

# Efficient Algorithms for Dynamic Curing and Network Design in SIS Epidemic Processes

Yuhao Yi , Liren Shan , Shijie Wang , Philip E. Paré , *Senior Member, IEEE*,  
and Karl H. Johansson , *Fellow, IEEE*

**Abstract**—This paper studies efficient algorithms for dynamic curing policies and the corresponding network design problems to guarantee the fast extinction of epidemic spread in a susceptible-infected-susceptible (SIS) model. We consider a Markov process-based SIS epidemic model. We provide a computationally efficient curing algorithm based on the curing policy proposed by Drakopoulos, Ozdaglar, and Tsitsiklis (2014). Since the corresponding optimization problem is NP-hard, finding optimal policies is intractable for large graphs. We provide approximation guarantees on the curing budget of the proposed dynamic curing algorithm. We also present a curing algorithm fair to demographic groups. When the total infection rate is high, the original curing policy includes a waiting period in which no measurements are taken to mitigate the spread until the rate slows down. By utilizing network design strategies that either delete edges or reduce their weights, we limit the total infection rate, allowing the curing process to proceed continuously without requiring a waiting period. We provide algorithms with provable guarantees for the considered network design problems. In summary, the proposed curing and network design algorithms together provide an effective and computationally efficient approach that mitigates SIS epidemic spread in networks.

**Index Terms**—Epidemic, processes, cutwidth, network.

Received 28 November 2024; revised 24 June 2025; accepted 30 July 2025. Date of publication 7 August 2025; date of current version 2 December 2025. The work of Yuhao Yi was supported in part by the National Natural Science Foundation of China under Grant 62303338 and in part by the Fundamental Research Funds for the Central Universities, through Sichuan University under Grant YJ202285. The work of Liren Shan was supported by the National Science Foundation under Grant CCF-1955351 and Grant CCF-1934931. The work of Philip E. Paré was supported by the National Science Foundation under Grant NFS-ECCS-2238388. The work of Karl H. Johansson was supported in part by Knut and Alice Wallenberg Foundation and in part by Swedish Research Council. This work was supported by the Natural Science Foundation of Sichuan Province under Grant t2024NSFSC1939. Recommended for acceptance by Prof. Daniel Sadoc Menasche. (*Corresponding author: Yuhao Yi.*)

Yuhao Yi is with the Institute of Clinical Pathology, West China Hospital, Sichuan University, Chengdu 610065, China, and also with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: yuhaoyi@scu.edu.cn).

Liren Shan is with the Toyota Technological Institute at Chicago, Chicago, IL 60637 USA (e-mail: lirenschan@ttic.edu).

Shijie Wang was with Sichuan University, Chengdu 610065, China. He is now with the Chinese University of Hong Kong, Hong Kong SAR, China (e-mail: michael1003@foxmail.com).

Philip E. Paré is with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-2035 USA (e-mail: philpare@purdue.edu).

Karl H. Johansson is with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden, and also with Digital Futures, SE-100 44 Stockholm, Sweden (e-mail: kallej@kth.se).

Digital Object Identifier 10.1109/TNSE.2025.3596585

## I. INTRODUCTION

THE modeling and control of contagious processes have received significant research interest due to the overwhelming societal cost of widespread epidemics. Extensive mathematical models have been proposed to characterize the behavior of human pathogens, computer malware, or misinformation. The susceptible-infected-susceptible (SIS) model [1] and the susceptible-infected-recovered (SIR) model [2] are the simplest and most popular models, for which the analysis and control of epidemic spread are investigated accordingly. In an SIS model, individuals can be infected multiple times. Network-based compartmental models have been studied for SIS processes, including stochastic models [3], [4] and their mean-field approximations [5], [6].

In a wave of an epidemic, the demand for medical services could exceed local resources including staff, supplies, equipment, hospital space, vaccines, and antivirals. Under such circumstances, policymakers sometimes need to coordinate regional resource allocation and usage to improve medical services [7], [8]. Therefore, optimal resource allocation problems have been studied for various epidemic models. In practice, medical resource allocation strategies are complemented by non-pharmaceutical strategies [9] to reduce the number of infections and therefore the demand for medical resources. For example, travel restrictions, mask wearing, mass screening, school closure, and stay-at-home policies. Simulation studies have shown the advantages of combined medical and non-pharmaceutical countermeasures in effectively mitigating epidemic spread [9]. However, the problem of allocating curing resources and designing the contact network optimally in an SIS model remains to be investigated from a theoretical viewpoint to attain a better understanding of their interaction.

In this paper, we study a Markov process-based SIS model [4] on a weighted, undirected graph. Each edge represents the contact between two individuals, and the weight of the edges is proportional to its infection rate. We further investigate algorithms for intervention that are both effective and computationally efficient to guarantee the fast extinction of SIS processes. We provide algorithms for various interventions, including curing policies and contact restrictions. For curing policies, the resource constraint is the curing budget. To achieve fast extinction, an ordering of curing must be computed such that the total infection rate is maintained relatively small [10]. Thus, we aim to calculate a near-optimal ordering with an efficient

algorithm. We also consider contact restrictions to mitigate the spread concurrently with curing policies. The contact restriction problems can be formulated as network design problems. The restriction of contact between two individuals is modeled as the reduction of the edge weight. The cost of contact restriction is the sum of edge weight reductions. For network design problems with the cost constraint, we present approximation algorithms to the optimal solutions. In the end, the algorithms for curing policies and network designs can be combined to guarantee the fast extinction of the epidemic spread, which shows the superiority of the combined medical and non-pharmaceutical countermeasures.

Our paper is developed based on the dynamic curing policy in [10]. The CURE policy provides insights into curing rate allocation. However, it can be improved in various aspects. First of all, as mentioned by [10], computing the impedance is NP-hard. Therefore efficient approximation algorithms are needed. Secondly, fairness of curing policies is considered a major factor in public health decision-making, which may be compromised by an algorithm with minimum resource consumption.

Apart from effectiveness and efficiency, fairness has become a concern in algorithmic decision-making. In the real world, fairness is also an important aspect of healthcare policy making. For example, the fair distribution of scarce medical resources was identified as a key concern by the world health organization (WHO) during the COVID-19 pandemic [11]. Recently, Dinitz et al. [12] studied fair graph cuts for disaster containment. The disaster model considered in [12] is equivalent to a susceptible-infected (SI) epidemic model. In this paper, we study fair curing algorithms in a stochastic SIS model.

Moreover, the impedance of an infected set can be greater than the curing capability. When the impedance is large, the policy sets a waiting period. We address the first two issues by designing computationally efficient curing algorithms under resource and fairness constraints, and the last issue by allowing a controller to modify network connections.

### A. Contributions

The main contribution of this paper is providing efficient algorithms for curing policies and network design. We extend the  $O(\log^2 n)$  approximation algorithm for cutwidth in [13] to the calculation of impedance, which was mentioned in [10] as an open problem. We further propose a variant of the problem that takes into consideration demographic fairness. We introduce the fair impedance and provide  $O(\log^2 k)$  approximations for the fair impedance. The algorithms are constructed by designing a dynamic programming solution based on a tree embedding [14].

In addition, we study the problem of dynamically modifying the graph structure to ensure that the given curing budget is adequate to achieve fast extinction and avoid the waiting period. We consider two settings: the SIS model with and without targeted curing rate allocation. For the first case, we propose algorithms to minimize the cut between infected nodes and susceptible nodes for a fixed order of nodes. For the latter case, we propose a 1.44-approximation algorithm for the problem of minimizing the maximum cut. By combining the proposed

curing and network design algorithms, we provide a comprehensive solution to control SIS processes in networks.

### B. Related Work

Both curing and network design algorithms for epidemic interventions have been proposed for various models. While the problems have been studied from various perspectives, most of the proposed policies are static, which means that the policy is determined given the initial configuration.

Preciado et al. [15] studied a static resource allocation problem to minimize the cost to bound the spectral radius of a matrix in a mean-field approximation of an SIS model. The cost was defined as a convex function of the infection rate reductions. The authors formulated the problem as a semi-definite program that can be solved in polynomial time. Recently, [16] presented an algorithm with lower time and space complexity than the SDP-based algorithm in [15]. The authors of [17] proposed an optimal strategy to stabilize SIS processes by allocating curing resources to a single node. Spread minimization problems are also studied in SIS models without using mean-field approximation. Borgs et al. [18] proposed to allocate curing rates according to the degree of each node. Using this policy, logarithmic extinction time is achieved by allocating total curing rate proportional to the number of edges in the network. The authors of [19] improved the algorithm by using the personalized PageRank vectors, which exploit the local structure of networks. Zhu, Lam, and Ogura [20] solve an optimal control problem of finding piecewise-constant matrix functions to minimize infections in a simplified SIS model.

The approaches mentioned above use a static policy. In contrast, a dynamic curing policy can adapt to the observed history of epidemic spread. Bang-bang control and feedback control policies were studied in [21]. Ogura and Preciado [22] considered an adaptive network, but the resource allocation problem and optimal network design under budget constraint are not investigated. A feedback control policy [23] using information from direct neighbors was proposed to suppress the SIS process. However, efficient algorithms to calculate optimal curing resource allocation in stochastic SIS processes need to be further studied. In this paper, we consider the same setting as in [10], where the curing budget is limited at any given moment, and a controller optimally distributes resources among all nodes. Drakopoulos et al. proposed a dynamic curing policy for the stochastic SIS model with budget constraint, namely the CURE policy [10]. The approach seeks to cure individuals in an optimal order. Following this order, the total infection rate is always bounded by the *impedance* of the initial infected set in the network. The impedance is the largest infection rate among any remaining subset of the initial infected set following the optimal order. Thus, the curing budget required by the policy depends on the *cutwidth* of the network, which is the impedance of the whole network.

For the network design, algorithms have been proposed for various epidemic models. For SI models, a  $O(\log^2 n)$ -approximation algorithm was proposed by [24] to find the minimum weight of edges such that a given number of vertices

are separated from the initially infected nodes by removing these edges. Constant factor bi-criteria algorithms for this problem were later given by [25], [26]. Two modified problems considering demographic and individual fairness were investigated recently in [12]. For the independent cascade susceptible-infected-removed (IC-SIR) model, recent studies by [27], [28] gave approximations to minimize the expected number of infections by deleting edges, under various assumptions about the underlying graph. For the SIS model, [29] provided approximation algorithms for spectral radius reduction problems by edge or node deletion.

### C. Outline

The remainder of the paper is organized as follows. In Section II we introduce notations and the considered epidemic model. In Section III we propose the considered policies and corresponding optimization problems. In Section IV we propose algorithms for calculating crusades for the CURE policy and its provably fair counterpart. Section V presents algorithms for the edge weight reduction problems with and without targeted curing policies. Section VI shows the effectiveness of the proposed algorithms via numerical simulations using real network data. Section VII summarizes the presented results, and discusses open problems.

## II. PRELIMINARIES

In this section we first introduce some frequently used notations and definitions. Then we describe the probabilistic networked SIS model.

### A. Concepts and Notations

We consider an SIS epidemic process in an undirected weighted graph  $G = (V, E, w)$ , in which  $V$  is the set of nodes and  $E$  is the set of edges with a weight function  $w : E \mapsto [0, 1]$ . The degree  $d_u$  of a node  $u$  is defined as the sum of weights of all its incident edges. We denote by  $d_{\max}$  the maximum degree of nodes in the graph. We adopt some of the terms used in [10], [30]. We refer to a subset of  $V$  as a *bag*. For any bag  $A$  of  $V$ , we use  $A^c \stackrel{\text{def}}{=} V \setminus A$  to denote the complement of  $A$ . For any bag  $A$  and any node  $u \in A^c$  and  $v \in A$ , let  $A + u \stackrel{\text{def}}{=} A \cup \{u\}$  and  $A - v \stackrel{\text{def}}{=} A \setminus \{v\}$ . We then define the cut of the graph  $G$  with respect to a bag. We denote by  $G[A]$  the subgraph of  $G$  supported on the bag  $A$ , i.e.,  $G[A] = (A, E', w')$  where  $E' = \{(u, v) \mid u, v \in A\} \subset E$  and  $w'(e) = w(e)$  for  $e \in E'$ .  $\mathbf{1}\{\cdot\}$  is defined as the indicator function of a given statement, taking the value of 1 if the statement is true and 0 otherwise.

**Definition 1:** A cut of the graph  $G$  is defined for a bag  $A$  as the vertex partition  $(A, A^c)$ . The size of the cut is defined as  $c(A) \stackrel{\text{def}}{=} \sum_{(u,v) \in E^c} w_{uv}$ , where  $E^c$  is the set of edges  $(u, v) \in E$ ,  $u \in A$ ,  $v \in A^c$ , and  $w_{uv}$  is the weight of the edge  $(u, v)$ .

We also use the standard definition for the balanced cut in [31], where a cut is  $\alpha$ -balanced if  $\min\{|A|, |A^c|\} \geq \alpha|V|$ . We introduce the following definition of maximum restricted cut.

**Definition 2:** Given a graph  $G$  and a bag  $A$ , the maximum restricted cut of the bag  $A$  is defined by

$$\phi(A) \stackrel{\text{def}}{=} \max_{Q \subseteq A} c(Q). \quad (1)$$

We further recall some concepts defined in [10], [30].

**Definition 3:** For any two bags  $A$  and  $B$  satisfying  $B \subseteq A$ , a monotone crusade from  $A$  to  $B$  is a sequence of bags  $p(A, B) \stackrel{\text{def}}{=} (p_0, p_1, \dots, p_k)$  where  $p_0 = A$ ,  $p_k = B$ ;  $p_i \subseteq p_{i-1}$  and  $|p_{i-1} \setminus p_i| = 1$  for any  $i \in [k]$ . We denote by  $\mathcal{C}(A, B)$  the set of all crusades from  $A$  to  $B$ .

**Definition 4:** The *width* of a crusade  $p = (p_0, p_1, \dots, p_k)$  is defined by  $z(p) \stackrel{\text{def}}{=} \max_{0 \leq i \leq k} \{c(p_i)\}$ .

**Definition 5:** Given a bag  $A$ , its *impedance*  $\delta(A)$  is defined as  $\delta(A) \stackrel{\text{def}}{=} \min_{p \in \mathcal{C}(A, \emptyset)} z(p)$ , namely the minimum width of a crusade from  $A$  to  $\emptyset$ .

The impedance  $\delta(V)$  is referred to as the *cutwidth* of a graph, denoted by  $\mathcal{W}$ . The problem of calculating the cutwidth is also referred to as the minimum cut linear arrangement problem [13]. Sometimes we use a superscript to indicate the associated network of a quantity, for example,  $z_G(p)$ ,  $\delta_G(A)$ , and  $\phi_G(A)$ .

To discuss the fairness of curing policies, we assign vertices in the graph to demographic groups. Suppose we are given sets  $V_1, V_2, \dots, V_\ell$ , which composite the vertex set  $V$ . We assume that the number of groups  $\ell$  is a constant. Let  $p = (p_0, p_1, \dots, p_k)$  be a crusade from  $A$  to  $B$ . We consider a set  $\mathcal{T}$  of checkpoints  $0 < \tau_1 < \tau_2 < \dots < \tau_s < k$ , on which we want to check the fairness of curing policies. Let  $\tau_0 = 0$  be a dummy checkpoint. For every  $i \in \{0, 1, 2, \dots, s-1\}$ , we use  $S_i = p_{\tau_i} \setminus p_{\tau_{i+1}}$  to denote the set of nodes between checkpoints  $\tau_i$  and  $\tau_{i+1}$ . For any  $\gamma \geq 1$ , the crusade  $p$  is a  $\gamma$ -fair crusade with respect to the set of checkpoints  $\mathcal{T}$  if for every  $i \in \{0, 1, 2, \dots, s-1\}$ , the set of nodes  $S_i$  satisfies for every  $h \in [\ell]$ ,

$$|S_i \cap V_h| < \gamma \cdot \frac{|p_0 \cap V_h|}{|p_0|} \cdot |S_i| + 1, \quad (2)$$

which means the curing ratio of a group  $h$  in the set between checkpoints  $\tau_i$  and  $\tau_{i+1}$  is preserved within  $\gamma$  times its initial ratio. We also refer to the sets  $S_0, S_1, \dots, S_s$  as a  $\gamma$ -fair partitioning of the bag  $A \setminus B$ , where  $S_i$  satisfies (2) for  $i \in \{0, \dots, s-1\}$ .

We denote by  $\mathcal{C}_{\gamma, \mathcal{T}}(A, B)$  the set of all  $\gamma$ -fair crusades from  $A$  to  $B$  with respect to checkpoints  $\mathcal{T}$ , and we employ this notation to define the notion of  $\gamma$ -fair impedance.

**Definition 6:** Given a bag  $A$  and a set of checkpoints  $\mathcal{T}$ , its  $\gamma$ -fair impedance  $\delta_{\gamma, \mathcal{T}}(A)$  is defined as  $\delta_{\gamma, \mathcal{T}}(A) \stackrel{\text{def}}{=} \min_{p \in \mathcal{C}_{\gamma, \mathcal{T}}(A, \emptyset)} z(p)$ , the minimum width of a  $\gamma$ -fair crusade from  $A$  to  $\emptyset$  w.r.t.  $\mathcal{T}$ .  $\square$

We note that  $\mathcal{C}_{\gamma, \mathcal{T}}(A, \emptyset) \subseteq \mathcal{C}(A, \emptyset)$  always holds, therefore  $\delta_{\gamma, \mathcal{T}}(A) \geq \delta(A)$  for all  $\mathcal{T}$ .

### B. Epidemic Model

We consider a networked SIS model in which each node can be in one of two states: susceptible or infected. The infection spreads according to a continuous-time Markov process  $\{I(t)\}_{t \geq 0}$  on the state space  $\{0, 1\}^V$  where  $I_u(t) = 0$ , if node  $u$  is susceptible, and  $I_u(t) = 1$ , if it is infected. Without ambiguity, we also use  $I(t)$  to denote the set of all infected nodes at time  $t$ .

The process starts with a given initial state  $I(0)$ . Let  $w_{uv}$  be the infection rate for each edge  $(u, v) \in E$ . At any time  $t \geq 0$ , the transition probability of a susceptible node  $v$  to the infected state is defined by  $\sum_{(u,v) \in E} w_{uv} I_u(t)$ , which is the sum of infection rates  $w_{uv}$  over all infected neighbors of  $v$ .

On the other hand, the curing rate of an infected node  $u$  is denoted by  $\rho_u(t) \geq 0$ . We consider two typical scenarios: (a)  $\rho_u(t)$  is decided by a network controller with the budget constraint  $\sum_{u \in V} \rho_u(t) \leq r$  where  $r$  is the curing budget; (b) the allocation of  $\rho_u(t)$  for each  $u$  is decided by the environment and can be adversarial. We define the total curing capacity to be  $r(t) \stackrel{\text{def}}{=} \sum_{u \in V} \rho_u(t)$ .

### III. PROBLEM FORMULATIONS

In this section we present formal definitions for the optimization problems that arise in determining the curing order and reducing the infection rate.

#### A. Curing Policies

The general strategy of the considered curing policies is to assure that the number of infected nodes has a downward drift during the process to guarantee short extinction time. Since curing infected nodes may increase the cut between the infected and susceptible nodes, the ordering of the nodes being cured has a significant impact on the process. The CURE policy proposed in [10] seeks to find paths along which the cuts are maintained as small as possible such that the spread stops in sublinear time with high probability. We briefly recall the CURE policy as follows:

- Wait until  $c(I(t)) \leq r/8$ . Let  $A$  be the set of infected nodes right after the waiting period.
- Start a segment. Calculate the optimal crusade and obtain an ordering  $\{v_1, \dots, v_{|A|}\}$  in the beginning of a segment. Let  $C \stackrel{\text{def}}{=} \{v_2, \dots, v_{|A|}\}$ . At any  $t$  before the segment ends, allocate the entire curing budget to cure an arbitrary node in the set  $D(t) \stackrel{\text{def}}{=} I(t) \setminus C$ . A segment ends when  $I(t) = C$  or  $|D(t)| \geq r/(8d_{\max})$ . If  $I(t) = C$ , a new segment will be started. If  $|D(t)| \geq r/(8d_{\max})$ , a waiting period will be started.

The paper [10] does not specify an algorithm to calculate the optimal crusade. A dynamic programming algorithm with exponential time and space complexity is obtained by directly applying the Bellman equation given in the paper. In [10], improving the computational complexity of the algorithm was listed as an open problem. In this paper we study the following optimization problem.

*Problem 1:* Given a network  $G$  and a bag  $A$ , find a crusade  $p$  from  $A$  to  $\emptyset$  such that

$$\underset{p \in C(A, \emptyset)}{\text{minimize}} \quad z(p). \quad (3)$$

It has been shown that the problem of calculating the cutwidth (and hence the impedance) of a graph is **NP-hard** [32]. Therefore we resort to approximation algorithms which compute a crusade whose width is bounded within a certain factor compared to the minimum width.

By integrating demographic information, a  $\gamma$ -fair curing policy only considers  $\gamma$ -fair crusades as feasible target paths of curing. Apart from this distinction, the policy follows a similar procedure to that of the CURE policy. Thus we also study approximation algorithms for the following problem.

*Problem 2:* Given a network  $G$ , a bag  $A$ , a number  $\gamma \geq 1$ , and a set of checkpoints  $\mathcal{T}$ , find a crusade  $p$  from  $A$  to  $\emptyset$  such that

$$\underset{p \in C_{\gamma, \mathcal{T}}(A, \emptyset)}{\text{minimize}} \quad z(p). \quad (4)$$

#### B. Network Design

A drawback of the CURE policy is that when  $c(I(t)) > r/8$ , the policy starts a waiting period. This is clearly undesired since the epidemic spreads to more nodes while no measures are taken. For similar reasons, the CURE policy only works when the budget is  $\Omega(\mathcal{W})$ , where  $\mathcal{W}$  is the cutwidth of the network. Moreover, the authors have shown in subsequent work [30], [33] that there exists a phase transition such that when  $r = o(\mathcal{W})$  the extinction time is exponential regardless of the curing policy.

For public health decision-making, it is advisable to consider both medical and non-pharmaceutical intervention strategies. Real-world epidemic response is often a composition of medical resource allocation and contact restrictions. Previous work has shown that, by applying, along with preferentially vaccinating urban locations, travel restrictions can postpone the arrival of the peak in the spread of influenza [34]. School closures and social distancing are combined with antiviral treatment or vaccination to reduce an influenza outbreak [35]. A simulation of smallpox shows the effectiveness of combining vaccination with school closures [36]. Motivated by these simulation studies, we investigate the problem of reducing network connections to achieve fast extinction with insufficient medical resources.

Given a graph  $G$  and a budget  $r$ , if  $r$  is insufficient for any policy to suppress the spread, we ask the question of how to modify the network to effectively reduce its impedance and cutwidth. Since modifying the network results in the change of optimal curing order, and changing the curing order affects the optimal weight reduction of the network, optimizing the two policies simultaneously is challenging. We consider a simplified problem with any fixed curing order.

Given the CURE policy, we study an alternative problem, in which we modify the graph to effectively reduce the width of a *fixed* crusade computed in the beginning of each curing segment. Specifically, we modify the graph by reducing the weight of each edge  $(u, v) \in E$  to  $w'_{uv} = w_{uv} - \Delta_{uv}$ . Our goal is to minimize the total weight reduction such that the width of the current crusade is no more than a threshold  $b$ .

*Problem 3:* Given a graph  $G$  with the edge weight function  $w$ , a bag  $A$  and a crusade  $p$  from  $A$  to  $\emptyset$ , a threshold  $b \in \mathbb{R}_+$ , find the weight reduction  $\Delta_{uv}$  of each edge  $(u, v) \in E$  for the following optimization program:

$$\underset{\Delta}{\text{minimize}} \quad \sum_{(u,v) \in E} \Delta_{uv},$$

$$\begin{aligned} \text{subject to } & 0 \leq \Delta_{uv} \leq w_{uv}, \forall (u, v) \in E, \\ & z_{G'}(p) \leq b, \end{aligned} \quad (5)$$

where  $G' = (V, E, w')$  is the modified graph with weight  $w'_{uv} = w_{uv} - \Delta_{uv}$  for  $(u, v) \in E$ .

We also consider this problem under the constraint that weight reductions  $\Delta_{uv} \in \{0, w_{uv}\}$ , for all edges, which means each edge is either kept or removed completely.

*Problem 4:* Given a graph  $G$  with the edge weight function  $w$ , a bag  $A$ , a crusade  $p$  from  $A$  to  $\emptyset$ , and a threshold  $b \in \mathbb{R}_+$ , find the weight reduction  $\Delta_{uv}$  of each edge  $(u, v) \in E$  for the following optimization program:

$$\begin{aligned} \text{minimize } & \sum_{(u,v) \in E} \Delta_{uv}, \\ \text{subject to } & \Delta_{uv} \in \{0, w_{uv}\}, \forall (u, v) \in E, \\ & z_{G'}(p) \leq b, \end{aligned} \quad (6)$$

where  $G' = (V, E, w')$  is the modified graph with weight  $w'_{uv} = w_{uv} - \Delta_{uv}$  for  $(u, v) \in E$ .

When the curing policy is decided by the environment, the ordering of curing is arbitrary. Therefore, a variant of the maximum cut of the network decides the extinction time of the SIS process. We also investigate the problem of minimizing the maximum restricted cut of a given bag.

*Problem 5:* Given a graph  $G$  with the edge weight function  $w$ , a bag  $A$ , and a threshold  $b \in \mathbb{R}_+$ , find the weight reduction  $\Delta_{uv}$  of each edge  $(u, v) \in E$  for the following optimization program:

$$\begin{aligned} \text{minimize } & \sum_{(u,v) \in E} \Delta_{uv}, \\ \text{subject to } & 0 \leq \Delta_{uv} \leq w_{uv}, \forall (u, v) \in E, \\ & \phi_{G'}(A) \leq b. \end{aligned} \quad (7)$$

where  $G' = (V, E, w')$  is the modified graph with weight  $w'_{uv} = w_{uv} - \Delta_{uv}$  for  $(u, v) \in E$ .  $\square$

#### IV. CURING ALGORITHMS

In this section we provide algorithms for approximating impedance and  $\gamma$ -fair impedance of a bag. The definitions of these problems are given in Section III-A.

##### A. Algorithm for CURE Policy

In this section, we provide a polynomial-time curing algorithm in the network SIS model. Specifically, we combine a polynomial-time approximation algorithm for computing the crusade with the CURE policy proposed in [10]. We present the following theorem.

*Theorem 1:* Given a graph  $G$ , suppose the curing budget  $r \geq \max\{\alpha \mathcal{W} \log^2 n, 8d_{max} \log n\}$ , where  $\mathcal{W}$  is the cutwidth of graph  $G$  and  $\alpha$  is a fixed constant. Then, there exists a polynomial-time curing algorithm such that the expected extinction time is at most  $O(n \log^2 n/r)$ .

To prove this theorem, we use a polynomial-time algorithm that, given any bag  $A$  of size  $k$ , finds a crusade of  $A$  with

---

##### Algorithm 1: ApprImpe( $G, A$ ).

---

**Input** : a graph  $G = (V, E, w)$ , a bag  $A$  with  $|A| = k$ ;  
**Output**: a crusade  $p \in \mathcal{C}(A, \emptyset)$ ;  
**if**  $|A| = 1$  **then**  
  | **return**  $p \leftarrow (\{u\}, \emptyset)$ ;  
**else**  
   $(V_1, V_2) \leftarrow \text{BalancedCut}(G[A], A)$ ;  
   $(q_0, \dots, q_{|V_1|}) \leftarrow \text{ApprImpe}(G[V_1], V_1)$ ;  
   $(p_{|V_1|}, \dots, p_{|A|}) \leftarrow \text{ApprImpe}(G[V_2], V_2)$ ;  
  **for**  $i \leftarrow 0$  **to**  $|V_1| - 1$  **do**  
    |  $p_i \leftarrow q_i \cup V_2$ ;  
  **end**  
  **return**  $p \leftarrow (p_0, \dots, p_{|A|})$ ;  
**end**

---

width at most  $O(\log^2 k)$  times the impedance of  $A$ . This algorithm is shown in Algorithm 1. Our algorithm follows the approaches of approximation algorithms for multiway cut [37] and cutwidth [13]. The crux of the algorithm is an approximation algorithm for the balanced cut problem [13]. The algorithm recursively calculates a balanced cut on the subgraph supported on a bag. In Algorithm 1, the subroutine `BalancedCut` returns a partition  $(V_1, V_2)$  of the bag  $A$ , i.e.,  $V_1 \cup V_2 = A$  and  $V_1 \cap V_2 = \emptyset$ . Lemma 1 guarantees that this partition  $(V_1, V_2)$  satisfies two key properties: both sets are of size at least a constant fraction of  $|A|$ , and the number of edges between them is small, which is at most  $O(\log k)$  times the size of the minimum balanced cut, where  $k = |A|$ .

The algorithm then recursively computes crusades for the subgraphs  $G[V_1]$  and  $G[V_2]$  until the subgraphs cannot be further divided. Without loss of generality, we assume that the crusade first cures all nodes in  $V_1$  before proceeding to  $V_2$ . To ensure consistency with the crusade structure, every node in  $V_2$  is added to all preceding bags (those used in the  $V_1$  sub-crusade). This process continues until the subgraphs can no longer be partitioned.

Since each recursive step reduces the bag size by at least a constant factor, the depth of the recursion is at most  $O(\log k)$ . Moreover, each bag in the resulting crusade includes cuts across at most  $O(\log k)$  recursive levels.

*Lemma 1 (Section 3.3 in [13]):* Given a graph  $G$ , there exists a polynomial-time algorithm that finds a  $1/3$ -balanced cut with size at most  $O(\log k) \text{OPT}_G(1/2)$ , where  $\text{OPT}_G(1/2)$  is the size of the minimum balanced cut.

*Theorem 2:* Algorithm 1 takes the input of a graph  $G$  and an infected bag  $A$  with  $|A| = k$  and outputs a crusade  $p \in \mathcal{C}(A, \emptyset)$  with width  $z(p)$ , such that

$$\delta(A) \leq z(p) \leq O(\log^2 k) \delta(A). \quad (8)$$

*Proof:* By the definition of impedance, we have the width  $z(p) \geq \delta(A)$ . For the other side, we only need to upper bound the size of the cut given by the bag  $p_i$  for every  $i \in \{0, 1, 2, \dots, k\}$ . Since the bag  $p_0 = A$ , we have  $c(p_0) = c(A) \leq \delta(A)$ . We use  $v_i$  to denote the node in  $p_{i-1} \setminus p_i$  for each  $i \in \{1, 2, \dots, k\}$ . Each edge in the cut given by  $p_i$  is contained in one of the following

two cases: (a) the cut  $(A, A^c)$ ; (b) a balanced cut generated by our algorithm over a bag containing the node  $v_i$ . Consider any balanced cut over a bag  $B$  containing  $v_i$  in our algorithm. Let  $(B_1, B_2)$  be the  $1/3$ -balanced cut over the bag  $B$  returned by the call of  $\text{BalancedCut}(G[B], B)$ . By Lemma 1, we have

$$c_{G[B]}(B_1) \leq O(\log k) \text{OPT}_{G[B]}(1/2).$$

Suppose  $q = (q_0, q_1, \dots, q_{|A|})$  is an optimal crusade from the bag  $A$  to  $\emptyset$  with width  $z(q) = \delta(A)$ . We find the bag  $q_j$  in this crusade such that  $q_j \cap B$  provides a  $1/2$ -balanced cut over the set  $B$ . Thus, we have

$$\text{OPT}_{G[B]}(1/2) \leq c_{G[B]}(q_j \cap B) \leq c_G(q_j) \leq \delta(A).$$

Therefore, every balanced cut in our algorithm has the size at most  $O(\log k)\delta(A)$ . Then, we show that the total number of recursive balanced cuts containing the node  $v_i$  is at most  $O(\log k)$ . Our recursive algorithm starts from the bag  $A$  with  $|A| = k$ . Since the subroutine  $\text{BalancedCut}$  returns a  $1/3$ -balanced cut, the size of the bag containing the node  $v_i$  shrinks by a factor of  $2/3$  after each recursive call of  $\text{BalancedCut}$ . Thus, the total number of recursive calls containing the node  $v_i$  is at most  $O(\log k)$ . For every  $i \in \{1, 2, \dots, k\}$ , the size of the cut given by the bag  $p_i$  is at most

$$c(p_i) \leq c(A) + \sum_{B: v_i \in B} c_{G[B]}(B_1) \leq O(\log^2 k)\delta(A),$$

which completes the proof.  $\square$

We note that a  $O(\log^2 n)$  approximation algorithm for the cutwidth of a graph is also shown in [38] by using a  $O(\log n)$  approximation for the minimum linear arrangement (MLA). By using an improved  $O(\sqrt{\log n} \log \log n)$  approximation for MLA in [39], an  $O(\log^{3/2} n \log \log n)$  approximation for the cutwidth can be given, which can also be generalized to a  $O(\log^{3/2} k \log \log k)$  approximation for the impedance problem. To prove Theorem 1, we show that the following lemma holds by choosing the crusade given by Algorithm 1.

*Lemma 2:* For every time  $t$  during a segment, the cut given by the set of infected nodes is at most  $c(I(t)) \leq r/2$ .

*Proof:* For each segment, we want to reach the target bag  $C$  on the current crusade provided by Algorithm 1 with respect to an input bag  $B$ . Let  $D(t) = I(t) \setminus C$ . Recall that the curing policy will start a waiting period if  $|D_t| \geq r/(8d_{max})$ . Then, we have

$$c(I(t)) \leq c(C) + c(D(t)) \leq c(C) + d_{max} \cdot |D(t)| \leq c(C) + r/8,$$

where the last inequality holds since  $|D_t| < r/(8d_{max})$ . According to Lemma 1 in [10], the impedance of the bag  $B$  is upper bounded by  $\mathcal{W} + c(B)$ . By Theorem 2, we have

$$c(C) \leq \alpha/4 \cdot \log_2^2 n \cdot \delta(B) \leq \alpha/4 \cdot \log_2^2 n (\mathcal{W} + c(B)).$$

Since the bag  $B$  satisfies our new ending condition for the waiting period, we have  $c(B) \leq r/(2\alpha \log_2^2 n)$ . By our assumption  $r \geq \alpha \mathcal{W} \log_2^2 n$ , we have  $c(C) \leq 3r/8$ , which implies  $c(I(t)) \leq r/2$ .  $\square$

Then, we prove Theorem 1 by modifying the analysis of the CURE policy in [10].

*Proof of Theorem 1:* We slightly modify the CURE policy as follows. Let the waiting period end when  $c(I(t)) \leq r/(2\alpha \log_2^2 n)$ , where  $\alpha/4$  is the fixed constant in the approximation factor in Theorem 2. In a segment, we choose the crusade given by Algorithm 1 instead of the optimal crusade. We assume that the curing budget  $r \geq \alpha \mathcal{W} \log_2^2 n$  and  $r \geq 8d_{max} \log_2 n$ .

Then, we upper bound the expected extinction time. Since the ending condition for the waiting period is  $c(I(t)) \leq r/(2\alpha \log_2^2 n)$ , the expected length of a waiting period is at most  $(2\alpha n \log_2^2 n)/r$ . By Lemma 2, the expected length of a segment is at most  $2/r$ . The number of segments is at most  $n$ . Similar to the analysis in [10], we get that the expected extinction time is  $O(n \log^2 n/r)$ .  $\square$

## B. Fair Curing Algorithms

Prioritizing utility alone can inadvertently lead to unfair healthcare policies. During the COVID-19 pandemic, American Indians and African Americans experienced significantly higher hospitalization rates at various stages in the US [40], [41]. In the U.K., Black and Asian populations also faced disproportionately higher hospital admission rates [42]. In northern Italy, during a surge of infections, medical equipment was allocated to younger patients, a decision that was arguably unfair to the elderly [43].

The following example illustrates how an optimal solution to Problem 1 can inadvertently be unfair to demographic groups. Consider a network consisting of two  $n$ -node cliques connected by a random bipartite graph, with edges appearing independently with probability  $1/n$ . Suppose all  $2n$  nodes are initially infected. The optimal curing strategy is to cure all nodes in one clique first, followed by curing nodes in the other clique. If these cliques represent distinct demographic communities, such a policy would be deemed unfair under our definition when the only checkpoint is  $n$ . This example highlights how optimizing purely for utility may inadvertently lead to unfair healthcare policies and ultimately result in inequitable outcomes.

Next, we show an approximation algorithm for finding a fair curing policy. Given a graph  $G$ , a bag  $A$ , a fair parameter  $\gamma$ , and a set of checkpoints  $\mathcal{T}$ , our algorithm sequentially finds fair partitions at these checkpoints. Specifically, we start from the first checkpoint  $\tau_1$  and find the  $\gamma$ -fair partition of bag  $A$  at checkpoint  $\tau_1$ . Then, we pick a crusade for nodes before  $\tau_1$  with Algorithm 1. After that, we repeat this process on the remaining nodes after  $\tau_1$  until there is no checkpoint. Our algorithm is shown in Algorithm 2.

We first show that our algorithm achieves a  $O(\log^2 k)$  approximation for  $\gamma$ -fair impedance when there is only one checkpoint.

*Theorem 3:* Given a bag  $A$  with size  $k$ , we have a  $O(\log^2 k)$ -approximation algorithm for  $\gamma$ -fair impedance of a bag  $A$  in a graph  $G$  with respect to groups  $V_1, \dots, V_\ell$  and one checkpoint  $\mathcal{T} = \{\tau_1\}$ .

For multiple checkpoints,  $s > 1$ , we reconstruct the crusade after each checkpoint by using our algorithm. We show that when checkpoints satisfy a doubling condition specified in the following theorem, our algorithm still achieves a bi-criteria approximation to the  $\gamma$ -fair impedance.

**Algorithm 2:** FairApprImpe( $G, A, \gamma, \mathcal{T}$ ).

---

**Input** : a graph  $G = (V, E, w)$ , a bag  $A$  with  $|A| = k$ , a parameter  $\gamma$ , a set of checkpoints  $\mathcal{T}$ ;

**Output:** a fair crusade  $p \in \mathcal{C}(A, \emptyset)$ ;

**if**  $\mathcal{T}$  is not empty **then**

Find the nearest checkpoint  $\tau_i$ ;

Find a  $\gamma$ -fair partitioning of  $A$ ,  $S_{i-1}$  and  $A \setminus S_{i-1}$  of sizes  $\tau_i$  and  $k - \tau_i$  by using dynamic programming on HST;

$(q_0, \dots, q_{|S_{i-1}|}) \leftarrow \text{ApprImpe}(G[S_{i-1}], S_{i-1})$ ;

$(p_{|S_{i-1}|}, \dots, p_{|A|}) \leftarrow$

FairApprImpe( $G[A \setminus S_{i-1}], A \setminus S_{i-1}, \gamma, \mathcal{T} \setminus \{\tau_i\}$ );

**for**  $i \leftarrow 0$  **to**  $|S_{i-1}| - 1$  **do**

$p_i \leftarrow q_i \cup (A \setminus S_{i-1})$ ;

**end**

**return**  $p \leftarrow (p_0, \dots, p_{|A|})$ ;

**else**

$(p_0, \dots, p_{|A|}) \leftarrow \text{ApprImpe}(G[A], A)$ ;

**return**  $p \leftarrow (p_0, \dots, p_{|A|})$ ;

**end**

---

*Theorem 4:* Given a bag  $A$  with size  $k$ , suppose the set of checkpoints  $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_s\}$  satisfy that  $\tau_i - \tau_{i-1} \geq \tau_{i-1}$ , for every  $i = 2, 3, \dots, s$ . We have an algorithm that finds a  $2\gamma$ -fair crusade of bag  $A$  that satisfies: the width of this crusade is at most  $O(\log^2 k)$  times the  $\gamma$ -fair impedance of a bag  $A$ .

Analysis for Theorems 3 and 4 are shown in Appendix A.

## V. NETWORK DESIGN

In this section we present algorithms for the network design problems posed in Section III-B.

### A. Network Design for Curing Policies

In this section, we consider the problem of modifying the network structure to reduce the width of a fixed crusade, which in turn improves the performance of the CURE policy. Specifically, our goal is to reduce edge weights in the graph so that the width of the given crusade falls below a target threshold, while minimizing the total weight reduction. This corresponds to adjusting the graph topology by weakening or removing connections in a way that facilitates more efficient curing. We consider two versions of this problem: a fractional version where edge weights can be partially reduced, and an integral version where edges are either kept fully or removed entirely, corresponding to Problems 3 and 4, respectively.

For a fixed crusade, Problem 3 is a linear program with a cost function that is linear in the edge weights and  $k$  linear constraints for each cut to be no more than the curing budget. Therefore, we obtain the optimal solution in polynomial time.

*Theorem 5:* There exists a polynomial time algorithm which finds the optimal solution for Problem 3.

Next we consider Problem 4, in which  $\Delta_{uv}$  takes the value of either  $w_{uv}$  or 0. We provide an algorithm with the following guarantee:

**Algorithm 3:** WidthOpt( $G, A, p$ ).

---

**Input** : a graph  $G = (V, E, w)$ , a bag  $A$  with  $|A| = k$ , a crusade  $p$  from  $A$  to  $\emptyset$

**Output:**  $\Delta_{uv} \in \{0, w_{uv}\}$  for  $(u, v) \in E$

Let  $\{(u, v), \Delta_{uv}^* : (u, v) \in E\}$  be an optimal solution for the LP in Problem 3;

$v_i \leftarrow p_{i-1} \setminus p_i$  for  $i \in [k]$ ;

Set an arbitrary ordering  $\{v_{k+1}, v_{k+2}, \dots, v_n\}$  for nodes in  $A^c$ ;

**for**  $i \leftarrow 1$  **to**  $k$  **do**

Let  $\vec{E}_i$  be the sorted list of  $\{(v_i, v_j) : j > i\}$  in non-increasing order of  $j$ ;

$x \leftarrow 0$ ;

**while**  $x < \sum_{(u,v) \in \vec{E}_i} \Delta_{uv}^*$  **do**

Let the edge  $(u, v)$  be the first edge in  $\vec{E}_i$ ;

$x \leftarrow x + w_{uv}$ ;

$\Delta_{uv} \leftarrow w_{uv}$  and remove  $(u, v)$  from  $\vec{E}_i$ ;

**end**

$\Delta_{uv} \leftarrow 0$  for the remaining edges  $(u, v)$  in  $\vec{E}_i$ ;

**end**

---

*Theorem 6:* Given a graph  $G$ , a bag  $A$  with  $|A| = k$ , and a crusade  $p$  from  $A$  to  $\emptyset$ , Algorithm 3 provides a solution of Problem 4 with the total weight reduction  $\text{OPT}(G, A, p) + k$ , where  $\text{OPT}(G, A, p)$  is the optimal solution of Problem 4.

*Proof:* Algorithm 3 is an LP rounding algorithm. It begins by computing the optimal fractional solution to Problem 3. Then, it performs a greedy rounding procedure, starting from the earliest node in the fixed crusade. At each step, it removes edges corresponding to cuts with the largest span in the crusade, continuing until the width at the current node falls below the desired threshold. We first show that the total weight reduction returned by Algorithm 3 is at most  $\text{OPT}(G, A, p) + k$ . Let  $\Delta_{uv}^*$  be the optimal solution for the LP used in Algorithm 3. Since the edge weight is at most one, we have for each  $i \in \{1, 2, \dots, k\}$ ,

$$\sum_{(u,v) \in \vec{E}_i} \Delta_{uv} \leq \sum_{(u,v) \in \vec{E}_i} \Delta_{uv}^* + 1.$$

Since Problem 3 is a relaxation of Problem 4, the total weight reduction in the optimal LP solution is at most  $\text{OPT}(G, A, p)$ . Thus, we have

$$\sum_{(u,v) \in E} \Delta_{uv} \leq \sum_{(u,v) \in E} \Delta_{uv}^* + k \leq \text{OPT}(G, A, p) + k.$$

Then, we show that the solution returned by Algorithm 3 is a feasible solution for Problem 4. We say that an edge  $(u, v)$  covers  $v_i$  if exactly one of  $u$  and  $v$  is in  $p_i$ . For each  $i \in \{1, 2, \dots, k\}$ , let  $E_i$  denote all edges covering node  $v_i$ . By the rounding process, we have for any  $i \in \{1, 2, \dots, k\}$ ,

$$\sum_{(u,v) \in E_i} \Delta_{uv} \geq \sum_{(u,v) \in E_i} \Delta_{uv}^*. \quad (9)$$

Note that the edges in  $E_i \cap \vec{E}_j$  are the first  $|E_i \cap \vec{E}_j|$  edges in the sorted list  $\vec{E}_j$  for any  $j \leq i \leq k$ . We claim that for any

$j \leq i \leq k$ ,

$$\sum_{(u,v) \in E_i \cap \vec{E}_j} \Delta_{uv} \geq \sum_{(u,v) \in E_i \cap \vec{E}_j} \Delta_{uv}^*. \quad (10)$$

We prove this claim by considering the following two cases. If all edges  $(u, v)$  in  $E_i \cap \vec{E}_j$  are rounded to  $w_{uv}$ , then the inequality (10) holds. If there is an edge  $(u, v)$  in  $E_i \cap \vec{E}_j$  that is rounded to 0, then all edges in  $\vec{E}_j \setminus E_i$  are rounded to 0 according to the rounding process. So, we get

$$\begin{aligned} \sum_{(u,v) \in E_i \cap \vec{E}_j} \Delta_{uv} &= \sum_{(u,v) \in \vec{E}_j} \Delta_{uv} \geq \\ \sum_{(u,v) \in \vec{E}_j} \Delta_{uv}^* &\geq \sum_{(u,v) \in E_i \cap \vec{E}_j} \Delta_{uv}^*, \end{aligned}$$

where the first inequality is due to the inequality in (9).

Therefore, we have for the edge set  $E_i$  covering node  $i$ ,  $i \in \{1, 2, \dots, k\}$ ,

$$\sum_{(u,v) \in E_i} \Delta_{uv} \geq \sum_{(u,v) \in E_i} \Delta_{uv}^*,$$

which implies that  $\Delta_{uv}$  satisfies the width constraint.  $\square$

*Remark 1:* Let  $\text{LP}^* = \sum_{(u,v) \in E} \Delta_{uv}^*$  where  $\Delta_{uv}^*$  is an optimal solution for Problem 3. We provide an example that shows the existence of an  $O(\text{LP}^* + k)$  additive integrality gap for any algorithm of Problem 4 by rounding the solution of Problem 3. We let  $G$  be a path with all edges weighted one and nodes are sorted from one end to the other. Let  $p$  be the crusade where  $p_i$  consists of the last  $k - i$  nodes. Let  $b = 0.9$ , then the LP returns a solution where  $\Delta_{uv}^* = 0.1$  for all edges. The only feasible integral solution, however, is  $\Delta_{uv} = 1$  for all edges. Therefore, the additive integrality gap is equal to  $9 \cdot \text{LP}^*$  (and also  $0.9k$ ).

We then consider an unweighted version of Problem 4 where  $w_{uv} = 1$  and  $\Delta_{uv} \in \{0, 1\}$  for all  $(u, v) \in E$ , which we refer to as the Unweighted Crusade Width Cost Minimization Problem (UWCMP). We show an algorithm which finds the optimal solution of UWCMP in polynomial time. Details are shown in Appendix A.

*Remark 2:* By applying the proposed network design algorithms, we can discard the waiting period in the CURE policy. Instead we modify the network such that  $z_{G'}(p) \leq r/4$  and start a segment. The target path is given by  $p$ . Then we start a new segment. When  $|D(t)| \geq z_{G'}(p) \leq r/(4d_{\max})$ , we calculate a new nearly optimal crusade  $p$  and a modified graph  $G'$  with  $z_{G'}(p) \leq r/4$  and start a new segment.

### B. Network Design for Minimizing the Maximum Restricted Cut

In this section, we study a network design problem where the goal is to minimize the maximum restricted cut associated with a given infected set, namely Problem 5. Intuitively, a restricted cut measures the total weight of edges connecting infected nodes to healthy ones. Reducing this cut makes it easier to contain the spread of infection. We show that by reducing the restricted cut of the infected set below a certain threshold using a limited edge weight reduction budget, we can guarantee sublinear expected extinction time, even when the curing order is arbitrary.

Given an algorithm for Problem 5, we propose a dynamic curing policy, where the network is modified adaptively based on the current infected set  $I(t)$ . Let  $r(t)$  be the total curing rate allocated to the infected nodes  $I(t)$  at time  $t$ . Suppose  $r(t)$  is always greater than a fixed constant  $r'$  for all  $t$ . The policy proceeds as follows:

- *Network design:* modify the graph such that  $\phi_{G'}(I(t)) \leq r'/4$  and start a segment. Let  $A = I(t)$  be the set of infected nodes at time  $t$ .
- *Segment:* Define  $D(t) = I(t) \setminus A$ . If  $|D(t)| \geq r'/(4d_{\max}) - 1$  then start a new network design period.

*Theorem 7:* If  $r' \geq 2 \log n$  always holds, by modifying  $G'$  such that  $\phi_{G'}(I(t)) \leq r'/4$ , the policy achieves sublinear expected extinction time.

*Proof:* We let the curing process start at  $t = 0$ . Let  $M_t \stackrel{\text{def}}{=} |I(t)|$ . We define the process  $P_t$  on the space  $\{0, 1, 2, \dots\}$  as follows: Let  $P_0 = M_0$ . The process  $P_t$  has a downward transition rate of  $r$  and a rate  $r/2$  of upward transition. Let  $T$  be the first time that  $P_t$  takes the value 0. We note that  $M_t$  is stochastically dominated by  $P_t$ .

We further let  $H_t \stackrel{\text{def}}{=} M_t + \frac{r}{2} \cdot t$ , and  $\hat{H}_t$  be the truncation of  $H_t$  which stops at time  $T$  and keeps the value  $H_T$  afterwards. Since the cut between the infected nodes and the healthy nodes is less than  $r/2$ , and the downward drift is at least  $r$ , the total downward drift of  $\hat{H}_t$  is greater than  $r/2$ . Therefore  $\hat{H}_t$  is a supermartingale.

By Doob's optional stopping theorem,

$$k = \mathbb{E}[M_0] = \mathbb{E}[\hat{H}_0] \geq \mathbb{E}[\hat{H}_T] + \frac{r}{2} \mathbb{E}[T] \geq \frac{r}{2} \mathbb{E}[T].$$

Therefore, we attain  $\mathbb{E}[T] \leq \frac{2k}{r}$ . Since  $r \geq 2 \log n$ , we arrive at  $\mathbb{E}[T] \leq \frac{k}{\log n} \leq \frac{n}{\log n}$ .  $\square$

Now we present the algorithm for Problem 5. We consider the following minimax program for the modified graph  $G' = (V, E, w')$  where  $w'_{uv} = w_{uv} - \Delta_{uv}$ :

$$\begin{aligned} &\text{minimize}_{G'} \quad \phi_{G'}(A) \\ &\text{subject to} \quad 0 \leq \Delta_{uv} \leq w_{uv}, \forall (u, v) \in E \\ &\quad \quad \quad \sum_{(u,v) \in E} \Delta_{uv} \leq b', \end{aligned} \quad (11)$$

where  $b'$  is a given edge reduction budget. We note that by running a binary search on  $b'$  we solve Problem 5 with any precision.

*Theorem 8:* There exists a polynomial time algorithm which computes a graph  $G'$  with weight reductions  $\Delta_{uv}$ , in which  $\phi_{G'}(A) \leq 1.14 \cdot \phi_{\hat{G}}(A)$  where  $\hat{G}$  is any modified graph with the same reduction budget  $b$ .

The maximum restricted cut  $\phi_G(A)$  for a fixed  $G$  and  $A$  is given by the following integer program:

$$\begin{aligned} &\text{maximize}_{y_i} \quad \frac{1}{2} \left( \sum_{\substack{i < j \\ i, j \in A}} w_{ij}(1 - y_i y_j) + \sum_{\substack{i \in A \\ j \in (V \setminus A)}} w_{ij}(1 - y_i) \right) \\ &\text{subject to} \quad |y_i| = 1, \end{aligned} \quad (12)$$

where  $y_i \in \{-1, 1\}$  for  $i \in A$  provides the partition of nodes in the maximum restricted cut.

Equivalently, we can combine all the nodes in  $V \setminus A$  as a new node  $s$  and obtain a modified graph  $\hat{G}$  with  $|A| + 1$  nodes  $\hat{V} = A + s$ . The edge set  $\hat{E}$  in  $\hat{G}$  contains all the edges  $(u, v)$  in the graph  $G[A]$  with their original weights. We add edge  $(u, s)$  to  $\hat{E}$  with the weight  $\hat{w}_{us} = \sum_{(u,v) \in E, v \in A^c} w_{uv}$  if there exists an edge  $(u, v)$  for  $v \in A^c$ . Then we study the equivalent maximum cut problem

$$\begin{aligned} & \underset{y_i}{\text{maximize}} && \frac{1}{2} \sum_{\substack{i < j \\ i, j \in \hat{V}}} \hat{w}_{ij} (1 - y_i y_j) \\ & \text{subject to} && |y_i| = 1. \end{aligned} \quad (13)$$

Let  $L_{\hat{G}}$  be the Laplacian matrix of the graph  $\hat{G}$ . The above problem has a well-known Semidefinite Programming (SDP) relaxation [44]:

$$\begin{aligned} & \underset{X}{\text{maximize}} && \frac{1}{4} \text{Tr}(L_{\hat{G}} X) \\ & \text{subject to} && X \succeq 0. \end{aligned} \quad (14)$$

Therefore, a relaxation of the program in (11) is given by

$$\begin{aligned} & \min_{\Delta} \max_X && \text{Tr}(B^\top (W - \text{diag}(\Delta)) B X) \\ & \text{subject to} && X \succeq 0, W \succeq \text{diag}(\Delta) \succeq 0 \\ & && \mathbf{1}^\top \Delta \leq b, \end{aligned} \quad (15)$$

where  $B$  is the edge-node incident matrix of  $\hat{G}$  and  $W$  is the diagonal weight matrix for edges in  $\hat{G}$ .

*Proof of Theorem 8:* For any maximum cut instance, the optimal value of the SDP relaxation is upper bounded by 1.14 times the maximum cut [44]. Thus, we have that the optimal value of program (15) is at most 1.14 times the optimal value of program (11). Let  $(\Delta^*, X^*)$  be the optimal solution of program (15). Let the graph  $G'$  be the modified graph corresponding to the weight reduction  $\Delta^*$ . By the minimax theorem [45], [46], we have  $\phi_{G'}(A) \leq \text{OPT}(\Delta^*, X^*)$ , where  $\text{OPT}(\Delta^*, X^*)$  is the optimal value of program (15). Thus, we find a modified graph  $G'$  such that  $\phi_{G'}(A)$  is at most 1.14 times the optimal solution of program (11).

It suffices to find an algorithm that solves (15) in polynomial time. The dual form of (14) is given by

$$\begin{aligned} & \underset{u, t}{\text{minimize}} && \frac{1}{4} n \cdot t \\ & \text{subject to} && tI - (L + \text{diag}(u)) \succeq 0, \\ & && \mathbf{1}^\top u = 0. \end{aligned} \quad (16)$$

The primal (14) and dual (16) both have feasible solutions. In addition the primal has a strictly feasible solution. Therefore, weak and strong duality hold for the pair, indicating that the primal and the dual have the same optimal solution.

The program in (15) can be further written as

$$\underset{\Delta, u, t}{\text{minimize}} \quad \frac{1}{4} n \cdot t$$

$$\text{subject to} \quad tI + B^\top \text{diag}(\Delta) B - \text{diag}(u) \succeq B^\top W B,$$

$$\mathbf{1}^\top u = 0, W \succeq \text{diag}(\Delta) \succeq 0, \mathbf{1}^\top \Delta \leq b. \quad (17)$$

It is easy to observe that the objective is linear and the constraints are convex in  $\Delta$ ,  $u$ , and  $t$ . Therefore, (17) is an SDP which can be solved in polynomial time.  $\square$

## VI. NUMERICAL SIMULATIONS

In this section, we examine the effectiveness of the proposed curing policies as well as the network design algorithms through numerical simulations. The model for the SIS process is described in Section II-B. For all numerical examples, the original infection rates of the edges in the network are sampled from a uniform distribution of the range  $[0.4, 1.6]$ . Curing rates of the nodes are allocated by the policies with a budget  $r$ .

### A. Simulations for Curing Policies

We compare Algorithm 1 against 4 baseline curing policies, including two static policies and their dynamic variations.

- 1) *Uniform (static)*: the curing budget is uniformly allocated to all nodes,  $\rho_u(t) = 1/n, \forall u \in V$ .
- 2) *Degree (static)* [18]: the curing rate of a node  $u \in V$  is set to  $r \cdot d_u / (\sum_{u \in V} d_u), \forall u \in V$ .
- 3) *Uniform (dynamic)*: the curing budget is uniformly allocated to all *currently infected* nodes, i.e.  $\rho_u(t) = 1/|I(t)|, \forall u \in I(t)$ , and otherwise set to 0.
- 4) *Degree (dynamic)*: the curing rate of a node  $u \in I(t)$  is set to  $\rho_u(t) = r \cdot d_u I_u(t) / (\sum_{u \in V} d_u I_u(t))$ , and otherwise set to 0.

We implement the CURE policy by using Algorithm 1 to calculate the ordering of curing. We note that instead of using the 1/3-balanced cut algorithm of [13], we use a heuristic for balanced cut based on the eigenvector corresponding to the second smallest eigenvalue of the graph Laplacian. We sort the nodes using their corresponding values in the eigenvector, and run a sweep algorithm to find the best 1/3-balanced cut, similar to the SweepCut algorithm for the sparsest cut problem [47]. Our study on the fair curing policy shows the existence of polynomial-time approximations to the fair curing problems. However, the construction of the tree embedding [14] remains a theoretical tool, not a practical algorithm. Therefore, we do not implement Algorithm 2.

To test the effectiveness of the CURE policy using Algorithm 1, we simulate the SIS process on two typical networks: the locally connected network and the binary tree network. The locally connected network is augmented from a path of  $n = 3000$  vertices. Suppose the nodes in a path are labeled using its distances to one of the endpoint of the path, then we add edges  $\{i, i + 2\}, i = 0, 1, \dots, (n - 2)$  to the path to get the locally connected network with  $m = 5997$  edges. The binary tree network is a perfect binary tree with 11 layers, therefore, it has  $n = 2047$  nodes and  $m = 2046$  edges.

Fig. 1 shows the performance of the CURE policy and the baselines. The figure shows the average results across 10 runs for all methods. The curing budgets are set to 800 and 500 for the locally connected network and binary tree network, respectively.

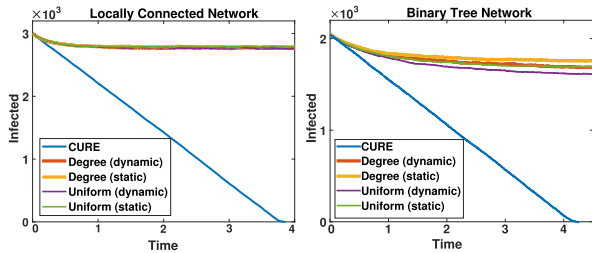


Fig. 1. Comparison of CURE (using Algorithm 1) with baseline policies.

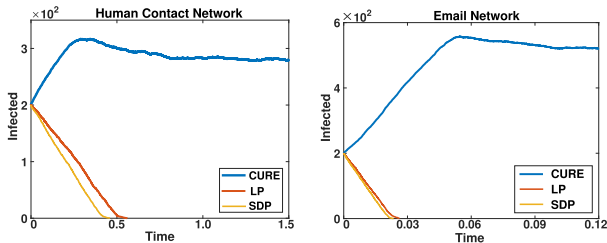


Fig. 2. Comparison of network design algorithms: 1) CURE without network design; 2) CURE with linear programming network design; 3) Random curing rate allocation with SDP.

For the same network, all policies use the same curing budget. We assume all nodes in the networks are infected at the beginning of the process.

In both networks, the CURE policy is the only one that succeeds in curing all nodes in all 10 runs. All baseline methods fail to cure the network in any run. The figure clearly shows the advantage of the CURE policy against the considered heuristics, whether they are static or dynamic.

### B. Simulations for Network Design Algorithms

For the network design algorithms, we simulate the process on a human contact network and an email network. We use the networks to simulate the spread of human viruses and internet worms respectively. The human contact network is constructed by collecting data using mobile technologies [48], [49]. The network has 405 nodes and 1258 edges. We let the number of initial infections to be 200, and the curing budget be 500. All initial infected nodes are selected uniformly at random. The email network was generated using email data from a research institute [50]. We ignore the directions of edges, and then attain a network with 986 nodes and 16057 edges. We let the number of initial infected nodes to be 200, and  $r = 10000$  for the email network.

For both networks, we compare CURE policy with CURE policy augmented by linear programming (LP)-based network design, as well as random curing budget allocation complemented by semi-definite programming (SDP)-based network design. Fig. 2 shows the average trajectories of 10 runs for each method for both networks. Simulations on both networks show consistent results.

As for efficiency, we use the CVX package [51] to solve the LP and SDP problems. The average running time of the LP- and SDP-based network design algorithms is given in Table I.

TABLE I  
AVERAGE RUNNING TIME OF THE LP- AND SDP-BASED ALGORITHMS

Algorithm	Human Contact Network	Email Network
Nodes/Edges	405/1258	986/16057
LP-based	2.39s	21.57s
SDP-based	11.37s	837.81s

The results show the average running time of 10 runs of the simulation. The results show that the LP-based algorithm is more scalable. All experiments are conducted using Matlab 2022b. We run the algorithms on a laptop with an AMD R7 6800H CPU (3.2 GHz) using a single thread.

For the human contact network, the trajectory of the CURE policy clearly shows a waiting period at the beginning of the process. After the curing rates are dynamically allocated, the number of infections plateaus subsequent to a transient decline. With network design algorithms, the curing processes are continuous and quickly stop the SIS processes. We note that the SDP approach considers the worst curing rate allocation strategy and hence is more conservative. The SDP network algorithm achieves shorter extinction time with random curing targets at any given time, with an average of 890.24 network weight reduction for the human contact network, while the LP algorithm only reduces 419.11 edge weights in average. Results are similar for the email network, where SDP and LP costs 4994.05 and 2595.09 weight reduction in average, respectively. The averages are taken over all network design periods until the end of the processes.

## VII. CONCLUSION

In this paper, we have studied efficient algorithms for dynamic curing policies of SIS epidemic models. We have proposed approximation algorithms to the impedance and a fair impedance of an infected set in a given graph. We have also proposed algorithms for network design problems to help cure the network. Provable guarantees have been provided for all the studied algorithms. We have shown the effectiveness of the policy and network co-design approach by running simulations on real contact networks.

There are two open problems for consideration in future research. The first open problem is to find algorithms ensuring fairness in *every* interval longer than a given constant. The second is to devise algorithms for optimal network design to minimize impedance *without* fixing the order of curing. Future work should also study practical algorithms for fair curing, and improving the running time of current solutions.

## APPENDIX

We show a polynomial time reduction from UWCMP to the following problem.

*Problem 6 (Interval Scheduling Problem on  $\ell$  machines ( $\ell$ -ISP)):* Given a set of  $m$  intervals  $\Pi = \{\pi_1, \dots, \pi_m\}$  where each interval  $\pi_i \in \Pi$  has a start time  $s_i$  and a finish time  $f_i$  ( $s_i < f_i$ ), find a maximal subset of intervals  $Q \subseteq \Pi$  and a mapping  $\sigma : Q \mapsto [\ell]$  such that there is no overlap among intervals scheduled on every machine.

**Theorem 9:** There exists a polynomial time algorithm which finds the optimal solution for UWCMP.

*Proof:* For an instance of UWCMP, nodes are ordered according to the crusade  $p$ . In particular, the node in  $p_i \setminus p_{i-1}$  is assigned with the label  $v_i$  for  $i \in [k]$ . Then the ordered vertices  $\{v_i\}_{i=1}^k$  in UWCMP are mapped to timestamps in  $\ell$ -ISP. The edges are mapped to intervals. Specifically, for an edge  $e = (v_i, v_j)$ , we let  $s_e = \min\{v_i, v_j\}$  and  $f_e = \max\{v_i, v_j\}$ . We further map the threshold to the number of machines by defining  $\ell = b$ . Then we obtain the input for  $\ell$ -ISP from an instance of UWCMP. We note that  $\ell$ -ISP can be solved by a greedy algorithm in  $O(m \log m)$  running time [52], [53]. From the output of an instance of  $\ell$ -ISP, we can construct a feasible solution for the corresponding UWCMP instance. For the interval set  $Q$ , we find the corresponding edges  $E_Q$  in UWCMP and let  $\Delta_{uv} = 0$  for  $(u, v) \in E_Q$ . We further let  $\Delta_{uv} = 1$  for  $(u, v) \in E \setminus E_Q$ . The cost of UWCMP is given by  $|E| - |Q|$ . The running time of the algorithm for UWCMP is  $O(m \log m)$  where  $m$  is the number of edges of  $G$  in UWCMP.  $\square$

In Algorithm 2, We sequentially find fair partitions for the given checkpoints. For each checkpoint, we find a fair partition at this checkpoint. Then, we run balanced cuts recursively to pick the crusade for nodes before this checkpoint. Then we run the algorithm on the subgraph supported on the remaining nodes.

We reduce the problem to finding a fair partition with a small cut on tree instances. To this end, we recall the definition of hierarchically well-separated trees.

**Lemma 3 ([14]):** For any graph  $G = (V, E, w)$  with a weight function  $w : E \mapsto \mathbb{R}_{\geq 0}$ , there exists a collection of trees  $T_1 = (V, E_1), T_2 = (V, E_2), \dots, T_N = (V, E_N)$  with tree  $T_i$  having an edge weight function  $w^i : E_i \mapsto \mathbb{R}_{\geq 0}$ , and find non-negative multipliers  $(\lambda_1, \dots, \lambda_N)$ , such that  $\sum_{i=1}^N \lambda_i = 1$  and  $N = \text{poly}(|V|)$ . For any subset  $S \subseteq V$  let  $\mathcal{B}(S)$  be the set of edges in  $E$  with one endpoint in  $S$ , and  $\mathcal{B}_i(S)$  be the set of edges in  $E_i$  with one endpoint in  $S$ . Then, for any  $S \subseteq V$ :

- 1)  $w(\mathcal{B}(S)) \leq w^i(\mathcal{B}_i(S))$  for every  $i \in [N]$ ;
- 2)  $\sum_{i=1}^N \lambda_i w_i(\mathcal{B}_i(S)) \leq O(\log n)w(\mathcal{B}(S))$ .

*Proof of Theorem 3:* We prove a simple setting with two groups  $V_1, V_2$  and one checkpoint  $\tau_1$ . We first partition the bag  $A$  into  $\gamma$ -fair parts  $S_0$  and  $S_1$  by using the hierarchical well-separated trees and dynamic programming on trees. Then we partition  $S_0$  and  $S_1$  with balanced cuts recursively until the size of a set is 1.

Let  $S_0^*, S_1^*$  be the minimum weight  $\gamma$ -fair partitioning of bag  $A$ . By the hierarchical well-separated trees, there exists a tree embedding  $T^*$  such that the total weight of edges crossing the sets  $S_0^*, S_1^*$  is preserved up to a  $O(\log k)$  factor. Suppose we find the minimum weight  $\gamma$ -fair partitioning on the tree  $T^*$ , which is shown in Lemma 4. Then, we find a  $\gamma$ -fair partitioning which has a total cut weight at most  $O(\log k)$  times the optimal  $\gamma$ -fair impedance.

Finally, we pick the order for two parts  $S_0$  and  $S_1$  separately with recursive balanced cuts. Similar to Theorem 2, there are totally  $O(\log k)$  balanced cuts and each cut has a total weight at most  $O(\log k)$  times the optimal  $\gamma$ -fair impedance. Combining all these balanced cuts and the cut edges crossing  $S_0$  and  $S_1$ , we get the conclusion.  $\square$

**Lemma 4:** Given a tree  $T$ , there exists a polynomial time dynamic programming algorithm that finds the minimum cut that partitions the tree  $T$  into  $\gamma$ -fair parts.

To find a minimum cut that yields a  $\gamma$ -fair partition of a tree, we use dynamic programming to exploit the recursive structure of the tree. For each subtree, we enumerate all valid group assignments and store the minimum cut cost for each configuration. The algorithm processes the tree in a bottom-up manner, combining solutions from children and accounting for possible configurations at each node. This ensures an efficient and optimal partitioning that meets the desired fairness.

*Proof:* As explained in Lemma 15.18 in [31], we can assume without loss of generality that the tree is binary. Our dynamic programming algorithm is based on the value table

$$DP[v, l_v, k_1(0), k_2(0)],$$

where  $k_h(0), h \in [\ell]$  denotes the number of nodes in the sub-tree rooted at  $v$  and group  $h$  which are assigned to part  $S_0$ . The indicator variable  $l_v$  takes value 0 if  $v \in S_0$ , otherwise  $l_v = 1$ . We use  $DP[v, l_v, k_1(0), k_2(0)]$  to record the minimum cut in the subtree rooted at  $v$  that satisfies the given configuration.

Then we calculate the value table from the leaves of the tree to the root. We set  $DP[v, s_v, k_1(0), k_2(0)]$  to the minimum of the following four cases:

*Case 1:* When  $l_v = l_{vL} = l_{vR}$ , we obtain

$$\min \{ DP[v^L, l_{vL}, k_1^L(0), k_2^L(0)] \\ + DP[v^R, l_{vR}, k_1^R(0), k_2^R(0)],$$

$$\text{where } k_h(0) = k_h^L(0) + k_h^R(0) + \mathbf{1}\{v \in V_h, l_v = 0\}, \\ \forall h \in \{1, 2\}. \quad (18)$$

To simplify the notation, we define (18) to be  $f(l_v, l_{vL}, l_{vR})$ .

*Case 2:* When  $l_v = l_{vL} \neq l_{vR}$ , we have  $w(v, v^R) + f(l_v, l_{vL}, l_{vR})$ .

*Case 3:* When  $l_v = l_{vR} \neq l_{vL}$ , we have  $w(v, v^L) + f(l_v, l_{vL}, l_{vR})$ .

*Case 4:* When  $l_v \neq l_{vR} = l_{vL}$ , we have  $w(v, v^L) + w(v, v^R) + f(l_v, l_{vL}, l_{vR})$ .

The first case corresponds to cutting neither of  $\{v, v^L\}$  and  $\{v, v^R\}$ ; the second to cutting  $\{v, v^L\}$  but not  $\{v, v^R\}$ ; the third to cutting  $\{v, v^R\}$  but not  $\{v, v^L\}$ ; the last to cutting both edges.

To fill in the table, we initialize all leaves of the tree. Let  $v$  be a leaf, then  $DP[v, l_v, \mathbf{1}\{v \in V_1, l_v = 0\}, \mathbf{1}\{v \in V_2, l_v = 0\}] = 0$  for  $l_v \in \{0, 1\}$ . Let all other entries of leaves be infinity. We make a pass from the bottom to fill in the table. Then we check the root of the tree and find the minimum value in the table such that the partitioning is  $\gamma$ -fair. The running time of the algorithm is  $O(k^3)$ .  $\square$

Next, we show that our algorithm achieves a  $O(\log^2 k)$  approximation for a set of checkpoints  $\mathcal{T}$  when the set of checkpoints satisfies a doubling condition.

*Proof of Theorem 4:* Similar to Theorem 3, it is sufficient to show that we partition the bag  $A$  into  $\gamma$ -fair parts w.r.t. the set of checkpoints  $\mathcal{T}$ . In Theorem 3, we show that our algorithm finds a  $\gamma$ -fair partitioning w.r.t. one checkpoint. To find this  $s + 1$  partitioning, we iteratively call our algorithm at each checkpoint

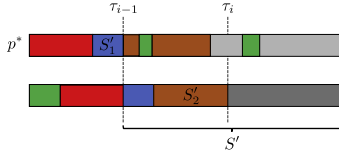


Fig. 3. The construction of a good partition in  $S'$ . The top figure corresponds to the optimal  $\gamma$ -fair crusade. The bottom figure corresponds to a good partition.

from  $\tau_1$  to  $\tau_s$ . Specifically, we first run our algorithm to find a  $\gamma$ -fair partitioning w.r.t.  $\tau_1$ ,  $S_1$  and  $A \setminus S_1$ . Then, we partition the set  $A \setminus S_1$  w.r.t. checkpoint  $\tau_2$ . We repeat this process until we get a  $s + 1$  partitioning of bag  $A$ .

For every  $i \in \{1, 2, \dots, s\}$ , we show that the total weight of cut edges crossing the checkpoint  $\tau_i$  in the algorithm solution,  $\bigcup_{j=0}^{i-1} S_j$  and  $\bigcup_{j=i}^s S_j$  is at most  $O(\log^2 k)$  times the optimal  $\gamma$ -fair impedance. By Theorem 3, it holds for the checkpoint  $\tau_1$ . Next, we consider any checkpoint  $\tau_i > \tau_1$ . We show that there exists a  $\gamma$ -fair partitioning of nodes in  $\bigcup_{j=0}^{i-1} S_j$  at checkpoint  $\tau_i$  such that the weight of cut edges is at most two times the optimal  $\gamma$ -fair impedance. By Theorem 3, our algorithm finds a  $O(\log k)$  approximation to this partitioning. Since there are at most  $\log_2 k$  such checkpoints due to the doubling condition, the total weight of cut edges in our solution is at most  $O(\log^2 k)$  times the optimal  $\gamma$ -fair impedance.

Now we show the existence of a good partitioning in the set of nodes  $S' = \bigcup_{j=i-1}^s S_j$  for every  $i > 1$ . Let  $p^*$  be the optimal  $\gamma$ -fair crusade w.r.t. the set of checkpoints  $\mathcal{T}$ . We denote the nodes contained in the optimal crusade  $p^*$  before checkpoint  $\tau_{i-1}$  but not contained in our solution before  $\tau_{i-1}$  by

$$S'_1 = (p^* \setminus p^*_{\tau_{i-1}}) \setminus \bigcup_{j=0}^{i-2} S_j.$$

We have  $S'_1$  is a subset of  $S'$  and  $|S'_1| \leq \tau_{i-1}$ . Since checkpoints satisfy  $\tau_i - \tau_{i-1} \geq \tau_{i-1}$ , we fully assign this set  $S'_1$  to the part between the checkpoints  $\tau_{i-1}$  and  $\tau_i$ . Note that the set  $S' \setminus S'_1$  is contained in  $p^*_{\tau_{i-1}}$  and at least  $\tau_i - \tau_{i-1} - |S'_1|$  among them are between checkpoints  $\tau_i$  and  $\tau_{i-1}$ . Thus, let  $S'_2$  be the set of the first  $\tau_i - \tau_{i-1} - |S'_1|$  nodes in  $p^*_{\tau_{i-1}}$  that are also contained in  $S' \setminus S'_1$ . Then  $S'' = S'_1 \cup S'_2$  is a valid partition of length  $\tau_i - \tau_{i-1}$ . Fig. 3 illustrates the construction of the partitioning.

Then, we show that the cut  $S'' = S'_1 \cup S'_2$  is a good partitioning in  $S'$ . The total weight of this cut  $S''$  in  $S'$  is at most the sum of the weight of edges from  $S'_1$  to  $(S'')^c$  and edges from  $S'_2$  to  $(S'')^c$ , which is at most two times the width of  $p^*$ . Note that  $S'_1$  is a subset of  $p^* \setminus p^*_{\tau_{i-1}}$ , which is a  $\gamma$ -fair set with size  $\tau_{i-1} \leq \tau_i - \tau_{i-1}$ . The set  $S'_2$  is a subset of  $p^*_{\tau_{i-1}} \setminus p^*_{\tau_i}$ , which is also a  $\gamma$ -fair set with size  $\tau_i - \tau_{i-1}$ . Therefore, we have  $S''$  is a  $2\gamma$ -fair set.  $\square$

## REFERENCES

- [1] W. O. Kermack and A. G. McKendrick, "Contributions to the mathematical theory of epidemics. II. the problem of endemicity," in *Proc. Roy. Soc. London, Ser. A*, vol. 138, no. 834, pp. 55–83, 1932.
- [2] M. Youssef and C. Scoglio, "An individual-based approach to SIR epidemics in contact networks," *J. Theor. Biol.*, vol. 283, no. 1, pp. 136–144, 2011.
- [3] P. V. Mieghem, J. Omic, and R. Kooij, "Virus spread in networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 1–14, Feb. 2009.

- [4] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies*, 2005, vol. 2, pp. 1455–1466.
- [5] H. J. Ahn and B. Hassibi, "Global dynamics of epidemic spread over complex networks," in *Proc. 52nd IEEE Conf. Decis. Control*, 2013, pp. 4579–4585.
- [6] P. E. Paré, C. L. Beck, and A. Nedić, "Epidemic processes over time-varying networks," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1322–1334, Sep. 2018.
- [7] J. W. Timbie et al., "Allocation of scarce resources during mass casualty events," in *Proc. Evidence Rep./Technol. Assessment, Agency Healthcare Res. Qual.*, 2012, pp. 1–305.
- [8] A. Devereaux et al., "Optimizing scarce resource allocation during COVID-19: Rapid creation of a regional health-care coalition and triage teams in San Diego county, California," *Disaster Med. Public Health Preparedness*, vol. 16, no. 1, pp. 321–327, 2022.
- [9] S. Hempel et al., "Resource allocation and pandemic response: An evidence synthesis to inform decision making," *Rapid Evidence Product, Agency Healthcare Res. Qual. (US)*, pp. 1–305, 2020. [Online]. Available: [https://www.ncbi.nlm.nih.gov/sites/books/NBK562921/pdf/Bookshelf\\_NBK562921.pdf](https://www.ncbi.nlm.nih.gov/sites/books/NBK562921/pdf/Bookshelf_NBK562921.pdf)
- [10] 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, Jul.–Dec. 2014.
- [11] World Health Organization, "A global framework to ensure equitable and fair allocation of COVID-19 products," Jun. 2020. [Online]. Available: [https://apps.who.int/gb/COVID-19/pdf\\_files/18\\_06/Global%20Allocation%20Framework.pdf](https://apps.who.int/gb/COVID-19/pdf_files/18_06/Global%20Allocation%20Framework.pdf)
- [12] M. Dimitz, A. Srinivasan, L. Tsepeneas, and A. Vullikanti, "Fair disaster containment via graph-cut problems," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, Mar. 2022, vol. 151, pp. 6321–6333.
- [13] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *J. ACM*, vol. 46, no. 6, pp. 787–832, Nov. 1999.
- [14] H. Räcke, "Optimal hierarchical decompositions for congestion minimization in networks," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, pp. 255–264.
- [15] V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. J. Pappas, "Optimal resource allocation for network protection against spreading processes," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 99–108, Mar. 2014.
- [16] S. Jafarizadeh and D. Veitch, "Optimal curing resource allocation for epidemic spreading processes," *Automatica*, vol. 150, 2023, Art. no. 110851.
- [17] D. Wang et al., "Controlling a networked SIS model via a single input over undirected graphs," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 10981–10986, 2020.
- [18] C. Borgs, J. Chayes, A. Ganesh, and A. Saberi, "How to distribute antidote to control epidemics," *Random Structures Algorithms*, vol. 37, no. 2, pp. 204–222, 2010.
- [19] F. Chung, P. Horn, and A. Tsiatas, "Distributing antidote using pagerank vectors," *Internet Math.*, vol. 6, no. 2, pp. 237–254, 2009.
- [20] B. Zhu, J. Lam, and M. Ogura, "Log-log convexity of an optimal control problem for positive linear systems," *Automatica*, vol. 146, 2022, Art. no. 110553. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109822004149>
- [21] A. L. Krause, L. Kurowski, K. Yawar, and R. A. V. Gorder, "Stochastic epidemic metapopulation models on networks: SIS dynamics and control strategies," *J. Theor. Biol.*, vol. 449, pp. 35–52, 2018.
- [22] M. Ogura and V. M. Preciado, "Epidemic processes over adaptive state-dependent networks," *Phys. Rev. E*, vol. 93, Jun. 2016, Art. no. 062316. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.93.062316>
- [23] Y. Wang, S. Gracy, H. Ishii, and K. H. Johansson, "Suppressing the endemic equilibrium in SIS epidemics: A state dependent approach," *IFAC-PapersOnLine*, vol. 54, no. 15, pp. 163–168, 2021.
- [24] Z. Svitkina and É. Tardos, "Min-max multiway cut," in *Proc. 2004 Approximation, Randomization, Comb. Optim. Algorithms Tech.*, K. Jansen, S. Khanna, J. D. P. Rolim, and D. Ron, Eds., 2004, pp. 207–218.
- [25] A. Hayrapetyan, D. Kempe, M. Pál, and Z. Svitkina, "Unbalanced graph cuts," in *Proc. Eur. Symp. Algorithms*, 2005, pp. 191–202.
- [26] S. Eubank et al., "Modelling disease outbreaks in realistic urban social networks," *Nature*, vol. 429, no. 6988, pp. 180–184, 2004.
- [27] Y. Yi, L. Shan, P. E. Paré, and K. H. Johansson, "Edge deletion algorithms for minimizing spread in SIR epidemic models," *SIAM J. Control*, vol. 60, no. 2, pp. S246–S273, 2022.

- [28] A. E. Babay, M. Dinitz, A. Srinivasan, L. Tsepenekas, and A. Vullikanti, "Controlling epidemic spread using probabilistic diffusion models on networks," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, Mar. 2022, vol. 151, pp. 11641–11654.
- [29] S. Saha, A. Adiga, B. A. Prakash, and A. K. S. Vullikanti, "Approximation algorithms for reducing the spectral radius to control epidemic spread," in *Proc. 2015 SIAM Int. Conf. Data Mining*, 2015, pp. 568–576.
- [30] K. Drakopoulos, A. Ozdaglar, and J. N. Tsitsiklis, "When is a network epidemic hard to eliminate?," *Math. Operations Res.*, vol. 42, no. 1, pp. 1–14, 2017.
- [31] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
- [32] F. Gavril, "Some NP-complete problems on graphs," in *Proc. Conf. Inf. Sci. Syst.*, 1977, pp. 91–95.
- [33] K. Drakopoulos, A. Ozdaglar, and J. N. Tsitsiklis, "A lower bound on the performance of dynamic curing policies for epidemics on graphs," in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 3560–3567.
- [34] S. Engebretsen, K. Engø-Monsen, A. Frigessi, and B. Freiesleben de Blasio, "A theoretical single-parameter model for urbanisation to study infectious disease spread and interventions," *PLoS Comput. Biol.*, vol. 15, no. 3, 2019, Art. no. e1006879.
- [35] J. K. Kelso, N. Halder, and G. J. Milne, "Vaccination strategies for future influenza pandemics: A severity-based cost effectiveness analysis," *BMC Infect. Dis.*, vol. 13, pp. 1–13, 2013.
- [36] T. Zenihana and H. Ishikawa, "Effectiveness assessment of countermeasures against bioterrorist smallpox attacks in Japan using an individual-based model," *Environ. Health Prev. Med.*, vol. 15, pp. 84–93, 2010.
- [37] N. Garg, V. V. Vazirani, and M. Yannakakis, "Approximate max-flow min-(multi) cut theorems and their applications," *SIAM J. Comput.*, vol. 25, no. 2, pp. 235–251, 1996.
- [38] C. F. Bornstein and S. Vempala, "Flow metrics," *Theor. Comput. Sci.*, vol. 321, no. 1, pp. 13–24, 2004.
- [39] U. Feige and J. R. Lee, "An improved approximation ratio for the minimum linear arrangement problem," *Inf. Process. Lett.*, vol. 101, no. 1, pp. 26–29, 2007.
- [40] National Center for Immunization and Respiratory Diseases (U.S.), Division of Viral Diseases and United States. Department of Homeland Security. Cybersecurity and Infrastructure Security Agency, "COVID-19 in racial and ethnic minority groups," Jun. 2020, pp. 1–16. [Online]. Available: <https://stacks.cdc.gov/view/cdc/89820>
- [41] S. Magesh et al., "Disparities in COVID-19 outcomes by race, ethnicity, and socioeconomic status: A systematic review and meta-analysis," *JAMA Netw. Open*, vol. 4, no. 11, p. e2134147, 2021.
- [42] L. Platt and R. Warwick, "Are some ethnic groups more vulnerable to COVID-19 than others?," Accessed: Jun. 21, 2025. [Online]. Available: <https://www.nejm.org/doi/full/10.1056/NEJMs2005114>
- [43] E. J. Emanuel et al., "Fair allocation of scarce medical resources in the time of COVID-19," *New England J. Med.*, vol. 382, no. 21, pp. 2049–2055, 2020. [Online]. Available: <https://www.nejm.org/doi/full/10.1056/NEJMs2005114>
- [44] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.
- [45] J. V. Neumann, "Zur theorie der gesellschaftsspiele," *Mathematische Annalen*, vol. 100, no. 1, pp. 295–320, 1928.
- [46] M. Sion, "On general minimax theorems," *Pacific J. Math.*, vol. 8, no. 1, pp. 171–176, 1958.
- [47] F. R. Chung, *Spectral Graph Theory*, vol. 92. Providence, RI, USA: American Mathematical Soc., 1997.
- [48] P. Klepac, S. Kissler, and J. Gog, "Contagion! The BBC four pandemic—the model behind the documentary," *Epidemics*, vol. 24, pp. 49–59, 2018.
- [49] J. A. Firth, J. Hellewell, P. Klepac, S. Kissler, A. J. Kucharski, and L. G. Spurgin, "Using a real-world network to model localized COVID-19 control strategies," *Nature Med.*, vol. 26, pp. 1–7, 2020.
- [50] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, pp. 1–41, 2007.
- [51] I. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0," Aug. 2012. [Online]. Available: <https://cvxr.com/cvx>
- [52] U. Faigle and W. M. Nawijn, "Note on scheduling intervals on-line," *Discrete Appl. Math.*, vol. 58, no. 1, pp. 13–17, 1995.
- [53] M. C. Carlisle and E. L. Lloyd, "On the  $k$ -coloring of intervals," *Discrete Appl. Math.*, vol. 59, no. 3, pp. 225–235, 1995.



**Yuhao Yi** received the Ph.D. degree from Fudan University, Shanghai, China, in 2019. From 2019 to 2020, he was a Postdoc with Rensselaer Polytechnic Institute, Troy, NY, USA. From 2020 to 2022, he was with the KTH Royal Institute of Technology, Stockholm, Sweden. He is currently a Researcher with the Institute of Clinical Pathology, West China Hospital and College of Computer Science, Sichuan University, Chengdu, China. His research interests include network systems, machine learning, and bioinformatics.



**Liren Shan** received the Ph.D. degree from Northwestern University, Evanston, IL, USA, in 2023. He is currently a research Assistant Professor with Toyota Technology Institute at Chicago, Chicago, IL. His research interests include approximation algorithm, graph algorithm, and algorithmic game theory.



**Shijie Wang** received the bachelor's degree in computer science from Sichuan University, Chengdu, China. He is currently working toward the master's degree in computer science with The Chinese University of Hong Kong, Hong Kong.



**Philip E. Paré** (Senior Member, IEEE) received the B.S. degree in mathematics in 2012, the M.S. degree in computer science from Brigham Young University, Provo, UT, USA, in 2014, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2018. From 2019 to 2020, he was with the KTH Royal Institute of Technology, Stockholm, Sweden. He is currently the Rita Lane and Norma Fries Assistant Professor with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. His research interests include networked control systems, namely modeling, analysis, and control of virus spread over networks.



**Karl H. Johansson** (Fellow, IEEE) received the M.Sc. degree in electrical engineering and the Ph.D. degree in automatic control from Lund University, Lund, Sweden. He held visiting positions with UC Berkeley, Caltech, NTU, HKUST Institute of Advanced Studies, and NTNU. He is currently a Professor with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden, and the Director of Digital Futures. His research interests include networked control systems and cyber-physical systems with applications in transportation, energy, and automation networks. He is an IEEE Control Systems Society Vice President Diversity, Outreach and Development, past President of the European Control Association, and a member of the IFAC Council. He was on the IEEE Control Systems Society Board of Governors and the Swedish Scientific Council for Natural Sciences and Engineering Sciences. He was the recipient of several best paper awards and other distinctions from IEEE, IFAC, and ACM. He has been awarded the Swedish Research Council Distinguished Professor, Wallenberg Scholar with the Knut and Alice Wallenberg Foundation, Future Research Leader Award from the Swedish Foundation for Strategic Research, the triennial IFAC Young Author Prize, and IEEE Control Systems Society Distinguished Lecturer. He is Fellow of the Royal Swedish Academy of Engineering Sciences.