LINEAR NETWORK OPERATORS USING NODE-VARIANT GRAPH FILTERS

Santiago Segarra[†]*, Antonio G. Marques*[*]*, and Alejandro Ribeiro*[†]

[†]Dept. of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA
[*]Dept. of Signal Theory and Communications, King Juan Carlos University, Madrid, Spain

ABSTRACT

We introduce *node-variant* graph filters, which allow the simultaneous implementation of multiple (regular) graph filters at different nodes, and study their design to implement arbitrary linear transformations between graph signals. Node-variant graph filters can be implemented distributedly, making them suitable for networked settings. We determine spectral conditions under which a specific linear transformation can be implemented perfectly and, for the cases where perfect implementation is infeasible, the design of optimal approximations for different error metrics is analyzed. We demonstrate the practical relevance of the developed framework by studying the application of node-variant graph filters for analog network coding.

Index Terms— Graph signal processing, Graph filter, Network operator, Analog network coding.

1. INTRODUCTION

Networks and graphs can be understood as structures that encode pairwise relationships between elements of a set. Often, networks have intrinsic value and are themselves the object of study. In other occasions, the network defines an underlying notion of proximity or dependence, but the object of interest is a signal defined on top of the graph, i.e., data associated with the nodes of the network. This is the matter addressed in the field of graph signal processing [1, 2].

The problem that we investigate is how to design *node-variant* graph filters – a novel, more flexible class of graph filters – to implement a pre-specified linear transformation. Since graph filters can be implemented *distributedly*, especial focus will be devoted to *network* linear transformations. Without leveraging the particular structure of a graph filter, several works have addressed the problem of *distributed* implementation of different linear operators in the context of graph signal processing. Relevant examples are the projection of graph-signals onto low-dimensional spaces [3], the approximation of graph Fourier multipliers using Chebyshev polynomials [4], or graph-signal inpainting [5,6]. Graph filters have been used to implement distributedly *specific* linear transformations such as fast consensus [7], projections onto the low-rank space of graph-bandlimited signals [8], or interpolation of graph-bandlimited signals [9,10], and, more recently, the implementation of *general* linear transformations using regular (node-invariant) graph filters was addressed [11].

The contribution of this paper is threefold. Firstly, we introduce the notion of a node-variant graph filter, a generalization of regular graph filters that preserves locality and increases design flexibility by allowing filter coefficients to vary across nodes (Section 3). Secondly, conditions under which a specific linear transformation can be implemented perfectly using node-variant graph filters are identified and, for the cases where the conditions are not met, the design of optimal approximations for different error metrics is addressed (Section 4). This extends the analysis in [11], where node-invariant graph filters were considered, and allows us to implement a larger class of linear network transformations. Lastly, to demonstrate the relevance of our results for the design of distributed network operators, Section 5 applies the developed framework to the problem of analog network coding (ANC) [12, 13] and shows that node-variant graph filters clearly outperform their node-invariant counterpart.

Work in this paper is supported by USA NSF CCF-1217963 and Spanish MINECO TEC2013-41604-R.

2. GRAPH SIGNALS AND FILTERS

Let $\mathcal{G}$ denote a directed graph with a set of N nodes or vertices $\mathcal{N}$ and a set of links $\mathcal{E}$, such that if node i is connected to j, then $(i,j) \in \mathcal{E}$. The (incoming) neighborhood of i is defined as the set of nodes $\mathcal{N}_i = \{j \,|\, (j,i) \in \mathcal{E}\}$ connected to i. For any given graph we define the adjacency matrix $\mathbf{A}$ as a $N \times N$ matrix with nonzero elements A_{ji} if and only if $(i,j) \in \mathcal{E}$. The value of A_{ji} captures the strength of the connection from i to j. The focus of this paper is not on analyzing $\mathcal{G}$, but graph signals defined on $\mathcal{N}$. Formally, each of these signals can be represented as a vector $\mathbf{x} = [x_1, ..., x_N]^T \in \mathbb{R}^N$ where the i-th component represents the value of the signal at node i.

The graph $\mathcal{G}$ is endowed with a *graph-shift operator* $\mathbf{S}$ [2]. The operator $\mathbf{S}$ is a $N \times N$ matrix whose entry S_{ji} can be nonzero only if $i = j$ or if $(i,j) \in \mathcal{E}$. The sparsity pattern of the matrix $\mathbf{S}$ captures the local structure of $\mathcal{G}$, but we make no specific assumptions on the values of the nonzero entries of $\mathbf{S}$. Possible choices for $\mathbf{S}$ are the adjacency [2] and the Laplacian [1] matrices of the graph. The intuition behind $\mathbf{S}$ is to represent a linear transformation that can be computed locally at the nodes of the graph. More rigorously, if $\mathbf{y}$ is defined as $\mathbf{y} = \mathbf{S}\mathbf{x}$, then node i can compute y_i provided that it has access to the value of x_j at $j \in \mathcal{N}_i$. We assume henceforth that $\mathbf{S}$ is diagonalizable, so that there exist matrices $\mathbf{V}$ and $\mathbf{\Lambda}$, the latter being diagonal, that can be used to decompose $\mathbf{S}$ as $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$. Given a graph signal $\mathbf{x}$, we refer to $\widehat{\mathbf{x}} := \mathbf{V}^{-1}\mathbf{x}$ as the frequency representation of $\mathbf{x}$ [2]. A linear graph operator is a transformation $\mathbf{B} : \mathbb{R}^N \to \mathbb{R}^N$ between graph signals, which can be represented by a square $N \times N$ matrix.

2.1. Graph filters

Graph filters $\mathbf{H} : \mathbb{R}^N \to \mathbb{R}^N$ are particular linear graph operators of the form

$$\mathbf{H} := \sum_{l=0}^{L-1} c_l \mathbf{S}^l, \tag{1}$$

i.e., polynomials (of degree $L-1$) of the graph-shift operator [2]. The graph filter $\mathbf{H}$ can also be written as $\mathbf{H} = \mathbf{V}\big(\sum_{l=0}^{L-1} c_l \mathbf{\Lambda}^l\big)\mathbf{V}^{-1}$. The diagonal matrix $\widehat{\mathbf{H}} := \sum_{l=0}^{L-1} c_l \mathbf{\Lambda}^l$ can then be viewed as the frequency response of $\mathbf{H}$ and it can be alternatively written as $\widehat{\mathbf{H}} = \text{diag}(\widehat{\mathbf{c}})$, where vector $\widehat{\mathbf{c}}$ contains the N frequency re-

sponses of the filter. Let λ_k denote the k-th eigenvalue of $\mathbf{S}$ and define the $N \times L$ Vandermonde matrix $\mathbf{\Psi}$ such that $\Psi_{ij} = \lambda_i^{j-1}$. Upon defining the vector containing the coefficients of the filter as $\mathbf{c} := [c_0, \ldots, c_{L-1}]^T$, it holds that $\widehat{\mathbf{c}} = \mathbf{\Psi}\mathbf{c}$ and therefore

$$\mathbf{H} = \sum_{l=0}^{L-1} c_l \mathbf{S}^l = \mathbf{V}\text{diag}\big(\mathbf{\Psi}\mathbf{c}\big)\mathbf{V}^{-1} = \mathbf{V}\text{diag}(\widehat{\mathbf{c}})\mathbf{V}^{-1}. \quad (2)$$

This implies that if $\mathbf{y}$ is defined as $\mathbf{y} = \mathbf{H}\mathbf{x}$, its frequency representation $\widehat{\mathbf{y}} = \mathbf{V}^{-1}\mathbf{y}$ satisfies $\widehat{\mathbf{y}} = \text{diag}\big(\mathbf{\Psi}\mathbf{c}\big)\mathbf{V}^{-1}\mathbf{x} = \text{diag}\big(\widehat{\mathbf{c}}\big)\widehat{\mathbf{x}}$, which demonstrates that the output at a given frequency depends only on the value of the input and the filter response at that given frequency. Note that while for the time-varying case the operator that transforms the signals and the filter coefficients into the frequency domain is the same – the Discrete Fourier Transform (DFT) –, when a generic $\mathbf{S}$ is considered, matrices $\mathbf{V}^{-1}$ and $\mathbf{\Psi}$ are different.

A particularity of graph filters is that they can be implemented locally, e.g., with $L-1$ exchanges of information among neighbors. To see this, it is convenient to define the l-th shifted input signal as $\mathbf{z}^{(l)} := \mathbf{S}^l\mathbf{x}$. Notice that node i can compute $[\mathbf{z}^{(l)}]_i$ locally based on the values of $[\mathbf{z}^{(l-1)}]_j$ at $j \in \mathcal{N}_i$. To emphasize this local property, we define $\mathbf{z}_i$ as an $L \times 1$ vector collecting the entries of $\{\mathbf{z}^{(l)}\}_{l=0}^{L-1}$ that are known by node i, so that $[\mathbf{z}_i]_l := [\mathbf{z}^{(l)}]_i$. With $\mathbf{y}$ denoting the output of a graph filter for the input signal $\mathbf{x}$, it follows from (1) that $\mathbf{y} = \mathbf{H}\,\mathbf{x} = \sum_{l=0}^{L-1} c_l\mathbf{S}^l\mathbf{x} = \sum_{l=0}^{L-1} c_l\mathbf{z}^{(l)}$. Hence, the i-th entry of vector $\mathbf{y}$ can be computed as $y_i = \sum_{l=0}^{L-1} c_l[\mathbf{z}^{(l)}]_i = \mathbf{c}^T\mathbf{z}_i$, showing that if the nodes know the value of the filter coefficients, y_i can be computed using solely information available at node i.

3. NODE-VARIANT GRAPH FILTERS

This paper proposes a generalization of graph filters, called *node-variant* graph filters, defined as operators $\mathbf{H}_{\text{nv}} : \mathbb{R}^N \to \mathbb{R}^N$ of the form [cf. (1)]

$$\mathbf{H}_{\text{nv}} := \sum_{l=0}^{L-1} \text{diag}(\mathbf{c}^{(l)})\mathbf{S}^l. \quad (3)$$

Whenever the $N \times 1$ vectors $\mathbf{c}^{(l)}$ are constant, i.e. $\mathbf{c}^{(l)} = c_l\mathbf{1}$ for all l where $\mathbf{1}$ is the all-one vector, the node-variant filter reduces to the regular (node-invariant) graph filter. However, for general $\mathbf{c}^{(l)}$, when $\mathbf{H}_{\text{nv}}$ is applied to a signal $\mathbf{x}$, each node applies different weights to the shifted signals $\mathbf{S}^l\mathbf{x}$. This additional flexibility enables the design of more general operators without undermining the local implementation. For notational convenience, the filter coefficients associated with node i are collected in the $L \times 1$ vector $\mathbf{c}_i$, such that $[\mathbf{c}_i]_l = [\mathbf{c}^{(l)}]_i$, and we define the $L \times N$ matrix $\mathbf{C} := [\mathbf{c}_1, \ldots, \mathbf{c}_N]$. The graph filters in (3) can be viewed as a generalization of linear time-varying filters whose impulse response changes with time.

Since $\mathbf{S}^l$ and $\text{diag}(\mathbf{c}^{(l)})$ are not simultaneously diagonalizable (i.e., their eigenvectors are not the same), the neat frequency interpretation succeeding (1) does not hold true for the filters in (3). However, the spectral decomposition of $\mathbf{S}$ can still be used to understand how the output of the filter at a given node i depends on the frequency components of the input. To be specific, let us write the filter in (3) as $\mathbf{H}_{\text{nv}} = \sum_{l=0}^{L-1} \text{diag}(\mathbf{c}^{(l)})\mathbf{V}\mathbf{\Lambda}^l\mathbf{V}^{-1}$. Next, to analyze the effect of $\mathbf{H}_{\text{nv}}$ on the value of the output signal at node i, consider the i-th row of $\mathbf{H}_{\text{nv}}$, given by

$$\mathbf{h}_i^T := \mathbf{e}_i^T\mathbf{H}_{\text{nv}} = \sum_{l=0}^{L-1} [\mathbf{c}_i]_l\mathbf{e}_i^T\mathbf{V}\mathbf{\Lambda}^l\mathbf{V}^{-1}, \quad (4)$$

where $\mathbf{e}_i$ is the i-th $N \times 1$ canonical basis vector (all entries of $\mathbf{e}_i$ are zero except for the i-th one, which is one). Defining the vectors $\mathbf{u}_i := \mathbf{V}^T\mathbf{e}_i$ and $\widehat{\mathbf{c}}_i := \mathbf{\Psi}\mathbf{c}_i$, we can rewrite (4) as

$$\mathbf{h}_i^T = \sum_{l=0}^{L-1} [\mathbf{c}_i]_l\mathbf{u}_i^T\mathbf{\Lambda}^l\mathbf{V}^{-1} = \mathbf{u}_i^T\Big(\sum_{l=0}^{L-1} [\mathbf{c}_i]_l\mathbf{\Lambda}^l\Big)\mathbf{V}^{-1}$$
$$= \mathbf{u}_i^T\text{diag}(\mathbf{\Psi}\mathbf{c}_i)\mathbf{V}^{-1} = \mathbf{u}_i^T\text{diag}(\widehat{\mathbf{c}}_i)\mathbf{V}^{-1}. \quad (5)$$

The expression in (5) reveals that the output of the filter at node i, which can be written as $\mathbf{h}_i^T\mathbf{x}$, can be viewed as an inner product of $\mathbf{V}^{-1}\mathbf{x}$ (the frequency representation of the input) and $\mathbf{u}_i$ (how strongly the different frequencies are expressed by node i), modulated by $\widehat{\mathbf{c}}_i$ (the frequency response associated with the coefficients used by node i).

4. IMPLEMENTATION OF LINEAR OPERATORS

The objective in this section is to implement pre-specified linear transformations using node-variant graph filters [cf. (3)]. More specifically, given a desired linear transformation $\mathbf{B}$ we want to design the coefficient vectors $\mathbf{c}^{(l)}$ for $l = 0, \ldots, L-1$ so that

$$\mathbf{B} = \sum_{l=0}^{L-1} \text{diag}(\mathbf{c}^{(l)})\mathbf{S}^l. \quad (6)$$

We first identify the conditions under which (6) can be solved exactly, and then analyze approximate solutions.

4.1. Conditions for perfect implementation

Defining the vectors $\mathbf{b}_i := \mathbf{B}^T\mathbf{e}_i$ and $\tilde{\mathbf{b}}_i := \mathbf{V}^T\mathbf{b}_i$, the conditions under which the equivalence in (6) can be achieved are given in the following proposition and the subsequent corollary.

Proposition 1 *The linear transformation $\mathbf{B}$ can be implemented using the node-variant graph filter $\mathbf{H}_{\text{nv}}$ in* (3) *if the three following conditions hold for all i:*
a) $[\tilde{\mathbf{b}}_i]_k = 0$ for those k such that $[\mathbf{u}_i]_k = 0$.
b) For all (k_1,k_2) such that $\lambda_{k_1} = \lambda_{k_2}$, it holds that $[\tilde{\mathbf{b}}_i]_{k_1}/[\mathbf{u}_i]_{k_1} = [\tilde{\mathbf{b}}_i]_{k_2}/[\mathbf{u}_i]_{k_2}$.
c) The degree $L-1$ of $\mathbf{H}_{\text{nv}}$ satisfies $L \geq D$.

Proof: We use (5) to write the row-wise equality between $\mathbf{H}_{\text{nv}}$ and $\mathbf{B}$

$$\mathbf{b}_i^T = \mathbf{h}_i^T = \mathbf{u}_i^T\text{diag}(\mathbf{\Psi}\mathbf{c}_i)\mathbf{V}^{-1}, \quad (7)$$

for all i. Multiplying (7) from the right by $\mathbf{V}$, transposing the equality, and using the definition of $\tilde{\mathbf{b}}_i$, we get

$$\tilde{\mathbf{b}}_i = \text{diag}(\mathbf{\Psi}\mathbf{c}_i)\mathbf{u}_i = \text{diag}(\mathbf{u}_i)\mathbf{\Psi}\mathbf{c}_i, \quad (8)$$

for all i. Thus, for $\mathbf{B}$ to be implementable, for every node i we need to find the vector $\mathbf{c}_i$ that satisfies (8).

For each i, partition the set $\{1, \ldots, N\}$ intro three subsets of indices $\mathcal{K}_1^i$, $\mathcal{K}_2^i$, and $\mathcal{K}_3^i$ such that i) $\mathcal{K}_1^i$ contains the indices k such that $[\mathbf{u}_i]_k = 0$; ii) $\mathcal{K}_2^i$ contains the indices of all the eigenvalues in $\{1, \ldots, N\} \setminus \mathcal{K}_1^i$ that are distinct as well as one index per repeated eigenvalue λ_k, and; iii) $\mathcal{K}_3^i = \{1, \ldots, N\} \setminus (\mathcal{K}_1^i \cup \mathcal{K}_2^i)$. Thus, divide the set of N linear equations in (8) into three groups

$$\mathbf{E}_{\mathcal{K}_1^i}^T\tilde{\mathbf{b}}_i = \text{diag}(\mathbf{E}_{\mathcal{K}_1^i}^T\mathbf{u}_i)\mathbf{E}_{\mathcal{K}_1^i}^T\mathbf{\Psi}\mathbf{c}_i, \quad (9)$$

$$\mathbf{E}_{\mathcal{K}_2^i}^T\tilde{\mathbf{b}}_i = \text{diag}(\mathbf{E}_{\mathcal{K}_2^i}^T\mathbf{u}_i)\mathbf{E}_{\mathcal{K}_2^i}^T\mathbf{\Psi}\mathbf{c}_i, \quad (10)$$

$$\mathbf{E}_{\mathcal{K}_3^i}^T\tilde{\mathbf{b}}_i = \text{diag}(\mathbf{E}_{\mathcal{K}_3^i}^T\mathbf{u}_i)\mathbf{E}_{\mathcal{K}_3^i}^T\mathbf{\Psi}\mathbf{c}_i. \quad (11)$$

If condition *a)* holds, then (9) is true for any $\mathbf{c}_i$ since both sides of the equality are 0. Since $\mathcal{K}_2^i$ contains no repeated eigenvalues, the matrix $\text{diag}(\mathbf{E}_{\mathcal{K}_2^i}^T \mathbf{u}_i)\mathbf{E}_{\mathcal{K}_2^i}^T \boldsymbol{\Psi}$ in (10) has full row-rank. Given that $|\mathcal{K}_2^i| \leq D$, condition *c)* ensures that $\mathbf{c}_i$ has at least as many elements as equations in (10), guaranteeing the existence of a solution. Denoting by $\mathbf{c}_i^*$ a solution of (10), condition *b)* implies that this same vector also solves (11). To see why this is true, notice that for every equation in (11) (of index k_3) there is one in (10) (of index k_2) such that $[\boldsymbol{\Psi}\mathbf{c}_i^*]_{k_3} = [\boldsymbol{\Psi}\mathbf{c}_i^*]_{k_2} = [\tilde{\mathbf{b}}_i]_{k_2}/[\mathbf{u}_i]_{k_2}$, where the last equality follows from the fact that $\mathbf{c}_i^*$ solves (10). Imposing condition *b)*, we arrive to the conclusion that $[\boldsymbol{\Psi}\mathbf{c}_i^*]_{k_3} = [\tilde{\mathbf{b}}_i]_{k_3}/[\mathbf{u}_i]_{k_3}$ and, thus, (11) is satisfied. The simultaneous fulfillment of (9), (10), and (11), implies (8), concluding the proof. ■

The conditions in Proposition 1 detail how the spectral properties of $\mathbf{S}$ impact the set of linear transformations that can be implemented. Condition *a)* states that if *node* i is unable to express a given frequency k, only linear operators whose i-th row is orthogonal to the k-th frequency basis vector can be implemented. Condition *b)* states that if two frequencies k_1 and k_2 are indistinguishable for the *graph-shift* operator, then the projection of the i-th row of $\mathbf{B}$ onto these two frequency basis vectors must be proportional to how strongly node i expresses frequencies k_1 and k_2 for every node i. Finally, condition *c)* requires that the order of the *filter* has to be high enough to have enough degrees of freedom to design the linear operator and to allow the original signal $\mathbf{x}$ to percolate through the network. Conditions *a)* and *b)* are necessary while *c)* details a sufficient filter degree for general implementation. However, filters with lower degree may be able to implement a specific linear transformation. The following result follows as a corollary of Proposition 1.

Corollary 1 *Any linear transformation* $\mathbf{B}$ *can be implemented by a node-variant filter of the form* $\sum_{l=0}^{N-1} \text{diag}(\mathbf{c}^{(l)})\mathbf{S}^l$ *if the graph-shift operator* $\mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$ *satisfies the following two properties:*
a) all the entries of $\mathbf{V}$ *are non-zero.*
b) all the eigenvalues $\{\lambda_k\}_{k=1}^N$ *are distinct.*

Proof: Conditions *a)* and *b)* immediately guarantee the corresponding conditions in Proposition 1 for any $\mathbf{B}$. Condition *c)* in Proposition 1 is satisfied from the fact that the corollary sets $L = N$ and, by definition, $N \geq D$. ■

Under the conditions in Corollary 1, there is a unique set of filter coefficients $\{\mathbf{c}_i\}_{i=1}^N$ leading to perfect implementation, which can be found as

$$\mathbf{c}_i = \boldsymbol{\Psi}^{-1}\text{diag}(\mathbf{u}_i)^{-1}\mathbf{V}^T\mathbf{b}_i, \tag{12}$$

for all i. Proposition 1 and Corollary 1 confirm that the class of linear transformations that can be implemented using (3) is significantly broader than the one that can be implemented using (1) [11]. More specifically, the restrictive condition of simultaneous diagonalization in [11, Proposition 1] is dropped when focusing on the more general node-variant filters.

4.2. Approximate implementation

In general, if the conditions in Proposition 1 are not satisfied, perfect implementation of $\mathbf{B}$ is not feasible. In such cases, the filter can be designed to minimize a pre-specified error metric. Upon defining the error *matrix* as the difference $\mathbf{D} := \mathbf{H}_{\text{nv}} - \mathbf{B}$, two prevalent error norms used in the context of matrix estimation and reconstruction are the 2-norm, which is related to the worst-case error (WCE), and the Frobenius norm, which is related to the mean squared error (MSE) [14]. Hence, we will cast our approximate-filter-design problem as that of finding the coefficients $\mathbf{c}$ that minimize either $\|\mathbf{D}\|_2$ or $\|\mathbf{D}\|_{\text{F}}$.

To minimize the Frobenius norm of $\mathbf{D}$, let us start by defining the per-node vector difference $\mathbf{d}_i := \mathbf{D}^T\mathbf{e}_i = \mathbf{h}_i - \mathbf{b}_i$, so that we can write $\|\mathbf{D}\|_{\text{F}} = \sum_i \|\mathbf{d}_i\|_2^2$. This shows that the optimal coefficients $\mathbf{c}_i^*$ can be designed separately across nodes. We do this in the following proposition, where we also present the design to minimize $\|\mathbf{D}\|_2$. Recall that $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N]$ and define the matrices $\tilde{\mathbf{U}} := [\text{diag}(\mathbf{u}_1), \text{diag}(\mathbf{u}_2), \ldots, \text{diag}(\mathbf{u}_N)]^T$ and $\boldsymbol{\Phi}_i := (\mathbf{V}^{-1})^T\text{diag}(\mathbf{u}_i)\boldsymbol{\Psi}$.

Proposition 2 *The optimal filter coefficients defined as* $\{\mathbf{c}_{i,\text{F}}^*\}_{i=1}^N := \text{argmin}_{\{\mathbf{c}_i\}_{i=1}^N} \|\mathbf{H}_{\text{nv}} - \mathbf{B}\|_{\text{F}}$ *and* $\{\mathbf{c}_{i,2}^*\}_{i=1}^N := \text{argmin}_{\{\mathbf{c}_i\}_{i=1}^N} \|\mathbf{H}_{\text{nv}} - \mathbf{B}\|_2$ *are, respectively, given for all* i *by*

$$\mathbf{c}_{i,\text{F}}^* = \boldsymbol{\Phi}_i^\dagger\mathbf{b}_i = (\boldsymbol{\Phi}_i^T\boldsymbol{\Phi}_i)^{-1}\boldsymbol{\Phi}_i^T\mathbf{b}_i, \tag{13}$$

where the second equality holds if $\boldsymbol{\Phi}_i$ *has full column rank, and*

$$\{\mathbf{C}_2^*, s^*\} = \underset{\{\mathbf{C},s\}}{\text{argmin}} \quad s \tag{14}$$

$$\text{s. to} \begin{bmatrix} s\mathbf{I} & (\mathbf{I}\odot\boldsymbol{\Psi}\mathbf{C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1} - \mathbf{B} \\ ((\mathbf{I}\odot\boldsymbol{\Psi}\mathbf{C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1} - \mathbf{B})^T & s\mathbf{I} \end{bmatrix} \succeq 0,$$

where $\odot$ *denotes the Khatri-Rao product.*

Proof: From (5), we have that $\|\mathbf{d}_i\|_2^2 = \|\mathbf{u}_i^T\text{diag}(\boldsymbol{\Psi}\mathbf{c}_i)\mathbf{V}^{-1} - \mathbf{b}_i^T\|_2^2$. By transposing the vector inside the norm and recalling that for general vectors $\text{diag}(\mathbf{a})\mathbf{b} = \text{diag}(\mathbf{b})\mathbf{a}$, we obtain that

$$\|\mathbf{d}_i\|_2^2 = \|(\mathbf{V}^{-1})^T\text{diag}(\mathbf{u}_i)\boldsymbol{\Psi}\mathbf{c}_i - \mathbf{b}_i\|_2^2 = \|\boldsymbol{\Phi}_i\mathbf{c}_i - \mathbf{b}_i\|_2^2, \tag{15}$$

which can be minimized using the pseudoinverse, and the result in (13) follows.

To show (14), we begin by showing that $(\mathbf{I}\odot\boldsymbol{\Psi}\mathbf{C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1} = \mathbf{H}_{\text{nv}}$. To see why this is true, transpose (5) to obtain $\mathbf{h}_i = (\mathbf{V}^{-1})^T\text{diag}(\mathbf{u}_i)\boldsymbol{\Psi}\mathbf{c}_i$, from where it follows that

$$[\mathbf{h}_1, ..., \mathbf{h}_N] = (\mathbf{V}^{-1})^T[\text{diag}(\mathbf{u}_1), ..., \text{diag}(\mathbf{u}_2)] \begin{bmatrix} \boldsymbol{\Psi}\mathbf{c}_1 & \mathbf{0} & \cdots \\ \mathbf{0} & \boldsymbol{\Psi}\mathbf{c}_2 & \\ \vdots & \vdots & \ddots \end{bmatrix}. \tag{16}$$

Noting that the rightmost matrix can be written as $\mathbf{I}\odot\boldsymbol{\Psi}\mathbf{C}$ and that $\mathbf{H}_{\text{nv}} = [\mathbf{h}_1, ..., \mathbf{h}_N]^T$, the result follows. Finally, notice that the constraint in (14) can be rewritten in terms of its Schur complement as $s^2\mathbf{I} \succeq (\mathbf{H}_{\text{nv}} - \mathbf{B})^T(\mathbf{H}_{\text{nv}} - \mathbf{B})$ to obtain $s^2 \geq \|\mathbf{H}_{\text{nv}} - \mathbf{B}\|_2^2$. This implies that s is an upper bound on $\|\mathbf{H}_{\text{nv}} - \mathbf{B}\|_2$. Since the constraint in (14) is tight, minimizing the upper bound is equivalent to minimizing $\|\mathbf{H}_{\text{nv}} - \mathbf{B}\|_2$, concluding the proof. ■

The choice of an element-wise error metric, such as the Frobenius norm, leads to a decoupled optimization of the filter coefficients at each node. By contrast, if the objective is to minimize $\|\mathbf{D}\|_2$, then the filter coefficients across nodes must be optimized jointly.

Whenever prior knowledge of the input signal $\mathbf{x}$ is available, it can be incorporated into the design of the filter coefficients. In such a case, the goal is to minimize an error metric of the difference *vector* $\mathbf{d} := \mathbf{H}_{\text{nv}}\mathbf{x} - \mathbf{B}\mathbf{x}$. A case of particular interest is when $\mathbf{x}$ is drawn from a zero-mean distribution with known covariance $\mathbf{R}_\mathbf{x} := \mathbb{E}[\mathbf{x}\mathbf{x}^T]$. In this case, the error covariance is given by $\mathbf{R}_\mathbf{d} := \mathbb{E}[\mathbf{d}\mathbf{d}^T] = (\mathbf{H}_{\text{nv}} - \mathbf{B})\mathbf{R}_\mathbf{x}(\mathbf{H}_{\text{nv}} - \mathbf{B})^T$. Our objective is to pick filter coefficients to minimize some metric of the error covariance matrix $\mathbf{R}_\mathbf{d}$. Two commonly used approaches are the minimiza-

tion of $\text{Trace}(\mathbf{R_d})$ and $\lambda_{\max}(\mathbf{R_d})$, where the former is equivalent to minimizing the MSE of $\mathbf{d}$ and the latter minimizes the WCE achievable by all possible realizations of $\mathbf{x}$ [15].

Proposition 3 *The optimal filter coefficients defined as* $\{\mathbf{c}^*_{i,\text{Tr}}\}_{i=1}^N := \text{argmin}_{\{\mathbf{c}_i\}_{i=1}^N} \text{Trace}(\mathbf{R_d})$ *and* $\{\mathbf{c}^*_{i,\lambda}\}_{i=1}^N := \text{argmin}_{\{\mathbf{c}_i\}_{i=1}^N} \lambda_{\max}(\mathbf{R_d})$ *are, respectively, given for all* i *by*

$$\mathbf{c}^*_{i,\text{Tr}} = (\mathbf{R_x}^{1/2}\mathbf{\Phi}_i)^\dagger \mathbf{R_x}^{1/2}\mathbf{b}_i = (\mathbf{\Phi}_i^T\mathbf{R_x}\mathbf{\Phi}_i)^{-1}\mathbf{\Phi}_i^T\mathbf{R_x}\mathbf{b}_i, \quad (17)$$

where the second equality holds if $\mathbf{R_x}^{1/2}\mathbf{\Phi}_i$ *has full column rank, and*

$$\{\mathbf{C}^*_\lambda, s^*\} = \underset{\{\mathbf{C},s\}}{\text{argmin}} \quad s \quad (18)$$

$$\text{s. to} \begin{bmatrix} s\mathbf{I} & (\mathbf{I}\odot\mathbf{\Psi C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1} - \mathbf{B} \\ ((\mathbf{I}\odot\mathbf{\Psi C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1} - \mathbf{B})^T & s\mathbf{R_x}^{-1} \end{bmatrix} \succeq 0.$$

Proof: Given that for any real matrix $\mathbf{A}$ we have that $\|\mathbf{A}\|_\text{F}^2 = \text{Trace}(\mathbf{A}\mathbf{A}^T)$, it follows that

$$\text{Trace}(\mathbf{R_d}) = \sum_{i=1}^N \|\mathbf{h}_i^T\mathbf{R_x}^{1/2} - \mathbf{b}_i^T\mathbf{R_x}^{1/2}\|_2^2. \quad (19)$$

Since $\mathbf{h}_i$ depends on $\mathbf{c}_i$ and not on $\mathbf{c}_j$ for $i \neq j$, we may minimize the summands in (19) separately. Following the same reasoning as in the proof of Proposition 2, we write $\mathbf{h}_i$ in terms of $\mathbf{c}_i$ to obtain

$$\text{Trace}(\mathbf{R_d}) = \sum_{i=1}^N \|\mathbf{R_x}^{1/2}\mathbf{\Phi}_i\mathbf{c}_i - \mathbf{R_x}^{1/2}\mathbf{b}_i\|_2^2. \quad (20)$$

The result in (17) follows from minimizing the norm in (20) via the Moore-Penrose pseudoinverse.

In order to show (18), recall that $\mathbf{H}_\text{nv} = (\mathbf{I}\odot\mathbf{\Psi C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1}$ (cf. proof of Proposition 2) and note that the constraint in (18) can be rewritten in terms of its Schur complement as $s^2\mathbf{I} \succeq (\mathbf{H}_\text{nv} - \mathbf{B})\mathbf{R_x}(\mathbf{H}_\text{nv} - \mathbf{B})^T$. In terms of eigenvalues, this is equivalent to $s^2 \geq \lambda_{\max}(\mathbf{R_d})$, as wanted. ■

Notice that whenever the signal entries are uncorrelated, i.e. $\mathbf{R_x} = \sigma^2\mathbf{I}$, Propositions 2 and 3 are equivalent. The ensuing section illustrates the use of node-variant filters for the implementation of analog network coding.

5. ANALOG NETWORK CODING

In the context of multi-hop communication networks, network coding is a scheme where routing nodes, instead of simply relaying the received information, combine the packets (symbols) received from different sources to perform a single transmission. Even though network coding was originally conceived for transmission of digital data in the form of packets [16,17], extensions to the transmission of analog signals have been developed under the name of ANC [12,13]. In this section, we show how node-variant graph filters can be leveraged to design ANC schemes. Apart from traditional communication networks, the results presented here are also relevant for setups where there exists an inherent diffusion dynamic that percolates the information across the network, as, for example, in the case of molecular and nano communication networks [18,19].

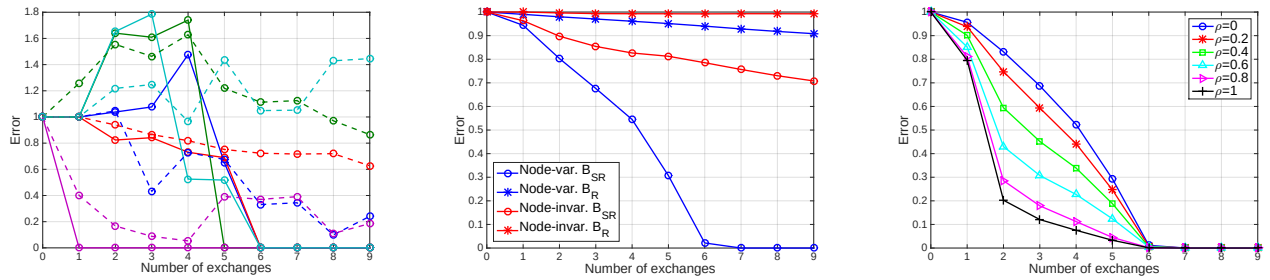
The filter design framework presented has to be slightly modified to account for the particularities of ANC. So far, we have considered $\mathbf{B}$ to be a desired transformation encompassing the *whole* set of nodes. However, in ANC we are interested in the transmission of information from sources to sinks that are, in general, a subset of the nodes in a graph. To be precise, denote by $\mathcal{S} := \{s_1, \ldots, s_S\}$ the set of S sources and by $\mathcal{R} := \{r_1, \ldots, r_R\}$ the set of R sinks or receivers. Since every source can have one or more receivers, we also define the surjective function $s: \mathcal{R} \to \mathcal{S}$, which identifies the source for each receiver. In ANC one is interested in transformations $\mathbf{B} = \mathbf{B}_\text{anc}$ where the i-th row of $\mathbf{B}_\text{anc}$ is equal to the canonical vector $\mathbf{e}_{s(i)}^T$ for all $i \in \mathcal{R}$. Since the values of $\mathbf{B}_\text{anc}$ for rows $i \notin \mathcal{R}$ are not relevant for the performance of ANC, one can define the reduced $R \times N$ matrix $\mathbf{B}_\mathcal{R} := \mathbf{E}_\mathcal{R}^T\mathbf{B}_\text{anc}$, where $\mathbf{E}_\mathcal{R} := [\mathbf{e}_{r_1}, \ldots, \mathbf{e}_{r_R}]$. Hence, the goal of ANC boils down to designing a filter $\mathbf{H}_\text{nv}$ such that $\mathbf{E}_\mathcal{R}^T\mathbf{H}_\text{nv}$ is as close to $\mathbf{B}_\mathcal{R}$ as possible. Most ANC setups consider that only source nodes have signals to be transmitted, so that the input signal at all other nodes can be considered zero. The graph-filter design can leverage this fact to yield a better approximation. Indeed, in such a case, upon defining matrices $\mathbf{E}_\mathcal{S} := [\mathbf{e}_{s_1}, \ldots, \mathbf{e}_{s_S}]$ and $\mathbf{B}_{\mathcal{SR}} := \mathbf{B}_\mathcal{R}\mathbf{E}_\mathcal{S} \in \mathbb{R}^{S\times R}$, the goal for ANC is to design $\mathbf{H}_\text{nv}$ such that $\mathbf{E}_\mathcal{R}^T\mathbf{H}_\text{nv}\mathbf{E}_\mathcal{S}$ is as close to $\mathbf{B}_{\mathcal{SR}}$ as possible.

Although the propositions presented throughout the paper need to be modified to accommodate for the introduction of $\mathbf{B}_\mathcal{R}$ and $\mathbf{B}_{\mathcal{SR}}$, the main results and the structure of the proofs remain the same. To be specific, consider first the case where the nodes that are not sources do not inject any input, so that the goal is to approximate the $S \times R$ matrix $\mathbf{B}_{\mathcal{SR}}$. Then, defining $\mathbf{\Phi}_{r_i,\mathcal{S}} := \mathbf{E}_\mathcal{S}^T\mathbf{\Phi}_{r_i}$ and denoting the i-th row of $\mathbf{B}_{\mathcal{SR}}$ as $\mathbf{b}_{i,\mathcal{S}}^T$, we can find the coefficients that minimize $\|\mathbf{E}_\mathcal{R}^T\mathbf{H}_\text{nv}\mathbf{E}_\mathcal{S} - \mathbf{B}_{\mathcal{SR}}\|_\text{F}$ as [cf. (13)]

$$\mathbf{c}^*_{r_i,\text{F}} = \mathbf{\Phi}^\dagger_{r_i,\mathcal{S}}\mathbf{b}_{i,\mathcal{S}} = (\mathbf{\Phi}^T_{r_i,\mathcal{S}}\mathbf{\Phi}_{r_i,\mathcal{S}})^{-1}\mathbf{\Phi}^T_{r_i,\mathcal{S}}\mathbf{b}_{i,\mathcal{S}}, \quad (21)$$

for all $r_i \in \mathcal{R}$, where the second equality holds if $\mathbf{\Phi}_{r_i,\mathcal{S}}$ has full column rank. For a given filter length, the L coefficients in $\mathbf{c}^*_{r_i,\text{F}}$ specify the optimal weights that sink node r_i must give to the original signal and the first $L-1$ shifted versions of it to resemble as close as possible (in terms of MSE) the desired linear combination of source signals $\mathbf{b}_{i,\mathcal{S}}$. When it cannot be assumed that the initial input signal at the routing nodes is zero, the goal is to approximate the $R \times N$ matrix $\mathbf{B}_\mathcal{R}$. In that case, the previous expressions for the optimal filter coefficients still hold true. The only modifications required is to substitute $\mathbf{B}_{\mathcal{SR}} = \mathbf{B}_\mathcal{R}$ and $\mathbf{E}_\mathcal{S} = \mathbf{I} \in \mathbb{R}^{N\times N}$ into the definitions of $\mathbf{\Phi}_{r_i,\mathcal{S}}$ and $\mathbf{b}_{i,\mathcal{S}}$. Further, when every node acts as both a source and a sink, (21) reduces to the original formulation in (13). Although not presented here, expressions analogue to that in (21) for the remaining optimal filter-design criteria can be derived too.

Illustrative simulations: In the following experiments, we consider 100-node Erdős-Rényi graphs with edge probability 0.1 and where every edge is assigned a random weight drawn from a uniform distribution with support $[0.5, 1.5]$. We randomly select $S = 5$ sources and, to each of these sources, we assign a sink so that $R = 5$ and $\mathbf{B}_{\mathcal{SR}} = \mathbf{I}$. We set the graph-shift operator $\mathbf{S} = \mathbf{A}$ equal to the weighted adjacency matrix. Denoting by $\mathbf{x}$ the S-sparse input signal containing the values to be transmitted by the source nodes (drawn from a standard Gaussian distribution) and by $\mathbf{y} = \mathbf{H}\mathbf{x}$ the filtered signal, we define the error e_i at sink node r_i as $e_i := |y_{r_i} - x_{s_i}|/|x_{s_i}|$. Figure 1a shows one realization of the evolution of e_i as a function of the filter degree for $i = 1, \ldots, 5$ (different colors) for node-variant filters (solid lines) and compares it with the error achieved using regular node-invariant (dashed lines) filters [11]. For the node-variant case, we obtain perfect recovery at every sink node after 6 local interactions. For the node-invariant case, there is an overall error reduction as the filter degree increases, but its performance is markedly weaker than that of the node-variant case. This is not surprising since the number of degrees of freedom to design the



(a) (b) (c)

Fig. 1: ANC applied to 100-node Erdős-Rényi graphs with $S = R = 5$. (a) Error at each of the sinks (different colors) as a function of the filter degree for node-variant (solid line) and node-invariant (dashed line) filters. (b) Mean recovery error for node-variant and node-invariant filters when approximating $\mathbf{B}_{\mathcal{R}}$ and $\mathbf{B}_{\mathcal{SR}}$. (c) Mean recovery error for different levels of correlation ρ among the values at different sources.

coefficients of a node-invariant filter is much lower.

The next goal is to evaluate the difference in terms of error performance between approximating $\mathbf{B}_{\mathcal{R}}$ and $\mathbf{B}_{\mathcal{SR}}$. Recall that the later assumes that the nodes that are not sources inject a zero input, while the former does not. Since the size of $\mathbf{B}_{\mathcal{R}}$ is N/S times larger than the size of $\mathbf{B}_{\mathcal{SR}}$, the performance is expected to be considerably lower. To corroborate this, in Figure 1b we plot the mean error across 1,000 graphs for node-invariant [11] (red) and node-variant (blue) filters when the coefficients are designed to approximate $\mathbf{B}_{\mathcal{R}}$ ($*$ marker) and $\mathbf{B}_{\mathcal{S},\mathcal{R}}$ ($\circ$ marker). The remaining parameters are the same than in the previous test-case ($S = R = 5$) and the error is defined as $e := \|\mathbf{E}_{\mathcal{R}}\mathbf{y} - \mathbf{E}_{\mathcal{S}}\mathbf{x}\|_2/\|\mathbf{E}_{\mathcal{S}}\mathbf{x}\|_2$x, i.e. the normalized difference between the signal injected at the sources and the one recovered at the sinks. Notice that if the sources are known and the relay nodes inject a zero signal, after 7 local exchanges the mean error for node-variant filters is 0. For this same filter degree the error when approximating $\mathbf{B}_{\mathcal{R}}$ is 0.93. Eventually, the latter error also vanishes, but filters of degree close to $N = 100$ are needed. By contrast, when node-invariant filters are used to approximate $\mathbf{B}_{\mathcal{R}}$ [11], the error improvement associated with increasing filter degree for the selected interval ($0 \leq L - 1 \leq 9$) is negligible. Finally, by comparing the plots for node-variant and node-invariant filters when the coefficients are designed to approximate $\mathbf{B}_{\mathcal{SR}}$, we corroborate the trend observed in Figure 1a, where node-variant filters achieve zero error after a few interactions, while node-invariant filters exhibit a slow reduction of the overall error with the filter degree.

Correlation among the injected signals at source nodes can be leveraged to reduce the error at the receivers. To illustrate this, for different values of $\rho \in \{0, 0.2, \ldots, 1\}$, we build a $S \times S$ covariance matrix $\mathbf{R}_{\mathbf{x},\rho}$ defined as $\mathbf{R}_{\mathbf{x},\rho}^{1/2} := \mathbf{I} + \rho(\mathbf{1}\mathbf{1}^T - \mathbf{I}) + 0.1\rho\mathbf{Z}$ where the elements in the *symmetric* matrix $\mathbf{Z}$ are drawn from a standard multivariate Gaussian distribution. In this way, the correlation between injected signals increases with ρ and, in each realization, is corrupted by additive zero-mean random noise. In Figure 1c we plot the mean error across 1,000 graphs as a function of the node-variant filter degree parametrized by ρ. Notice that, as ρ increases, the achieved error for a given filter degree markedly decreases. The reason for this is that high correlation allows sink nodes to build accurate estimates of their objective source signals based on *all* injected signals.

6. CONCLUSIONS

The optimal design of node-variant graph filters – a novel generalization of regular graph filters – for distributed implementation of linear network operators was investigated. We stated conditions for perfect implementation of linear operators and, for the cases where these conditions were not met, we provided optimal designs for the minimization of different error metrics. Finally, the practical relevance of our approach for distributed setups was emphasized by particularizing our analysis to the implementation of analog network coding.

7. REFERENCES

[1] D. Shuman, S. 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, Mar. 2013.

[2] A. Sandryhaila and J. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[3] S. Barbarossa, G. Scutari, and T. Battisti, "Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2009, pp. 2893–2896.

[4] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *CoRR*, vol. abs/1111.5239, 2011.

[5] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, Sept. 2015.

[6] S. Chen, A. Sandryhaila, and J. Kovacevic, "Distributed algorithm for graph signal inpainting," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 19-24, 2015.

[7] A. Sandryhaila, S. Kar, and J. Moura, "Finite-time distributed consensus through graph filters," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, May 2014, pp. 1080–1084.

[8] S. Safavi and U. Khan, "Revisiting finite-time distributed algorithms via successive nulling of eigenvalues," *IEEE Signal Process. Lett.*, vol. 22, no. 1, pp. 54–57, Jan. 2015.

[9] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Interpolation of graph signals using shift-invariant graph filters," in *European Signal Process. Conf. (EUSIPCO)*, Nice, France, Aug. 31 - Sept. 4, 2015.

[10] ——, "Reconstruction of graph signals through percolation from seeding nodes," *arXiv preprint arXiv:1507.08364*, 2015.

[11] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed implementation of network linear operators using graph filters," in *53rd Allerton Conf. on Commun. Control and Computing*, Univ. of Illinois at U-C, Monticello, IL, Sept. 30- Oct. 2 2015.

[12] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 397–408, Oct 2007.

[13] S. Zhang, S. C. Liew, and P. P. Lam, "Physical-layer network coding," in *Intl. Conf. Mobile Comp. and Netw. (MobiCom)*, 2006, pp. 358–365.

[14] M. Gavish and D. L. Donoho, "Optimal shrinkage of singular values," *arXiv preprint arXiv:1405.7511*, 2014.

[15] F. Pukelsheim, *Optimal Design of Experiments*. SIAM, 1993, vol. 50.

[16] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[17] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[18] T. Nakano, M. Moore, F. Wei, A. Vasilakos, and J. Shuai, "Molecular communication and networking: Opportunities and challenges," *IEEE Trans. Nanobiosci.*, vol. 11, no. 2, pp. 135–148, June 2012.

[19] M. S. Kuran, H. B. Yilmaz, T. Tugcu, and B. Ozerman, "Energy model for communication via diffusion in nanonetworks," *Nano Communication Networks*, vol. 1, no. 2, pp. 86 – 95, 2010.