Network Synchronizability Enhancement Using Convex Optimization

Iman Shames, Brian D. O. Anderson, Xiaofan Wang, Barış Fidan

Abstract— **In this paper we propose a method for enhancing synchronizability using convex optimization. This method is based on adding new edges to the network, later the performance of the proposed method is tested through providing some numerical examples. Furthermore, a comparison with another method for adding edges, namely edge addition using an eigenvector criterion is presented. Moreover, the extension of the proposed method to the networks with different edge weights is described.**

I. Introduction

Recently the study of behaviour of networked systems has gained a lot of attention, and researchers from different disciplines were interested in different aspects of this behaviour, e.g. rendezvous problem in multiagent systems [1], localization of sensor networks [2], etc. A very important dynamical behaviour of an interconnected network is synchronization, and many groups around the globe focused on the relationship between synchronization and the structural topology of a network, see [3], [4] and references therein.

In particular, the community is interested to find ways to enhance the network synchronizability. For instance, in [5] the authors suggested that network synchronizability can be enhanced if the coupling strength from a node is inverse to its degree. The works in [6], [7] introduced inserting weights into the network while keeping the network topology unchanged to improve the network synchronizability. Moreover, in [8] it is shown that synchronizability can be enhanced by reducing the maximal betweenness centrality (BC) of the node, while in [9] it is shown that this end can be achieved by minimizing the maximal BC of an edge. In [10], which has the closest result to this paper, the authors proposed a heuristic method for adding one edge at a time with small weight to increase synchronizability of a network.

Another closely related set of works are those concerning consensus problems in distributed systems; [11] provides an overview of existing convergence results for reaching consensus. Furthermore, in [12] the authors study convergence speed for consensus problem in both linear time-invariant and time-varying topologies. In [13] the continuous consensus functions of the initial state of the network agents are considered and the necessary and sufficient conditions characterizing any algorithm that asymptotically achieves consensus are presented. Later, this characterization is the building block to obtain various design results for networks with weighted, directed interconnection topologies.

In this paper we consider a network with undirected edges corresponding to bidirectional information links. In addition, in the beginning we assume that the weights of the edges in the network are all equal to one. However, later in the paper we show that this is not an essential assumption and we relax the equality assumption. We propose two methods to enhance synchronizability of the network. First, we try to do so by adding extra edges to the network, i.e. by changing the topology of the network in a way that this addition optimizes an optimality index. Second, we do it by changing the weights of the edges in the network in order to optimize the same optimality index. In Section II we introduce this optimality index to be the second largest eigenvalue of a matrix called the coupling matrix. Note here that in [14] a way to optimize the same optimality index is proposed for networks of mobile agents, where the agents are required to move to positions forming a setting with the optimality index acquiring its extremum. The approach in [14] used algebraic methods along with an iterative algorithm for seeking to achieve optimization. The problem is not a convex one, and the algorithm [14] is essentially a greedy one.

The structure of the paper is as follows: in the next section the necessary background is presented. In Section III we consider the problem of how to add some unit-weight edges to enhance synchronizability of the network, and present some numerical examples. In Section IV we solve the same problem but allowing the edges to be added to have predetermined arbitrary weights, and demonstrate the applicability of the method by introducing some numerical examples. In Section V we extend the results of Sections III and IV to synchronization enhancement of networks having links (edges) with arbitrary weights. Finally in Section VI we present some concluding remarks.

II. Background

Consider a network, $\mathcal{N}$, of n interconnected nodes where the node interconnection is represented by an undirected graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, with vertex set $\mathcal{V}\{i\}_{i=1}^{|\mathcal{V}|=n}$, each vertex representing a node with the same label, and edge set $\mathcal{E}=\{e_i\}_{i=1}^{|\mathcal{E}|=p}$. An edge connects two vertices if the nodes associated with those vertices have an information link, e.g. have communication, can sense each other, etc. If e_i connects vertices j and k, in the rest of this paper for simplicity we denote e_i by $\{j,k\}$. The graph $\mathcal{G}$ is called the *underlying graph* of $\mathcal{N}$. Additionally, for each node labelled i one has

This work is supported by NICTA, which is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

Iman Shames, Brian. D. O. Anderson, and Barış Fidan are with Australian National University and National ICT Australia. Xiaofan Wang is with School of Electronics, Information and Electrical Engineering Shanghai Jiao Tong University. {Iman.Shames, Brian.Anderson, Baris.Fidan}@anu.edu.au, xfwang@sjtu.edu.cn

ISBN 978-963-311-369-1

the state dynamics

$$\dot{x}_i = f(x_i) + c\sum_{j=1}^{n} a_{ij}x_j, \quad i = 1, \cdots, n \qquad \text{(II.1)}$$

where $x_i = [x_{i1}, \cdots, x_{id}]^\top \in \mathbb{R}^d$, $d \in \mathbb{N}$, is the state variable of node i, $c > 0$ is the coupling strength [4], [3], and the differentiable function f represents the dynamics of an isolated node. For the time being, for the sake of simplicity and without loss of generality, we assume that the weight for each edge is equal to one. We extend the method proposed here to the networks with different weights for different edges in Section V. Furthermore, we define coupling matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$ to be $\mathcal{A} = -\mathcal{L}$, where $\mathcal{L} = \mathcal{H}^\top\mathcal{H}$ is the Laplacian matrix associated with $\mathcal{G}$, where $\mathcal{H}$ is the incidence matrix associated with $\mathcal{G}$. One can construct $\mathcal{H} = [h_{ij}] \in \mathbb{R}^{p\times n}$ in the following way. If edge e_i is incident on nodes (vertices) j and k, where $j < k$, then $h_{ij} = 1$, and $h_{ik} = -1$, and $h_{il} = 0$ for any $l \notin \{j, k\}$. In other words, the i-th row of the matrix $\mathcal{H}$ corresponds to edge e_i. Furthermore, we make the following definition for the sake of clarity.

Definition 2.1: The complement of a graph $\mathcal{G}$ is the graph $\mathcal{G}^c$ with the same vertex set but whose edge set consists of the edges not present in $\mathcal{G}$ (i.e., the complement of the edge set of $\mathcal{G}$ with respect to all possible edges on the vertex set of $\mathcal{G}$).

For connected $\mathcal{G}$, the following relation between the eigenvalues of $\mathcal{A}$ holds [15], [16].

$$0 = \lambda_1(\mathcal{A}) > \lambda_2(\mathcal{A}) \geq \cdots \geq \lambda_n(\mathcal{A}) \qquad \text{(II.2)}$$

The system (II.1) is known [3] to obtain synchronization exponentially fast, i.e. $x_1(t) = \cdots = x_n(t) = z(t)$ as $t \to \infty$, if $\mathcal{G}$ is connected and

$$c \geq \frac{|\bar{q}|}{|\lambda_2|}, \qquad \text{(II.3)}$$

$\bar{q}$ is the largest value of $q < 0$ that for some $d \times d$ diagonal positive definite matrix D and $\tau > 0$ satisfies:

$$[Df(z(t)) + qI_d]^\top D + D[Df(z(t)) + qI_d] \leq -\tau I_d.$$

Here, $Df(z(t)) \in \mathbb{R}^{d\times d}$ is the Jacobian of f evaluated at $z(t)$, and $I_d \in \mathbb{R}^{d\times d}$ is an identity matrix. Note that $z(t) \in \mathbb{R}^d$ can be a stable equilibrium point, a limit cycle or a chaotic attractor of an isolated node:

$$\dot{z}(t) = f(z(t)). \qquad \text{(II.4)}$$

If the network obtains synchronization with a small c, then it is considered to have strong synchronizability. Inequality (II.3) indicates that network synchronizability depends on the second largest eigenvalue of $\mathcal{A}$, and consequently one can enhance this synchronizability (with c fixed) by decreasing the value of $\lambda_2(\mathcal{A})$. So we identify $\lambda_2(\mathcal{A})$ as an optimality index and aim to minimize it for the rest of this paper. This idea is described in the next section in more detail.

III. Enhancing Synchronizability by Adding Multiple Unit-Weight Edges

Partitioning $\mathcal{H}$ as

$$\mathcal{H} = \begin{bmatrix} h_1 \\ \vdots \\ h_p \end{bmatrix} \qquad \text{(III.1)}$$

where $h_i = [h_{i1}, \cdots, h_{in}]$, we have

$$\mathcal{A} = -\mathcal{H}^\top\mathcal{H} = -\sum_{i=1}^{p} h_i^\top h_i \qquad \text{(III.2)}$$

In addition, we define $\mathcal{G}^c(\mathcal{V}, \mathcal{E}^c)$ as the complement graph of $\mathcal{G}$, where $\mathcal{E}^c = \{e_i^c\}_{i=1}^{|\mathcal{E}^c|=p^c}$, and denote $\mathcal{H}^c = \begin{bmatrix} h_1^c \\ \vdots \\ h_{p^c}^c \end{bmatrix}$ to be its incidence matrix, where $p^c = \frac{n(n-1)}{2} - p$. Now we formally define the following problem.

Problem 3.1: Consider a network of n interconnected nodes with underlying undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The goal is to add m edges to $\mathcal{G}$, call the new graph $\mathcal{G}'$ with associated coupling matrix $\mathcal{A}'$, to obtain the smallest possible $\lambda_2(\mathcal{A}')$. Where should one add these edges to reach this goal?

First we describe $\mathcal{A}'$ in more detail. The new graph, $\mathcal{G}'$ can be described as a graph with the vertex set $\mathcal{V}$ as before and edge set $\mathcal{E}'$, where $\mathcal{E}' = \mathcal{E} \cup \mathcal{E}^+$, $\mathcal{E}^+ \subset \mathcal{E}^c$, and $|\mathcal{E}^+| = m$. Additionally, let $\mathcal{G}^+$ be the graph with $\mathcal{V}$ as vertex set and $\mathcal{E}^+$ as edge set. Call the incidence matrix and Laplacian matrix associated with this graph, $\mathcal{H}^+$ and $\mathcal{L}^+$ respectively. Thus, the Laplacian matrix of $\mathcal{G}'$, $\mathcal{L}'$ can be calculated as, $\mathcal{L}' = \mathcal{L} + \mathcal{L}^+$, and consequently $\mathcal{A}' = -\mathcal{L}'$.

We can cast Problem 3.1 as the following minimization problem.

$$\begin{array}{ll} \text{minimize} & \lambda_2\left(\mathcal{A} - \sum_{i=1}^{p^c} s_i {h_i^c}^\top h_i^c\right) \\ \text{subject to} & \mathbf{1}^\top s = m \\ & s_i \in \{0, 1\}, \quad i = 1, \cdots, p^c. \end{array} \qquad \text{(III.3)}$$

where $s = [s_1, \cdots, s_{p^c}]$, and $\mathbf{1}$ is a vector with 1 entries.

Remark 3.1: Note that in (III.3), the objective function is actually $\lambda_2(\mathcal{A}')$, and $-\sum_{i=1}^{p^c} s_i {h_i^c}^\top h_i^c = \mathcal{A}^+$. Because the s_i are all 0 or 1, the matrix $\sum_{i=1}^{p^c} s_i {h_i^c}^\top h_i^c$ is obtained by selecting certain edges (corresponding to $s_i = 1$) to constitute $\mathcal{E}^+$ and the matrix $\mathcal{A} - \sum_{i=1}^{p^c} s_i {h_i^c}^\top h_i^c$ is $\mathcal{A}'$. The constraint $\mathbf{1}^\top s = m$ ensures that precisely m additional edges are selected.

Before continuing further we introduce the following proposition:

Proposition 3.1: Minimization of the objective function of (III.3) is the same as minimization of $\sum_{j=1}^{2} \lambda_j\left(\mathcal{A} - \sum_{i=1}^{p^c} s_i {h_i^c}^\top h_i^c\right)$.

Proof: In the light of (II.2), we know $\lambda_1(\mathcal{A}') = 0$ and the result is straightforward. ■

Using Proposition 3.1 we rewrite (III.3) as

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{2} \lambda_j \left(\mathcal{A} - \sum_{i=1}^{p^c} s_i h_i^{c\top} h_i^c \right) \\ \text{subject to} & \mathbf{1}^\top s = m \\ & s_i \in \{0,1\}, \quad i = 1, \cdots, p^c \end{array} \tag{III.4}$$

Furthermore, we have the following proposition,

Proposition 3.2: Given arbitrary positive integers k and M satisfying $k \leq M$ and an arbitrary $M \times M$ symmetric matrix X, let $\Gamma_k(X)$ denote the sum of k largest eigenvalues of X. Then $\Gamma_k(.)$ is a convex function over the set of $M \times M$ symmetric matrices.

Proof: From [17] we know

$$\begin{aligned} \Gamma_k(X) &= \sum_{i=1}^{k} \lambda_i(X) \\ &= \sup\left\{ tr(Z^\top X Z) | Z \in \mathbb{R}^{M\times k}, Z^\top Z = I \right\} \end{aligned}$$

where $tr(.)$ is the trace function and $k \leq M$. For arbitrary matrices X and Y, and constant $0 \leq \theta \leq 1$, we have

$$\begin{aligned} \Gamma_k(\theta X + (1-\theta)Y) &= \sup_{Z^\top Z = I} \left[tr(Z^\top(\theta X + (1-\theta)Y)Z) \right] \\ &= \sup_{Z^\top Z=I} \left[tr(Z^\top \theta X Z) + tr(Z^\top (1-\theta) Y Z) \right] \\ &\leq \sup_{Z^\top Z=I} \left[tr(Z^\top \theta X Z) \right] + \sup_{Z^\top Z=I} \left[tr(Z^\top (1-\theta) Y Z) \right] \\ &\leq \theta \sup_{Z^\top Z=I} \left[tr(Z^\top X Z) \right] + (1-\theta) \sup_{Z^\top Z=I} \left[tr(Z^\top Y Z) \right] \end{aligned}$$

Hence $\Gamma_k(.)$ is a convex function over the set of $M \times M$ symmetric matrices. ■

Because of Proposition 3.2, the objective function in (III.4) is convex. However (III.4) has Boolean constraints as well, as a result, it cannot be solved using convex optimization techniques [18]. However, one can solve the *relaxed* version of it, which is

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{2} \lambda_j \left(\mathcal{A} - \sum_{i=1}^{p^c} s_i h_i^{c\top} h_i^c \right) \\ \text{subject to} & \mathbf{1}^\top s = m \\ & s_i \in [0,1], \quad i = 1, \cdots, p^c \end{array} \tag{III.5}$$

This minimization problem has convex constraints and can be solved using standard convex optimization techniques, as described in more detail below. Assume s^* is the solution to this problem; it is not necessarily a solution to the original problem (III.4), since s_i^* can take a non-integer value. However, one can say that the value of the objective function for s^* is a *lower bound* for the value of the objective function at a solution of the original problem, because the feasible set for the relaxed problem contains the solution to the original problem as well. Call this lower bound L_s. To generate a suboptimal solution to the nonrelaxed problem we can proceed as follows: Let $s_{i_1}^*, \cdots, s_{i_{p^c}}^*$ denote the elements of s^* rearranged in descending order; compose the p^c-vector $\hat{s}$ with entries $\hat{s}_i \in \{0,1\}$ such that $\hat{s}_{i_k} = 1$ for $k \in \{1, \cdots, m\}$ and $\hat{s}_{i_k} = 0$ for $k \in \{m+1, \cdots, 2n\}$. This way the entries of $\hat{s}$ with indices corresponding to the m largest elements of s^* are assigned to be unity and the rest to be zero. The associated objective value with $\hat{s}$, call it U_s, is an *upper bound* for the optimal objective function. We define a gap between the upper bound and the lower bound as, $\delta_s = U_s - L_s$. For the cases that δ_s is negligible one can be sure that $\hat{s}$ is the solution of the original problem. However, if δ_s is not negligible a local optimization method can be used if desired to seek other better solutions [18], [19].

Standard convex optimization perhaps may appear not immediately applicable to solving the convex optimization problem (III.5), however if we cast the problem in the Semidefinite Programming (SDP) framework it can be solved easily. Assume $F(s)$ is the affine function of $s \in \mathbb{R}^\mu$: $F(s) = F_0 + s_1 F_1 + \cdots + s_\mu F_\mu$, where $\mu \in \mathbb{N}$, $s = [s_1, \cdots, s_\mu]^\top$, and $F_0, F_1, \cdots, F_\mu$ are η-by-η symmetric matrices, and $\eta \in \mathbb{N}$. To minimize the sum of the k largest eigenvalues of $F(s)$ we can solve the following semidefinite programming problem, in t, S, s:

$$\begin{array}{ll} \text{minimize} & kt + tr(S) \\ \text{subject to} & tI + S - F(s) \geq 0 \\ & S \geq 0 \\ & S = S^\top \end{array}$$

where $S \in \mathbb{R}^{\eta\times\eta}$. For further information see [20] and references therein. For a proof the reader may refer to [21], however, we provide another simple proof here in case the aforementioned reference is not available to the reader. We state the following lemma.

Lemma 3.1: Let s^* achieve $\min \sum_{i=1}^{k} \lambda_i(F(s))$, where $\lambda_1 \geq \lambda_2 \geq \cdots, \lambda_\eta$, and let t and S satisfy

$$\begin{aligned} tI + S - F(s) &\geq 0 \\ S &\geq 0 \\ S &= S^\top \end{aligned}$$

Then

$$kt + tr(S) \geq \sum_{i=1}^{k} \lambda_i(F(s)),$$

and there exist t^* and S^* attaining the lower bound.

Proof: Without loss of generality, using diagonalization by an orthogonal matrix if necessary, suppose

$$F(s^*) = diag(\lambda_1, \cdots, \lambda_n).$$

Let $E = [I_k \; 0_{n-k}]$; then

$$\begin{aligned} tI + S - F(s^*) \geq 0 &\Rightarrow E(tI+S)E^\top - EF(s^*)E^\top \geq 0 \\ &\Rightarrow tI_k + S_{11} - diag(\lambda_1, \cdots, \lambda_k) \geq 0 \end{aligned}$$

where S_{11} is the first k-by-k diagonal block of S. Hence, $kt + tr(S_{11}) - \sum_{i=1}^{k} \lambda_i(F(s)) \geq 0$. Since $S \geq 0$, $tr(S_{11}) \leq$

$tr(S)$, so

$$kt + tr(S) \geq \sum_{i=1}^{k} \lambda_i(F(s)).$$

Next, let

$$S^* = diag(\lambda_1 - \lambda_{k+1}, \cdots, \lambda_k - \lambda_{k+1}, 0, \cdots, 0)$$
$$, t^* = \lambda_{k+1}.$$

Then we see that

$$t^* I + S^* - F(s^*) =$$
$$diag(0, \cdots, 0, 0, \lambda_{k+1} - \lambda_{k+2}, \cdots, \lambda_{k+1} - \lambda_n) \geq 0.$$

Further $kt^* + tr(S^*) = \sum_{i=1}^{k} \lambda_i(F(s))$, i.e. the lower bound is attained. ■

A. Numerical Examples

Here we introduce three numerical examples. The software package used to solve them is CVX, for more information see [22] .

Example 3.1: Consider the network with a graph as depicted in Fig. 1 with,

$$\mathcal{H} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

We are interested to add one edge; after solving the optimization problem (III.5), using the method described for determining $\hat{s}$, we set the value corresponding to $\{2, 10\}$ to one and calculate $\delta_s = 0.0873$, which is negligible compared to the lower bound $L_s = -1.3230$. Therefore, the result obtained obtained is very close to the global optimum of the cost function and the added edge is the optimum edge addition.

Example 3.2: In this example we consider the network in Example 3.1 but we aim to add 5 edges. The edges to be added are obtained by doing the same procedure in Example 3.1 are, $\{4, 10\}$, $\{1, 9\}$, $\{2, 8\}$, $\{4, 7\}$, and $\{2, 7\}$. Furthermore, $L_s = -1.1677$, and $\delta_s = 0.0543$, which is negligible.

Example 3.3: In this example we consider a network with 50 nodes, and 151 edges. It is desired to add 20 more edges. The optimization procedure used in Examples 3.2 and 3.3 is applied. The values for L_s and δ_s are respectively, -2.2986 and 1.4922, and in this case the latter is not negligible. Fig. 2 shows the network with the added edges.

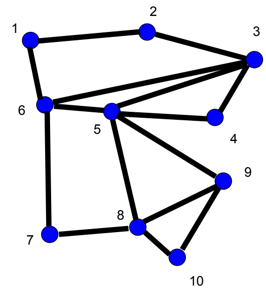

Fig. 1. The graph of the network studied in example 3.1.

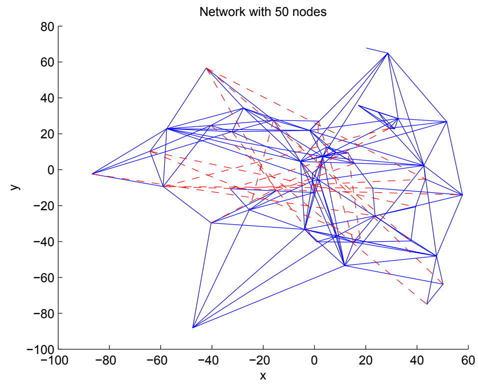

Fig. 2. The network studied in example 3.3. The solid blue lines correspond to the existing edges and the dashed red lines correspond to the newly added edges.

IV. Enhancing Synchronizability by Adding Edges with the Same Non-Unit Weights

In this section we consider the case where the edges that are going to be added to the network all have the same prescribed weight ω, which is not necessarily equal to one. As a result the optimization problem (III.5) will become:

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{2} \lambda_j \left(\mathcal{A} - \omega \sum_{i=1}^{p^c} s_i h_i^{c\top} h_i^c \right) \\ \text{subject to} & \mathbf{1}^\top s = m \\ & s_i \in [0, 1], \quad i = 1, \cdots, p^c \end{array} \tag{IV.1}$$

In [10] a method to select where to add one edge when ω is sufficiently small is proposed and it is suggested that the added edge be retained when ω is not necessarily small, but, say, 1 ; In the next example we compare the result obtained by solving (IV.1) and the one obtained by using the method in [10]. We use the same method here as used in Section III, to solve (IV.1). However, the results are not necessarily the same, since here the objective function has a parameter ω which may affect its optimum.

A. Numerical Examples

Example 4.1: Consider the network introduced in Example 3.1. It is desirable to add an edge with weight of $\omega = 0.1$ to the network. Solving (IV.1), the edge to be added is calculated to be $\{2, 10\}$, which is the same edge suggested

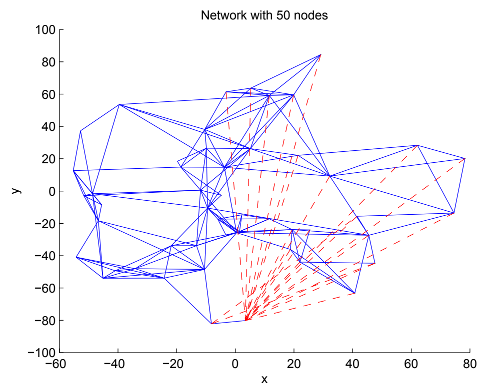

Fig. 3. The network studied in example 4.2. The solid blue lines correspond to the existing edges and the dashed red lines correspond to the newly added edges.

by the method in [10]. The δ_s associated with the solution of (IV.1) is equal to zero which means the solution indeed is the optimal solution to the problem.

Example 4.2: In this example and similarly to Example 3.3, we consider a network with 50 nodes and 150 edges, and we are interested to add 20 new edges each with a weight $\omega = 0.1$. The values for L_s and δ_s are respectively, -0.7557 and 0.0148, where the value for δ_s is negligible. Fig. 3 shows the network with the added edges.

V. Networks with Different Edge Weights

There are scenarios where the edges connecting different vertices have different weights. These different weights may be the result of having edges (links) with different communication bandwidth, different importance, etc. To consider these scenarios we have to redefine the matrix $\mathcal{A}$. Consider a graph $\mathcal{G}_W(\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V}$ and $\mathcal{E}$ are the vertex and edge set as described before, and $W = diag(w_1, \cdots, w_p)$ is the edge weight matrix, where $w_i > 0$ is the weight of edge e_i. Hence, we have the following definition for $\mathcal{A}_W$:

$$\mathcal{A}_W = -\mathcal{H}^\top W \mathcal{H}, \tag{V.1}$$

or equivalently,

$$\mathcal{A}_W = -\sum_{i=1}^{p} w_i h_i^\top h_i. \tag{V.2}$$

Having a definition for $\mathcal{A}_W$ one can apply the methods introduced in Section III and Section IV for selecting the edges to be added to the network, replacing $\mathcal{A}$ with $\mathcal{A}_W$. So the minimization problem for adding m edges with equal prescribed weight ω will be:

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{2} \lambda_j \left(\mathcal{A}_W - \omega \sum_{i=1}^{p^c} s_i \, h_i^{c\top} h_i^c \right) \\ \text{subject to} \quad & \mathbf{1}^\top s = m \\ & s_i \in [0,1], \quad i = 1, \cdots, p^c. \end{aligned} \tag{V.3}$$

The difference between (V.3) and (III.5) is presence of $\mathcal{A}_W$ and ω in (V.3), which affects the solution of the minimization problem

In what follows we provide an example to show how one can add a few edges with weights equal to ω to a network with weight matrix W.

A. Numerical Example

Example 5.1: In this example we consider the same network as Example 3.1. However, we assume the following weight matrix:

$$\begin{aligned} W = diag(&2.0961,\ 1.8646,\ 1.6613,\ 1.0408,\ 1.7916 \\ &1.6499,\ 0.7607,\ 0.6582,\ 0.7261,\ 0.0689 \\ &0.7797,\ 1.1219,\ 1.3396,\ 2.9646,\ 0.6798), \end{aligned}$$

It is desired to add 5 edges with weight $\omega = 0.25$ in order to enhance the synchronizability. Solving (V.3) we get $L_s = -1.1302$ and $\delta_s = .0098$ which is negligible. The proposed edges to be added are, $\{2, 10\}$, $\{2, 9\}$, $\{4, 9\}$, $\{4, 10\}$, and $\{1, 10\}$. This addition decreases the second largest eigenvalue from -0.4784 to -1.1302.

VI. Concluding Remarks

In this paper a method for enhancing synchronizability of a network of interconnected nodes is presented. This method is heuristically appealing and can be computed in short time intervals (On a 2.0GHz computer it takes 1 second to calculate a candidate for edge addition). The proposed method is based on adding new edges to increase synchronizability of the network, and its applicability is demonstrated via several numerical examples. In addition, the result obtained using this method is compared with the result obtained using another method introduced in [10]. While the latter requires small weights for added edges to the network, the former does not require such condition.

References

[1] J. Lin, A. S. Morse, and B. D. O. Anderson, “The multi-agent rendezvous problem: an extended summary,” in *Cooperative Control, Lecture Notes in Control and Information Sciences, Volume 309*, V. Kumar, N. Leonard, and A. S. Morse, Eds. Springer-Verlag, New York, 2004, pp. 257–289.

[2] J. Aspnes, T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, “A theory of network localization,” *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1663–1678, 2006.

[3] X. F. Wang and G. Chen, “Synchronization in scale-free dynamical networks: Robustness and fragility,” *IEEE Transactions on Circuits and Systems-I*, vol. 49, pp. 54–62, 2002.

[4] M. Barahona and L. Pecora, “Synchronization in small-world systems,” *Physical Review Letters*, vol. 89, pp. 054 101–(1–4), 2002.

[5] A. E. Motter, C. S. Zhou, and J. Kurths, “Enhancing complex-network synchronization,” *Europhysics Letters*, vol. 69, pp. 334–340), 2005.

[6] M. Chavez, D.-U. Hwang, A. Amann, H. G. E. Hentschel, and S. Boccaletti, “Synchronization is enhanced in weighted complex networks,” *Physical Review Letters*, vol. 94, p. 218701, 2005.

[7] D.-U. Hwang, M. Chavez, A. Amann, and S. Boccaletti, “Synchronization in complex networks with age ordering,” *Physical Review Letters*, vol. 94, p. 138701, 2005.

[8] M. Zhao, T. Zhou, B. Wang, and W. Wang, “Enhanced synchronizability by structural perturbations,” *Physical Review E*, vol. 72, p. 057102, 2005.

[9] K. Dai, X. F. Wang, and X. Li, “Two novel methods to enhance network synchronizability,” *Commun. Theor. Phys.*, vol. 49, pp. 1064–1068), 2008.

[10] K. Dai, X. F. Wang, and B. D. O. Anderson, “An eigenvector criterion to enhance network synchronizability,” 2008, preprint.

[11] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, Seville, Spain, Dec. 2005.
[12] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," 2006, SIAM Journal on Control and Optimization, in press.
[13] J. Cortes, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, pp. 726–737, 2008.
[14] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, pp. 116–120, 2006.
[15] C. Wu and L. Chua, "Synchronization in an array of linearly coupled dynamical systems," *IEEE Transactions on Circuits and Systems-I*, vol. 42, pp. 430–447, 1995.
[16] ——, "Application of graph theory to the synchronization in an array of coupled nonlinear oscillators," *IEEE Transactions on Circuits and Systems-I*, vol. 42, pp. 494–497, 1995.
[17] R. A. Horn and C. P. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991, reprinted 1999.
[18] S. Joshi and S. P. Boyd, "Sensor selection via convex optimization," 2008, to appear in IEEE Trans. Signal Processing.
[19] I. Shames, B. Fidan, and B. D. O. Anderson, "Minimization of the effect of noisy measurements on localization of multi-agent autonomous formations," *Automatica*, vol. 45, no. 4, pp. 1058–1065, 2009.
[20] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, pp. 49–95, 1996.
[21] Y. Nesterov and A. Nemirovsky, *Interior-point polynomial methods in convex programming*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1994.
[22] M. Grant and S. P. Boyd, "CVX: Matlab software for disciplined convex programming (web page and software)," August 2008, http://stanford.edu/ boyd/cvx.