Journal of Complex Networks (2013) Page 1 of 17
doi:10.1093/comnet/cnt009

Fast response to infection spread and cyber attacks on large-scale networks

Sven Leyffer
Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA

and

Ilya Safro*
School of Computing, Clemson University, Clemson, SC 29634, USA
*Corresponding author: isafro@clemson.edu

Edited by: Ernesto Estrada

[Received on 31 January 2013; accepted on 2 June 2013]

We present a strategy for designing fast and practical methods of response to cyber attacks and infection spread on complex weighted networks. In these networks, vertices can be interpreted as primitive elements of the system, and weighted edges reflect the strength of interaction among these elements. The proposed strategy belongs to the family of multiscale methods whose goal is to approximate the system at multiple scales of coarseness and to obtain a solution of microscopic scale by combining the information from coarse scales. In recent years, these methods have demonstrated their potential for solving optimization and analysis problems on large-scale networks. We consider an optimization problem that is based on the susceptible-infected-susceptible (SIS) epidemiological model. The objective is to detect the network vertices that have to be secured (or immunized) in order to keep a low level of infection in the system.

Keywords: mathematical and numerical analysis of networks; structural analysis of networks; networks and epidemiology; technological and infrastructural networks; infection spread.

1. Introduction

Networks are a widely used type of abstraction for complex data of scientific interest for which one may want to emphasize the relationships between primitive components of the system (such as vertices) by connecting them with edges (either directed or undirected). Examples can be found in domains such as cyber networks, food webs, social and metabolic networks. Recent growth of large-scale, real-world network data available for scientific analysis has promoted significant theoretical and practical advances in many areas of natural sciences and engineering [1]. Optimization of different quantitative objectives on networks often plays a crucial role in network science, not only when a practical solution is needed, but also for a general understanding of structural and statistical features of networks.

In recent decades, a significant amount of research in the networks science has been done in analysing infection spreading. Examples can be found in domains such as epidemiology [2], cyber security [3] and social sciences [1]. Developing strategies and formulations of response policies to the infection propagation is crucial for real-life applications. Usually, such response strategies can be formulated as optimization models that consider applying some operation (e.g. rerouting of network connections, updating the antivirus software, immunization of individuals) on network primitives (such

as vertices, edges and communities) at different resolutions. Often, such optimization models must consider that the number of available resources is *limited*, and performing *all* possible operations for better response results is not feasible.

Cybersecurity in open grids and peer-to-peer networks is a typical motivating example of an emerging area and corresponding real-life situations in which solving such optimization models on large-scale data are vitally important. Open grids and peer-to-peer networks represent collaborations among thousands of users and hundreds of organizations. Spreading the malicious attacks over them through the grid middleware is not the hardest task for the attacker because the grid middleware is designed to cross the boundaries between collaborators and their organizations seamlessly. When an unanticipated attack occurs, it is hard to mobilize the immediate response at all vertices of the network as well as it is often impossible to shut down the entire network. For details, see [3]. Another example is an infrastructure network massively damaged by the real attack or disaster. It can be hard to send engineers to all damaged places. Similar problems happen when epidemics occurs, i.e. it is impossible to immunize everybody immediately. Instead, we have to formulate the policies that can help to achieve a particular goal given partial immunization with limited resources. In this paper, we propose an approach for designing efficient and practical methods for response optimization problems on large-scale networks. Implementation of the method is available for download at [4].

The proposed strategy belongs to the family of multiscale methods whose goal is to approximate the system at multiple scales of coarseness and to obtain a final solution by combining the information from different scales. Heterogeneity is one of the key advantages of the multiscale framework that is, in particular, relevant to the discussed model. In the context of optimization model solvers, it is expressed in the ability to incorporate external optimization algorithms in the main framework at different scales. In other words, if for a particular model exceptionally suitable algorithm of higher than linear complexity was developed, one may consider to apply it as a local refinement in the multiscale framework (will be demonstrated in Section 3.2) for a better complexity and possibly better numerical quality (for details see [5]). The contribution of this work is providing a scalable method that is able to obtain numerical results of usually better quality but orders of magnitude faster than the existing methods. Computational complexity of the proposed optimization model along with the lower bounds and relevance to the previous deterministic models is presented in [6].

We introduce the optimal response model, notation and necessary definitions in Section 2. The multiscale framework and the algorithm are described in Section 3. Evaluation of the method on different artificial and real-world networks is presented in Section 4. In Section 5, we conclude and provide future research directions.

2. Optimal response model

We consider a traditional infection spread model in which network vertices can be in one of the two possible states, namely *infected* and *susceptible* (SIS model; see [7]), and each vertex i is associated with a probability of being infected at time t, $\phi_{i,t}$. Introduced as a simplification of the susceptible-infected-recovered (SIR) model in [8], the SIS model has been extensively analysed in epidemiology and adapted in the cyber security area for analysis of computer virus propagation [9]. In this paper we follow that general model and the probabilistic version of the optimal response model (formulated in [6]) that takes into account the status of all individuals in the network at one particular snapshot of the network.

The SIS model considers the following quantities: S, number of susceptible vertices; I, number of infected vertices; β, infection transmission rate; and δ, rate of recovery from infection. The model

describes an evolution of the two classes of population of infected vertices I and susceptible vertices S at time t as

$$\begin{cases} \dfrac{\mathrm{d}I}{\mathrm{d}t} = \lambda S - \delta I \\ \dfrac{\mathrm{d}S}{\mathrm{d}t} = \delta I - \lambda S, \end{cases} \tag{2.1}$$

where $\lambda = \beta\langle k\rangle I/(S+I)$ reflects the rate at which susceptible vertices become infected and $\langle k\rangle$ is average vertex degree. One of the most important consequences of (2.1) is the notion of an *epidemic threshold* τ, a measure to predict when the infection outbreak disappears, that is, the value that has to be compared with β/δ. Chakrabarti *et al.* proposed a topology-independent, non-linear dynamical system model of SIS [10]. Their model,

$$1 - \phi_{i,t} = (1 - \phi_{i,t-1})h_{i,t} + \delta\phi_{i,t-1}h_{i,t}, \quad i = 1 \ldots n, \tag{2.2}$$

describes the probability that vertex i is susceptible, where $h_{i,t}$ is the probability that vertex i is not infected from its neighbours at time t. We assume that the probabilities of vertices being infected in the previous round $t-1$ are independent; see [10] for more details. Often, the infection transfer rate cannot be represented by a single parameter β; thus, without loss of generality it is replaced by a matrix $P^{n\times n} = \{p_{ij}\}$, where p_{ij} is the probability of vertex i being infected by vertex j. The probability of an uncompromised vertex i not being infected at time t is

$$h_{i,t} = \prod_{j\in\Gamma_i}(1 - p_{ij}\phi_{j,t-1}), \tag{2.3}$$

where Γ_i is a set of neighbours of i.

In [3], Altunay *et al.* modelled the optimal response to network attacks for one snapshot of the network at time t in which each vertex can be in one of the two possible states (similar to SIS). Their multiobjective optimization model has two competing goals: reduction of the infection at uncompromised vertices as much as possible, and minimization of the impact of the response on the grid (or maximizing the utility of the network). In that model the infection accumulated at each vertex was estimated by the weighted sum of infections from neighbours.

We now formulate a more complicated version of their model in which the amount of infection is measured probabilistically based on (2.3). In our optimization problem, the goal is to maximize the (weighted) number of connections between those vertices that will not be considered by the policy as 'requiring special attention' while keeping the level of infection[1] at each vertex low. This is motivated by the infection-spread response policies in different domains that are often driven by the number of resources available for the realization. Given an appropriate definition of vertex weight, one may also consider similar maximization objectives for the vertices. If we define a vertex weight to be proportional to the weight of working edges, then the problems will be just equivalent. On the other hand, a simple counting of open nodes in the objective may lead to a contradiction between good solution of the maximization problem and the connectivity of a network. We note, however, that even if such model changes can be more appropriate for some real-life applications, it is likely that they will not change the principles of the proposed strategy.

[1] Note that the level of infection can accumulate both the existing infection at vertex i and the infection received from neighbours. Both can be represented in constant b_i in (2.4).

We denote the graph underlying a given network by $G = (V, E, w)$, where V is a set of vertices, E is a set of edges and $w : E \to \mathbb{R}_{\geqslant 0}$ is a weighting function on E that represents the strength of connectivity between two vertices (such as the number of shared users between sites in a cyber system). Assuming that the probabilities of infection transition from Γ_i to i are independent, the problem is formulated as

$$\begin{aligned} \text{maximize}_x \quad & \sum_{ij \in E} w_{ij} x_i x_j \\ \text{subject to} \quad & x_i - \prod_{j \in \Gamma_i} (1 - p_{ij} \phi_{j,t-1} x_j) \leqslant b_i \quad \forall i \in V, \\ & x_i \in \{0, 1\} \quad \forall i \in V, \end{aligned} \tag{2.4}$$

where w_{ij} is the edge weight between vertices i and j; b_i is a threshold for bounding the level of allowed probability of infection at vertex i; and x_i are binary variables (1, if we decide to leave the vertex i functioning, 0 closed, requiring special attention). If for some $ij \in E$, one of the vertices i or j is decided to be closed, that edge does not contribute its weight w_{ij} to the objective. In general, (2.4) is a non-convex integer non-linear programme and known to be NP-complete [6]. There exist a number of deterministic solvers for (2.4), including BARON [11] based on a branch-and-reduce strategy that employs piecewise linear underestimators of the multilinear functions on (2.4) to construct a convex relaxation, and then branches either on an integer variable or on a non-convexity creating a branch-and-bound tree-search. The open-source solver Couenne [12] implements a similar strategy. However, these solvers cannot handle problems with tens of thousands of vertices (typically they work well for problems with a few hundred variables and non-linear expressions), because the search tree explodes exponentially. Therefore, in this paper, we propose a strategy for designing fast, suboptimal multiscale methods for this class of problems. Such methods are often more useful in practice than optimal ones that take a long time to converge even for small instances. It is important to mention that, in practice, models with linear constraint in (2.4) (such as in [6]) poorly approximate the optimal solution on many real large-scale networks and do not describe the independence of events to absorb an infection propagation from different neighbours. If such event are dependent, the model constraint should be reformulated appropriately (see details in [3,6]).

3. Multiscale strategy

In many practical situations, it is often noticeable that when elementary parts of a system have a complicated behaviour, their ensembles can often be much more structured. The multiscale computational methodology [5,13] is a systematic approach for achieving efficient calculations of systems containing many degrees of freedom (such as network vertices, image pixels and particles). From the relationships among the given microscopic parts of the system, the rules for the system at increasingly coarser scales are derived. The idea behind multiscale methods is to collect the relevant information regarding the system at different scales and then to obtain the solution at the microscopic scale by adapting the information inherited from coarse scales. Realizations of the multiscale frameworks are attractive in practice, because they can be naturally combined with other computational and analysis techniques, making them suitable for applying on large-scale instances [5]. For many applications with underlying computational structural problems on networks (or graphs), the introduction of multiscale methods has led to breakthroughs in the quality of computational and data analysis results. Examples include

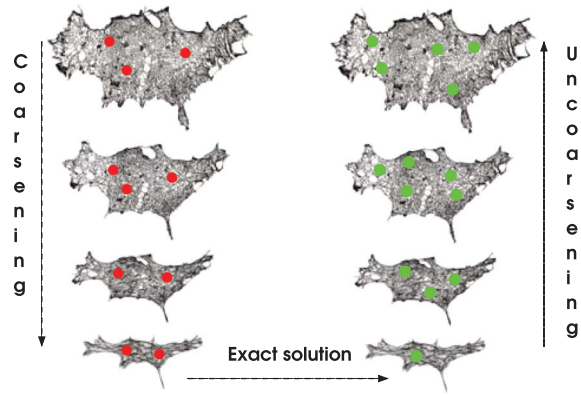

FIG. 1. V-cycle scheme for US roads infrastructure network. Circles in the left and right columns of images illustrate initially compromised and closed areas (solutions of (2.4)), correspondingly. The network along with its compromised areas is coarsened by graduate AMG-like projections (see (3.2)).

structural analysis problems [14,15], graph partitioning [16], clustering [17], segmentation [18], VLSI design [19] and linear arrangement [20].

Our method is inspired by *algebraic multigrid* (AMG) [5] approach for solving optimization problems on large-scale graphs [20]. In this framework, a hierarchy of decreasing-size network graph Laplacians $\{L_i\}_{i=0}^{k}$ is created by a process called *coarsening*, starting from the original network L_0. When a small-enough (or easy-to-solve) Laplacian L_k is created, the problem is solved exactly for L_k; and the solution is prolongated to the original L_0 by interpolating it scale after scale. Each interpolation is then followed by refinement, that is, by local processing that improves the solution (see Fig. 1).

Addressing the optimization problem at multiple scales of the complex network is beneficial, in particular, when it is known that the topology of many complex networks is hierarchical and thus might be described at multiple resolutions. The structural properties of the network can often be different at different scales, as evidenced by the finding that complex networks are self-dissimilar across scales [21–23]. These dissimilarities can naturally be reflected in the proposed multiscale framework.

3.1 *Coarse problem*

The construction of a coarse problem on a network consists of two main phases: defining the sets of coarse vertices and edges. For both phases, it is important to be able to describe how 'close' the two given fine-level vertices are to each other at the stage of switching to the coarse level. In particular, it is crucial in the context of infection spread on real networks when the weights on the edges can be noisy and the measure of closeness must take into account the neighbourhoods of vertices, instead of looking at one particular direct connection. A recently introduced approach of algebraic distance between vertices [20,24] has proved itself being successful in AMG-based methods [25]. We define the distance between vertices i and j as

$$\varrho_{ij}^{(k)} := \left(\sum_{r=1}^{R} |\chi_i^{(k,r)} - \chi_j^{(k,r)}|^p \right)^{1/p}, \tag{3.1}$$

where the superscript$^{(k,r)}$ refers to the kth iteration on the rth initial random vector, namely $\chi^{(k,r)} = H^k \chi^{(0,r)}$, and H is a Jacobi over-relaxation iterator of the Laplacian (see Appendix A.1). This approach substitutes edge weights and redefines the 'closeness' between two vertices by measuring how well their values are correlated at the coarsening stage. Technically this is done by several relaxation sweeps which take into account the connectivity of the respective neighbourhoods of two vertices [20]. In the context of our optimization problem, two correlated vertices (i.e. well connected) are more likely to exhibit a higher diffusion conductance of an infection spread (which can also lead to similar solutions) and, thus, can be located in the same coarse aggregate. The coarse-level solution of this aggregate will be interpolated to its components as initialization of the current level.

The set of coarse vertices V_c is created by the aggregation of fine vertices V_f into small clusters based on the strength of connectivity estimated by using the algebraic distance. Initially $V_f = V$ and $V_c = \emptyset$. The vertices from V_f are traversed one by one and divided into two sets C and F such that (a) $C \cup F = V_f$; and (b) vertices in F are strongly coupled to C (see Appendix A.2). Then vertices in F are divided among some of their neighbours in C to form future coarse vertices. The vertices are traversed in the ascending order of the infection level. By doing so we localize small areas of high connectivity that can potentially propagate the infection rapidly (and, thus, have to be be determined as 'closed' in the solution).

The coarse network Laplacian L_c is defined by the restriction operator $L_c = R^T L_f R$, where $R \in \mathbb{R}^{n \times N}$ is a matrix of connections between variables in F and C, where $n = |V_f|$ and $N = |C|$. Finally, the optimization problem is formulated for aggregated (coarse) variables as

$$\begin{aligned} \text{maximize}_X \quad & \sum_{IJ \in E_c} W_{IJ} X_I X_J + \sum_{I \in V_c} A_I X_I \\ \text{subject to} \quad & X_I - \prod_{J \in \Gamma_I} (1 - P_{IJ} \Phi_{J,t-1} X_J) \leqslant B_I \quad \forall I \in V_c, \\ & X_I \in \{0,1\} \quad \forall I \in V_c, \end{aligned} \tag{3.2}$$

where X_I are binary decision variables that correspond to coarse vertices; W_{IJ} and P_{IJ} are accumulated strengths of connectivity and infection spread probabilities between aggregates I and J in V_c, respectively; Φ_I are infection probabilities for coarse vertices; and B_I are accumulated thresholds for infection level for coarse vertices (see Appendix A.3).

The main difference between the fine and the coarse problems is the new linear term $\sum_{I \in V_c} A_I X_I$. It takes into account the fine-level edges between vertices aggregated into the same coarse vertex. Indeed, when the small subset of vertices represented by aggregate $I \in V_c$ remains open (i.e. contains low level of infection and does not spread too much of it) in the solution of (3.2), we assume that at the next-finer scale the endpoints of contracted edges will be (at least initially) open as well.

3.2 *Uncoarsening*

During the coarsening process, we recursively form the hierarchy of smaller problems until a small-enough level is reached. The size of this level depends on the external optimization solver one can use (see Appendix A.5). After the coarsest problem is solved, the solution is gradually prolongated back to the original scale. It consists of three phases: C-vertices interpolation, F-vertices interpolation and refinement.

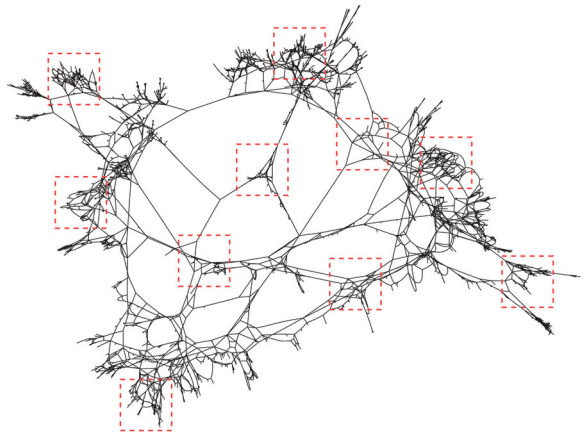

FIG. 2. Localized refinement. Dashed squares correspond to subgraphs induced by small subsets of vertices for localized refinements.

Initially, all seed vertices $i \in C$ are initialized by $x_i = X_I$, where X_I is a corresponding coarse variable seeded by vertex $i \in C$. Next, all F-vertices are interpolated by maximizing their contribution to the current objective. This is equivalent to solving (2.4) when all x_j's are fixed except the vertex i that is being currently interpolated. As a result of these two phases, we obtain the first solution of the fine-scale problem. This solution is then improved by Gauss–Seidel-like relaxation in which for every vertex the contribution of the opposite solution to the objective (or/and number of satisfied constraints) is compared with its current contribution. This process is realized by (sequential or parallel) graph traversal sweeps in which the contribution of all nodes is maximized deterministically.

The refinement phase consists of the collective improvement of the solution for sufficiently small subsets of variables. For this purpose we have formulated a *localized refinement* procedure, which extracts from the entire system small subproblems and solves each separately. We note that this phase can easily be performed in parallel by using for example the red-black order of the refinements [5] (see Fig. 2). Single subset refinement solves problem (3.3) for a subset of vertices S by choosing a connected subgraph and fixing the *boundary conditions* for the rest of the vertices. The single localized refinement is formulated as

$$\begin{aligned} \text{maximize}_x \quad & \sum_{i,j\in S} w_{ij}x_i x_j + \sum_{i\in S, j\notin S} w_{ij}x_i\tilde{x}_j + \sum_{i\in S} a_i x_i \\ \text{subject to} \quad & x_i - k_i \prod_{\substack{j\in\Gamma_i \\ j\in S}} (1 - p_{ij}\phi_{j,t-1}x_j) \leqslant b_i \quad \forall i \in V, \\ & x_i \in \{0,1\} \quad \forall i \in V, \end{aligned} \tag{3.3}$$

where $\tilde{x}_j$ is a fixed solution for vertex $j \notin S$,

$$k_i = \prod_{j\in\Gamma_i, j\notin S} (1 - p_{ij}\phi_{j,t-1}\tilde{x}_j),$$

and a_i are the accumulated edges in node i at the current level (similar to A_I, with $a_i = 0$ for all i at the finest level). We realized this by traversing all vertices and randomly fixing a small subset of neighbours S to solve (3.3) around each vertex. The localized refinement was solved with external optimization toolkit described in Appendix A.5.

4. Computational results

We evaluate our method on a set of small networks with known optimal solutions, two case studies (HIV spread and cyber infrastructure networks), and one large-scale data set. The two case study networks are typical complex network instances on which solving this particular optimization is of great practical importance. The connection between epidemiological models and analysis of cyber attacks has been extensively investigated during the past two decades [9,10,26]. The massive data set evaluation contains networks of different structures and sources, including some that arise in applications that are not related immediately to the response problem but can potentially represent hard structures for the method. In all experiments, we compared average results of optimization objectives for feasible solutions only, namely $\sum_{ij\in E} w_{ij}x_ix_j$. Each average was calculated over 50 evaluations with different random seeds.

We compare our computational results to those produced by a combination of different iterated local searches (ILSs). This was the only (linear) scalable method to produce the objectives that are comparable to those computed by the multilevel algorithm. We note that even if one would find some efficient and effective method of solving (2.4) we can always use it at the refinement stage of the framework and, thus, potentially improve it even more by inheriting its own solution from the coarse scales. One of the main goals of this work was to demonstrate the effectiveness of the multiscale method on this class of problems. The combination of different strategies in our ILS includes deterministic and randomized Gauss–Seidel-like point-wise relaxations, localized refinements of small subsets of nodes described in (3.3) and their heat-bath and simulated annealing versions. Switching between different types of ILSs almost always helped to escape local attraction basins (see example in Section 4.2).

We have implemented and tested the algorithm using standard C++, and LEDA libraries [27] on Linux 2.4 GHz machine. The implementation is non-parallel and has not been optimized. The results (objectives and running times) should only be considered qualitatively and can certainly be further improved by a more advanced implementation.

4.1 *Networks with known optimal solutions*

Before analysing the proposed method on large-scale instances that cannot be solved to prove optimality (even using commercial solvers) reasonably fast, we evaluated how good the results of the multiscale method are on small networks (up to 70 vertices and 350 edges) that can be solved exactly. Although these networks are too small to demonstrate the power of the multiscale approach, we can still create up to five levels in the hierarchy of the algorithm. For this purpose, we generated Erdös and Rényi [28], Barabasi and Albert [29] and R-MAT [30] networks (200 of each type) with randomly initiated w_{ij} and $\phi_{i,t-1}$ (see (2.4)). The results are demonstrated in Fig. 3. Typically in such settings almost half of the instances are optimally solved while others are close enough to the optimum (between 90 and 100%). Barabasi and Albert and R-MAT models are solved slightly better than Erdös and Rényi model.

4.2 *HIV spread network*

We demonstrate our algorithm on a network created from data collected by Potterat *et al.* [31] related to the HIV spread over individuals who were in contact through sex or injection drug use. The original data

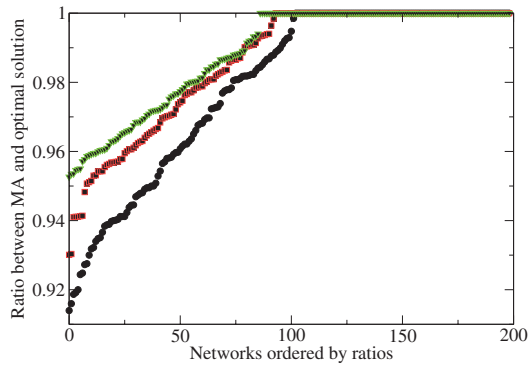

FIG. 3. Comparison with optimal solutions for 200 small networks. Each point represents a ratio between the objectives of MA and optimal solutions, respectively, for one network. Solutions of MA are feasible. Circles, squares, and triangles correspond to Erdös-Rényi, R-MAT, and Barabasi-Albert models, respectively.

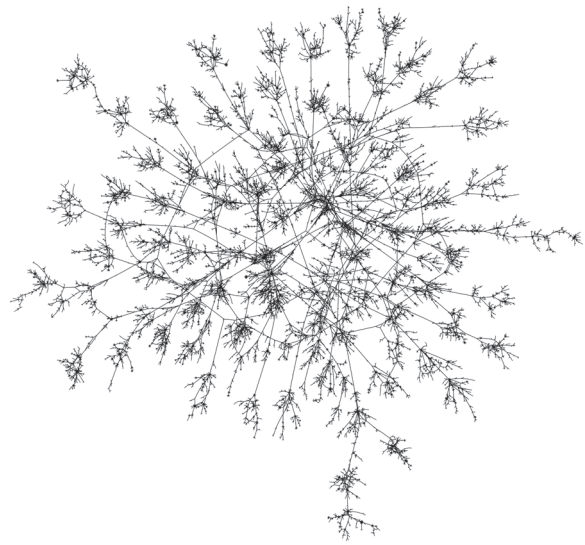

FIG. 4. Infection spread network ($|V| = 25{,}090$, $|E| = 28{,}284$) constructed by sparse random connections among 100 generated networks that are similar to real HIV spread data.

contains a network with 250 vertices, where each vertex corresponds to an individual. We generated 100 similar networks by using a multiscale network generator [32] and connected them by several random edges in order to create one big network (see Fig. 4). We simulated an immediate outbreak of the infection in which initially 5% of vertices were associated with high level of infection ($\phi_i \in [0.8, 1]$) and each edge had the same rate of infection transmission. Then five iterations of the infection spread were performed; at each iteration, all vertices released their infection to the neighbours, and the updated ϕ was

$$\forall i \in V \quad \phi_i^{\text{new}} = \min\left(1, \phi_i^{\text{old}} + \sum_{j \in \Gamma_i} \frac{p_{ij}}{\sum_{k \in \Gamma_i} p_{ik}} \phi_j^{\text{old}}\right).$$

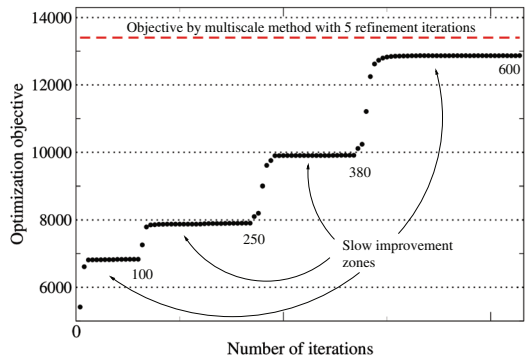

FIG. 5. Computational results on the infection spread network. Each point corresponds to the feasible solution of ILS. The dashed line corresponds to the objective found by MA.

Typical computational results comparing the multiscale algorithm (MA) and a combination of different types of ILS with restarts are presented in Fig. 5. We experimented with several state of the art algorithms implemented in and linked to MINOTAUR optimization toolkit [33]. We found that no single algorithm (other than MA) was able to provide better result than ILS (in a reasonably comparable time) on large-scale scale instances.

The MA reached the objective 13,404 in just five iterations of the refinement, while ILS was able to achieve the objective 12,870 being more than 100 times slower than MA. We note that the content of the two solutions was different. The number of vertices suggested to consider as 'closed' by MA (8159) was bigger than those chosen by ILS (7864). In contrast to ILS, MA left 'open' more high-degree vertices. We observe that introducing the linear penalty term $-\sum_{i\in V}\sum_{j\in\Gamma_i} w_{ij}x_i$ to the objective of (2.4) may reverse this situation in favour of closing more high-degree vertices. Such a term can be coarsened similarly to the aggregated edge coefficients A_i in (3.2).

4.3 *Peer-to-peer network*

Peer-to-peer systems (P2Ps) are a type of technology for collaborative environments in which each participating computer can play roles of both server and client. At the core of such systems lies an infrastructure for sharing computational resources such as storage space and CPU time. Data streams in such networks are often associated with mutually anonymous (for users) source and target vertices which brings the realization of a strong cyber security system to one of the system's central issues. Examples of P2P systems include Napster, Gnutella and SETIHome. Altunay *et al.* analysed one such system [3], namely the Open Science Grid, and proposed an optimization model for manipulating collaboration policies to prevent the fast spread of cyber attacks. Unfortunately, methods proposed in [3] are too slow for large-scale networks.

We evaluated our method on the biggest connected component of the Gnutella P2P network [34,35]. As in the previous case, we compared our results with those of ILS. We observed that ILS rapidly reaches slow improvement zones; however, in contrast to the previous case there was a significant gap in the objective between MA and ILS on this type of network, and thus the evaluation consists of 30 trials with different random initial seeds. The results are shown in Fig. 6. Each bar corresponds to the ratio between MA and ILS objectives for one initial random seed. The difference in running time is

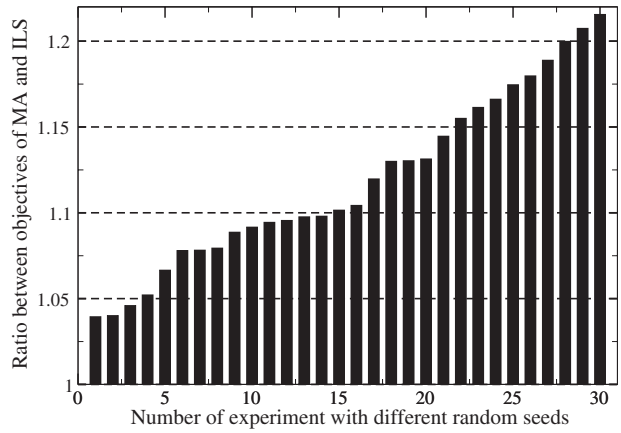

FIG. 6. Computational results on the Gnutella P2P network. Each bar represents a ratio between the objectives of MA and ILS solutions, respectively, for one network. Solutions of MA and ILS are feasible.

similar to the previous case (between 100 and 200 times) in favour of MA. In addition, MA typically finds a better solution, as shown in Fig. 6. The difference in solution quality can be as much as 20%.

4.4 *Massive simulations*

We also demonstrated the robustness of the proposed method on a test set of 100 large-scale graphs (up to $|V| + |E| \approx 10M$) taken from different sources such as [34,36,37]. Most of the graphs can be downloaded at [4]. In contrast to HIV and P2P networks, the connection of many of these graphs to the infection spread response problem is not straightforward (if at all), but their structural complexity presents a particular difficulty for optimization methods. The results of comparison with ILS are presented in Fig. 7, where each point corresponds to the ratio between objectives of MA and ILS.

For approximately one-third of the test set, the difference in the objective is practically significant (more than 10%) while the running time of MA is still between 50 to 200 times faster. The difference in running time depends mostly on the size of refinement subproblems (3.3) because in many cases the external solvers such as [33] that ensure upper bounds are not of linear complexity.

We note that, according to the results, the most difficult for ILS instances are networks with high average degree. The biggest difference detected between MA and ILS ratios was 132 for a graph with average degree 240. We generated 100 graphs by high-entropic multiscale editing [32] and confirmed that MA still improves the objective over ILS with a factor between 70 and 150 for all of them.

4.5 *High-degree nodes immunization*

Immunization of high-degree nodes (in the situations when the amount of resources is limited) is one of the well-known target immunization strategies. In contrast to computationally difficult epidemic threshold-based methods [10] in which the largest eigenvalue of the adjacency matrix has to be decreased, the target strategies (that are based on the degree and other similar centrality metrics) are usually much faster. We experimented with the degree-based target immunization strategy in two settings.

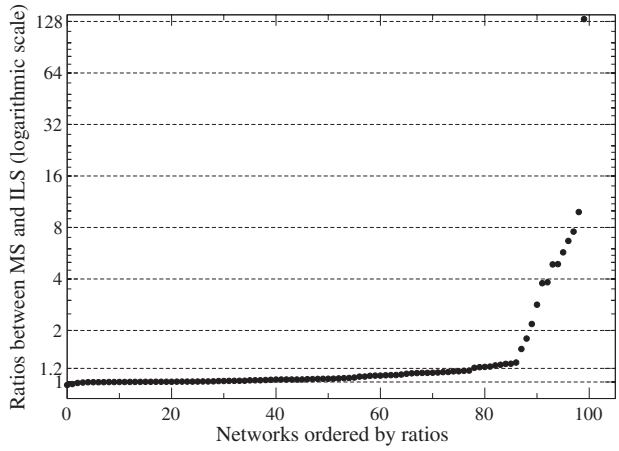

FIG. 7. Large-scale data experiments on various graph structures. For a better visualization, the networks are ordered by the ratios of resulting objectives. Each point represents a ratio between the objectives of MA and ILS solutions, respectively, for one network. Solutions of MA and ILS are feasible.

In the first setting (HDAlg) the algorithm included initial assignment $x_i = 1$ for all $i \in V$ with one pass over all nodes in the order of decreasing degree. At each step of the pass if node i did not satisfy the constraint in (2.4) the node was closed ($x_i \leftarrow 0$). The computed objectives of this algorithm were at least twice (and usually even more) worse than those obtained by MA. In the second setting the results of HDAlg were used as initial assignment for ILS. However, no significant differences in the objective in comparison to other initializations of the ILS were observed.

5. Conclusions

We propose a fast multiscale method for optimizing the response policies to infection spread in large-scale, complex weighted networks. The method is flexible and can be easily adapted for different changes in the model formulation such as changing the model to *link-based* immunization [3,38] and adding penalty function to the objective and new constraints. Similar to many methods in the large family of MAs, our approach is scalable and suitable for parallelization on HPC systems.

As key future work directions we identify two branches: theoretical and applied. Theoretical work involves rigorous analysis in order to identify upper and lower bounds. In the applied branch we suggest to introduce similar optimization problems for the SIR model, where the 'recovered' states of vertices will be introduced and PDE-based constraints will describe time series of the data. Other interesting prospective directions include an application of the multiscale strategy to minimize the expected number of infections in the network and to delay the epidemic peak.

Acknowledgements

We thank two anonymous referees for providing constructive comments and help in improving the paper. The submitted manuscript has been created in part by UChicago Argonne, LLC, Operator of Argonne National Laboratory ('Argonne'). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said

Funding

This work is supported in part by the U.S. Department of Energy, Basic Energy Sciences, Office of Science, under contract DE-AC02-06CH11357.

References

1. Newman, M. (2010) *Networks: An Introduction*. New York: Oxford University Press.
2. Eubank, S., Guclu, H., Kumar, V., Marathe, M., Srinivasan, A., Toroczkai, Z. & Wang, N. (2004) Modelling disease outbreaks in realistic urban social networks. *Nature*, **429**, 180–184.
3. Altunay, M., Leyffer, S., Linderoth, J. T. & Xie, Z. (2011) Optimal response to attacks on the Open Science Grid. *Comput. Netw.*, **55**, 61–73.
4. Safro, I. (2012) Fast solver for optimal response to infection spread. http://www.cs.clemson.edu/isafro/optresp/.
5. Brandt, A. & Ron, D. (2003) Chapter 1: multigrid solvers and multilevel optimization strategies. *Multilevel Optimization and VLSICAD* (J. Cong & J. R. Shinnerl eds). Springer.
6. Goldberg, N., Leyffer, S. & Safro, I. (2012) Optimal response to epidemics and cyber attacks in networks. *Technical Report ANL/MCS-1992-0112*. Argonne National Laboratory.
7. Keeling, M. J. & Eames, K. T. D. (2005) Networks and epidemic models. *J. R. Soc. Interface*, **2**, 295.
8. Kermack, W. & Mckendrick, A. (1927) A contribution to the mathematical theory of epidemics. *Proc. R. Soc. Lond. A*, **115**, 700–721.
9. Kephart, J. O. & White, S. R. (1993) Measuring and modeling computer virus prevalence. *Proceedings of the 1993 IEEE Symposium on Security and Privacy*, SP '93, Washington, DC, USA: IEEE Computer Society, pp. 2–14.
10. Chakabarti, D., Wang, Y., Wang, C., Leskovec, J. & Faloutsos, C. (2008) Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Security*, **10**, 1–26.
11. Sahinidis, N. V. (1996) BARON: a general purpose global optimization software package. *J. Global Optim.*, **8**, 201–205.
12. Belotti, P. (2009) Couenne: a user's manual. *Technical Report*. Lehigh University.
13. Brandt, A. (2001) Multiscale scientific computation: review 2001. *Multiscale and Multiresolution methods (Proceeding of the Yosemite Educational Symposium, October 2000)* (T. Barth, R. Haimes & T. Chan eds). Springer.
14. ren Fang, H., Sakellaridi, S. & Saad, Y. (2010) Multilevel manifold learning with application to spectral clustering. *CIKM* (J. Huang, N. Koudas, G. J. F. Jones, X. Wu, K. Collins-Thompson & A. An eds). New York: ACM, pp. 419–428.
15. Serrano, M. Á., Bogu ná, M. & Vespignani, A. (2009) Extracting the multiscale backbone of complex weighted networks. *Proc. Natl Acad. Sci. USA*, **106**, 6483–6488.
16. Karypis, G. & Kumar, V. (1998) A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, **20**, 359–392.
17. Dhillon, I., Guan, Y. & Kulis, B. (2005) A fast kernel-based multilevel algorithm for graph clustering. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD'05. New York: ACM, pp. 629–634.
18. Sharon, E., Galun, M., Sharon, D., Basri, R. & Brandt, A. (2006) Hierarchy and adaptivity in segmenting visual scenes. *Nature*, **442**, 810–813.
19. Nam, G.-J. & Cong, J. (2007) *Modern Circuit Placement: Best Practices and Results*. Springer.
20. Ron, D., Safro, I. & Brandt, A. (2011) Relaxation-based coarsening and multiscale graph organization. *Multiscale Model. Simul.*, **9**, 407–423.

21. Carlson, J. & Doyle, J. (2002) Complexity and robustness. *Proc. Natl Acad. Sci. USA*, **99**(Suppl 1), 2538.
22. Itzkovitz, S., Levitt, R., Kashtan, N., Milo, R., Itzkovitz, M. & Alon, U. (2005) Coarse-graining and self-dissimilarity of complex networks. *Phys. Rev. E*, **71**, 016127.
23. Wolpert, D. & Macready, W. (2007) Using self-dissimilarity to quantify complexity. *Complexity*, **12**, 77–85.
24. Chen, J. & Safro, I. (2011) Algebraic distance on graphs. *SIAM J. Scientific Computing*, **33**, 3468–3490.
25. Brandt, A., Brannick, J. J., Kahl, K. & Livshits, I. (2011) Bootstrap AMG. *SIAM J. Scientific Comput.*, **33**, 612–632.
26. Kephart, J. O., Sorkin, G. B., Arnold, W. C., Chess, D. M., Tesauro, G. & White, S. R. (1995) Biologically inspired defenses against computer viruses. *IJCAI (1)*, pp. 985–996.
27. Mehlhorn, K. & Näher, S. (1995) LEDA: a platform for combinatorial and geometric computing. *Commun. ACM*, **38**, 96–102.
28. Erdos, P. & Renyi, A. (1960) On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, **5**, 17–61.
29. Barabási, A.-L. & Albert, R. (1999) Emergence of scaling in random networks. *Science*, **286**, 509–512.
30. Chakrabarti, D., Zhan, Y. & Faloutsos, C. (2004) R-MAT: a recursive model for graph mining. *SDM* (M. W. Berry, U. Dayal, C. Kamath & D. B. Skillicorn eds). Philadelphia: SIAM.
31. Potterat, J. J., Phillips-Plummer, L., Muth, S. Q., Rothenberg, R. B., Woodhouse, D. E., Maldonado-Long, T. S., Zimmerman, H. P. & Muth, J. B. (2002) Risk network structure in the early epidemic phase of HIV transmission in Colorado Springs. *Sex Transmit Infect*, **78**(Suppl I), 159–163.
32. Gutfraind, A., Meyers, L. A. & Safro, I. (2012) Multiscale network generation. *CoRR*. abs/1207.4266.
33. Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A. & Munson, T. (2012) MINOTAUR, a toolkit for solving mixed-integer nonlinear optimization problems. http://wiki.mcs.anl.gov/minotaur/index.php/Main_Page.
34. Leskovec, J. (2012) Stanford Network Analysis Package (SNAP). http://snap.stanford.edu/index.html.
35. Ripeanu, M., Foster, I. & Iamnitchi, A. (2002) Mapping the Gnutella network: properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Comput. J.*, **6**, 2002.
36. Cong, J. (2012) Optimality Study Project. http://cadlab.cs.ucla.edu/pubbench/.
37. Davis, T. (1997) University of Florida sparse matrix collection. *NA Digest*, **97**.
38. Bishop, A. & Shames, I. (2011) Link operations for slowing the spread of disease in complex networks. *EPL (Europhysics Letters)*, **95**, 180–185.
39. Safro, I., Sanders, P. & Schulz, C. (2012) Advanced Coarsening schemes for graph partitioning. *SEA* (R. Klasing ed.). Lecture Notes in Computer Science 7276. Springer, pp. 369–380.
40. Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F. & Sawaya, N. (2008) An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optim.*, **5**, 186–204.

Appendix: Technical details

For complete understanding of the coarsening process, we recommend the reader to begin with [20].

A.1 *Algebraic distance*

The algebraic distance can be based on different types of stationary iterative relaxations such as Gauss–Seidel and Jacobi [24]. We experimented with the Jacobi overrelaxation iterator $H = (1 - \omega)I + \omega D^{-1}W$ in order to validate the ability to make the implementation fully parallel (in contrast to Gauss–Seidel which is difficult to parallelize). Here I is a diagonal matrix with ones on the diagonal, W is a weighted adjacency matrix with entries w_{ij} and D is a diagonal matrix with entries $d_{ii} = \sum_{j \in V} w_{ij}$.

In general, the Jacobi overrelaxation converges on graph Laplacians $L = D - W$ which allows to take into account long-range edges. However, since we work in the multiscale framework, we postpone capturing long-range distances to coarse scales and allow the overrelaxation to work for only a very limited number of iterations based on Theorem 8 in [24] that ensures that the relaxed values $\chi^{(k,r)}$ are stabilized quickly and no significant change is expected between two iterations after sufficiently small number of iterations.

A.2 *F–C coupling and restriction operator*

The selection of vertices to C is done by traversing all vertices starting with those that are highly infected. Similar to other multigrid methods, we observed that an exact sorting of nodes by their corresponding ϕ_i values does not play an important role. Results of the same quality can be obtained by using rough sort with buckets. In our experiments (with networks up to 10M nodes) the influence of an exact sort to the algorithm's running time was negligible.

The set C is formed as follows. Initially we set $F = V_f$ and $C = \emptyset$. Then the vertex is added to C if

$$\sum_{j \in C} \rho_{ij}^{-1} \Big/ \sum_{j \in V_f} \rho_{ij}^{-1} \geqslant \Theta.$$

The default value for the parameter Θ is 0.5. Increasing it will usually lead to slower coarsening and potentially better results, as more scales are created during the coarsening and more refinement is done. Decreasing Θ will lead to the opposite effects. However, unless one fixes extremely small value for Θ no significant change will be observed.

Similar to many multigrid methods, our coarsening depends on several random factors. We experimented with series of tests in which each setting was solved 50 times. The standard deviations of the results were small. Same results were confirmed in many AMG methods for combinatorial optimization problems [5,20,39]. Moreover, we can state that our method is 'more deterministic' than the AMG algorithms that are based on the real edge weights. This is because in the real-world problems the distribution of edge weights can be less wide than the distribution of algebraic distances (not to mention unweighted graphs).

A.3 *Aggregated variables and constants*

Coarse vertices represent small clusters of fine vertices and, thus, edge weights between coarse vertices are

$$W_{IJ} = \sum_{k \neq l} R_{kI} w_{kl} R_{lJ},$$

where R is an operator of restriction reinforced by the algebraic distance (see Equation (3.7) and Algorithm 3 in [20]). In order to reduce the complexity of the coarse problems, the number of non-zeros in rows of R has to be bounded by a sufficiently small number. In the experiments we determined that one non-zero entry with the strongest algebraic distance coupling [20] is enough for practical purposes. Coefficients P_{IJ} can be derived similarly if p_{ij} are not inverses of w_{ij}, namely,

$$P'_{IJ} = \sum_{k \neq l} R_{kI} p_{kl} R_{lJ} \quad \text{and} \quad P_{IJ} = P'_{IJ} \Big/ \sum_{K \in \Gamma_J} P'_{KJ}.$$

In the presented experimental settings p_{ij} are normalized inverses of the given connection strengths. We also experimented with random p_{ij}. The results were not principally different than the presented ones.

Coefficients A_I in (3.2) accumulate weights of edges w_{ij} whose endpoints are aggregated into I, namely,

$$A_I = \sum_{i,j \in I} w_{ij}.$$

Aggregation of scalars $\phi_{j,t-1}$ and b_I into $\Phi_{J,t-1}$ and B_I, respectively, has to be done according to the application, because in some situations the probability of infection in coarse vertex J may not depend linearly on those in fine-level vertex. In our simplified models, they are accumulated from the corresponding fine-level vertices as their restricted contribution to the aggregates. For example,

$$\Phi'_{J,t-1} = \sum_{j \in J} R_{jJ} \phi_{j,t-1} \quad \text{and} \quad \Phi_{J,t-1} = \Phi'_{J,t-1} / \max_J(\Phi'_{J,t-1}),$$

where the sum runs over all fine nodes j that are aggregated into J, and R_{jJ} are the corresponding interpolation weights. In our experiments, each fine node was entirely attached to some seed, i.e. $R_{jJ} = 1$. However, any construction of R with higher interpolation order is possible. We experimented with higher orders of interpolation, namely 2 and 3, and found no significant difference in the results.

A.4 *Interpolation*

Algorithm A.4 describes the stage of interpolation given a solution for coarse level. It consists of initial assignment of all seeds at the current level with the solutions of the respective coarse variables (lines 1–3) and one pass over the fine nodes with attempts to leave them working ($x_i = 1$). If either ith or one of its neighbour's constraints are not satisfied (line 7) then we close node i ($x_i = 0$).

Interpolation

1: **for all** $i \in C$ **do**
2: $x_i \leftarrow X_I$, where X_I is a corresponding aggregate seeded by i
3: **end for**
4: **for all** $i \in F$ **do**
5: $\Gamma'(i) \leftarrow \{j \in V \mid x_j \neq -1\}$
6: $x_i \leftarrow 1$
7: **if** $x_i - \prod_{j \in \Gamma'_i} \left(1 - p_{ij}\phi_j x_j\right) > b_i$ or $\exists j \in \Gamma'_i$ s.t. $x_j - \prod_{k \in \Gamma'_j} \left(1 - p_{jk}\phi_k x_k\right) > b_j$ **then**
8: $x_i \leftarrow 0$
9: **end if**
10: **end for**

A.5 *External optimization solver*

The recently developed mixed-integer non-linear optimization toolkit MINOTAUR [33] has proved itself as particularly suitable for such problems. MINOTAUR compares favourably with other state-of-the-art MINLP solvers such as BONMIN [40]. We used MINOTAUR as both a coarsest level and

local-processing solver for the problems in (3.3). The sizes of the coarsest level and local processing problems were 40 and 15 variables, respectively. During our experiments, all small problems of the type (3.3) were solved exactly. The number of sweeps for solving (3.3) for all nodes and the respective small neighbourhoods was 3 in MA.

A.6 *Refinement and relaxation*

In our problem, the Gauss–Seidel-like relaxation is a point-wise optimization process in which the contribution of each node to the total objective is maximized. Its complexity is significantly better than that of the localized refinement defined in (3.3). The difference between them depends on the sizes of subsets S in (3.3) and on the external solver. We found that the best results can be achieved by applying both relaxation and localized refinement. If the running time of the algorithm is critical, one can omit the localized refinement as the most time-consuming component of the framework. In this case, we observed worsening of final results by 7% on the average but the running time is improved by factor 10.