Designing Asymmetric Shift Operators for Decentralized Subspace Projection

Siavash Mollaebrahim, Baltasar Beferull-Lozano, Senior Member, IEEE

Abstract

A large number of applications in wireless sensor networks includes projecting a vector of noisy observations onto a subspace dictated by prior information about the field being monitored. In general, accomplishing such a task in a centralized fashion, entails a large power consumption, congestion at certain nodes, and suffers from robustness issues against possible node failures. Computing such projections in a decentralized fashion is an alternative solution that solves these issues. Recent works have shown that this task can be done via the so-called graph filters where only local inter-node communication is performed in a distributed manner using a graph shift operator. Existing methods designed the graph filters for symmetric topologies. However, in this paper, the design of the graph shift operators to perform decentralized subspace projection for asymmetric topologies is studied.

Index Terms

Wireless sensor networks, subspace projection, graph signal processing, graph filters.

I. INTRODUCTION

Processing and analysis of data-sets gathered in different settings such as social and economic networks, information networks, as well as infrastructure networks like Wireless Sensor Networks (WSN) is of high importance in many applications where decentralized methods are required. These data-sets are structured and can be represented over graphs. Graph signal Processing (GSP) is a powerful tool that enables us to process and analyze the graph-supported signals in different applications such as denoising and reconstruction of sensor data. In this area, traditional signal processing processing tools defined on regular domains, are re-designed and developed to operate on the graph domain. For instance, the notations such as sampling and frequency analysis have been extended to the graph-supported signals [1]–[4]. In [4], graph filters (GFs) have been introduced as polynomials of the so-called graph-shift operator, which is a local operator. Moreover, in [5], the design of GFs to implement a pre-specific linear transformation has been studied.

In addition, processing data in general over WSNs, Internet of Things (IoT) or other types of multi-agent networks is one of the main goals of decentralized signal processing. In many applications of interest, processing data should be performed in a decentralized fashion i.e only by exchanging local information among nodes. There are several reasons for this. For example, in centralized methods where there is fusion center, a node failure or the disruption of some communication link (either direct transmission or multi-hop communication) to the fusion center or sink node, supposes a clear limitation in terms of robustness with respect to a possible failure of the sink node or a node close to the sink node. Indeed, when a node failure or link disruption occurs for a node that is close to the sink node, this causes the loss of a large part of the sensor data. Furthermore, in some applications where there exist privacy issues, the agents are not allowed to send their local data to the sink node. Moreover, since in WSNs there are usually stringent power budget constraints, having continuous communication to a remote fusion center may also be unfeasible. On the other hand, decentralized processing enables us to perform in-network computations,
allowing for instance some decisions to be made by intelligent nodes or agents. This property is vital for enabling real-time distributed wireless control in many industrial applications.

Subspace projection is one of the most important instances of distributed data processing. This problem not only can be regarded as a denoising or noise reduction task, but it is also directly connected to many other estimation problems [6]. Let us assume a WSN network with a certain number of nodes \((N)\), where the gathered sensor measurements are assumed to be unreliable because of observation noise or erroneous data (e.g., sensor malfunction), leading to a discrete noisy signal. In many physical fields of interest, the useful signal belongs typically to a subspace that has a dimension \(r\) that is much smaller than \(N\). The subspace projection problem is to estimate the useful signal given the received noisy signal and the given subspace (i.e., expressed in terms of a matrix whose columns span the useful signal subspace).

GFs can be designed and implemented to perform decentralized subspace projection. In [5], [7], by using GFs, some methods have been proposed for a special case of subspace projection task (average consensus). Theses methods are capable of converging after a finite number of iterations. In addition, more general scenarios of subspace projection have been considered in [5]. However, the proposed schemes need knowledge of the graph shift operator or matrix (e.g., Laplacian or adjacency matrix have been used as the default graph shift operators), or restrict themselves to design GFs for rank-1 projections. To address these limitations, in [8], [9], a symmetric graph shift operator has been designed to compute GFs that can accomplish decentralized subspace projection after a finite number of iterations. The graph shift operator is obtained by optimizing a criterion that yields convergence to the subspace projection in a nearly minimal number of iterations.

On the other hand, the methods in [5], [6], [8], [9] have been proposed for undirected graph networks which means that they consider symmetric topologies. However, in this paper, we consider the problem of decentralized subspace projection via GFs for asymmetric topologies. For this, we develop a new methodology based on the Schur matrix decomposition and formulating an optimization problem that exploits this decomposition, in order to minimize the number of iterations (i.e., number of inter-node exchanges) until convergence.

The contributions of this paper can be enumerated as follows: a) characterization of the existence and properties of an asymmetric graph shift matrix, such that it is possible to construct a graph filter for asymmetric topologies, exploiting the Schur Decomposition, and which implements the projection operator, b) formulation of an optimization problem to design a fast graph filter c) ADMM-based algorithm to find efficiently the solution to our graph filter design problem, d) extensive experimental results showing that our proposed method compute exactly the projection operator.

The rest of the paper is structured as follows. In section II notations and reviews some existing results on decentralized subspace projection with graph filters is introduced. Section III presents the proposed algorithm. Finally, Section IV validates its performance through numerical experiments and section V concludes the paper.

II. DECENTRALIZED SUBSPACE PROJECTION PROBLEM

Consider \(N\) networked sensor nodes or agents that can exchange messages with their connected neighbors. The network is modeled as a directed connected graph \(G(V, E)\), where the vertex set \(V := 1, 2, \cdots, N\) correspond to the network agents, and \(E\) represents the set of edges. The \(n\)-th vertex \(v_n\) is connected to \(v_n\) if there is a directed edge from \(v_{n'}\) to \(v_n\) \((v_{n'}, v_n) \in E\), but this does not mean that \(v_n\) is connected to \(v_{n'}\) unless \((v_{n'}, v_n) \in E\). The in-neighborhood of the \(n\)-th node is defined as the set of nodes connected to it, which can be denoted as \(N_n = \{(v_{n'}, (v_{n'}, v_n) \in E)\}\).

The observation vector \(y = x + n\) \((y \in \mathbb{R}^N)\) contains noisy information gathered by nodes, where \(x\) and \(n\) are the useful signal and additive noise, respectively. The \(n\)-th entry of \(y = [y_1, y_2, \cdots, y_N]\) \((y_n)\) denotes information gathered by the \(n\)-th node. In the subspace projection context, the useful signal typically lies on a subspace of a dimension \(r\), much smaller than \(N\), which means that \(x = U\alpha\) where \(U \in \mathbb{R}^{N \times r}\) is a matrix whose columns span the useful signal subspace and \(\alpha \in \mathbb{R}^r\) [6], [8].
Noise reduction can be obtained by projecting \( y \) onto the useful signal subspace which equals the least-squares estimate of \( x \), denoted by \( \hat{x} \), given by:

\[
\hat{x} = U_{\parallel} U_{\parallel}^T y \triangleq Py
\]

where \( P \in \mathbb{R}^{N \times N} \) is the projection matrix. Estimating the useful signal \( x \) from the observation signal \( y \) and the knowledge of the subspace matrix \( U_{\parallel} \), is the subspace projection problem.

As stated earlier, decentralized subspace projection can be performed by using GFs. In this case, we are also given a certain network connectivity topology so that each node has a certain set of neighbour nodes it can reach and also there is a certain set of neighbour nodes from which it can be reached. Let us first introduce the concept of graph shift matrix. Any matrix that satisfies \( S_{(n,n')} = 0 \) if \( (v_n, v_n') \notin \mathcal{E} \) is a feasible graph shift matrix, which implies also that \( S \) characterises the underlying network topology.

A graph filter is a linear combination of successively shifted graph signals i.e. \( H := \sum_{l=0}^{L} c_l S_l \) where \( \{c_l\}_{l=0}^{L-1} \) are the filter coefficients, and \( L \) is the order of the filter. The procedure is that all nodes exchange their information with their neighbours (\( y_n \) is the \( n \)-th node observed noisy signal sample), so that the signal information of all nodes is updated via \( y^{(1)} = Sy \). For the next iteration, we have \( y^{(2)} = S y^{(1)} = S^2 y \). This procedure is repeated for \( L \) iterations. Thus, a graph filter can be computed in a decentralized fashion. To compute subspace projection via GFs, we need to have \( P = H \). It has been shown in previous work \([5]\) that with some appropriate given choices of shift matrix (typically Laplacian or Adjacency matrices), the filter coefficients for \( P = \sum_{l=0}^{L} c_l S_l \) can be found by solving a linear system of equations. GFs have been also designed for rank-1 projections in \([5]\), which is a special case of subspace projection. In \([8]\), valid symmetric shift matrices are found by optimizing a criterion related to minimizing the filter order \( (L) \), i.e. number of iterations or node information exchanges required for convergence to the projection matrix after a finite number of iterations. In this paper, we extend the formulation so that our optimization problem aims at finding the best graph shift non-necessarily symmetric matrix \( S \) that minimizes the required filter order, that is, removing the constraint of symmetry in the graph shift operator. As we show next, allowing for asymmetric matrices, makes it necessary to develop a different design method.

### III. Problem Formulation and Proposed Method

Our main problem can be formally stated as follows:

**Given:** i) A matrix \( U_{\parallel} \in \mathbb{R}^{N \times r} \) whose columns span the subspace of interest and ii) the set of edges \( \mathcal{E} = \{(n_1, n_1'), (n_2, n_2'), \ldots, (n_{|\mathcal{E}|}, n_{|\mathcal{E}|}')\} \) defining the network topology.

**Find:** An asymmetric matrix \( S \in \mathbb{R}^{N \times N} \) (the graph shift matrix) and vector \( c = [c_0, c_1, \ldots, c_L] \in \mathbb{R}^{L+1} \) (the filter coefficients) such that:

- \( \sum_{l=0}^{L} c_l S_l = U_{\parallel} U_{\parallel}^T \Rightarrow \) Polynomial shift matrix
- \( S_{n,n'} = 0 \) for all \( (v_n, v_{n'}) \notin \mathcal{E} \) \Rightarrow \) Topological shift matrix

The first and the second condition mean that \( S \) is polynomially and topologically feasible, respectively. Finding polynomial and topological shift matrices, which provide fast convergence, is the main goal of this paper. In order to achieve this, we need to introduce the following Schur matrix factorization.

**Schur Decomposition:** If \( S \in \mathbb{R}^{N \times N} \), then it can be decomposed as follows \([10]\): \n
\[
S = W TW^T
\]

where \( T \) is upper quasi-triangular and \( W \) is orthogonal. In fact, \( T \) has the following form:

\[
\begin{bmatrix}
B_1 & * & \cdots & * \\
0 & B_2 & \cdots & * \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & B_k
\end{bmatrix}
\]
The diagonal blocks $B_i$ are either $1 \times 1$ or $2 \times 2$ matrices. A $1 \times 1$ block corresponds to a real eigenvalue, and a $2 \times 2$ block corresponds to real matrix whose eigenvalues are a pair of complex conjugate eigenvalues. Consequently, $S$ can be decomposed as:

$$S = W(D + Q)W^T$$

(4)

where $D$ is diagonal, $Q$ is upper triangular with diagonal elements being zero and $W$ is orthogonal. The next Theorem states that the problem of finding polynomial asymmetric shift matrices is feasible.

**Theorem 1:** Given a matrix $U_\perp \in \mathbb{R}^{N \times N-r}$ with orthonormal columns that satisfy $R(U_\perp) = R^\perp(U_\parallel)$ and $P = U_\parallel U_\perp^T$, then there exists $S \in \mathbb{R}^{N \times N}$ with real distinct eigenvalues and $\{c_l\}$ such that $\sum_{l=0}^{L} c_l S^l = P$ with $L \leq N - 1$.

**Proof:** As we show next, it is sufficient to consider the set of matrices $S$ with real eigenvalues for finding a valid polynomial shift matrix. From the Schur decomposition of $S$, we have that $\sum_{l=0}^{L} c_l S^l = \sum_{l=0}^{L} c_l W(D + Q)^l W^T$, and the condition for $S$ to be a polynomial shift matrix can be re-written as:

$$\sum_{l=0}^{L} c_l S^l = \sum_{l=0}^{L} c_l [W \parallel W_\perp](D + Q)^l [W \parallel W_\perp]^T = [U \parallel U_\perp] \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} [U \parallel U_\perp]^T$$

(5)

where: $R(U_\perp) = R^\perp(U_\parallel)$. If $\exists \{c_l\} : \sum_{l=0}^{L} c_l(D + Q)^l = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}$ and $W_\parallel W_\perp^T = U_\parallel U_\perp^T$, this implies directly that the ranges of $W_\parallel$ and $U_\parallel$ are the same, where $W = [W \parallel W_\perp]$ and $W_\parallel \in \mathbb{R}^{N \times r}$. Then, we have that:

$$\sum_{l=0}^{L} c_l S^l = \sum_{l=0}^{L} c_l W(D + Q)^l W^T = W_\parallel W_\perp^T = U_\parallel U_\perp^T = P$$

(6)

which implies that $S$ is a polynomial shift matrix. Let us now consider the required coefficients $\{c_l\}$. From (5) and (6), to find a graph shift matrix $S$, we need coefficients $\{c_l\}$ such that:

$$\sum_{l=0}^{L} c_l(D + Q)^l = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}$$

(7)

To find the filter coefficient vector $(c = [c_0, c_1, \ldots, c_{L-1}]^T)$, we can rewrite (7) as a set of equations in the form $Tc = b$ where $T \in \mathbb{R}^{N^2 \times L}$ obtained via $\sum_{l=0}^{L} (D + Q)^l$ and $b = \text{vec} \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}$. Since $D$ is diagonal and $Q$ is upper triangular, $(N^2 - N)/2$ entries of $T$ are always zero. Thus, we can remove them reducing the size of the matrix $T$ to another matrix $T' \in \mathbb{R}^{(N^2 + N)/2 \times L}$. If $L = (N^2 + N)/2$ and the columns of $T'$ are independent, the filter coefficients can be found easily since in this case, $T'$ is square and full rank, which means that it is also invertible. To show this, let us consider a matrix $D = \text{diag}\{[\lambda_1, \lambda_2, \ldots, \lambda_N]\}$ with all the diagonal elements being real values and we show that it is possible to find a valid solution in this case. A number of $T'$ rows consist of just the eigenvalues of $S$, and the other ones are combinations of $Q$ and $D$ entries. Then, the set of rows of $T'$ associated just to $S$ eigenvalues, can be grouped as:

$$\begin{bmatrix}
1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^L \\
1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^L \\
: & : & : & \cdots & : \\
1 & \lambda_N & \lambda_N^2 & \cdots & \lambda_N^L
\end{bmatrix}$$

(8)

which is a Vandermonde matrix. Then, notice that it is possible to choose the values $\lambda_1, \ldots, \lambda_N$ to be distinct, in which case the columns of (8) become independent, making columns of $T'$ independent. Therefore, there exists a matrix $S$ with different real eigenvalues which is polynomially feasible with
\[ L = (N^2 + N)/2 \]. To finish the proof, we use next the Cayley-Hamilton Theorem [11].

Based on the Cayley-Hamilton Theorem, if there is a matrix \( S \) for which \( \exists \{c_i\}_{i=1}^{L} \) with \( L = (N^2 + N)/2 \), such that \( S \) is polynomially feasible, then there exists a set of coefficients \( \hat{c} \) such that \( \sum_{i=0}^{L} \hat{c}_i S^i = \mathbf{P} \) with \( \hat{L} \leq N - 1 \). Therefore, we can set \( L := \hat{L} \leq N - 1 \) and \( c := \hat{c} \).

\[ \square \]

Next, we propose a method to obtain a valid \( S \in \mathbb{R}^{N \times N} \) (with real eigenvalues) i.e. a topological and polynomial graph shift operator. First of all, to have a polynomial shift matrix, we need to have \( \sum_{i=0}^{L} c_i S^i = \mathbf{U}_\parallel \mathbf{U}_\parallel^\top \). If \( S \) is decomposed based on its Schur decomposition, from the polynomial condition (7), we have:

\[
\sum_{i=0}^{L} c_i S^i = \sum_{i=0}^{L} c_i (D + Q)^i W^\top = \mathbf{U}_\parallel \mathbf{U}_\parallel^\top
\]  

(9)

can be rewritten as:

\[
\sum_{i=0}^{L} c_i (W_\parallel W_\perp)(D + Q)^i [W_\parallel W_\perp]^\top = [\mathbf{U}_\parallel \mathbf{U}_\parallel] \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} [\mathbf{U}_\parallel \mathbf{U}_\parallel]^\top
\]  

(10)

where: \( \mathcal{R}(\mathbf{U}_\perp) = \mathcal{R}^\perp(\mathbf{U}_\parallel) \). Based on the proof of Theorem 1, we need to have: \( \mathbf{W}_\parallel \mathbf{W}_\parallel^\top = \mathbf{U}_\parallel \mathbf{U}_\parallel^\top \), with the elements of \( D \) being distinct. Thus, to design \( \mathbf{W} = [\mathbf{W}_\parallel \mathbf{W}_\perp] \), we can choose \( \mathbf{W}_\parallel = \mathbf{E}_\parallel \mathbf{U}_\parallel \) and \( \mathbf{W}_\perp = \mathbf{E}_\perp \mathbf{U}_\perp \) where \( \mathbf{E}_\parallel, \mathbf{E}_\perp \) are orthogonal matrices, which can be chosen arbitrarily. This guarantees that \( \mathbf{W}_\parallel \mathbf{W}_\parallel^\top = \mathbf{U}_\parallel \mathbf{U}_\parallel^\top \). In the next steps, we focus on the design of \( D \) and \( Q \). For the rest of paper, we assume, without loss of generality, that the matrix \( S \) has real eigenvalues.

From Theorem 1, we know that to compute the projection matrix exactly via the graph filter, the condition \( \sum_{i=0}^{L} \hat{c}_i (D + Q)^i = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} \) has to be satisfied.

Our goal is to minimize the order \( \hat{L} \) of the graph filter in order to converge as fast as possible to this condition. Let us consider \( D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix}, D_1 \in \mathbb{R}^{r \times r} \) and \( D_2 \in \mathbb{R}^{N-r \times N-r} \). Then, our approach is to make \( Q \) and \( D_2 \) close to \( 0_{N \times N} \) and \( 0_{N-r \times N-r} \), respectively, so that the undesired terms in the powers \( (D + Q)^i \) are mitigated as fast as possible (since \( D \) is diagonal and \( Q \) is upper triangular), promoting fast graph filters. This will also lead to an increase in numerical stability when obtaining the filter coefficients. Thus, \( \|Q\|_F^2 + \|D_2\|_F^2 \) is chosen as the cost function of our proposed method. Moreover, from the RHS of (7), we observe that the upper right hand side block matrix is a zero matrix; in addition, from the Schur decomposition of \( S \), we know that \( D \) and \( Q \) are diagonal and upper triangular, respectively. Thus, minimizing \( Q \) and \( D_2 \) in Frobenius norm, leads to obtaining faster graph filters.

Consequently, our optimization problem is stated as follows:

\[
\min_{S, D_1, D_2, Q} \|Q\|_F^2 + \|D_2\|_F^2
\]  

s. t. (S)_{n,n'} = 0 if \( (v_n, v_{n'}) \notin \mathcal{E}, n, n' = 1, \ldots, N \) \hspace{1cm} (11a)

\[
S = W_\parallel D_1 W_\parallel^\top + W_\perp D_2 W_\perp^\top + WQW^\top
\]  

(11b)

\[
(Q)_{l,l'} = 0, \ l' \leq l, \ l = 1, \ldots, N, \ l' = 1, \ldots, N
\]  

(11c)

\[
(D_1)_{l,l'} = 0, (D_2)_{l,l'} = 0, \ l' \neq l \quad \text{and} \quad l, l' = 1, \ldots, N
\]  

(11d)

\[
\lambda_{1i} \neq \lambda_{2j}, \ \forall i, j
\]  

(11e)

\[
\lambda_{1i} \neq \lambda_{1j}, \ \lambda_{2i} \neq \lambda_{2j}, \ \forall i, j, i \neq j
\]  

(11f)

where \( \lambda_1 = [\lambda_{11}, \ldots, \lambda_{1r}] \), \( \lambda_2 = [\lambda_{21}, \ldots, \lambda_{2(N-r)}] \) are the vectors containing the eigenvalues of \( D_1 \) and \( D_2 \), respectively. Notice that (11e) implies restricting the solution so that only real eigenvalues are allowed. Besides, (11f) and (11g) are added to the problem because based on Theorem 1, the eigenvalues
of $D_1, D_2$ should be distinct. Notice also that, as explained before, the matrix $W$ can be chosen in advance and is not part of the problem. In addition, since $S$ is a local operator, (11b) is also added to enforce the topological constraints.

However, optimization problem (11) is non-convex because of (11f) and (11g). In addition, $S = 0$ is one of its solutions. In order to tackle this problem and make the eigenvalues of $D_1$ and $D_2$ distinct, we can restrict the values of $D_1$ by adding $\text{tr}(D_1) = \epsilon r$ as a new term to the problem where $\epsilon$ can be chosen arbitrarily.

Thus, we have:

\[
\min_{S, D_1, D_2, Q} \|Q\|_F^2 + \|D_2\|_F^2 \tag{12a}
\]
\[
\text{s. t. } (S)_{n,n'} = 0 \text{ if } (v_n, v_{n'}) \notin \mathcal{E}, n, n' = 1, \ldots, N \tag{12b}
\]
\[
\text{tr}(D_1) = r\epsilon \tag{12c}
\]
\[
S = W_1 D_1 W_1^T + W_2 D_2 W_2^T + W Q W^T \tag{12d}
\]
\[
(Q)_{l,l'} = 0, l, l' = 1, \ldots, N \tag{12e}
\]
\[
(D_1)_{l,l'} = 0, (D_2)_{l,l'} = 0, l, l' = 1, \ldots, N \tag{12f}
\]

As we show next, optimization problem (12) can be solved via the Alternating Direction Method of Multipliers (ADMM) [12] effectively. For this, by substituting (12d) into (12b), the optimization problem (12) can be rewritten based on $D$ and $Q$, as follows:

\[
\min_{D, Q} \|\text{vec}(Q)\|_2^2 + \|F\text{vec}(D)\|_2^2 \tag{13a}
\]
\[
\text{s. t. } T\text{vec}(WDW^T + W Q W^T) = T(W \otimes W)\text{vec}(D + Q) = 0 \tag{13b}
\]
\[
P\text{vec}(D) = b, R\text{vec}(Q) = 0 \tag{13c}
\]

where we have the following matrices: $F$ is a matrix that satisfies $F\text{vec}(D) = \text{vec}(D_2)$, $T$ has a row $(e_n \otimes e_{n'})^\top$ for each pair $(n, n')$ such that $(v_n, v_{n'}) \notin \mathcal{H}$, and $e_n$ represents the $n$-th column of the identity matrix with the corresponding size. By applying the property $\text{tr}(BA) = (B^\top)\text{vec}(A)$, we can write (12c) in vector form as $\text{vec}^\top(Y_1)\text{vec}(D) = r\epsilon_1$ where $Y_1$ satisfies $Y_1 D = D_1$. Similarly, we can find $X_1, X_2, X_3$ such that $R\text{vec}(Q) = 0, X_1\text{vec}(D) = 0, X_2\text{vec}(D) = 0$ satisfy (12e) and (12f). Therefore, we have $P = [\text{vec}(Y_1); \text{vec}(Y_2); X_1; X_2], b = [r/\epsilon_1, (N - r)/\epsilon_2, 0, 0]$.

Notice that we have now an optimization problem with a convex cost function and linear constraints, making it possible to apply the scaled-form of ADMM to solve this problem efficiently. We have that:

\[
L_\rho(D, Q, v_1, v_2, v_3) = (\rho/2)\|\text{Mvec}(D + Q) + v_1\|_2^2 + \|\text{vec}(Q)\|_2^2 + (\rho/2)\|\text{Pvec}(D) - b + v_2\|_2^2
\]
\[
+ (\rho/2)\|\text{Rvec}(Q) + v_3\|_2^2 + \|F\text{vec}(D)\|_2^2 \tag{14}
\]

where $M = T(W \otimes W)$. The closed form solutions for each ADMM iteration is given as follows:

\[
\text{vec}(Q[k + 1]) = -(\rho(M^\top M + I + P^\top P))^{-1}(\rho(M\text{vec}(D[k]) + v_1[k]) + R^\top v_3[k]) \tag{15}
\]
\[
\text{vec}(D[k + 1]) = -(\rho M^\top M + \rho P^\top P + F^\top F)^{-1}(\rho(M\text{vec}(Q[k + 1]) + v_1[k]) + P^\top(-b + v_2[k]) - f) \tag{16}
\]
\[
v_1[k + 1] = v_1[k] + M\text{vec}(D[k + 1] + Q[k + 1]) \tag{17}
\]
\[
v_2[k + 1] = v_2[k] + P(\text{vec}(D[k + 1]) - b) \tag{18}
\]
\[
v_3[k + 1] = v_3[k] + R(\text{vec}(Q[k + 1])) \tag{19}
\]
The complete procedure is summarized in Algorithm 1, where $I_{\text{max}}$ denotes the maximum number of ADMM iterations.

**Algorithm 1** Proposed ADMM-based solver

**Require:** $I_{\text{MAX}}$, $U$.

1: for $i = 1$ to $I_{\text{MAX}}$ do
2:    initialize $D$, $v_1$, · · · , $v_2$, $v_3$
3:    update $Q$ based on (16)
4:    update $D$ based on (15)
5:    update $v_1$ based on (17)
6:    update $v_2$ based on (18)
7:    update $v_3$ based on (19)
8:    return $D$, $Q$, $v_1$, · · · , $v_2$, $v_3$
9: end for
10: Obtain $S = W(D + Q)W^T$

IV. NUMERICAL RESULTS

This section describes numerical experiments that validate the performance of the proposed algorithms by averaging the results over 100 different random networks of $N$ nodes. The subspace matrix $U$ is obtained by orthonormalizing an $N \times r$ matrix with i.i.d. standard Gaussian entries. Random signals $\alpha$ and noise signals $v$ were drawn from a normal distribution with zero mean and unit variance to generate input signals $z$ such that $z = \beta \sqrt{(N/r)}U\|\alpha + v = \xi + v$ where $\beta$ is Signal-to-Noise ratio (SNR). The graph topology is generated through the Erdos-Renyi model [13], where the presence of each directed edge is an i.i.d. Bernoulli random variable.

The considered performance metrics are the Normalized Mean Projection Error (NMPE) and the Normalized Mean Square Error (NMSE), which are given by: $\text{NMPE}(H_l) \triangleq \frac{E_x[\| (P_l - H_l)z \|_2^2]}{E_x[\| Pz \|_2^2]}$ and $\text{NMSE}(\hat{\xi}_k) = \frac{\mathbb{E}[\| \xi - \hat{\xi}_k \|_2^2]}{\mathbb{E}[\| \xi \|_2^2]}$, respectively. The expectation is taken over $G$, $U_\|$, $\alpha$, $v$.

The proposed method is compared with the other typical choices for the graph shift operator in previous works, such as the Laplacian matrix or the Adjacency matrix. Moreover, we also compare with a direct Least-Square method to find the asymmetric graph shift operator, by minimizing: $\| \text{vec}(H) - G \text{vec}(S) \|_2^2$, where $G$ is a matrix that satisfies the topology constraints, i.e. $S_{n,n'} = 0$ $\forall (v_n, v_{n'}) \notin \mathcal{H}$ via $G \text{vec}(S)$. For all the methods, the filter coefficients are found by solving directly the equation $H := \sum_{l=0}^L c_l S_l$. Besides, to alleviate problems associated with finite-precision arithmetic, each node uses a different set of filter coefficients [5]. It can be easily shown that all the results of the paper carry over also when each node uses a different set of filter coefficients. Moreover, for the proposed method, the identity matrix is chosen as $E_\perp$ and $E_\|$. Figure (1) depicts NMPE versus the number of local exchanges for different scenarios. It can be observed that the proposed method outperforms all the other methods, and achieves the exact projection after a small finite number of iterations, thus showing that it converges faster than the other methods.

Figure (2) shows the NMSE versus the number of local exchanges for sparser and larger networks, comparing our method with the other choices of graph shift operator. Our proposed method obtains lower NMSE in comparison with all the other methods.

V. CONCLUSION

This paper proposes an algorithm to design asymmetric graph shift operator to compute decentralized subspace projection. The proposed method is also capable to design other linear transformations. The results show that the proposed method obtain the projection matrix exactly after a finite number of iterations.
Fig. 1. NMPE as a function of number of local exchanges ($r = 3$, $\epsilon = 1$, $\beta = 5$, $\rho = 0.1$, $I_{\text{max}} = 1000$)

Fig. 2. NMSE as a function of number of local exchanges ($N = 40$, $r = 4$, $p_{\text{miss}} = .7$, $\epsilon = 1$, $\beta = 5$, $\rho = 0.1$, $I_{\text{max}} = 1000$)

REFERENCES

[1] A. Sandryhail and J. M. F. Moura, “Discrete signal processing on graphs: frequency analysis,” IEEE Trans. on Signal Processing, vol. 61, pp. 1644–1656, Apr 2013.

[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 Processing Magazine, vol. 30, pp. 83–98, May 2013.

[3] A. S. S. Chen, R. Varma and J. Kovaevi, “Discrete signal processing on graphs: Sampling theory,” vol. 63, pp. 6510–6523, Dec. 2015.

[4] A. Sandryhail and J. M. F. Moura, “Discrete signal processing on graphs,” IEEE Trans. on Signal Processing, vol. 62, pp. 3042–3054, Jun 2014.

[5] S. Segarra, A. G. Marques, and A. Ribeiro, “Optimal graph-filter design and applications to distributed linear network operators,” IEEE Trans. Sig. Process., vol. 65, pp. 4117–4131, May 2017.

[6] S. Barbarossa, G. Scutari, and T. Battisti, “Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints,” in International Conference on Acoustics, Speech, and Signal Processing (ICASSP), (Taipei,Taiwan), pp. 2893–2896, Apr. 2009.
[7] A. Sandryhail, S. Kar, and J. M. F. Moura, “Finite-time distributed consensus through graph filters,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Florence, Italy), pp. 1080–1084, May 2014.

[8] T. Weerasinghe, D. Romero, C. Asensio-Marco, and B. Beferull-Lozano, “Fast decentralized subspace projection via graph filtering,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Calgary,Canada), pp. 4639–4643, Apr. 2018.

[9] S. Mollaebrahim, C. Asensio-Marco, D. Romero, and B. Beferull-Lozano, “Decentralized subspace projection in large networks,” in *Global Conference on Signal and Information Processing (GlobalSIP)*, (Anaheim,USA), pp. 788–792, Feb. 2018.

[10] G. Golub and C. Van Loan, *Matrix computations*. Johns Hopkins University Press, 1996.

[11] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1995.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, pp. 1–122, 01 2011.

[13] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. Springer, New York, 2009.