Spectral Radius Minimization for Optimal Average Consensus and Output Feedback Stabilization [⋆]

Yoonsoo Kim [a], Da-Wei Gu [b], Ian Postlethwaite [b]

[a] *Department of Mechanical and Mechatronic Engineering, University of Stellenbosch, Private Bag X1, Matieland 7602, South Africa*

[b] *Department of Engineering, University of Leicester, Leicester, LE1 7RH, United Kingdom*

Abstract

In this paper, we consider two problems which can be posed as spectral radius minimization problems. Firstly, we consider the fastest average agreement problem on multi-agent networks adopting a linear information exchange protocol. Mathematically, this problem can be cast as finding an *optimal* $W \in \mathbf{R}^{n\times n}$ such that $x(k+1) = Wx(k)$, $W\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T W = \mathbf{1}^T$ and $W \in \mathcal{S}(\mathcal{E})$, where $x(k) \in \mathbf{R}^n$ is the value possessed by the agents at the kth time step, $\mathbf{1} \in \mathbf{R}^n$ is an all-one vector and $\mathcal{S}(\mathcal{E})$ is the set of real matrices in $\mathbf{R}^{n\times n}$ with zeros at the same positions specified by a network graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of agents and $\mathcal{E}$ is the set of communication links between agents. The optimal W is such that the spectral radius $\rho(W - \mathbf{1}\mathbf{1}^T/n)$ is minimized. To this end, we consider two numerical solution schemes: one using the qth-order spectral norm (2-norm) minimization (q-SNM) and the other gradient sampling (GS), inspired by the methods proposed in [3,20]. In this context, we theoretically show that when $\mathcal{E}$ is symmetric, i.e. no information flow from the ith to the jth agent implies no information flow from the jth to the ith agent, the solution $W_s^{(1)}$ from the 1-SNM method can be chosen to be symmetric and $W_s^{(1)}$ is a local minimum of the function $\rho(W - \mathbf{1}\mathbf{1}^T/n)$. Numerically, we show that the q-SNM method performs much better than the GS method when $\mathcal{E}$ is not symmetric. Secondly, we consider the famous static output feedback stabilization problem, which is considered to be a hard problem (some think NP-hard): for a given linear system (A,B,C), find a stabilizing control gain K such that all the real parts of the eigenvalues of $A + BKC$ are strictly negative. In spite of its computational complexity, we show numerically that q-SNM successfully yields stabilizing controllers for several benchmark problems with little effort.

Key words: Spectral radius, Distributed control, Communication networks, Output feedback.

1 Introduction and Problem Statement

A typical scenario in multi-agent missions is for the agents to agree upon a certain quantity or decision based on their current information. For example, suppose that a team of UAVs (unmanned air vehicles) is tracking a moving object and needs to sense continuously relevant data, e.g. position and heading of the object, and to communicate them to the team members in order to update the current status of the object. In practice, each UAV is likely to have limited data processing power and therefore the tracking must be done in a decentralized manner. In other words, a UAV is only capable of communicating with a limited number of adjacent UAVs. Therefore, one may easily conclude that the mission heavily hinges on the ramifications of the limited information exchange pattern. Such a decentralized tracking problem that requires each agent (processor) to do iterative weighted average operations in a decentralized manner is called *the average consensus problem*, and has been studied for numerous applications, e.g. mobile ad-hoc and wireless sensor networks. These applications include consensus with statically or dynamically changing information-exchange topologies [14], high-frequency channel noise [15], corrupted measurement data [18], network link failures [6], or state-dependent graph settings [11].

In this paper, we are particularly interested in the *optimal* matrix $W \in \mathbf{R}^{n\times n}$ (denoted by W^*) such that the

[⋆] A part of this paper was presented in the 17th IFAC Symposium on Automatic Control in Aerospace. Corresponding author Y. Kim. Tel. +27 (0)21 808 4265. Fax. +27 (0)21 808 4958.

Email addresses: `ykim@sun.ac.za` (Yoonsoo Kim), `dag@le.ac.uk` (Da-Wei Gu), `ixp@le.ac.uk` (Ian Postlethwaite).

following rule

$$x(k+1) = Wx(k), \tag{1}$$

allows $x_i(k)$ converge to $\mathbf{1}^T x(0)/n$ with minimum k^* within a prescribed tolerance for every i ($\in \{1, 2, \ldots, n\}$), where $x_i(k)$ is the value possessed by the ith agent at time step k on a network (graph) $\mathcal{G}$ with a *proper*[1] information exchange pattern $\mathcal{E}$, and k ($\in \{0, 1, 2, \ldots\}$) is the discrete-time step index. A network (graph) $\mathcal{G}$ consists of a set $\mathcal{V}$ of nodes (agents) v_i ($i = 1, 2, \ldots, n$) and a set $\mathcal{E}$ of edges (communication links) e_{ij} ($i, j = 1, 2, \ldots, n$, $i \neq j$) with weight w_{ij} on e_{ij}. The weighting factor w_{ij} is zero if no communication link exists from i to j, and can be any real (not necessarily positive) value otherwise. The position of zero weights (no communication links) defines different information exchange patterns and a set

$$\mathcal{S}(\mathcal{E}) \overset{\text{def}}{=} \{W = [w_{ij}] \in \mathbf{R}^{n \times n} \mid w_{ij} = 0 \ \text{ if } e_{ij} \notin \mathcal{E}\}.$$

We note that symmetric $\mathcal{E}$, i.e. $e_{ij} \in \mathcal{E}$ implies $e_{ji} \in \mathcal{E}$, does not necessarily imply symmetric weight W. For the sake of the average convergence of the rule (1), i.e.

$$\|W^q - \mathbf{1}\mathbf{1}^T/n\| < \epsilon \tag{2}$$

for large positive integer q and sufficiently small positive ϵ, the matrix $W \in \mathcal{S}(\mathcal{E})$ must have the following properties, as observed in [20]:

$$\begin{aligned} &W\mathbf{1} = \mathbf{1}, \ \mathbf{1}^T W = \mathbf{1}^T \\ \text{and } &\rho(W - \mathbf{1}\mathbf{1}^T/n) \overset{\text{def}}{=} \rho(\widetilde{W}) < 1, \end{aligned} \tag{3}$$

where $\mathbf{1} \in \mathbf{R}^n$ denotes an all-one vector and $\rho(X)$ is the spectral radius of matrix X. This implies that the optimal W^* is obtained when $\rho(\widetilde{W})$ or q is minimized.

[1] By a *proper* information exchange pattern at the kth time step, we mean an information exchange pattern such that (1) allows $x(k)$ to converge to $(\mathbf{1}\mathbf{1}^T/n)x_0$. In other words, for W to be associated with a proper information exchange pattern it must satisfy the three conditions in (3). For an example of an improper information exchange pattern, consider four agents whose information exchange pattern is a *one-way path*, i.e. the first agent only talks to the second, the second only to the third and the third only to the fourth. In this case, there is no W such that (3) is satisfied, and (1) thus fails to converge to the desired value. The characterization of proper information exchange patterns is under study and nontrivial, because the spectral radius condition (3) is not convex. As a special case, if one further restricts W by three linear constraints: (i) every entry of W is non-negative; (ii) the corresponding (directed) graph is *strongly connected* (see page 358 in [8] for the notion of strongly connectedness); (iii) one of the diagonal entries of W is positive; then one can drop the spectral radius condition in (3) and thus obtain at least a desired W (satisfying (3)) by solving, for example, $\mathcal{P}_s$ (to be introduced shortly) with the additional three linear constraints (see page 522 in [8]).

Finding W^* is an old problem, although the structure embedded in W may not be old. It can be translated into finding the most stable discrete-time linear system, or finding the fastest mixing Markov chain (discrete-time stochastic process) when W is non-negative (entry-wise). These areas have been popular research topics in the control community. However, finding W^* or minimizing the spectral radius matrix function $\rho(\cdot)$ is known as a very hard problem in general. This is because $\rho(\cdot)$ is continuous but neither convex nor locally Lipschtz [16]. For this reason, there are few works in the literature that directly address the problem in question. In [20], the authors approach the problem by solving the following program:

$$\begin{aligned} \mathcal{P}_s : \quad & \min_{W} \quad \|\widetilde{W}\| \\ & \text{s.t.} \quad W \in \mathcal{S}(\mathcal{E}), \ W\mathbf{1} = \mathbf{1}, \ \mathbf{1}^T W = \mathbf{1}^T \end{aligned}$$

for a given $\mathcal{G}$. The program $\mathcal{P}_s$ minimizes the largest singular value of $\widetilde{W}$, $\bar{\sigma}(\widetilde{W})$, instead of $\rho(\widetilde{W})$. Thus, the solution W_s to $\mathcal{P}_s$ only guarantees the well-known bound

$$\rho(\widetilde{W^*}) \leq \rho(\widetilde{W_s}) \leq \bar{\sigma}(\widetilde{W_s}),$$

where the gap between $\rho(\widetilde{W^*})$ and $\bar{\sigma}(\widetilde{W_s})$ can be unacceptably large in general. In [3], the authors propose the so-called *(unconstrained) gradient sampling method* to minimize the spectral abscissa $\alpha(\cdot)$ (the largest real part of the eigenvalues). The gradient sampling method is a variation of the reliable steepest descent method. It uses gradient information on the neighbourhood of each iteration point y_k, not just the gradient at the single point y_k. As a result, the gradient sampling method becomes particularly useful when it is applied to minimizing a non-smooth (non-differentiable) function such as $\rho(\cdot)$ (see [3] and §2.2 for details).

In this paper, we further develop the foregoing two ideas for finding W^*. In the following section §2.1, we first introduce *the q-th order spectral norm (2-norm) minimization method* (q-SNM) to improve the solution W_s to $\mathcal{P}_s$. We then show that 1-SNM can yield a symmetric solution $W_s^{(1)}$ such that $\bar{\sigma}(\widetilde{W_s^{(1)}}) = \rho(\widetilde{W_s^{(1)}})$ if $\mathcal{E}$ is symmetric, and, as a consequence, $W_s^{(1)}$ is a local minimizer of the objective functional in (4) with any positive integer q, and thus of the function $\rho(\widetilde{W})$. In §2.2, we propose *the constrained gradient sampling method* (CGSM) to accommodate non-smooth function minimization problems with general constraints, wherein we provide formula for computing the gradient of $\rho(\cdot)$. We finally compare the two methods by extensive numerical tests in §3. These numerical tests suggest that q-SNM is a better choice than CGSM when the information exchange pattern $\mathcal{E}$ is non-symmetric.

After the numerical tests for optimal average consensus, we further delineate the q-SNM's superiority in non-symmetric cases in §4. We consider the classical static output feedback stabilization problem which has long been considered a hard problem [19]: for a given state model (A,B,C), find a stabilizing K such that all the eigenvalues of $A+BKC$ are strictly negative. We consider several benchmark problems in which all associated $A+BKC$ have non-symmetric structures. It is then shown numerically that q-SNM successfully yields stabilizing controllers with little effort, in spite of the problems' notorious reputation. Concluding remarks follow in §5.

2 Optimal Average Consensus and Two Methods

2.1 q-SNM

The qth-order spectral norm minimization method (q-SNM) basically solves the following program: for a given $\mathcal{G}$ and a positive integer q

$$\mathcal{P}_s^{(q)}: \quad \min_{W} \quad \|W^q - \mathbf{1}\mathbf{1}^T/n\| \qquad \text{s.t.} \quad W \in \mathcal{S}(\mathcal{E}),\ W\mathbf{1}=\mathbf{1},\ \mathbf{1}^T W = \mathbf{1}^T, \tag{4}$$

where $\|\cdot\|$ denotes the 2-norm of a matrix. Note that $\mathcal{P}_s^{(1)} = \mathcal{P}_s$. If there exists W such that (2) is satisfied for some fixed q and ϵ, W must be a feasible solution to $\mathcal{P}_s^{(q)}$. Since we are interested in finding the least q, we first solve (4) with $q=1$, and then increase q to enlarge the feasible set $\mathcal{W}_q$, where

$$\begin{aligned}\mathcal{W}_q \ &= \{W \,|\, W\mathbf{1}=\mathbf{1},\ \mathbf{1}^T W = \mathbf{1}^T,\ W\in\mathcal{S}(\mathcal{E}),\\ &\quad \|W^q - \mathbf{1}\mathbf{1}^T/n\| < \epsilon\}\end{aligned}$$

for given ϵ and $\mathcal{G}$ (note that $\mathcal{W}_{q_1} \subseteq \mathcal{W}_{q_2}$ for $1 \le q_1 < q_2$). In other words, we aim to find the smallest q such that $\mathcal{W}_q$ is non-empty.

Unfortunately, $\mathcal{P}_s^{(q)}$ is not convex for $q>1$ and thus cannot be exactly solved using convex programming techniques. For this reason, we consider approaching the optimal solution to $\mathcal{P}_s^{(q)}$ from a feasible solution by iteratively solving the following *dynamic* version of $\mathcal{P}_s^{(q)}$, i.e.

$$\begin{aligned}&\min_{W(t)} \quad \|X(t) - \mathbf{1}\mathbf{1}^T/n\|\\ &\text{s.t.} \quad W(t)\in\mathcal{S}(\mathcal{E}),\ W(t)\mathbf{1}=\mathbf{1},\ \mathbf{1}^T W(t) = \mathbf{1}^T,\\ &\qquad X(t) = W(t)^q.\end{aligned}$$

For the sake of removing nonlinearity from the program, we note that (under the assumption that $W(t)$ is differentiable with respect to t)

$$X(t) = W(t)^q \tag{5}$$

if, and only if, $X(0) = W(0)^q$ and

$$\begin{aligned}X'(t) \ &= (W(t)^q)' \ = \ W'(t)W(t)^{q-1} + \\ &\quad W(t)W'(t)W(t)^{q-2} + \ldots + W(t)^{q-1}W'(t).\end{aligned}$$

We now discretize the program via $X_0 := X(0)$, $W_0 := W(0)$, $X'(t) := (X_l - X_{l-1})/\Delta t$ and $W'(t) := (W_l - W_{l-1})/\Delta t$, where Δt is a small positive constant and l is the time index. Note that the discretization is valid provided that $\|W_l - W_{l-1}\|$ is sufficiently small. In other words, the nonlinear constraint (5) can be treated as linear constraints via differentiation and linearization under certain restrictions. Thus, we actually solve the following $\overline{\mathcal{P}}_s^{(q)}$ for W_l and X_l, instead of $\mathcal{P}_s^{(q)}$: for given W_{l-1} and $X_{l-1} = W_{l-1}^q$,

$$\begin{aligned}\overline{\mathcal{P}}_s^{(q)}: \quad &\min_{W_l} \quad \|X_l - \mathbf{1}\mathbf{1}^T/n\|\\ &\text{s.t.} \quad W_l \in \mathcal{S}(\mathcal{E}),\ W_l\mathbf{1}=\mathbf{1},\ \mathbf{1}^T W_l = \mathbf{1}^T,\\ &\qquad X_l = X_{l-1} + (W_l - W_{l-1})W_{l-1}^{q-1} + \\ &\qquad\quad W_{l-1}(W_l - W_{l-1})W_{l-1}^{q-2} + \\ &\qquad\quad \ldots + W_{l-1}^{q-1}(W_l - W_{l-1})\end{aligned}$$

and

$$\|W_l - W_{l-1}\| \le \delta, \tag{6}$$

where $\delta > 0$ is a sufficiently small constant and varies dynamically (see the algorithm $\mathcal{A}_s$ below).

The choice of initial matrix W_0 is crucial for obtaining a high-quality solution to $\overline{\mathcal{P}}_s^{(q)}$. Since the program $\mathcal{P}_s^{(1)}$ (before linearization) can be globally solved using a convex program, we use the solution $W_s^{(1)}$ to $\mathcal{P}_s^{(1)}$ as the initial condition for $\overline{\mathcal{P}}_s^{(q)}$. Recalling (2), suppose we fix $\epsilon \in (0,1)$ and, after solving $\mathcal{P}_s^{(1)}$, we have

$$\|W_s^{(1)} - \mathbf{1}\mathbf{1}^T/n\| \overset{\text{def}}{=} \|\widetilde{W}_s^{(1)}\| \le \epsilon_1,$$

which implies

$$\begin{aligned}\|(W_s^{(1)})^q - \mathbf{1}\mathbf{1}^T/n\| &= \|(W_s^{(1)} - \mathbf{1}\mathbf{1}^T/n)^q\|\\ &\le \|W_s^{(1)} - \mathbf{1}\mathbf{1}^T/n\|^q\\ &= \|\widetilde{W}_s^{(1)}\|^q \le \epsilon_1^q.\end{aligned}$$

If $\epsilon_1^q < \epsilon$, then $W_s^{(1)}$ solves $\mathcal{P}_s^{(q)}$. Otherwise, we search for a better solution W near $W_s^{(1)}$ in the sense that $\rho(\widetilde{W}) < \rho(\widetilde{W}_s^{(1)})$. Since the least q such that (2) is satisfied for the fixed ϵ is unknown, we start with $q=2$ and thus solve $\overline{\mathcal{P}}_s^{(2)}$ with $W_{l-1} = W_s^{(1)}$ and $X_{l-1} = (W_s^{(1)})^2$, in the hope of finding $W_s^{(2)}$ such that $\|\widetilde{W}_s^{(2)}\|^q < \epsilon$. Once $W_s^{(2)}$ is obtained, we check if $\rho(\widetilde{W}_s^{(2)}) < \rho(\widetilde{W}_s^{(1)})$.

If so, we iterate again solving $\overline{\mathcal{P}}_s^{(2)}$ to improve the solution with new $W_{l-1} = W_s^{(2)}$ and $X_{l-1} = (W_s^{(2)})^2$. Otherwise, we solve the same problem with a smaller δ in (6) for a refined search or (if δ is too small) move on to $\overline{\mathcal{P}}_s^{(3)}$ and repeat the same procedure until q is sufficiently large (typically in our experience 7). The following summarizes the aforementioned algorithm $\mathcal{A}_s$:

```
Initialization: Fix $\delta := 10^{-3}$ and set $q := 2$.

Step 1: Solve $\mathcal{P}_s^{(1)}$, obtain $W_s^{(1)}$ and
        set $W^* := W_s^{(1)}$, $W_{l-1} := W_s^{(1)}$
        and $X_{l-1} := W_{l-1}^2$.

Step 2: Solve $\overline{\mathcal{P}}_s^{(q)}$ and obtain $W_s^{(q)}$.
        If $\rho(\widetilde{W}_s^{(q)}) < \rho(\widetilde{W}^*)$,
          set $W^* := W_s^{(q)}$, $W_{l-1} := W_s^{(q)}$
          and $X_{l-1} := W_{l-1}^q$,
          and proceed with Step 2;
        else
            if $q > 7$,
              terminate the algorithm;
            elseif $\delta < 10^{-5}$,
                set $q := q + 1$, $X_{l-1} := W_{l-1}^q$,
                $\delta := 10^{-3}$ and proceed with Step 2;
            else
                $\delta := 0.1\delta$ and proceed with Step 2.
```

For a fixed ϵ, there is no guarantee that the proposed algorithm finds a solution, $W \in \mathcal{W}_q$, and it is a challenge to determine how close the solution W is to the optimal solution W^*. However, the following result provides us with a partial answer to this question.

Proposition 1 *For a network graph having symmetric $\mathcal{E}$, $W_s^{(1)}$ can be chosen to be symmetric, so that*

$$\bar{\sigma}(W_s^{(1)} - \mathbf{1}\mathbf{1}^T/n) = \rho(W_s^{(1)} - \mathbf{1}\mathbf{1}^T/n).$$

Furthermore, the symmetric $W_s^{(1)}$ is a local minimum of the objective functional in (4) with any positive integer q.

Proof. The first claim easily follows by the convexity of the norm function. In fact, we first note that

$$\gamma W_s^{(1)} + (1-\gamma)(W_s^{(1)})^T$$

for any $\gamma \in [0,1]$, is a feasible solution to $\mathcal{P}_s^{(1)}$ when $\mathcal{E}$ is symmetric. In addition,

$$\begin{aligned}\|W - \mathbf{1}\mathbf{1}^T/n\| &= \gamma\|W - \mathbf{1}\mathbf{1}^T/n\| + \\ &\quad (1-\gamma)\|W^T - \mathbf{1}\mathbf{1}^T/n\| \\ &\geq \|\gamma W + (1-\gamma)W^T - \mathbf{1}\mathbf{1}^T/n\|.\end{aligned}$$

Thus, if $W_s^{(1)}$ is a global minimizer of the objective functional of $\mathcal{P}_s^{(1)}$, so is $\gamma W_s^{(1)} + (1-\gamma)(W_s^{(1)})^T$ for any $\gamma \in [0,1]$. This implies that $\|W_s^{(1)} - \mathbf{1}\mathbf{1}^T/n\| = \|\gamma W_s^{(1)} + (1-\gamma)(W_s^{(1)})^T - \mathbf{1}\mathbf{1}^T/n\|$. Setting $\gamma = 1/2$, we have another global symmetric minimizer, $(W_s^{(1)} + (W_s^{(1)})^T)/2$.

For the second claim, we only consider $q = 2$ because a similar argument can be applied to higher q. We first note that for a symmetric matrix V,

$$\|V^q\| = \|V\|^q.$$

Thus,

$$\begin{aligned}\|(W_s^{(1)})^q - \mathbf{1}\mathbf{1}^T/n\| &= \|(W_s^{(1)} - \mathbf{1}\mathbf{1}^T/n)^q\| \\ &= \|W_s^{(1)} - \mathbf{1}\mathbf{1}^T/n\|^q.\end{aligned}$$

This implies that if $W_s^{(1)}$ minimizes $\|W - \mathbf{1}\mathbf{1}^T/n\|$, then it also minimizes $\|W - \mathbf{1}\mathbf{1}^T/n\|^q$ over the set of symmetric W. Suppose now that there exists a non-symmetric perturbation Δ such that for a sufficiently small $\epsilon > 0$

$$\begin{aligned}\|(W_s^{(1)} + \epsilon\Delta)^2 - \mathbf{1}\mathbf{1}^T/n\| &< \|(W_s^{(1)})^2 - \mathbf{1}\mathbf{1}^T/n\| \\ &= \|\widetilde{W}_s^{(1)}\|^2,\end{aligned}$$

and therefore

$$\|(W_s^{(1)} + \epsilon\Delta^T)^2 - \mathbf{1}\mathbf{1}^T/n\| < \|\widetilde{W}_s^{(1)}\|^2.$$

Then, we have

$$\begin{aligned}&(\|(W_s^{(1)} + \epsilon\Delta)^2 - \mathbf{1}\mathbf{1}^T/n\| + \\ &\|(W_s^{(1)} + \epsilon\Delta^T)^2 - \mathbf{1}\mathbf{1}^T/n\|)/2 < \|\widetilde{W}_s^{(1)}\|^2,\end{aligned}$$

and consequently

$$\begin{aligned}&\|(W_s^{(1)} + \epsilon(\Delta + \Delta^T)/2)^2 + \epsilon^2(\Delta - \Delta^T)^2/4 - \mathbf{1}\mathbf{1}^T/n\| \\ &< \|\widetilde{W}_s^{(1)}\|^2.\end{aligned}$$

Since ϵ was chosen sufficiently small, we can conclude that

$$\|(W_s^{(1)} + \epsilon(\Delta + \Delta^T)/2)^2 - \mathbf{1}\mathbf{1}^T/n\| < \|\widetilde{W}_s^{(1)}\|^2,$$

which contradicts the assumption that $W_s^{(1)}$ minimizes $\|W^2 - \mathbf{1}\mathbf{1}^T/n\|$ when W is symmetric. This completes the proof.

Proposition 1 reveals two facts. The first part of the statement implies that one can further restrict the feasible matrices of $\mathcal{P}_s^{(1)}$ to be symmetric to obtain the same result, which was observed through numerical simulations in [20]. The second part of the statement indicates that $W_s^{(1)}$ is a local minimizer of the function $\rho(\widetilde{W})$, and thus q-SNM may not be better than 1-SNM when $\mathcal{E}$ is symmetric. This is just because of the linearization around $W_s^{(1)}$ introduced to approximately solve (4).

Note that the computational complexity of $\mathcal{A}_s$ is dominated by solving $\overline{\mathcal{P}}_s^{(q)}$ in **Step 2** at each iteration. As $\overline{\mathcal{P}}_s^{(q)}$ involves order n^2 variables, it costs order n^6L flops if one uses an interior-point method (Newton's method), where L is the length of a binary coding of the input data [17].

2.2 The Constrained Gradient Sampling Method

As briefly introduced earlier, the gradient sampling method (CGSM) seems quite attractive for optimization problems with non-smooth functions, e.g. [5]. This method subsumes and generalizes the classical steepest descent method by collecting more gradient information at each iterate. Once the iterates jam near the manifold on which the minimized functional is not differentiable, the method samples a bundle of gradients nearby the jamming point and finds a *way-out*, as opposed to the classical steepest descent method which fails to do so.

The original gradient sampling method proposed in [3] minimizes the spectral abscissa $\alpha(\cdot)$ (the largest real part of the eigenvalues) of a matrix $(\sum_{i=1}^m x_iW_i)$ with respect to scalar decision variables x_i. Here, $W_i \in \mathbf{R}^{n\times n}$ are fixed constant matrices and no constraints are imposed on x_i. Thus, for our purpose, we need to modify the method to accommodate constraints on variables and to calculate the gradient of the spectral radius function $\rho(\cdot)$ instead of $\alpha(\cdot)$. To this end, at each iteration matrix $X \in \mathbf{R}^{n\times n}$, we sample matrices Y from a uniform distribution such that $\|Y-X\| \le \epsilon$ for a sufficiently small $\epsilon > 0$, and $Y\mathbf{1} = \mathbf{1}$, $\mathbf{1}^TY = \mathbf{1}^T$ and $Y \in \mathcal{S}(\mathcal{E})$ for a network graph $\mathcal{G}$. Note that the number of decision variables (non-zero entries of X) x_i $(i = 1, 2, \ldots, m)$ is uniquely defined for each fixed $\mathcal{E}$. In order to satisfy the constraints $Y\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^TY = \mathbf{1}^T$, we need $2n$ linear constraints $h_j(x_i) = 0$ $(j = 1, 2, \ldots, 2n)$. Recalling the *constrained* steepest descent method, we therefore choose a direction d for each sampled Y such that

$$\nabla\rho(\widetilde{Y})^Td \le 0, \tag{7}$$

where $\nabla\rho(\widetilde{Y})$ is the gradient of $\rho(\cdot)$ at $Y - \mathbf{1}\mathbf{1}^T/n$, and

$$\nabla h_j(x_1, x_2, \ldots, x_m)^Td = 0 \quad \forall j = 1, 2, \ldots, 2n. \tag{8}$$

If there are multiple feasible directions d, we choose the one such that $\nabla\rho(\widetilde{Y})^Td$ is minimized.

We now present the gradient formula of $\rho(\cdot)$. The proof is motivated by that of Theorem 6.3.12 in [8], and could be deduced from existing works, e.g. [2].

Proposition 2 *Suppose $\rho(\cdot)$ is differentiable at X, and $\lambda = Re(\lambda) + iIm(\lambda)$ is the largest in magnitude eigenvalue of $X = \sum_i x_iX_i$ and it (and its conjugate pair) is algebraically simple, i.e. has multiplicity one. Then, for $k \in \{1, 2, \ldots, m\}$*

$$\frac{\partial\rho(X)}{\partial x_k} = \frac{Re(\lambda)}{|\lambda|}(u^TX_kv + \tilde{u}^TX_k\tilde{v}) + \frac{Im(\lambda)}{|\lambda|}(u^TX_k\tilde{v} - \tilde{u}^TX_kv),$$

where $u + i\tilde{u}$ and $v + i\tilde{v}$ are the left and right eigenvectors associated with λ, respectively.

Proof. The conditions given in the statement guarantee that $(u - i\tilde{u})^T(v + i\tilde{v}) \neq 0$ (see Lemma 6.3.10 in [8]), which allows the eigenvectors to be normalized such that $(u - i\tilde{u})^T(v + i\tilde{v}) = 1$. If we differentiate the normalized condition with respect to x_i, we have

$$(u' - i\tilde{u}')^T(v + i\tilde{v}) + (u - i\tilde{u})^T(v' + i\tilde{v}') = 0.$$

Consider

$$(u - i\tilde{u})^TX(v + i\tilde{v}) = \lambda(u - i\tilde{u})^T(v + i\tilde{v}) = \lambda$$

and differentiate the last equality; we then have

$$\begin{aligned}\lambda' &= (u' - i\tilde{u}')^TX(v + i\tilde{v}) + (u - i\tilde{u})^TX'(v + i\tilde{v}) + \\ &\quad (u - i\tilde{u})^TX(v' + i\tilde{v}') \\ &= \lambda\left((u' - i\tilde{u}')^T(v + i\tilde{v}) + (u - i\tilde{u})^T(v' + i\tilde{v}')\right) + \\ &\quad (u - i\tilde{u})^TX'(v + i\tilde{v}) = (u - i\tilde{u})^TX'(v + i\tilde{v},\end{aligned}$$

or

$$\begin{aligned}\frac{\partial\lambda}{\partial x_k} &= (u - i\tilde{u})^TX_k(v + i\tilde{v}) \\ &= u^TX_kv + \tilde{u}^TX_k\tilde{v} + i(u^TX_k\tilde{v} - \tilde{u}^TX_kv).\end{aligned}$$

Since $\rho(X) = |\lambda| = \sqrt{\mathrm{Re}(\lambda)^2 + \mathrm{Im}(\lambda)^2}$,

$$\begin{aligned}\frac{\partial\rho(X)}{\partial x_k} &= \frac{\partial\rho(X)}{\partial\mathrm{Re}(\lambda)}\frac{\partial\mathrm{Re}(\lambda)}{\partial x_k} + \frac{\partial\rho(X)}{\partial\mathrm{Im}(\lambda)}\frac{\partial\mathrm{Im}(\lambda)}{\partial x_k} \\ &= \frac{\mathrm{Re}(\lambda)}{|\lambda|}(u^TX_kv + \tilde{u}^TX_k\tilde{v}) \\ &\quad + \frac{\mathrm{Im}(\lambda)}{|\lambda|}(u^TX_k\tilde{v} - \tilde{u}^TX_kv).\end{aligned}$$

For numerical simulations, we use **Algorithm 1** proposed in [3], as shown in Table 1, with the following

Table 1
Algorithm 1 (Gradient Bundle) proposed in [3]

```
0. (initialization) Choose an initial feasible point $x \in \mathbf{R}^m$, an initial positive value for the sampling radius $\epsilon$, a positive radius factor $\theta < 1$, a positive integer $N$ defining the number of gradients per bundle, and two positive integers $M_1$ and $M_2$ to terminate the iterations. Initiate $k = 1$.

1. (Inner Iteration) Carry out minimization $k$, controlled by the sampling radius $\epsilon$, as follows. Initialize $j = 0$.

    (a) Define a bundle $G$ as the set of $N$ gradients $\{\nabla f(y)\}$, where $y$ takes on $N$ values: the current iterate $x$, and $N-1$ other vectors differing from $x$ by vectors whose entries are obtained by sampling from a uniform distribution on $[-\epsilon/2, \epsilon/2]$.
    (b) Define the search direction $d = -\arg\min\{\|v\|_2 : v \in \operatorname{conv} G\}$. If $d = 0$, go to Step 2.
    (c) Use a line search to find a positive step-length satisfying $f(x+td) < f(x)$ with $t \in (0, \bar{t}]$, where $\bar{t} = \arg\max\{t : \|x+td\|_\infty \leq \chi\}$ and $\chi$ is a large constant.
    (d) Replace $x$ by $x + td$. If $t = \bar{t}$, terminate.
        If $j < M_2$, increment $j$ and return to Step (a).

2. (Decrease Sampling Parameter) If $k < M_1$, replace $\epsilon$ by the smaller value $\theta_\epsilon$, increment $k$ and return to Step 1. Otherwise, terminate.
```

changes: (1) In Step 0, we choose, e.g. by solving a linear program, initial non-zero entries x_i $(i = 1, \ldots, m)$ of X such that $X\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T X = \mathbf{1}^T$ and $X \in \mathcal{S}(\mathcal{E})$ for a given network graph $\mathcal{G}$; (2) In Step 1-(a) and 1-(b), we sample Y matrices nearby the current iterate X such that $\|X - Y\| \leq \epsilon$, $Y\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T Y = \mathbf{1}^T$ and $Y \in \mathcal{S}(\mathcal{E})$, and then d is chosen such that (7) and (8) are satisfied, and $\nabla\rho(\widetilde{Y})^T d$ is minimized among Y. We modify the code (available on the internet site: *http://www.cs.nyu.edu/faculty/overton/papers/gradsamp/alg/* and written by one of the authors of [3]) accordingly.

Note that the computational complexity of the aforementioned procedure is dominated by evaluating $\nabla\rho(Y)$ N times. Hence, the number of flops per iteration is order mn^2N. More theoretical justifications for the gradient sampling algorithm may be found in [4].

Table 2
The average fractions of $\rho(\widetilde{W}_s^{(1)})$ for various n and m

n	m	2-SNM	3-SNM	4-SNM	5-SNM
5	5	0.6784	0.5225	0.3834	0.1791
5	10	0.5229	0.3996	0.2962	0.2350
10	20	0.6777	0.5988	0.2940	0.2202
10	30	0.5885	0.4807	0.2991	0.1101
10	40	0.8293	0.6960	0.5864	0.4053

3 Numerical Tests for Optimal Average Consensus

In this section, we present test examples to show the efficacy of the proposed two algorithms. We first show how much q-SNM can improve the solutions obtained via 1-SNM, and then compare q-SNM and CGSM. For each fixed number n of agents and the number m of zeros in the solution matrix W in (1), we randomly generate fifty symmetric and, respectively, nonsymmetric information exchange pattern $\mathcal{E}$ and apply the proposed algorithms to each case.

3.1 q-SNM versus 1-SNM

As discussed before, q-SNM is nothing but 1-SNM when $\mathcal{E}$ is symmetric. Therefore, all the simulations below are meant for non-symmetric $\mathcal{E}$. Table 2 tabulates the simulation results for $(n, m) = (5, 5), (5, 10), (10, 20), (10, 30)$ and $(10, 40)$, respectively. Each value represents the average fraction of $\rho(\widetilde{W}_s^{(1)})$, i.e. $\rho(\widetilde{W}_s^{(q)})/\rho(\widetilde{W}_s^{(1)})$, after iteratively applying q-SNM with different $q \in \{2, 3, 4, 5\}$, where $W_s^{(q)}$ is the solution obtained via q-SNM for each randomly generated pattern. As clearly shown, when $\mathcal{E}$ is non-symmetric, the solutions obtained via 1-SNM are greatly improved by up to over 80% after successively applying q-SNM.

3.2 q-SNM versus CGSM

Figs. 1 to 4 show comparison results between q-SNM and CGSM in terms of the spectral radii of the obtained solutions and the associated computational times. Figs. 1 and 2 are for $(n, m) = (5, 5)$, Figs. 3 and 4 for $(n, m) = (10, 20)$, Figs. 1 and 3 are for non-symmetric information exchange patterns $\mathcal{E}$ and Figs. 2 and 4 for symmetric information exchange patterns $\mathcal{E}$. The dotted lines correspond to CGSM and the solid lines to q-SNM. As the figures show, q-SNM finds the solutions with a lower spectral radius than CGSM for more than 90% of test cases when $\mathcal{E}$ is non-symmetric. In contrast, when $\mathcal{E}$ is symmetric, it is hard to compare the two methods in general. The two methods show similar performance for $(n, m) = (10, 20)$, but q-SNM is much inferior to CSGM for $(n, m) = (5, 5)$. Regarding the computational complexity of the methods, CGSM pertains to the number of

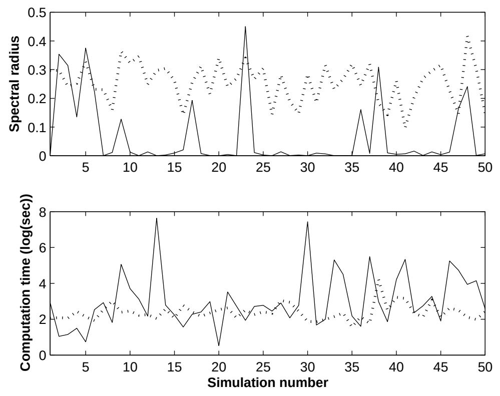

Fig. 1. Spectral radius values and computational times for $(n,m)=(5,5)$ and non-symmetric $\mathcal{E}$: solid line for q-SNM and dotted line for CGSM

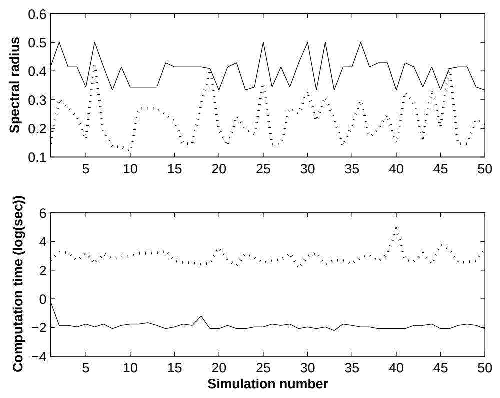

Fig. 2. Spectral radius values and computational times for $(n,m)=(5,5)$ and symmetric $\mathcal{E}$: solid line for q-SNM and dotted line for CGSM

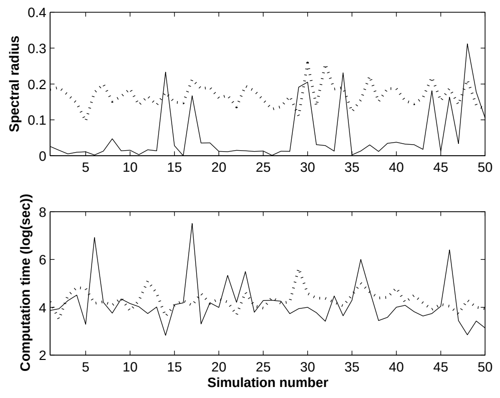

Fig. 3. Spectral radius values and computational times for $(n,m)=(10,20)$ and non-symmetric $\mathcal{E}$: solid line for q-SNM and dotted line for CGSM

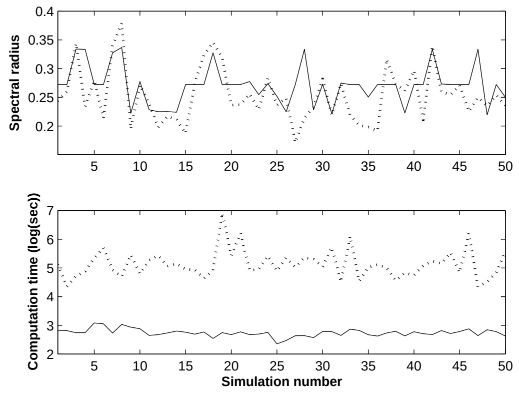

Fig. 4. Spectral radius values and computational times for $(n,m)=(10,20)$ and symmetric $\mathcal{E}$: solid line for q-SNM and dotted line for CGSM

samples for surveying gradient information and the size of the quadratic program for deciding the best direction to move at each iteration, whereas q-SNM mainly concerns the size of the quadratic or semi-definite program for solving (4). Since one can not predict which method requires the least number of mathematical programs to solve to reach a solution for a fixed information exchange pattern, the exact estimation of computational complexities of the two algorithms may not be obtained easily.

4 Static Output Feedback Stabilization and q-SNM

Having noticed that q-SNM performs well particularly for non-symmetric cases, we further test it for the famous static output feedback stabilization problem (SOFP). [2] The SOFP is stated as follows: for a given linear system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t),\\ y(t) &= Cx(t),\end{aligned}$$

find a gain K such that $u(t)=Ky(t)$ stabilzes the system, i.e. all the real parts of the eigenvalues of $A+BKC$ are strictly negative. This simply stated problem is a famous open problem in control theory [1]. With $u(t)=Ky(t)$, the closed-loop system becomes

$$\dot{x}(t) = (A+BKC)x(t)$$

or in discrete form

$$x(k+1) = (I + \Delta t(A+BKC))x(k)$$

[2] Although the SOFP is known to be theoretically difficult, many practical instances of the SOFP can be solved routinely using publicly available software, e.g. HIFOO (a MATLAB package for fixed-order controller design) available on the internet site: *http://www.cs.nyu.edu/overton/software/hifoo/*.

with $k = 0, 1, 2, \dots$ and a sufficiently small $\Delta t > 0$. Note that $\rho(I + \Delta t(A + BKC)) < 1$ implies all the real parts of the eigenvalues of $A + BKC$ are strictly negative i.e. $\max \mathrm{Re}(A + BKC) < 0$. Therefore, the SOFP can be posed as the minimization of $\rho(I + \Delta t(A + BKC))$ over a set of matrices K.

In order to handle the SOFP, the previously proposed algorithm $\mathcal{A}_s$ can be changed as follows:

Initialization: Set $\delta := 10^{-3}$, $\Delta t := 10^{-2}$, $q := 2$ and stabilizing control gain $K := \Phi$.

```
Step 1: Solve $\mathcal{P}_s^{(1)}$ and obtain $(W_s^{(1)}, K_s^{(1)})$.
        If max $\mathrm{Re}(A + BK_s^{(1)}C) < 0$,
          terminate the algorithm with $K := K_s^{(1)}$.
        Set $W^* := W_s^{(1)}$, $W_{l-1} := W_s^{(1)}$
        and $X_{l-1} := W_{l-1}^2$.

Step 2: Solve $\overline{\mathcal{P}}_s^{(q)}$ and obtain $(W_s^{(q)}, K_s^{(q)})$.
        If max $\mathrm{Re}(A + BK_s^{(q)}C) < 0$,
          terminate the algorithm with $K := K_s^{(q)}$.
        If $\rho(W_s^{(q)}) < \rho(W^*)$,
          set $W^* := W_s^{(q)}$, $W_{l-1} := W_s^{(q)}$
          and $X_{l-1} := W_{l-1}^q$,
          and proceed with Step 2;
        else
            if $q > 7$,
              terminate the algorithm;
            elseif $\delta < 10^{-5}$,
                  set $q := q + 1$, $X_{l-1} := W_{l-1}^q$,
                  $\delta := 10^{-3}$ and proceed with Step 2;
            else
                  $\delta := 0.1\delta$ and proceed with Step 2,
```

where

$$\mathcal{P}_s^{(q)} : \quad \min_{W = A + BKC} \|W^q\|$$

and $(W_s^{(1)}, K_s^{(1)})$ is the solution to $\mathcal{P}_s^{(1)}$. $\overline{\mathcal{P}}_s^{(q)}$ is similarly defined as we did in §2.1, i.e. the linearized version of $\mathcal{P}_s^{(q)}$, and $(W_s^{(q)}, K_s^{(q)})$ is the solution to $\overline{\mathcal{P}}_s^{(q)}$. If $K = \Phi$ after running the algorithm, then the considered system may have no stabilizing static output feedback controllers. As opposed to the previous version of $\mathcal{A}_s$ in §2.1, the current version can be terminated before q reaches its maximum value of 7, i.e. as soon as a stabilizing controller is found.

We now proceed with the following benchmark systems found in the literature [1,13,10,9,12]:

$$\text{Case 1: } A = \begin{bmatrix} 1 & 1.05 \\ -1.05 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; C = \begin{bmatrix} 0 & 1 \end{bmatrix};$$

$$\text{Case 2: } A = \begin{bmatrix} 1 & 1.05 & 0 \\ -1.05 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \; C = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix};$$

$$\text{Case 3: } A = \begin{bmatrix} 1 & 1.05 & 0 & 0 \\ -1.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix};$$

$$\text{Case 4: } A = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & -1.0100 & 0.0024 & -4.0208 \\ 0.1002 & 0.3681 & -0.7070 & 1.4200 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \end{bmatrix}, \quad B = \begin{bmatrix} 0.4422 & 0.1761 \\ 3.5446 & -7.5922 \\ -5.5200 & 4.4900 \\ 0.0000 & 0.0000 \end{bmatrix}, \; C = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}.$$

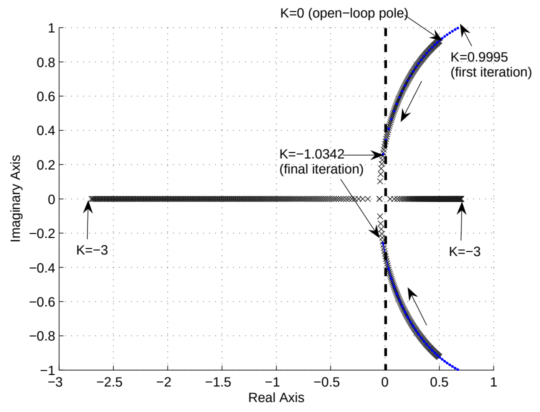

Fig. 5. The root locus plots for Case 1: the closed-loop ploes associated with different K (black crosses) and the closed-loop poles associated with each iteration of $\mathcal{A}_s$ (blue dots) [12]

$$\text{Case 5: } A = \begin{bmatrix} 1.38006 & -0.2077 & 6.7150 & -5.6760 \\ -0.5814 & -4.2900 & 0.0000 & 0.6750 \\ 1.0670 & 4.2730 & -6.6540 & 5.8930 \\ 0.0480 & 4.2730 & 1.3430 & -2.1040 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0000 & 0.0000 \\ 5.6790 & 0.0000 \\ 1.1360 & -3.1460 \\ 1.1360 & 0.0000 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The first case is particularly interesting in that the set of K which stabilizes the system is very small (see Fig. 5). As depicted in the figure, $\mathcal{A}_s$ starts out with $K = 0.9995$ and moves along the standard root-locus plot until it just enters the stabilizing zone and finds $K = -1.0342$. Table 3 summarizes the results for all the cases. Note that stabilizing gains are found in a couple of minutes for all the cases when a personal computer equipped with an Intel(R) Core(TM)2 CPU 2.00GHz is used.

It is observed that $\mathcal{A}_s$ shows stable convergence behaviour in that no internal parameters need to be changed for each case. In view of other approaches, this numerical stability is the most distinguishable feature of $\mathcal{A}_s$ and indicates the practical feasibility of the algorithm.

5 Concluding Remarks

We first considered finding the optimal $W \in \mathbf{R}^{n\times n}$ such that $W\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T W = \mathbf{1}^T$ and $W \in \mathcal{S}(\mathcal{E})$ for a given network graph $\mathcal{G}$. The optimal W is such that $\rho(W - \mathbf{1}\mathbf{1}^T/n)$ is minimized, and results in the fastest average agreement on multi-agent networks adopting the information exchange protocol $x(k+1) = Wx(k)$, where $x(k) \in \mathbf{R}^n$ is the value possessed by the agents at the kth time step. To this end, we considered two numerical solution schemes, the qth-order spectral norm minimization method (q-SNM) and the constrained gradient sampling method (CGSM). We showed that for symmetric information exchange patterns $\mathcal{E}$, the solution $W_s^{(1)}$ using 1-SNM can be chosen to be symmetric and q-SNM is nothing but 1-SNM. We also showed through extensive numerical simulations that q-SNM offers much better performance than CGSM when $\mathcal{E}$ is non-symmetric. The numerical simulation result was then elaborated in concert with the famous static output feedback problem, and was further supported by the application of q-SNM to several benchmark systems. Based on the aforementioned results, we believe that the proposed q-SNM method can offer a promising approach to many difficult optimization problems.

Table 3
The results for the five benchmark systems: Iter. and Elap. denote the number of iterations and the elapsed time (seconds) before the algorithm reaches the stabilizing gain K, respectively; $\bar{q}$ is the value of q when the algorithm is terminated

Case	K	$eig(A + BKC)$	Iter.	Elap.	$\bar{q}$
1	-1.0342	-0.0171$\pm j$0.2608	66	10.2	2
2	$\begin{bmatrix} -100.88 & -0.0920 \\ -0.0920 & -1.0372 \end{bmatrix}$	-0.0186$\pm j$0.2550, -100.88	58	10.0	2
3	$\begin{bmatrix} -109.11 & 7.2956 & 2.2455 \\ 7.2952 & -108.27 & -2.0756 \\ 2.2455 & -2.0756 & -1.1160 \end{bmatrix}$	-0.0173$\pm j$0.2584, -116.08, -101.38	46	10.2	2
4	$\begin{bmatrix} 11.151 & 28.688 \end{bmatrix}$	-179.30, -0.73, -0.0018$\pm j$0.3392	645	117	2
5	$\begin{bmatrix} 6.0984 & -21.513 \\ 7.7943 & -22.433 \end{bmatrix}$	-146.62, -9.9656, -1.7797, -0.0004	188	39.0	7

Acknowledgements

This research was supported by the UK Engineering and Physical Sciences Research Council and BAE Systems. The first author was partially supported by the Subcommittee B fund from the University of Stellenbosch in South Africa. We appreciate invaluable comments from

anonymous referees who helped us greatly improve the initial draft of this paper.

References

[1] Blondel, V., Sontag, E., Vidyasager, M., & Willems, J. (1999). *Open problems in mathematical systems and control theory.* London: Springer Verlag.

[2] Burke, J., & Overton, M. (2001). Variational analysis of non-Lipschitz spectral functions. *Mathematical Programming, 90*, 317–352.

[3] Burke, J., Lewis, A., & Overton, M. (2002). Two numertical methods for optimizing matrix stablity. *Linear Algebra and its Applications, 351-352*, 117–145.

[4] Burke, J., Lewis, A., & Overton, M. (2002). Approximating subdifferentials by random sampling of gradients. *Mathematics of Operations Research, 27*, 567–584.

[5] Burke, J., Henrion, D., Lewis, A., & Overton, M. (2006). Stabilization via nonsmooth, nonconvex optimization. *IEEE Transactions on Automatic Control, 51*(11), 1760–1769.

[6] Cortes, J., Martinez, S., & Bullo, F. (2006). Robust Rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control, 51*(8), 1289–1298.

[7] Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control, 48*(6), 988–1001.

[8] Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis.* Cambridge University Press.

[9] Leibfritz, F., & Mostafa, M. (2002). An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems. *SIAM Journal on Optimization, 12*(4), 1048–1074.

[10] Keel, L., Bhattacharyya, S., & Howze, J. (1988). Robust control with structured perturbations. *IEEE Transactions on Automatic Control, 33*(1): 68–78.

[11] Kim, Y., & Mesbahi, M. (2006). On maximizing the second smallest eigenvalue of a state-dependent Laplacian. *IEEE Transactions on Automatic Control, 51*(1), 116–120.

[12] Mesbahi, M. (2008). Personal communications.

[13] Nesterov, Y., & Nemirovskii, A. (1994). *Interior-point polynomial algorithms in convex programming.* Philadelphia: SIAM.

[14] Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control, 49*(9), 1520–1533.

[15] Olfati-Saber, R., & Shamma, J. S. (2005). Consensus filters for sensor networks and distributed sensor fusion. *Proceedings of the IEEE Conference on Desicion and Control and European Control Conference*, Seville, Spain (pp. 6698–6703).

[16] Overton, M. L., & Womersley, R. S. (1988). On miminizing the spectral radius of a nonsymmetric matrix function–optimality conditions and duality theory. *SIAM Journal on Matrix Analysis and Applications, 9*(1), 474–498.

[17] Potra, F. A., & Wright, S. J. (2000). Interior-point methods. *Journal of Computational and Applied Mathematics, 124*(1), 281–302.

[18] Ren, W., Beard, R. W., & Kingston, D. B. (2005). Multi-agent Kalman consensus with relative uncertainty. *Proceedings of the American Control Conference*, Portland, USA (pp. 1865–1870).

[19] Syrmos, V. L., Abdallah, C., Dorato, P., & Grigoriadis, K. (1997). Static output feedback: A survey. *Automatica, 33*(1), 125–137.

[20] Xiao, L., & Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems and Control Letters, 53*(1), 65–78.