Distributed Algorithm for Suppressing Epidemic Spread in Networks

Van Sy Mai, Abdella Battou and Kevin Mills

***Abstract*—This paper considers problems related to suppressing epidemic spread over networks given limited curing resources. The spreading dynamic is captured by a susceptible-infected-susceptible model. The epidemic threshold and recovery speed are determined by the contact network structure and the heterogeneous infection and curing rates. We develop a distributed algorithm that can be used for allocating curing resources to meet three potential objectives: minimize total curing cost while preventing an epidemic; maximize recovery speed given sufficient curing resources; or given insufficient curing resources, limit the size of an endemic state. The distributed algorithm is of the Jacobi type, and converges geometrically. We provide an upper bound on the convergence rate that depends on the structure and infection rates of the underlying network. Numerical simulations illustrate the efficiency and scalability of our distributed algorithm.**

***Index Terms*—optimization algorithms, agent-based systems, networked control systems.**

I. Introduction

EPIDEMIC spreading processes appear in a range of network phenomena, such as (online) social behaviors, diffusion of infections in people or computers, and cascading of failures. Dynamic behavior in such processes is among the most widely studied subjects in network science; see, e.g., [1]–[6], and also see [7], [8] for a recent survey. Many epidemic processes exhibit a critical threshold characterizing a network phase transition. When operating below the threshold, the network reaches an outbreak-free equilibrium; but otherwise enters an endemic state. The critical threshold is known [7], [9], [10] to relate to the contact graph in the network. Based on such information, many strategies for epidemic control have been proposed and studied [11]–[20]; see also [8] and references therein. Existing research in this direction focuses mostly on centralized approaches to determine efficient control policies. Clearly, centralized algorithms face scalability challenges when applied to very large networks.

In this paper, we develop distributed algorithms for optimally suppressing an epidemic spread in a network with heterogeneous population, and subject possibly to limited available resources. Specifically, we consider a *susceptible-infected-susceptible* (SIS) epidemic model on a weighted directed network. We deal with three problems: (i) find a cost-optimal resource distribution to cure the network (i.e., annihilate the epidemic spread) at a target rate, (ii) given sufficient resources, allocate them to maximize recovery speed, and (iii) given insufficient resources (i.e., an outbreak occurs), find a resource distribution that minimizes the size of the endemic state. Next, we review related, previous work.

Related works: Centralized methods for dealing with similar optimization problems include: projected gradient method for undirected unweighted graphs [14], semidefinite programming (SDP) and greedy heuristics for undirected symmetrically weighted graphs [15], [19], [20], and Geometric Programs for directed weighted networks [19]. As network size increases to be very large, these approaches become impractical to implement on a single machine. Further, these approaches require knowledge of all problem parameters, which are spread across the network. In [19], the authors rely on the community structure to reduce the problem dimension, and then extend the SDP approach to provide suboptimal solutions for directed weighted networks. In [21], a distributed algorithm based on an Alternating Direction of Multipliers Method is proposed, where every network node maintains and exchanges local estimates for full network state. Due to expensive communication cost, this algorithm does not scale well to very large networks. Further, because every node must solve a complicated convex sub-problem with dimension of at least network size, the algorithm's convergence rate has not been established. Recent work [22] develops distributed algorithms in a similar context, but with some caveats. First, all nodes must use a common step size, with an upper bound involving spectrum of L^2, where L is the network Laplacian matrix. Thus, this step size needs to be designed by a centralized unit. Second, the gradient directions for each optimization step are estimated by having the nodes implement an augmented power iteration algorithm, where a second, independent, distributed algorithm determines when to stop the iteration algorithm.

Our contributions: We develop an efficient, distributed algorithm that requires inexpensive local message-passing, but still exhibits fast convergence. Specifically, the algorithm requires each node to communicate only a subset of the state vector, and converges geometrically. We show that the algorithm output can be used to obtain optimal (or suboptimal) solutions to three essential problems regarding suppression of epidemics over a weighted directed network with some mild assumptions requiring an additional, independent consensus process to obtain a piece of centralized information.

The rest of the paper is organized as follows. In Section II, we describe the network epidemic model and associated problems of interest. Our main results are given in Sections III and IV, followed by a numerical example in Section V.

Notation and terminology: $\mathbb{R}$ denotes the set of real numbers. For a matrix $A = [a_{ij}]$, $[A]_{ij}$ and a_{ij} denote its (ij) element, A^T its transpose, and $\lambda_i(A)$ its i-th largest (in

*This work is supported by National Institute of Standards and Technology (NIST). Mention of commercial products does not imply NIST's endorsement. The authors are with the Information Technology Laboratory, NIST. {vansy.mai,abdella.battou,kevin.mills}@nist.gov

magnitude) eigenvalue. For two matrices A and B, we write $A \geq B$ if $A - B$ is a nonnegative matrix (i.e., a matrix with all nonnegative elements). Vectors are denoted by bold letters, e.g., $\mathbf{x} = [x_1, \ldots, x_n]^\mathsf{T}$, $\mathbf{1} = [1, \ldots, 1]^\mathsf{T}$ and $\mathbf{0} = [0, \ldots, 0]^\mathsf{T}$. If $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, ∇f denotes its gradient and $\partial_i f$ its partial derivative w.r.t. the i-th coordinate. A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$ and a set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ of direct edges. A directed path is a sequence of edges in the form $(i_1, i_2), (i_2, i_3), ..., (i_{k-1}, i_k)$. Node i is said to be reachable from node j if there exists a path from j to i. Each node is reachable from itself. Graph $\mathcal{G}$ is strongly connected if each node is reachable from any other node.

II. Problem Formulation and Assumptions

A. Modeling Epidemic Spread in Networks

We adopt a continuous time model, called the N-intertwined SIS model, proposed in [6], [23] in the context of directed networks similar to those considered in [10], [16]. Given a network of N agents with the underlying contact topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, each agent $i \in \mathcal{V} = \{1, 2, ..., N\}$ can be either susceptible or infected. Transition between these two states is characterized by its curing rate $\delta_i \geq 0$ and the infection rates of its neighbors $\beta_{ij} > 0, (ij) \in \mathcal{E}$. The whole network can thus be represented by a continuous-time Markov model with 2^N states. The analysis of such a model becomes difficult for large networks, as the state space grows exponentially; hence, the need for a good approximation, as described next.

Let $p_i(\tau) \in [0, 1]$ denote the probability of node i being infected at time $\tau \geq 0$ (here time is continuous). Using a mean-field approximation yields the following Markov differential equations for the infection states:

$$\frac{\mathrm{d}p_i}{\mathrm{d}\tau} = \big(1 - p_i(\tau)\big)\big(\textstyle\sum_{j \in \mathcal{N}_i^+} \beta_{ij} p_j(\tau)\big) - \delta_i p_i(\tau), \quad \forall i \in \mathcal{V}, \tag{1}$$

where $\mathcal{N}_i^+$ denotes the set of nodes that send information directly to agent i; a.k.a., in-neighbors of node i. Intuitively, the infection dynamic of each node $i \in \mathcal{V}$ is governed by two processes: (i) while healthy with probability $(1 - p_i(\tau))$, node i gets infected at rate β_{ij} from each infected neighbor $j \in \mathcal{N}_i^+$-an event with probability $p_j(\tau)$, and (ii) while infected with probability $p_i(\tau)$, node i is cured with rate δ_i. Depending on these parameters and the network structure, this nonlinear model exhibits two equilibria: a *disease-free state*, where the whole network is cured eventually (i.e., $\lim_{\tau \to \infty} p_i(\tau) = 0, \forall i \in \mathcal{V}$), and an *endemic state*, where each agent is infected with a strictly positive probability [10], [23] (i.e., $\lim_{\tau \to \infty} p_i(\tau) > 0, \forall i \in \mathcal{V}$). The phase transition can be characterized using

$$\lambda_c := \max_{1 \leq k \leq N} \mathsf{Re}\{\lambda_k(B - D)\}, \tag{2}$$

where $D = \mathrm{diag}(\delta_i)$, $B = [\beta_{ij}]$, and $\mathsf{Re}\{\lambda_k(B - D)\}$ denotes the real part of the eigenvalues of $B - D$; λ_c is also known as the stability modulus. Specifically, if $\lambda_c > 0$, the network enters an endemic state, while the disease-free state is the unique and stable equilibrium if $\lambda_c \leq 0$. In the latter case, λ_c is also an upper bound on the decay rate of $p_i(t)$; i.e., the smaller λ_c, the faster convergence to the disease-free state. Such analysis rests on the following assumption.

Assumption 1: $\mathcal{G}$ is fixed and strongly connected.

B. Problem Statements

Suppose that the cost per curing unit associated with each node $i \in \mathcal{V}$ is $w_i > 0$. Let $\mathbf{w} = [w_1, \ldots, w_N]^\mathsf{T}$. For a given curing profile $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_N]^\mathsf{T}$, we define

$$U(\boldsymbol{\delta}) := \mathbf{w}^\mathsf{T} \boldsymbol{\delta}$$

as the total network-wide cost. Note that this linear cost is similar to that in [19] and is more general than the uniform cost (or budget constraint) in [12]–[14], [17], [20], [22]. We are interested in the following three problems.

Problem 1: (Cost Minimization) *Given a desired decay rate $\lambda_0 \leq 0$, determine in a* distributed manner *an optimal curing profile $\boldsymbol{\delta}$ that minimizes the total cost:*

$$\min\nolimits_{\boldsymbol{\delta} \geq \mathbf{0}} \quad \big\{U(\boldsymbol{\delta}) \mid \lambda_c \leq \lambda_0\big\}. \tag{3}$$

The optimal cost clearly depends on the value λ_0 and will be denoted by $U^*_{\lambda_0}$. (Throughout, we use superscript * when referring to optimal values or solutions.) Note that the phase transition of the model (1) occurs at $\lambda_c = 0$ and thus U_0^* can be interpreted as the minimal cost to prevent the epidemic.

In practice, it is also often the case that a limited budget C is available for the whole network. Depending on whether the budget is sufficient to cure the network or not, it is natural to consider the following problems.

Problem 2: (Rate Maximization) *Given a total budget $C > U_0^*$, determine in a* distributed manner *a curing profile $\boldsymbol{\delta}$ that optimizes the convergence rate:*

$$\min\nolimits_{\boldsymbol{\delta} \geq \mathbf{0}} \quad \big\{\lambda_c \mid U(\boldsymbol{\delta}) \leq C\big\}. \tag{4}$$

Problem 3: (Damage Minimization) *Given an insufficient budget $C < U_0^*$, determine a (sub)optimal resource allocation to suppress the infected fraction at the endemic state:*

$$\min\nolimits_{\boldsymbol{\delta} \geq \mathbf{0}} \quad \big\{\mathbf{1}^\mathsf{T}\mathbf{v} \mid U(\boldsymbol{\delta}) \leq C\big\}, \tag{5}$$

where $\mathbf{v} = [v_1, \ldots, v_N]^\mathsf{T}$ with $v_i = \lim_{\tau \to \infty} p_i(\tau)$.

Here, Problems 2 and 3 correspond to two different scenarios with different objectives, depending on the comparative relation between C and U_0^*. We will show that such comparison can also be performed in a distributed manner.

In this paper, we assume that any distributed algorithm will be implemented in an overlay network of m controllers $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$, where $\tilde{\mathcal{V}} = \{1, ..., m\}$ and each controller $c \in \tilde{\mathcal{V}}$ has complete information about a subpart $\mathcal{C}_c$ of $\mathcal{G}$, and $\{\mathcal{C}_c\}_{c=1}^m$ is a partition of $\mathcal{G}$. We also denote by $\mathcal{C}_c$ the controller c. $\mathcal{C}_c$ and $\mathcal{C}_d$ are neighbors if $\exists (ij) \in \mathcal{E}$ with $i \in \mathcal{C}_c$ and $j \in \mathcal{C}_d$. A special case is $\tilde{\mathcal{G}} \equiv \mathcal{G}$; $\tilde{\mathcal{G}}$ can also be a network of virtual machines in the cloud. We allow two neighboring controllers to cooperate by information exchange. Our goal is to derive distributed algorithms so that each $\mathcal{C}_c$ can compute $\{\delta_i^*\}_{i \in \mathcal{C}_c}$.

III. Main Results for Problems 1 and 2

This section proceeds as follows. In Section III-A, we derive a reformulation of (3), which is an unconstrained Geometric Program. We then present our main assumptions and a distributed algorithm for solving the resulting problem. In Section III-B, we show this algorithm can also be used for dealing with Problem 2.

A. Assumptions, Algorithm and Convergence

First, note that (3) is equivalent to the following

$$\begin{aligned} &\overset{(a)}{\Leftrightarrow}\ \min_{\boldsymbol{\delta}\geq 0,\mathbf{x}>0}\{\mathbf{w}^{\mathsf{T}}\boldsymbol{\delta}\ |\ \textstyle\sum_{j\in\mathcal{V}}\beta_{ij}x_j/x_i\leq\delta_i+\lambda_0,\forall i\in\mathcal{V}\}\\ &\overset{(b)}{\Leftrightarrow}\ \min_{\mathbf{x}>0}\ G(\mathbf{x}):=\textstyle\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}}w_i\beta_{ij}x_j/x_i, \end{aligned}\tag{6}$$

where (a) holds by using the same technique in [24, §4.5.4] (see also Appendix A below) and (b) by variable eliminations since equalities hold in (a). Although (6) does not depend on λ_0, it is equivalent to (3) in the sense that an optimal solution of one problem can be inferred from the other:

$$G^*=U^*_{\lambda_0}+\lambda_0\mathbf{w}^{\mathsf{T}}\mathbf{1},\quad \delta_i^*=\textstyle\sum_{j\in\mathcal{V}}\beta_{ij}x_j^*/x_i^*-\lambda_0.\tag{7}$$

Note that when $\lambda_0=0$, (7) simply implies that $G^*=U_0^*$, i.e., (6) indeed amounts to finding the minimal cost to prevent the epidemic. Moreover, (6) is a Geometric Program; it is nonconvex in its current form but is convex in log-scale of the variables [24]. Applying a change of variables $y_i=\log x_i$ to (6) yields the following equivalent problem:

$$\min_{\mathbf{y}\in\mathbb{R}^N}\quad F(\mathbf{y}):=\textstyle\sum_{(ij)\in\mathcal{E}}w_i\beta_{ij}e^{y_j-y_i},\tag{8}$$

where the optimal value is denoted by F^*. Clearly, $F^*=G^*$. Let Y^* denote the set of optimal solutions. Y^* is nonempty; see Appendix A. Since (8) is an unconstrained convex problem with a smooth objective function, it follows that

$$\mathbf{y}^*\in Y^*\quad\Leftrightarrow\quad \partial_i F(\mathbf{y}^*)=0,\quad \forall i\in\mathcal{V}.\tag{9}$$

Clearly, $\mathbf{y}^*+c\mathbf{1}\in Y^*$ for any $\mathbf{y}^*\in Y^*$ and $c\in\mathbb{R}$.

The rest of this subsection is focused on solving (8) in a distributed manner based on condition (9). Compared to (3), (8) is more suitable for distributed implementation since each partial derivative

$$\partial_i F(\mathbf{y})=\textstyle\sum_{k\in\mathcal{N}_i^-}\tilde{b}_{ki}e^{y_i-y_k}-\sum_{j\in\mathcal{N}_i^+}\tilde{b}_{ij}e^{y_j-y_i},\tag{10}$$

where $\mathcal{N}_i^-$ denotes the set of out-neighbors of agent i and $\tilde{b}_{ij}=w_i\beta_{ij}\ \forall(ij)\in\mathcal{E}$, can be evaluated locally at the controller associated with node $i\in\mathcal{V}$ using local information from neighboring controllers that have node i's neighbors information. This will be key to developing our distributed algorithm, which differs significantly from existing ones [16], [21]. To enable local evaluations of partial derivatives, we assume the following:

Assumption 2: $\tilde{\mathcal{G}}$ is fixed, undirected and connected.

We now remark on key properties of F that will guide the development of our algorithms for solving (8). First, F is not strictly convex (hence not strongly convex) since $F(\mathbf{y})=F(\mathbf{y}+c\mathbf{1}),\forall\mathbf{y}\in\mathbb{R}^N,\forall c\in\mathbb{R}$. Second, F (as well as G) is continuously differentiable but its gradient ∇F (also ∇G) is not Lipschitz continuous and the level set $\{\mathbf{y}|F(\mathbf{y})\leq c\}$ is either empty or unbounded. Thus, most descent algorithms are not guaranteed to converge as their step size depends on either the Lipschitz parameter of ∇F or boundedness of level sets. Thus, we opt for nonlinear (or coordinate descent) algorithms.

We first consider the nonlinear Gauss-Seidel algorithm (see, e.g., [25]) applied to (8). We have the following:

Theorem 1: Let $\{\mathbf{y}(t)\}_{t\geq 0}$ be a sequence satisfying

$$\begin{aligned} y_i(t+1)=\arg\min_{\xi}F\big(&y_1(t+1),...,y_{i-1}(t+1),\xi,\\ &y_{i+1}(t),...,y_N(t)\big).\end{aligned}\tag{11}$$

Then every limit point of $\{\mathbf{y}(t)\}$ belongs to Y^*.

Proof: Since F is continuously differentiable and convex, and strictly convex in each y_i when the values of the other components of $\mathbf{y}$ are fixed, the result follows immediately from [25, Prop. 2.5, p. 208]. ∎

Here, (11) admits a closed form solution for $y_i(t+1)$ based on the partial derivative of F in (10). However, $\{\mathbf{y}(t)\}$ need not converge. Moreover, to parallelize (11) for implementation on $\tilde{\mathcal{G}}$, one has to design a coloring scheme for $\mathcal{G}$ such that the controllers can update corresponding estimates $\{y_j\}_{j\in\mathcal{C}_c},\forall c\in\tilde{\mathcal{V}}$; see, e.g., [25] for details.

To avoid such implementation requirement, we may consider the Jacobi update iteration: $y_i(t+1)=\arg\min_\xi F\big(y_1(t),...,y_{i-1}(t),\xi,y_{i+1}(t),...,y_N(t)\big),\forall i\in\mathcal{V}$, which obviously is much easier to implement in parallel. However, this update may not converge to any solution of problem (8) either. Consider, e.g., $F(\mathbf{y})=e^{y_2-y_1}+e^{y_1-y_2}$, where $\{\mathbf{y}(t)\}$ oscillates between $(y_1(0),y_2(0))$ and $(y_2(0),y_1(0))$. Below, we introduce a modified version of the Jacobi update that is guaranteed to converge geometrically.

Algorithm description: Suppose that each $\mathcal{C}_c$ is aware of local parameters $\{\tilde{b}_{ij}\}_{j\in\mathcal{N}_i^+}$ and $\{\tilde{b}_{ki}\}_{k\in\mathcal{N}_i^-}$ for all $i\in\mathcal{C}_c$. At $t=0$, each $\mathcal{C}_c$ initializes $\{y_i(0)\}_{i\in\mathcal{C}_c}$. At any $t>0$, each $\mathcal{C}_c$ updates its estimates $\{y_i\}_{i\in\mathcal{C}_c}$ using local information:

$$y_i(t+1)=y_i(t)+\alpha_i(y_i^+(t)-y_i(t)),\tag{12}$$

where $\alpha_i\in(0,1)$ is an arbitrary step size and

$$y_i^+(t):=\tfrac{1}{2}\log\big[\textstyle\sum_{j\in\mathcal{N}_i^+}\tilde{b}_{ij}e^{y_j(t)}/\sum_{k\in\mathcal{N}_i^-}\tilde{b}_{ki}e^{-y_k(t)}\big].\tag{13}$$

Subsequently, $\mathcal{C}_c$ sends $y_i(t+1)$ to its neighbors who need it. Detailed implementations are shown in Algorithm 1 below.

Algorithm 1: Distributed Cost Minimization

init: Each $\mathcal{C}_c$ selects $y_i(0)$ and collects $\{\tilde{b}_{ij}\}_{j\in\mathcal{N}_i^+}$, $\{\tilde{b}_{ki}\}_{k\in\mathcal{N}_i^-}$, $\{y_v(0)\}_{v\in\mathcal{N}_i^+\cup\mathcal{N}_i^-}$ for all $i\in\mathcal{C}_c$
for $t=1,2,\ldots$ **do**
  forall $\mathcal{C}_c\in\tilde{\mathcal{V}}$ **do**
    Update $\{y_i(t+1)\}_{i\in\mathcal{C}_c}$ by (12)–(13)
    Send $y_i(t+1)$ to neighboring controllers
$\mathcal{C}_c$: $\{y_i^*,\forall i\in\mathcal{C}_c\}$

The form of $y_i^+(t)$ is derived from condition (9), and α_i in (12) is a relaxation parameter used to avoid oscillations. Clearly, except for being synchronous, this algorithm does not require any other piece of centralized information. Its convergence and rate of convergence are shown next.

Theorem 2: Consider Algorithm 1 with any $\mathbf{y}(0)\in\mathbb{R}^N$ and $\alpha_i\in(0,1)$. Under Assumptions 1–2, $\lim_{t\to\infty}\mathbf{y}(t)=\mathbf{y}^*$ for some $\mathbf{y}^*\in Y^*$. Moreover, the convergence is geometric. Indeed, there exists $\Gamma>0$ such that $\forall i\in\mathcal{V},\forall t\geq 0$

$$|y_i(t+1)-y_i^*|\leq\Gamma\mu^t,\quad \mu=(1-2\gamma^d)^{\frac{1}{d}},\tag{14}$$

where d is the diameter of the graph obtained from $\mathcal{G}$ by replacing each directed edge with an undirected one, $\gamma = \min_{i\in\mathcal{V}}\{1-\alpha_i, \alpha_i\eta\}$ with $\eta = \frac{1}{2}\min_{i\in\mathcal{V}}\min_{\mathbf{z}\in\bar{Y}, l\in\mathcal{N}_i^+\cup\mathcal{N}_i^-}\big[\frac{\tilde{b}_{il}e^{z_l}}{\sum_{j\in\mathcal{V}}\tilde{b}_{ij}e^{z_j}} + \frac{\tilde{b}_{li}e^{-z_l}}{\sum_{k\in\mathcal{V}}\tilde{b}_{ki}e^{-z_k}}\big]$ and $\bar{Y}\subset\mathbb{R}^N$ the closed convex hull of $\{\mathbf{y}(t)\}_{t\geq 0}$.

Proof: See Appendix B. ∎

Clearly, the convergence rate μ given in (14) depends on the underlying network $\mathcal{G}$ through parameters d and η, the step sizes chosen by the controllers in $\tilde{\mathcal{G}}$, and initializations $\mathbf{y}(0)$ and weighted infection rates $\tilde{b}_{ij}$ through η. In general, solving for η exactly may not be practical since it requires not only $\tilde{B}$ but also $\bar{Y}$, which depends on $\{\mathbf{y}(t)\}$. A more practical lower bound of η, depending only on $\tilde{B}$, might be obtained as shown in the following corollary by noting that $\frac{a_1}{\sum_{i=1}^n a_i}+\frac{b_1}{\sum_{i=1}^n b_i}\geq\frac{a_1b_1}{\sum_{i=1}^n a_ib_i}, \forall\{a_i\}_{i=1}^n,\{b_i\}_{i=1}^n\subset\mathbb{R}_{\geq 0}$.

Corollary 1: If $\mathcal{N}_i^{\rm bi}:=\{j|(ij),(ji)\in\mathcal{E}\}\neq\varnothing, \forall i\in\mathcal{V}$, then

$$\eta\geq\frac{1}{2}\min_{i\in\mathcal{V}}\min_{l\in\mathcal{N}_i^{\rm bi}}\big(\tilde{b}_{il}\tilde{b}_{li}/\textstyle\sum_{j\in\mathcal{N}_i^{\rm bi}}\tilde{b}_{ij}\tilde{b}_{ji}\big)$$

Remark 1: (Solution to Problem 1) After Algorithm 1 executes[1], each $\mathcal{C}_c$ can determine $\{\delta_i^*\}_{i\in\mathcal{C}_c}$ as in (7), i.e., $\delta_i^*=\sum_{j\in\mathcal{N}_i^+}\beta_{ij}e^{y_j^*-y_i^*}-\lambda_0, \forall i\in\mathcal{C}_c$, provided λ_0 is known to all of them. This piece of information can be flooded to all the controllers from either a network designer or a controller.

B. Solution to Problem 2

We first demonstrate how the controllers can determine in a distributed manner if total budget C is sufficient, i.e., $C>G^*$. This is done by running another distributed algorithm on $\tilde{\mathcal{G}}$ (after Algorithm 1) to find the average $\frac{C-G^*}{m}$. Then, we provide a solution to Problem 2, provided $C>G^*$.

1) Distributed Computation of $\frac{C-G^*}{m}$: Assume that each $\mathcal{C}_c$ is aware of a fraction of budget, denoted by $C_{\mathcal{C}_c}$, such that $\sum_{c\in\tilde{\mathcal{V}}}C_{\mathcal{C}_c}=C$ (e.g., $C_{\mathcal{C}_c}=C/m$). Let the controllers finish Algorithm 1 (the implementation of which does not require knowledge of $C_{\mathcal{C}_c}$ or C). Then each $\mathcal{C}_c$ can compute $\tilde{f}_{\mathcal{C}_c}:=C_{\mathcal{C}_c}-\sum_{i\in\mathcal{C}_c}f_i^*$ with $f_i^*=\sum_{j\in\mathcal{N}_i^+}\tilde{b}_{ij}e^{y_j^*-y_i^*}$. Thus,

$$\frac{C-G^*}{m}=\frac{1}{m}\sum_{c=1}^m\tilde{f}_{\mathcal{C}_c}$$

can be computed in a distributed fashion by using an averaging consensus algorithm (e.g., [26]) on $\tilde{\mathcal{G}}$. Such an algorithm converges geometrically and is independent of Algorthm 1.

2) Optimal Solution: Our next result connects (4) and (6).

Theorem 3: Suppose $\mathbf{x}^*$ is an optimal solution to (6). Then, provided $C>G^*$, a solution to (4) is given by

$$\delta_i^*=\textstyle\sum_{j\in\mathcal{V}}\beta_{ij}x_j^*/x_i^*+(C-G^*)/\mathbf{1}^{\mathsf{T}}\mathbf{w},\quad\forall i\in\mathcal{V}\qquad(15)$$

with the corresponding optimal rate $\lambda_c(\boldsymbol{\delta}^*)=\frac{G^*-C}{\mathbf{1}^{\mathsf{T}}\mathbf{w}}$.

Proof: We prove by contradiction. Suppose $\boldsymbol{\delta}^*$ given by (15) is not optimal, i.e., $\exists\boldsymbol{\zeta}\geq\mathbf{0}$ such that $U(\boldsymbol{\zeta})\leq C$ and $\lambda_c(\boldsymbol{\zeta})<\lambda_c(\boldsymbol{\delta}^*)$. By Lemma 1 in Appendix A, $\exists\mathbf{x}>\mathbf{0}$ such that $(B-\mathrm{diag}(\boldsymbol{\zeta}))\mathbf{x}=\lambda_c(\boldsymbol{\zeta})\mathbf{x}$. Multiplying both sides by $\mathbf{w}^{\mathsf{T}}$ and rearranging terms yields $\sum_{(ij)\in\mathcal{E}}w_i\beta_{ij}x_j/x_i-\mathbf{w}^{\mathsf{T}}\boldsymbol{\zeta}=\lambda_c(\boldsymbol{\zeta})\mathbf{1}^{\mathsf{T}}\mathbf{w}$. Thus, $G(\mathbf{x})-U(\boldsymbol{\zeta})=\lambda_c(\boldsymbol{\zeta})\mathbf{1}^{\mathsf{T}}\mathbf{w}<\lambda_c(\boldsymbol{\delta}^*)\mathbf{1}^{\mathsf{T}}\mathbf{w}=G^*-C$. We then have $G(\mathbf{x})-G^*<U(\boldsymbol{\zeta})-C\leq 0$, which contradicts the fact that G^* is the optimal value of (6). ∎

It is now clear that both (3) and (4) are equivalent to (6), for which an optimal solution can be found by using Algorithm 1. It remains to specify how each $\mathcal{C}_c$ can determine its local optimal $\{\delta_i^*\}_{i\in\mathcal{C}_c}$ in a distributed manner. By (15), it suffices to show how each $\mathcal{C}_c$ estimates $\frac{C-G^*}{\mathbf{1}^{\mathsf{T}}\mathbf{w}}$, which equals

$$\big(\tfrac{C-G^*}{m}\big)/\big(\tfrac{1}{m}\textstyle\sum_{c=1}^m w_{\mathcal{C}_c}\big),\qquad(16)$$

where $w_{\mathcal{C}_c}=\sum_{i\in\mathcal{C}_c}w_i$ is locally known to $\mathcal{C}_c$. Clearly, (16) is the ratio of two average terms, the first of which has been computed earlier and the second can be found in a similar fashion by the controllers.

IV. Main Result for Problem 3

We now turn to the problem of suppressing the long term effect of the infection spread under an insufficient budget. Note that such insufficiency can be detected locally as shown in Section III-B1. Recall that $v_i=p_i(\infty)$ and at the steady state of (1) we have

$$\textstyle\sum_{j\in\mathcal{V}}\beta_{ij}v_j(1-v_i)=\delta_iv_i,\quad\forall i\in\mathcal{V}.\qquad(17)$$

Besides the trivial solution $\mathbf{v}=\mathbf{0}$, (17) admits a unique $\mathbf{v}>\mathbf{0}$ for each $\boldsymbol{\delta}\geq\mathbf{0}$ (see, e.g., [6], [23]) and thus $v_i(\boldsymbol{\delta})>0$ are well defined as functions of $\boldsymbol{\delta}$. However, such functions are implicit, and thus difficult to deal with. We consider the following inverse problem:

$$\min_{\mathbf{v}>\mathbf{0},\boldsymbol{\delta}\geq\mathbf{0}}\ \big\{U(\boldsymbol{\delta})\mid(17)\text{ and }\mathbf{1}^{\mathsf{T}}\mathbf{v}\leq\nu N\big\},\qquad(18)$$

where $\nu\in[0,1]$ is a constant representing a tolerable infected fraction. Here, (18) amounts to finding the minimum cost for a given level of damage. It is shown in [14] that (5) and (18) are indeed equivalent, in the sense that a solution of one problem implies a solution of the other. The latter is preferable here since we can express it as follows:

$$\min_{\mathbf{v}>\mathbf{0}}\big\{I(\mathbf{v}):=\textstyle\sum_{(ij)\in\mathcal{E}}(v_i^{-1}-1)\tilde{b}_{ij}v_j\ |\mathbf{1}^{\mathsf{T}}\mathbf{v}\leq\nu N\big\},\qquad(19)$$

where the optimal value is denoted by I_ν^*. Solving this nonconvex problem is challenging even using a centralized method. In [14], a projected gradient method is used to find a local minimum. We instead provide upper and lower bounds on I_ν^* based on the similarity between (19) and (6). For (6), a solution $\mathbf{x}^*$ can be obtained readily in a distributed fashion using Algorithm 1 above.

Theorem 4: Let $\mathbf{x}^*$ be an optimal solution of (6) and G^* the optimal value. Let $\bar{b}=\max_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}}\tilde{b}_{ji}$, $L_\nu=G^*-\nu N\bar{b}$, $U_\nu^1=G^*-\nu N\frac{\mathbf{1}^{\mathsf{T}}\tilde{B}\mathbf{x}^*}{\mathbf{1}^{\mathsf{T}}\mathbf{x}^*}$ and $U_\nu^2=(1-\nu)\mathbf{1}^{\mathsf{T}}\tilde{B}\mathbf{1}$. Then

$$L_\nu\leq I_\nu^*\leq\min\{U_\nu^1,U_\nu^2\},\quad\forall\nu\in(0,1),\qquad(20)$$

where $L_\nu=I_\nu^*=U_\nu^1$ iff $\sum_{j\in\mathcal{V}}\tilde{b}_{ji}=\bar{b}$ for all $i\in\mathcal{V}$.

Proof: For any feasible point $\mathbf{v}$ of (19), we have $I(\mathbf{v})=G(\mathbf{v})-\mathbf{1}^{\mathsf{T}}\tilde{B}\mathbf{v}\geq G(\mathbf{v})-\bar{b}\mathbf{1}^{\mathsf{T}}\mathbf{v}\geq G^*-\bar{b}\nu N$. The lower bound then follows. We now prove the upper bound. First, notice that $I_\nu^*\leq I(\nu\mathbf{1})=(1-\nu)\mathbf{1}^{\mathsf{T}}\tilde{B}\mathbf{1}$. Second, it can be seen that

$$\mathbf{u}_\nu:=\nu N\mathbf{x}^*/\mathbf{1}^{\mathsf{T}}\mathbf{x}^*\qquad(21)$$

[1]Although Algorithm 1 converges only in the limit as $t\to\infty$, its geometric convergence rate allows the controllers to terminate the algorithm after a sufficiently large number of steps with reasonable accuracy; designing such a number is out of the paper's focus and is left for future work.

is an optimal solution of (6) and is feasible for (19). Thus, $I_\nu^* \le I(\mathbf{u}_\nu) = G(\mathbf{u}_\nu) - \mathbf{1}^\mathsf{T}\tilde{B}\mathbf{u}_\nu = G^* - \nu N\mathbf{1}^\mathsf{T}\tilde{B}\mathbf{x}^*/\mathbf{1}^\mathsf{T}\mathbf{x}^*$, where equality holds if $\sum_{j\in\mathcal{V}}\tilde{b}_{ji} = \sum_{j\in\mathcal{V}}\tilde{b}_{jk}, \forall i,k \in \mathcal{V}$. It remains to show the "only if" statement. To this end, we prove that if $\sum_{j\in\mathcal{V}}\tilde{b}_{ji} < \sum_{j\in\mathcal{V}}\tilde{b}_{jk}$ for some $(ik) \in \mathcal{E}$, then $I(\mathbf{u}_\nu)$ is not optimal. Let $\mathbf{u}_\nu^\epsilon = \mathbf{u}_\nu + \epsilon(\mathbf{e}_i - \mathbf{e}_k)$, where $\epsilon \in \mathbb{R}$ and $\mathbf{e}_i$ denotes the i-th unit vector in $\mathbb{R}^N$. Since $\mathbf{u}_\nu > \mathbf{0}$, $\exists \epsilon_0 > 0$ such that $\mathbf{u}_\nu^\epsilon$ is a feasible point of (19) for all $\epsilon \in [-\epsilon_0, \epsilon_0]$. Using the optimality condition of G, we can show that $\frac{\mathrm{d}}{\mathrm{d}\epsilon} I(\mathbf{u}_\nu^\epsilon)|_{\epsilon=0} = \sum_j \tilde{b}_{ji} - \sum_j \tilde{b}_{jk} < 0$. Thus, $\exists \epsilon \in (0, \epsilon_0)$ such that $I(\mathbf{u}_\nu^\epsilon) < I(\mathbf{u}_\nu)$, i.e., $I(\mathbf{u}_\nu)$ is not minimal. ■

In general, U_ν^1 and L_ν get tighter as $\nu \to 0$, while U_ν^2 is tighter as $\nu \to 1$. Here, U_ν^2 corresponds to an in-degree-based solution $\delta_i = \sum_{j\in\mathcal{V}} \tilde{b}_{ij}(1-\nu)$, while U_ν^1 and L_ν correspond to out-degree-based ones. The regularity condition $\sum_{j\in\mathcal{V}}\tilde{b}_{ji} = \sum_{j\in\mathcal{V}}\tilde{b}_{jk}, \forall i,k \in \mathcal{V}$ yields optimality for any ν. We can interpret $\min_i U_\nu^i$ and L_ν, respectively, as sufficient and necessary costs to ensure that the infected fraction is less than ν. Similarly, for a given budget C, simply by solving $C = \min_i U_{\nu_u}^i$ and $C = L_{\nu_l}$, we can find an upper bound ν_u and lower bound ν_l on the final infected fraction.

We conclude this section by showing that the upper bound can be used to obtain a suboptimal solution $\boldsymbol{\delta}^\dagger$ to Problem 3. Given C, solving $C = U_\nu^1$ yields $\nu = \frac{G^*-C}{\mathbf{1}^\mathsf{T}\tilde{B}\mathbf{x}^*}\frac{\mathbf{1}^\mathsf{T}\mathbf{x}^*}{N}$, which can be ensured with $\mathbf{v} = \mathbf{u}_\nu$ as in (21). By (17), we have

$$\delta_i^\dagger = \sum_{j\in\mathcal{N}_i^+}\beta_{ij}\frac{x_j^*}{x_i^*} - \frac{G^*-C}{\sum_{i\in\mathcal{V}, j\in\mathcal{N}_i^+}\tilde{b}_{ij}x_j^*}\sum_{j\in\mathcal{N}_i^+}\beta_{ij}x_j^*.$$

Note that after running Algorithm 1, each $\mathcal{C}_c$ knows x_i^* and $h_i := \sum_{j\in\mathcal{N}_i^+}\tilde{b}_{ij}x_j^*, \forall i \in \mathcal{C}_c$. Then, the controllers can run additional average consensus algorithms (e.g., [26]) to compute $\frac{G^*-C}{\sum_{i\in\mathcal{V}}h_i} = \big(\frac{G^*-C}{m}\big)/\big(\frac{1}{m}\sum_{i\in\mathcal{V}}h_i\big)$; see also Section III-B1. Therefore, $\delta_i^\dagger$ can be obtained in a distributed manner.

V. Simulation Example

Consider a network based on the largest strongly connected component of a Wikipedia vote network studied in [27] (data available at http://snap.stanford.edu/data/wiki-Vote.html) with 1300 nodes and 39456 edges. We assume $\mathbf{w} = \mathbf{1}$ and generate the infection rate $b_{ij} \in (0,1)$ randomly for each edge $(ij) \in \mathcal{E}$. Our simulations are carried out in Matlab on a laptop computer with 8GB RAM and CPU $i5@2.4$GHz. We first solve (8) by using Algorithm 1 with $\tilde{\mathcal{V}} \equiv \mathcal{V}$ and compare the results with a centralized interior-point method (IPM). For the latter, we use a package [28], which we denote by OPTI-H when the exact Hessian matrix is supplied and by OPTI when the Hessian matrix is approximated by using a Quasi-Newton algorithm.

Fig. 1 shows convergence of Algorithm 1 after 100 iterations with $\mathbf{y}(0) = \mathbf{0}$ and $\alpha_i \equiv \alpha$ for different values of α (the case $\alpha_i = \mathrm{rand}$ means each α_i is chosen randomly in $[0.1, 1]$) and that of the IPM. While OPTI and OPTI-H take $245ms$ and $6.954s$, respectively, Algorithm 1 takes only $71ms$ on average. We notice that Algorithm 1 still converges for some $\alpha \ge 1$ but diverges if $\alpha \ge 1.3$. We also evaluate the bounds in Theorem 4 and compute an improved upper bound, namely $U_\nu^3 := I(\mathbf{v}^\dagger)$ where $\mathbf{v}^\dagger$ is a local optimizer of (19) obtained by using the centralized IPM with an initial guess as in (21) if $U_\nu^1 < U_\nu^2$, or $\mathbf{y}(0) = \log(\nu)\mathbf{1}$ otherwise. Here, L_ν is rather loose and U_ν^1–L_ν is large due to large variation of $\mathbf{1}^\mathsf{T}B$.

Appendix

A. Equivalence of (3) and (6) and Existence of Solutions

The following result, similar to [18, Lem. 3], is essentially an application of the Perron-Frobenius theorem [29].

Lemma 1: Let $B, D \in \mathbb{R}_{\ge 0}^{N\times N}$ such that B is irreducible and D is diagonal. Let $\nu = \max_i \mathsf{Re}\{\lambda_i(B-D)\}$. Then, $\exists \mathbf{x} > \mathbf{0}$ (unique up to a scaling factor) satisfying $(B-D)\mathbf{x} = \nu\mathbf{x}$.

Equivalence: Straightforward application of Lemma 1.

Existence of Solutions: Consider (3). Note that its feasible set is nonempty. This can be seen by choosing $\boldsymbol{\delta} = c\mathbf{1}$ for sufficiently large $c > 0$ and using Lemma 1. Now, fix such a number c and consider an equivalent problem $\min_{\boldsymbol{\delta}\ge 0}\ \{U(\boldsymbol{\delta}) \mid \mathbf{w}^\mathsf{T}\boldsymbol{\delta} \le cN, \lambda_c(B-D) \le \lambda_0\}$, where the cost function is continuous and the constraint set is bounded. Since $\lambda_c(B - \mathrm{diag}(\boldsymbol{\delta}))$ is continuous in $\boldsymbol{\delta}$, the constraint set is also closed. Thus an optimal solution $\boldsymbol{\delta}^*$ exists.

B. Proof of Theorem 2

Let $\{\mathbf{y}(t)\}$ be a sequence generated by Algorithm 1. We first show that $\{\mathbf{y}(t)\}$ is bounded. Then we will show that $\{\mathbf{y}(t)\}$ must converge geometrically to an optimal solution.

Let $g_i(\mathbf{y}) := \frac{1}{2}\log\big(\sum_{j\in\mathcal{V}}\tilde{b}_{ij}e^{y_j}/\sum_{k\in\mathcal{V}}\tilde{b}_{ki}e^{-y_k}\big), \forall i \in \mathcal{V}$. Clearly, g_i is continuously differentiable and $g_i(\mathbf{y}(t)) = y_i^+(t), \forall t \ge 0$; see (13). Now fix a $\mathbf{y}^\dagger \in Y^*$ (see Appendix A for the existence) and define $\boldsymbol{\varepsilon}(t) := \mathbf{y}(t) - \mathbf{y}^\dagger$. By (12),

$$\varepsilon_i(t+1) = (1-\alpha_i)\varepsilon_i(t) + \alpha_i\big(g_i\big(\mathbf{y}(t)\big) - y_i^\dagger\big).$$

Using the fact that $g_i\big(\mathbf{y}^\dagger\big) = y_i^\dagger$ (cf. the optimality condition (9)) and the Mean Value Theorem, we have

$$g_i(\mathbf{y}(t)) - y_i^\dagger = \nabla g_i\big(\mathbf{z}^{(i)}(t)\big)^\mathsf{T}(\mathbf{y}(t) - \mathbf{y}^\dagger),$$

with $\mathbf{z}^{(i)}(t) := (1-c_t^{(i)})\mathbf{y}(t) + c_t^{(i)}\mathbf{y}^\dagger$ and $c_t^{(i)} \in [0,1]$. Thus,

$$\varepsilon_i(t+1) = (1-\alpha_i)\varepsilon_i(t) + \alpha_i\textstyle\sum_{l\in\mathcal{V}}\partial_l g_i\big(\mathbf{z}^{(i)}(t)\big)\varepsilon_l(t)$$

with $\partial_l g_i(\mathbf{z}) = \frac{1}{2}\big[\frac{\tilde{b}_{il}e^{z_l}}{\sum_{j\in\mathcal{V}}\tilde{b}_{ij}e^{z_j}} + \frac{\tilde{b}_{li}e^{-z_l}}{\sum_{k\in\mathcal{V}}\tilde{b}_{ki}e^{-z_k}}\big], \forall \mathbf{z} \in \mathbb{R}^N$.

Let $A(t) = [a_{ij}(t)]$ satisfy $a_{ii}(t) = 1-\alpha_i, \forall i \in \mathcal{V}$, $a_{ij}(t) = \alpha_i\partial_j g_i\big(\mathbf{z}^{(i)}(t)\big), \forall j \in \mathcal{N}_i^+ \cup \mathcal{N}_i^-$; otherwise $a_{ij}(t) = 0$. Then,

$$\boldsymbol{\varepsilon}(t+1) = A(t)\boldsymbol{\varepsilon}(t). \tag{22}$$

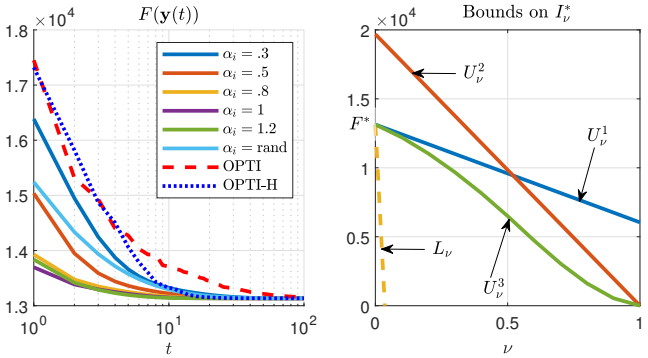

Fig. 1. (Color online) *Left:* Convergence of Algorithm 1 and IPM. *Right:* Upper bounds (solid lines) and lower bound (dashed line) of I_ν^* in Thm. 4.

Moreover, since $\partial_l g_i(\mathbf{z}) \geq 0, \forall l \in \mathcal{V}$ and $\sum_{l\in\mathcal{V}} \partial_l g_i(\mathbf{z}) = 1, \forall \mathbf{z} \in \mathbb{R}^N, \forall i \in \mathcal{V}$, we have $A(t) \geq \mathbf{0}$ and $A(t)\mathbf{1} = \mathbf{1}, \forall t \geq 0$, which means that $\varepsilon_i(t+1)$ is always a convex combination of $\{\varepsilon_j(t)\}_{j\in\mathcal{V}}$. As a result, the following hold for all $t \geq 0$

$$\min_{i\in\mathcal{V}} \varepsilon_i(t) \leq \min_{i\in\mathcal{V}} \varepsilon_i(t+1) \leq \max_{i\in\mathcal{V}} \varepsilon_i(t+1) \leq \max_{i\in\mathcal{V}} \varepsilon_i(t). \quad (23)$$

Thus, we conclude that $\{\mathbf{y}(t)\}_{t\geq 0}$ is a bounded sequence.

Next, we show that $\{\mathbf{y}(t)\}$ converges to some element in Y^*. The idea is to show that (22) is a consensus iteration. Let $\Omega \subset \mathbb{R}^N$ be any compact and convex set containing $\{\mathbf{y}(t)\}$ and $\mathbf{y}^\dagger$; boundedness of Ω is possible because of the boundedness of $\{\mathbf{y}(t)\}$. Note that $\mathbf{z}^{(i)}(t) \in \Omega$, since it is a convex combination of $\mathbf{y}(t)$ and $\mathbf{y}^\dagger$. Since $\partial_l g_i$ is continuous and $\partial_l g_i\big(\mathbf{z}\big) > 0$ for any $i \in \mathcal{V}, l \in \mathcal{N}_i^+ \cup \mathcal{N}_i^-$ and $\forall \mathbf{z} \in \mathbb{R}^N$, $\min_{\mathbf{z}\in\Omega}\{\partial_l g_i(\mathbf{z}) | l \in \mathcal{N}_i^+ \cup \mathcal{N}_i^-\}$ is achieved and positive. Thus

$$\tilde{\gamma} := \min_{i\in\mathcal{V},\mathbf{z}\in\Omega}\{1-\alpha_i, \alpha_i \partial_l g_i(\mathbf{z}) \mid l \in \mathcal{N}_i^+ \cup \mathcal{N}_i^-\} > 0.$$

Hence, $a_{ij}(t) \geq \tilde{\gamma}$ whenever $a_{ij}(t) > 0$. The graph $\mathcal{G}^A(t)$ generated by $A(t)$ is connected, and the weight of every directed edge is at least $\tilde{\gamma}$. Thus, consensus is reached as $t \to \infty$; see, e.g., [30]. Here, we can show convergence and a convergence rate (sharper than the bound in [30]) as follows.

Define $\Phi_{t,d} := A(t+d-1)A(t+d-2)...A(t)$ for any $t \geq 0$, where d is the diameter of $\mathcal{G}^A(t)$. Then $\Phi_{t,d}$ is a stochastic matrix (i.e., $\Phi_{t,d}\mathbf{1} = \mathbf{1}$), of which each element (ij) represents the total weight of all paths of length d from node i to node j, and thus is greater than or equal $\tilde{\gamma}^d$. Now let $\underline{\varepsilon}(t) := \min_{i\in\mathcal{V}} \varepsilon_i(t)$ and $\bar{\varepsilon}(t) := \max_{i\in\mathcal{V}} \varepsilon_i(t)$. Clearly, $\{\underline{\varepsilon}(t)\}$ is nondecreasing and $\{\bar{\varepsilon}(t)\}$ is nonincreasing; see (23). Denote by $\underline{k}$ the index satisfying $\varepsilon_{\underline{k}}(t) = \underline{\varepsilon}(t)$. By (22),

$$\begin{aligned}
\varepsilon_i(t+d) &= \textstyle\sum_{j\in\mathcal{V}\setminus\{\underline{k}\}} [\Phi_{t,d}]_{ij}\varepsilon_j(t) + [\Phi_{t,d}]_{i\underline{k}}\varepsilon_{\underline{k}}(t) \\
&\leq \big(\textstyle\sum_{j\in\mathcal{V}\setminus\{\underline{k}\}} [\Phi_{t,d}]_{ij}\big)\bar{\varepsilon}(t) + [\Phi_{t,d}]_{i\underline{k}}\underline{\varepsilon}(t) \\
&= \big(1 - [\Phi_{t,d}]_{i\underline{k}}\big)\bar{\varepsilon}(t) + [\Phi_{t,d}]_{i\underline{k}}\underline{\varepsilon}(t) \\
&\leq (1-\tilde{\gamma}^d)\bar{\varepsilon}(t) + \tilde{\gamma}^d\underline{\varepsilon}(t).
\end{aligned}$$

Similarly, we have $\varepsilon_i(t+d) \geq \tilde{\gamma}^d\bar{\varepsilon}(t) + (1-\tilde{\gamma}^d)\underline{\varepsilon}(t)$. Thus, $\bar{\varepsilon}(t+d) - \underline{\varepsilon}(t+d) \leq (1-2\tilde{\gamma}^d)\big(\bar{\varepsilon}(t) - \underline{\varepsilon}(t)\big)$. Since $\bar{\varepsilon}(t) - \underline{\varepsilon}(t)$ is positive and nonincreasing (see (23)), we conclude that $\bar{\varepsilon}(t) - \underline{\varepsilon}(t)$ decays to 0 geometrically at a rate bounded above by $(1-2\tilde{\gamma}^d)^{\frac{1}{d}}$. Thus $\exists c \in \mathbb{R}$ such that $\lim_{t\to\infty} \bar{\varepsilon}(t) = \lim_{t\to\infty} \underline{\varepsilon}(t) = \lim_{t\to\infty} \varepsilon_i(t) = c$, i.e., $\lim_{t\to\infty} \mathbf{y}(t) = \mathbf{y}^\dagger + c\mathbf{1} := \mathbf{y}^*$. Clearly, $\mathbf{y}^* \in Y^*$. Moreover, it can be seen that the convergence is also geometric with rate $(1-2\tilde{\gamma}^d)^{\frac{1}{d}}$.

Finally, since $\mathbf{y}^* \in \bar{Y}$ and the above argument holds for any $\mathbf{y}^\dagger \in Y^*$, the convergence rate can be refined by selecting $\mathbf{y}^\dagger \in \bar{Y}$. Thus, $\Omega = \bar{Y}$ and $\tilde{\gamma} = \gamma$, as in the theorem statement.

REFERENCES

[1] M. E. Newman, "Spread of epidemic disease on networks," *Phys. Rev. E*, vol. 66, no. 1, p. 016128, 2002.

[2] M. Boguná and R. Pastor-Satorras, "Epidemic spreading in correlated complex networks," *Phys. Rev. E*, vol. 66, no. 4, p. 047104, 2002.

[3] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in *Proc. 22nd Int. Symp. Reliable Dist. Syst.*, 2003, pp. 25–34.

[4] M. Garetto, W. Gong, and D. Towsley, "Modeling malware spreading dynamics," in *Proc. IEEE 22th Annu. Conf. Comp. Comm. Societies*, vol. 3, 2003, pp. 1869–1879.

[5] A. Ganesh, L. Massoulié, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. IEEE 24th Annu. Conf. Comp. Comm. Societies*, vol. 2, 2005, pp. 1455–1466.

[6] P. Van Mieghem, J. Omic, and R. Kooij, "Virus spread in networks," *IEEE/ACM Trans. Net.*, vol. 17, no. 1, pp. 1–14, 2009.

[7] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Rev. Mod. Phys.*, vol. 87, no. 3, p. 925, 2015.

[8] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks," *IEEE Contr. Syst.*, vol. 36, no. 1, pp. 26–46, 2016.

[9] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos, "Epidemic thresholds in real networks," *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 4, p. 1, 2008.

[10] A. Khanafer, T. Başar, and B. Gharesifard, "Stability of epidemic models over directed graphs: A positive systems approach," *Automatica*, vol. 74, pp. 126–134, 2016.

[11] R. Cohen, S. Havlin, and D. Ben-Avraham, "Efficient immunization strategies for computer networks and populations," *Phys. Rev. Lett.*, vol. 91, no. 24, p. 247901, 2003.

[12] Y. Wan, S. Roy, and A. Saberi, "Designing spatially heterogeneous strategies for control of virus spread," *IET Syst. Bio.*, vol. 2, no. 4, pp. 184–201, 2008.

[13] C. Borgs, J. Chayes, A. Ganesh, and A. Saberi, "How to distribute antidote to control epidemics," *Random Struct. Algor.*, vol. 37, no. 2, pp. 204–222, 2010.

[14] E. Gourdin, J. Omic, and P. Van Mieghem, "Optimization of network protection against virus spread," in *8th Int. Workshop Design Reliable Comm. Netw.*, 2011, pp. 86–93.

[15] V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. Pappas, "Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks," in *IEEE 52nd Conf. Decision Contr.*, 2013, pp. 7486–7491.

[16] ——, "Optimal resource allocation for network protection against spreading processes," *IEEE Trans. Contr. Netw. Syst.*, vol. 1, no. 1, pp. 99–108, 2014.

[17] K. Drakopoulos, A. Ozdaglar, and J. N. Tsitsiklis, "An efficient curing policy for epidemics on graphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 2, pp. 67–75, 2014.

[18] J. Liu, P. E. Paré, A. Nedić, C. Y. Tang, C. L. Beck, and T. Başar, "On the analysis of a continuous-time bi-virus model," *arXiv preprint arXiv:1603.04098*, 2016.

[19] S. Ottaviano, F. De Pellegrini, S. Bonaccorsi, and P. Van Mieghem, "Optimal curing policy for epidemic spreading over a community network with heterogeneous population," *J. Complex Netw.*, p. cnx060, 2017.

[20] P. E. Paré, J. Liu, C. L. Beck, A. Nedić, and T. Başar, "Multi-competitive viruses over static and time-varying networks," in *American Contr. Conf.* IEEE, 2017, pp. 1685–1690.

[21] C. Enyioha, A. Jadbabaie, V. Preciado, and G. J. Pappas, "Distributed resource allocation for epidemic control," *arXiv preprint arXiv:1501.01701*, 2015.

[22] E. Ramírez-Llanos and S. Martínez, "Distributed discrete-time optimization algorithms with applications to resource allocation in epidemics control," *Optim. Control Appl. Methods*, pp. 1–21, 2017.

[23] P. Van Mieghem and J. Omic, "In-homogeneous virus spread in networks," *arXiv preprint arXiv:1306.2588*, 2013.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[25] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall Englewood Cliffs, NJ, 1989.

[26] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65–78, 2004.

[27] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proc. 19th Int. Conf. on World Wide Web*, 2010, pp. 641–650.

[28] J. Currie and D. I. Wilson, "Opti: lowering the barrier between open source optimizers and the industrial matlab user," *Foundations of computer-aided process operations*, vol. 24, p. 32, 2012.

[29] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[30] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. 44th IEEE Conf. Decision Contr.*, 2005, pp. 2996–3000.