106, no. 5, pp. 808–828, 2018.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” IEEE Signal Proc. Magazine, vol. 30, no. 3, pp. 83–98, 2013.

[3] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs,” IEEE Trans. Signal Processing, vol. 61, no. 7, pp. 1644–1656, 2013.

[4] ——, “Discrete signal processing on graphs: Frequency analysis.” IEEE Trans. Signal Processing, vol. 62, no. 12, pp. 3042–3054, 2014.

[5] ——, “Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure,” IEEE Signal Processing Magazine, vol. 31, no. 5, pp. 80–90, 2014.

[6] M. Coutino, E. Isufi, and G. Leus, “Advances in distributed graph filtering,” IEEE Transactions on Signal Processing, vol. 67, no. 9, pp. 2320–2333, 2019.

[7] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, “Graph signal processing for machine learning: A review and new perspectives,” IEEE Signal Processing Magazine, vol. 37, no. 6, pp. 117–127, 2020.

[8] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, “Multitask learning over graphs: An approach for distributed, streaming machine learning,” IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 14–25, 2020.

[9] L. Le Magoarou, R. Gribonval, and N. Tremblay, “Approximate fast graph fourier transforms via multilayer sparse approximations,” IEEE transactions on Signal and Information Processing over Networks, vol. 4, no. 2, pp. 407–420, 2017.

[10] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” Applied and Computational Harmonic Analysis, vol. 30, no. 2, pp. 129–150, 2011.

[11] S. Kruzick and J. M. Moura, “Graph signal processing: Filter design and spectral statistics,” in 2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP). IEEE, 2017, pp. 1–5.

[12] S. Segarra, A. G. Marques, and A. Ribeiro, “Optimal graph-filter design and applications to distributed linear network operators,” IEEE Transactions on Signal Processing, vol. 65, no. 15, pp. 4117–4131, 2017.

[13] A. Loukas, A. Simonetto, and G. Leus, “Distributed autoregressive moving average graph filters,” IEEE Signal Processing Letters, vol. 22, no. 11, pp. 1931–1935, 2015.

[14] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, “Filtering random graph processes over random time-varying graphs,” IEEE Transactions on Signal Processing, vol. 65, no. 16, pp. 4406–4421, 2017.

[15] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Stationary graph processes and spectral estimation,” IEEE Transactions on Signal Processing, vol. 65, no. 22, pp. 5911–5926, 2017.

[16] O. Teke and P. Vaidyanathan, “Linear systems on graphs,” in 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, 2016, pp. 385–389.

[17] L. Chen, S. Cheng, V. Stankovic, and L. Stankovic, “Shift-enabled graphs: Graphs where shift-invariant filters are representable as polynomials of shift operations,” IEEE Signal Processing Letters, vol. 25, no. 9, pp. 1305–1309, 2018.

[18] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” Applied and Computational Harmonic Analysis, vol. 30, no. 2, pp. 129–150, 2011.

[19] A. Gadde, A. Anis, and A. Ortega, “Active semi-supervised learning using sampling theory for graph signals,” in Proc. 20th ACM SIGKDD Intl. Conf. Knowledge Discovery & Data Mining, 2014, pp. 492–501.

[20] C. Yang, G. Cheung, and V. Stankovic, “Estimating heart rate and rhythm via 3d motion tracking in depth video,” IEEE Transactions on Multimedia, vol. 19, no. 7, pp. 1625–1636, 2017.

[21] K. He, L. Stankovic, J. Liao, and V. Stankovic, “Non-intrusive load disaggregation using graph signal processing,” IEEE Transactions on Smart Grid, vol. 9, no. 3, pp. 1739–1747, 2018.

[22] L. Hogben, “Spectral graph theory and the inverse eigen-

value problem of a graph," Electronic Journal of Linear Algebra, vol. 14, no. 1, p. 3, 2005.

[23] M. Trefois and J.-C. Delvenne, "Zero forcing number, constrained matchings and strong structural controllability," Linear Algebra and its Applications, vol. 484, pp. 199–218, 2015.

[24] S. M. Fallat and L. Hogben, "The minimum rank of symmetric matrices described by a graph: a survey," Linear Algebra and its Applications, vol. 426, no. 2-3, pp. 558–582, 2007.

[25] S. Pemmaraju and S. Skiena, "Cycles, stars, and wheels," Computational Discrete Mathematics Combinatiorics and Graph Theory in Mathematica, pp. 284–249, 2003.

[26] P. Lancaster and M. Tismenetsky, The theory of matrices: with applications. Elsevier, 1985.

[27] R. A. Horn and C. R. Johnson, Matrix analysis. Cambridge University Press, 2012.

LIYAN CHEN received the B.S. degree in mathematics and applied mathematics from Wenzhou Normal University, China, in 2002 and the M.S. degree in basic mathematics from Hangzhou Normal University, China, in 2005. She is currently pursuing the Ph.D. degree in computer science at Tongji University.

Since 2005, she has been with the School of Information Engineering, Zhejiang Ocean University in China, where she is currently a lecturer. Her research interests include graph signal processing and graph theory.

SAMUEL CHENG (S'00–M'04–SM'12) received the B.S. degree in electrical and electronic engineering from the University of Hong Kong, the M.Phil. degree in physics from the Hong Kong University of Science and Technology, the M.S. degree in electrical engineering from the University of Hawaii, Honolulu, and the Ph.D. degree in electrical engineering from Texas A&M University, in 2004.

He was with Microsoft Asia, China, and Panasonic Technologies Company, NJ, in the areas of texture compression and digital watermarking in the summers of 2000 and 2001. In 2004, he joined Advanced Digital Imaging Research, a research company-based near Houston, TX, as a Research Engineer to perform biomedical imaging research, and was promoted to Senior Research Engineer in 2005. Since 2006, he has been with the School of Electrical and Computer Engineering, University of Oklahoma, where he is currently an Associate Professor. He visited Tongji University, Shanghai, as a Professor in 2015. He holds six U.S. patents in miscellaneous areas of signal processing. His research interests include machine learning, image/signal processing, and pattern recognition. He is a member of ACM.

KANGHANG HE received the M.Eng. degree and the Ph.D. degree in electronic and electrical engineering from the University of Strathclyde, Glasgow, U.K., in 2015 and 2020. His research interests in graph signal processing applications includes power disaggregation, eye trackers, and electroencephalography.

VLADIMIR STANKOVIC (M'03–SM'10) received the Dr.Ing. (Ph.D.) degree from the University of Leipzig, Leipzig, Germany, in 2003.

From 2003 to 2006, he was with Texas A&M University, College Station, first as a Research Associate and then as a Research Assistant Professor. From 2006 to 2007, he was with Lancaster University. Since 2007, he has been with the Department Of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, where he is currently a Reader. He has co-authored four book chapters and over 160 peer-reviewed research papers, and holds five U.S. patents in signal processing. His research interests include multimedia processing for health monitoring, user-experience driven image processing, and communications and energy disaggregation. He was an IET TPN Vision and Imaging Executive Team Member, an Associate Editor of the IEEE COMMUNICATIONS LETTERS, a member of the IEEE Communications Review Board, and a Technical Program Committee Co-Chair of EUSIPCO in 2012. He was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and is currently an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Editor-at-Large of the IEEE TRANSACTIONS ON COMMUNICATIONS, and Area Editor of Elsevier Signal Processing: Image Communication.

LINA STANKOVIC (M'03–SM'12) received the B.Eng. (Hons.) degree in electronic communications engineering and the Ph.D. degree from Lancaster University, in 1999 and 2003, respectively. Her Ph.D. was sponsored by QinetiQ on developing an RF receiver, including coding, equalization, and synchronization for combat radio.

She is currently a Lecturer/Assistant Professor with the University of Strathclyde, Glasgow. Before joining Strathclyde in 2007, she was with Lancaster University as a Research Associate and then has been a Lecturer since 2006. She has been with BT Labs, Martlesham Heath, on digital video streaming, and Philips Research Eindhoven, on signal acquisition and processing from 2-D optical discs. She has authored four book chapters and 107 peer-reviewed research articles. Her main research areas lie in smart monitoring from sensor network platforms and meaningful information extraction that is user-centric as opposed to technology-centric. Her key focus areas are signal and image processing of biomedical data for motion assessment and understanding human behavior, and algorithmic analysis of energy data to understand how people use appliances in the home.