Simulation-Optimization Approaches for the Network Immunization Problem with Quarantining

Hoogervorst, Rowan; van der Hurk, Evelien; Pisinger, David

Published in:
arXiv physics e-prints

Publication date:
2025

Document Version
Early version, also known as pre-print

Simulation-Optimization Approaches for the Network Immunization Problem with Quarantining

Rowan Hoogervorst[1], Evelien van der Hurk[1], and David Pisinger[1]

[1]Department of Technology, Management and Economics, Technical University of Denmark, Akademivej, Building 358, Kongens Lyngby, 2800, Denmark

Abstract

Vaccination has played an important role in preventing the spread of infectious diseases. However, the limited availability of vaccines and personnel at the roll-out of a new vaccine, as well as the costs of vaccination campaigns, might limit how many people can be vaccinated. Network immunization thus focuses on selecting a fixed-size subset of individuals to vaccinate so as to minimize the disease spread. In this paper, we consider simulation-optimization approaches for this selection problem. Here, the simulation of disease spread in an activity-based contact graph allows us to consider the effect of contact tracing and a limited willingness to test and quarantine. First, we develop a stochastic programming algorithm based on sampling infection forests from the simulation. Second, we propose a genetic algorithm that is tailored to the immunization problem and combines simulation runs of different sizes to balance the time needed to find promising solutions with the uncertainty resulting from simulation. Both approaches are tested on data from a major university in Denmark and disease characteristics representing those of COVID-19. Our results show that the proposed methods are competitive with a large number of centrality-based measures over a range of disease parameters and that the proposed methods are able to outperform them for a considerable number of these instances. Finally, we compare network immunization against our previously proposed approach of limiting distinct contacts. Although, independently, network immunization has a larger impact in reducing disease spread, we show that the combination of both methods reduces the disease spread even further.

Keywords— Network Immunization, Targeted Immunization, Simulation, Stochastic Programming, Sample Average Approximation, Genetic Algorithm

1 Introduction

Immunization of a population through vaccination has been shown to play a vital role in reducing the spread of infectious diseases. Examples include the vaccination campaigns against smallpox,

influenza, and, more recently, COVID-19. The latter has, e.g., reduced the health risks for individuals [Tartof et al., 2021, Vasileiou et al., 2021, McNamara et al., 2022] and allowed for a greater degree of opening up society [Bauer et al., 2021, Olivera Mesa et al., 2022]. However, immunization is often costly due to the cost of acquiring and administering vaccines. Moreover, time constraints and limited availability of vaccines during the roll-out of a new vaccine often make it necessary to prioritize some individuals for vaccination. Therefore, finding efficient immunization strategies that target the most influential individuals and achieve the greatest reduction in disease burden has shown to be an important topic of research.

In this paper, we look at the immunization of a population that is represented through an activity-based contact hypergraph. Each individual is represented by a node in this graph, and hyperarcs represent planned activities that individuals are involved in over time. These activities could, e.g., be school classes, sports classes, or work meetings. During these activities, contacts occur between those individuals partaking in the activity, i.e., between some of the individuals that are part of the same hyperarc, allowing the disease to spread. The network immunization problem then focuses on selecting nodes from the graph to immunize, given a budget on how many individuals can be immunized. Our aim is hereby to limit the spread of the disease and thus to minimize the number of individuals that are infected over a given time horizon.

Compared to the existing literature on network immunization, we consider a richer disease spreading model that considers the quarantining of infected individuals and the quarantining of exposed contacts as a result of contact tracing. Moreover, we take into account a limited willingness to test and quarantine, corresponding to the fact that not all individuals will opt to get tested and quarantine after becoming infected or being informed of an infected contact. To measure the disease spread over the network under this disease spreading model with contact tracing, we rely on a simulation approach instead of direct graph measures. In particular, we use a simulation approach similar to the one in Bagger et al. [2022], which we integrate into simulation-optimization methods for the network immunization problem.

We consider an application focusing on the case of higher education for one of Denmark's largest universities, using data that was introduced in Bagger et al. [2022]. Here, the population consists of students who would like to attend sessions for the classes they have subscribed to. The activities considered are thus scheduled course classes in which students meet and during which the disease can spread. Immunization in our setting corresponds to offering vaccination to individuals, representing a setting in which it would be possible to offer vaccines on an individual basis to a given number of students at the university.

The contributions of this paper are fourfold. First, we formally define the network immunization problem with quarantining and contact tracing and describe how simulation can be used to determine the disease spread. Second, we propose a stochastic programming approach based on sampling infection forests from the simulation model. Third, we propose a parallelized genetic algorithm to solve the problem, which extensively makes use of the graph characteristics to find efficient solutions and combines small and large simulation runs. Fourth, we perform an extensive numerical study in which we show that our proposed methods are competitive with a large number of existing centrality measures and show the benefits of network immunization for our DTU application by comparing and contrasting to a scheduling policy that minimizes the number of distinct contacts [Bagger et al., 2022].

The paper is organized as follows. In Section 2, we discuss the relevant literature. In Section 3, we introduce the disease spreading model and formally define the considered network immunization

problem. We propose both a stochastic programming approach and a genetic algorithm to solve the problem in Section 4. The data that we use from a major university in Denmark is described and analyzed in Section 5. We discuss the results obtained by our network immunization approach for this application in Section 6, where we both benchmark the proposed algorithms and evaluate the extent to which immunization can reduce the disease spread. Lastly, the paper is concluded in Section 7.

2 Literature Review

The study of network immunization policies for limiting disease spread is part of a larger stream of research on vaccine allocation, which considers the allocation of vaccines over, e.g., geographical, age, and social groups. See, e.g., Medlock and Galvani [2009], Enayati and Özaltın [2020], and Liu and Lou [2022], who focus on the allocation of influenza and COVID-19 vaccines to different groups in the population, respectively. Network immunization problems characterize themselves by considering detailed contact networks, accounting for the fact that diseases tend to spread differently on realistic contact networks than in random graphs [Pastor-Satorras and Vespignani, 2001, Newman, 2002]. Moreover, network immunization generally focuses on immunizing influential individuals in the population rather than on targeting groups as a whole based on, e.g., age or social characteristics. It should be noted that network immunization problems can also be found in other application areas, such as when looking at the spread of computer viruses [Gao et al., 2011] and the spread of harmful information in social networks [Peng et al., 2019].

Traditionally, network immunization has focused on sequentially eliminating nodes from a network by ranking the nodes and choosing those nodes with the highest rank [Pastor-Satorras and Vespignani, 2002]. The centrality of a node is a commonly used indicator of its importance in the network, and different types of centrality measures have been proposed. For example, *degree centrality* looks at the number of neighbors adjacent to a node, where the assumption is that nodes with more neighbors are more likely to spread a disease [Pastor-Satorras and Vespignani, 2002]. Another common centrality measure is *betweenness centrality*, which looks at the number of times a given node is on the shortest path between any other two nodes [Freeman, 1977, Anthonisse, 1971]. Other centrality concepts include those of *eigenvector centrality* [Bonacich, 1972] and *closeness centrality* [Freeman, 1978].

While the above measures are general indicators of a node's importance and not specific to preventing disease spread, an increasing number of papers are now focusing on measures that consider the specifics of disease spreading models. For example, it has been shown by Chakrabarti et al. [2008] that the epidemic threshold, i.e., the value below which the epidemic dies out, for a SIS-epidemiological model equals the inverse of the largest eigenvalue of the adjacency matrix of the network. Therefore, multiple papers focus on maximizing the eigenvalue drop that is achieved by removing a node, where Chen et al. [2016] suggest approximating the eigenvalue drop using a so-called shield value and Van Mieghem et al. [2011] suggest different heuristic strategies for selecting a node. Another approach to better consider the disease spread dynamics is taken by Piraveenan et al. [2013], who suggest explicitly taking the current health state of each node into account when ranking the nodes, a measure they refer to as percolation centrality.

Opposed to selecting nodes sequentially, which in general does not provide an optimal solution, authors have also looked at selecting the set of nodes to remove in an integrated way. Emmerich et al. [2020] use quadratic optimization to find the set of nodes to immunize that minimizes the

shield value and the cost of immunization. Saha et al. [2015] propose approximation algorithms for maximizing the eigenvalue drop by either removing nodes or edges from the graph. Moreover, Nandi and Medal [2016] propose algorithms for removing edges in a network to minimize the connectivity and, hence, disease spread in the graph. Another strategy used to minimize disease spread is to disconnect the graph into multiple connected components. For example, Schneider et al. [2011] propose an algorithm to minimize the size of the largest connected components over the duration of the immunization process. Moreover, Ventresca and Aleman [2014] propose a randomized rounding algorithm for finding the smallest possible subset of vertices to remove such that the graph is split into disconnected components of a given maximum cardinality.

Among the approaches that try to find the set of nodes in an integrated way, multiple use common metaheuristics from the mathematical programming literature. For example, Deng et al. [2016] use tabu search to minimize the size of the largest connected component of the graph after node removal. Especially relevant for this study, Maulana et al. [2017] propose a genetic algorithm to maximize the drop in the largest eigenvalue after node removal. They benchmark their method against the algorithm presented by Chen et al. [2016], which is based on shield value, and show that better results can be found by means of their algorithm.

Compared to the literature described above, we focus on a richer epidemiological model in this study. In particular, we focus on a SEIR epidemiological model and take into account the effect of contact tracing and the willingness of people to test and quarantine. As a result, we do not focus on a direct graph metric in assessing the objective during optimization but use simulation to assess the effect of immunization. To the best of the authors' knowledge, this is the first study to look at optimization approaches for network immunization in such a rich epidemiological setting that requires simulation.

3 Problem Description

To model the spread of a disease through a population, we consider an activity-based contact hypergraph $G = (V, H, T)$ like in Bagger et al. [2022]. Each node $v \in V$ in this graph represents an individual in the population that can be exposed to the disease. The hyperedges $h \in H$ indicate planned activities, thus connecting those individuals participating in the activity. Moreover, the set T gives the time periods during which these activities take place, where $H_t \subseteq H$ denotes those activities that take place at time $t \in T$. An example of this hypergraph structure is depicted in Figure 1, which shows 12 individuals represented by nodes and three planned activities represented by hyperarcs. Note, in particular, how each hyperarc connects all individuals participating in the activity.

We use a compartmental model to describe the current health state of each individual in the population, i.e., each node in the graph. In such a compartmental model, each individual is categorized to be in one of the compartments based on its current health state. We consider the following health states in our model:

- **S**: The individual is susceptible to the disease, i.e., can potentially become infected in the future.
- **E**: The individual has been exposed to the disease but is not yet infectious.
- **I**: The individual is infectious and can transmit the infection to other individuals.

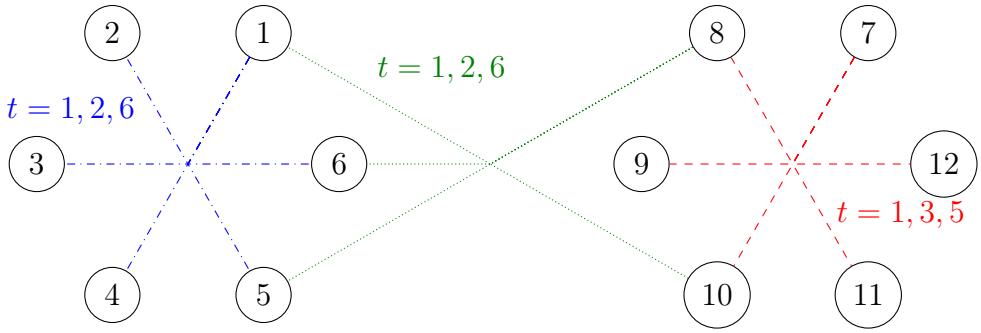


Figure 1: A visualization of the activity-based contact hypergraph for an example in which 12 individuals (nodes) attend three activities (blue/dash-dotted, green/dotted, and red/dashed hyperarcs) spanning six time periods. The periods in which a hyperarc is active are denoted next to the hyperarc.

- **R**: The individual has recovered from the disease.

Transitions between health states for each individual occur according to a discrete time Markov chain, where the probabilities to move between states are influenced by the contacts in the graph G. The possible transitions in our Markov chain are given in Figure 2. Here, an individual moves with probability β_t from being susceptible (S) to being exposed (E) in time period $t \in T$ or stays susceptible with probability $1 - \beta_t$. The value of β_t depends on the previous contacts that the individual had within the graph G. Individuals move with probability μ from state E to state I and with probability γ from state I to state R. Note that state R is an absorbing state, meaning that we assume that recovered individuals become immune to the disease. This assumption is motivated by the relatively short time period that we consider in our simulation.

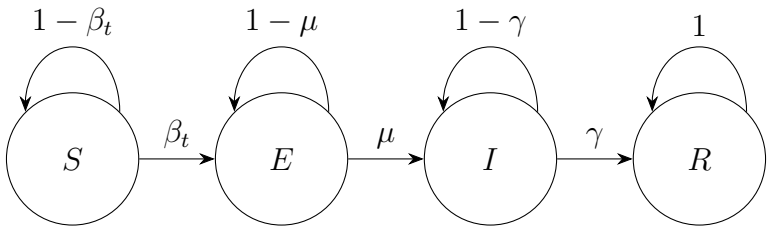


Figure 2: Transition rates for the discrete time Markov chain describing the health states of individuals at any timestep $t \in T$.

We assume that the probability β_t depends on two different components. First, infections can take place due to close contacts that the individual $v \in V$ had with infected individuals in the previous period, i.e., during the activities $\{h \in H_{t-1} \mid v \in h\}$. As such activities can be large in size, we assume that each individual has, on average, close contacts to N_{close} others in each activity. Each close contact with an infected individual then has a probability of β_{con} to spread the disease. Second, infection can occur due to contacts not explicitly modeled through the contact graph, i.e., contacts that occur outside of the modeled activities. In our application for university education, these spontaneous infections could, e.g., correspond to infections that occur outside of the university

setting and from random encounters at the university that we did not model. We assume that such spontaneous infection occurs with probability β_{spon}.

We also consider the effect of quarantining and contact tracing in the evaluation of the spread of the disease. Here, people go into quarantine after transitioning to the infectious state (I) with probability p^Q_{self}. Moreover, it is assumed that people are made aware of any close contacts they had with people who tested positive over the last $t^{max}_{trace} - t^{delay}_{notify}$ days. Here, t^{max}_{trace} represents the maximum number of days during which contacts are traced, while t^{delay}_{notify} represents the delay as a result of waiting to be tested, getting the result of the test, and communicating the result to close contacts. A close contact of an infectious person goes into quarantine with probability $p^Q_{neighbor}$, which incorporates both the willingness of individuals to quarantine after being informed as well as the likelihood of contact tracing being successful. Persons who are in quarantine are unable to spread the disease, corresponding to removing them from all activities, i.e., hyperarcs, that occur during the time periods in which they are quarantined.

Regarding the immunization of individuals, we assume that all selected individuals are immunized at the start of the time horizon. Moreover, we assume that immunization is fully immunity-inducing, meaning that immunized individuals are no longer susceptible to the disease. Considering these assumptions, immunized individuals will no longer play a role in the spread of the disease. While these assumptions seem strong in practice, they resemble that we look at relatively short time horizons of weeks to months. Moreover, it should be noted that our simulation approach is not dependent on these assumptions and can be easily extended to take both a partially effective vaccine and immunization throughout the time horizon into account.

We can now formally define the network immunization problem. Let $f(G, X)$ denote the number of infected individuals that is obtained for some hypergraph $G = (V, H, T)$ and a set of immunized nodes $X \subseteq V$ over all time periods T. This number will be estimated by means of simulation based on the above-discussed disease spreading model, where a set of initially infected nodes is chosen in such a way that each node $v \in V$ has a chance of β_{spon} of being infected at the start of the time horizon. Moreover, let k be the immunization budget, i.e., the number of nodes in the graph G that can be immunized. We can then state the problem as:

Definition 1 *(Network Immunization Problem) Select a set $X \subseteq V$ with $|X| \leq k$, such that the disease spread $f(G, X)$ is minimized.*

It should be noted that, as $f(G, X)$ is obtained using simulation, this is a *simulation-optimization* problem where there is uncertainty about the true objective value when choosing a set of nodes X to immunize.

4 Solution Methodology

In this section, we propose both a stochastic programming approach and a genetic algorithm for the Network Immunization Problem. We first describe the stochastic programming approach and afterwards describe the genetic algorithm.

4.1 Stochastic Programming Approach

The first solution approach that we consider for the Network Immunization Problem is based on stochastic programming. In particular, we apply sample average approximation, in which we sample

infection forests from our simulation. The main idea behind this method is that the sampled infection forests reflect the infection dynamics within the proposed disease spreading model and should thus give an accurate representation of the most influential nodes when the sample size is large enough. This can then be used to decide upon the nodes to immunize. Note that sample average approximation has been successfully used to solve a wide number of simulation-based optimization problems, see, e.g., Kim et al. [2015].

In our approach, we start by running a total of σ_p simulation runs in which none of the nodes in the population are assumed to be immunized. As we assume that recovered nodes cannot be infected again, each initial infection that is present at the start of the simulation and each spontaneous (outside) infection leads to a tree of further infected nodes. This means that the infections in each simulation run $i \in \{1, \ldots, \sigma_p\}$ are captured by a forest $\mathcal{F}_i = \{F_{i1}, \ldots, F_{im}\}$, as there can be multiple initial and outside infections. Here, $F_{i1}, \ldots, F_{im}$ are the trees in the forest, and m is the sum of the total number of initial and spontaneous infections in the simulation run. We will use $V_{\mathcal{F}_i}$ to denote the nodes contained in some forest $\mathcal{F}_i$.

As immunized nodes cannot be infected in our setting, they can also not pass on the infection to other individuals. Hence, given a fixed infection forest, immunizing a node in any of the trees in the infection forest corresponds to removing a sub-tree of nodes from the forest. This idea is illustrated in Figure 3, where the immunization of node 7 leads to the nodes below it no longer being infected and the shown blue (dashed) sub-tree being deleted from the forest. When a sample of infection forests is available, the immunization of a node will thus lead to a sub-tree being cut off in each forest $\mathcal{F}_i$ that contains that node, where the size of the cut-off sub-tree depends on the placement of the node in the respective tree.

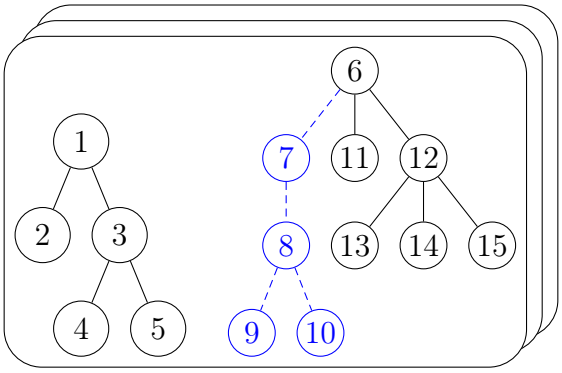


Figure 3: Illustration of the effect of immunizing node 7 on the infection forests

Our aim in the stochastic programming approach is now to minimize the number of infections that occur over the sampled infection forests. We solve this problem as an Integer Programming (IP) problem, where each variable x_v indicates whether individual $v \in V$ is immunized. Moreover, we use the auxiliary variables y_{vi} to denote whether individual $v \in V_{\mathcal{F}_i}$ is infected in forest $\mathcal{F}_i$ given the immunization decisions x. Let P_{vi} denote all the nodes that lie on the path from $v \in V$ to the root node of the tree in which v is contained in simulation run $i \in \{1, \ldots, \sigma_p\}$. We then obtain the

following IP formulation:

$$\min \sum_{i=1}^{\sigma_p} \frac{1}{\sigma_p} \left(\sum_{v \in V_{F_i}} y_{vi} \right) \tag{1}$$

$$\text{s.t.} \sum_{v \in V} x_v \leq k, \tag{2}$$

$$y_{vi} \geq 1 - \sum_{v' \in P_{vi}} x_{v'} - x_v \qquad \forall i \in \{1, \ldots, \sigma_p\}, v \in V_{\mathcal{F}_i}, \tag{3}$$

$$x_v \in \{0, 1\} \qquad \forall v \in V, \tag{4}$$

$$y_{vi} \in \{0, 1\} \qquad \forall i \in \{1, \ldots, \sigma_p\}, v \in V_{\mathcal{F}_i}. \tag{5}$$

The objective (1) minimizes the average number of infections that occur over all infection forests, where each scenario has equal probability. Constraints (2) ensure that the immunization budget is satisfied, i.e., that not too many individuals are immunized. Constraints (3) determine if a node is infected given the set of immunized nodes. Here, the variable y_{vi} can only become zero, i.e., v is not infected, if node v is immunized itself or if it is contained in a subtree of another node that is immunized. This corresponds precisely to what we earlier saw in Figure 3, where the immunization of a node leads to the deletion of the subtree in the infection forest below it. The remaining constraints (4) – (5) enforce the domain of the decision variables.

We solve formulation (1) – (5) using a commercially available IP solver. To speed up the solution process, we provide the solver with a starting solution that is determined based on the degree centrality measure [Pastor-Satorras and Vespignani, 2002]. Here, the k best variables are selected based on the degree centrality measure and the variables x_v and y_{vi} are set to the corresponding values in the starting solution.

It should be noted that the above method is only exact for a given set of infection forests from the simulation, as the spread of the disease in the simulation might change after nodes are immunized. This could, for example, mean that the nodes in a deleted subtree will still be infected by some other infectious individual when running the simulation with the immunized individuals. Moreover, quarantine may also impact the time moments during which an individual engages in activities, meaning that nodes might become exposed to the disease at different time points. Hence, this approach can be seen as a heuristic for the Network Immunization Problem.

4.2 Genetic Algorithm

We additionally developed a metaheuristic approach based on a *Genetic Algorithm (GA)* for the Network Immunization Problem. A GA is a metaheuristic inspired by evolution that mimics the process of natural selection by modifying a *population* of individual solutions [Sivanandam and Deepa, 2008]. In particular, a GA typically combines existing solutions through crossover to find new solutions (offspring) and incorporates mutation to create diversity in the solution population.

As we require simulation to evaluate the found solutions, it can take considerable time to evaluate the solutions in the solution population. Therefore, we developed an adapted GA framework in which we combine smaller and larger simulation runs. Here, the smaller simulations are used to quickly identify the most promising solutions from the solution population, while the larger simulation runs evaluate these solutions to get a better estimate of the true disease spread and thus

reduce simulation variance. By additionally running the small simulations in parallel, a significantly larger number of iterations can be executed in this way. Our GA framework is illustrated in Figure 4. In the remainder of this section, we explain the different components of our GA algorithm.

Representation of Solutions Consider the contact hypergraph $G = (V, H, T)$ and an immunization budget allowing for the vaccination of k nodes. Each solution I in our solution population then consists of k genes, each of which represents a node $v \in V$ selected for immunization in solution I. As nodes with (very) low centrality are unlikely to be good candidates for immunization, we limit the search space to nodes with a high centrality on at least one of several centrality measures. For each of these considered centrality measures, denoted by the set M, we calculate the centrality score for each node at the start of the algorithm. Let $V_m \subseteq V$ be the set of nodes selected according to centrality measure $m \in M$, where V_m contains the $t \leq N$ nodes with highest ranking on measure m. Each solution in the solution population is then of the form

$$I = \{v_1, v_2, \ldots, v_k\} \quad \text{where } v_i \in \bigcup_{m \in M} V_m, \tag{6}$$

meaning that only those genes are considered that are ranked among the t best nodes for at least one of the centrality measures.

Initial solution population In each iteration of the algorithm, we consider a solution population consisting of N solutions. At the start of the algorithm, an initial population is generated in which a few solutions are selected based on the considered centrality measures M, while the other solutions are randomly selected. Here, we add a solution for each centrality measure $m \in M$ and let the genes of this solution be the first k nodes in the ranking provided by that centrality measure. In this way, we ensure that there are solutions in the initial solution population that likely lead to a low disease spread. The remaining $N - |M|$ solutions are then chosen randomly from the search space $\cup_{m \in M} V_m$ to ensure a diverse initial solution population.

Fitness Function Each individual solution is evaluated based on a fitness score, which is computed using the proposed simulation model from Section 3. As we use simulation to estimate the number of infections in the SEIR model for a particular contact graph, we consider the average of all simulation runs. Therefore, the fitness score is

$$\mathit{fitness}(I) = \frac{\sum_{i=1}^{\sigma} CI(i)}{\sigma}, \tag{7}$$

where $CI(i)$ is the number of infections in simulation run $i \in \{1, \ldots, \sigma\}$. The number of simulation runs depends on the phase of the genetic algorithm, as illustrated in Figure 4. In particular, we use a smaller number of simulations σ_s in evaluating all solutions in the population to identify the λ most promising ones, for which a more accurate fitness score is then computed using σ_l simulation runs. Here, it holds that $\sigma_l \gg \sigma_s$. In conclusion, the objective in our GA is to find a solution $I^* \subseteq \bigcup_{m \in M} V_m$ that minimizes the expected number of infections, i.e.,

$$I^* = \underset{I \subseteq \bigcup_{m \in M} V_m, |I| = k}{\arg\min} \mathit{fitness}(I). \tag{8}$$

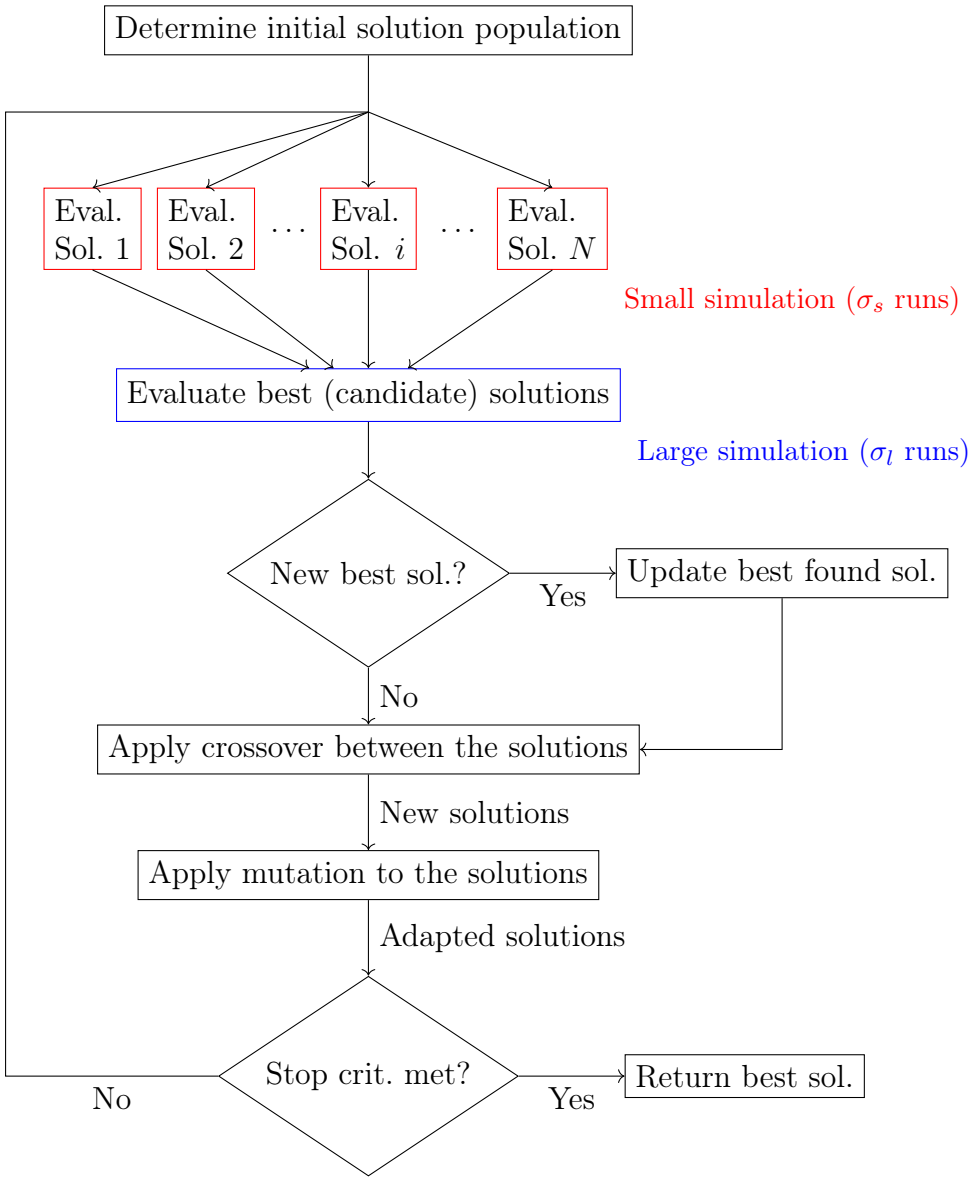


Figure 4: Illustration of our GA framework. Here, each solution in the solution population is first evaluated in parallel using smaller simulation runs in each iteration, after which the most promising solutions are evaluated using a larger simulation run. Moreover, crossover and mutation are applied to obtain the solution population for the next iteration.

Selection The selection process includes elitist selection, where the best ϵ individual solutions from the current generation are directly moved to the next generation without crossover or mutation. This ensures that good solutions remain present throughout the search process. The remaining $N-\epsilon$ solutions are generated by breeding pairs of solutions from the current generation, chosen through tournament selection. In such a tournament selection, τ solutions are randomly selected from the solution population and the one among them with the best fitness score is chosen.

Crossover and Mutation In each iteration, every pair of parent solutions produces two children solutions. The mating process uses an adaptation of uniform crossover. Here, genes that are included in both parents are first assigned to be part of both children. The other nodes for each child are then uniformly selected from the remaining genes of each parent, meaning that each gene has an equal chance of ending in any of the two children. Therefore, the approach prevents duplicates, i.e., a child cannot have two of the same genes.

To ensure genetic diversity, mutation occurs during each generation. Here, each gene is mutated with probability ρ, leading to ρk genes, on average, being mutated per solution in each iteration. When a gene is mutated, a random node is selected uniformly from the set $\cup_{m \in M} V_m$ in such a way that the new gene is not already present in the solution.

Stopping Criterion We use a time-based stopping criterion for the genetic algorithm, meaning that the genetic algorithm is continued until a certain wall clock time limit is reached. Note that this implies that the number of iterations will depend on the instance and computing infrastructure.

5 Data

To test the proposed methods, we use course data from the *Technical University of Denmark (DTU)*, one of the 8 major universities in Denmark. This dataset was first introduced by Bagger et al. [2022]. The course data describes the classes that students have subscribed to for the fall semester of 2020 and the teaching sessions, such as lectures and exercise classes, that have been planned for these classes. In total, the data describes the preferences of over 9500 individual students who subscribed to about 650 courses. On average, each student takes 3–4 classes, leading to about 34500 individual course subscriptions. Moreover, most courses have one or more sessions each week, meaning that the total number of contacts over the whole 13-week semester is higher.

The course subscriptions of students lead to a contact hypergraph G, where each hyperarc represents a course session being attended by a certain group of students. As each hyperarc connects all individuals in the activity, each hyperarc can also be represented as a complete sub-graph connecting these individuals, which provides a regular graph G'. In this graph, an arc is present between any two students if they are participating together in at least one course session, meaning that we aggregate over the different time periods T. This graph G' is depicted in Figure 5, where nodes are colored according to the number of contacts they have. Moreover, summary statistics for this graph G' are given in the upper part of Table 1.

The results in Table 1 show that the average degree is large, which is explained by the fact that students are connected with all other students in the classes they attend. Moreover, both the diameter and the average shortest path length are low, indicating that a disease can generally spread rapidly in the network. Together, these properties show that the studied network exhibits

many properties of a small world [Watts and Strogatz, 1998]. It should be noted, though, that the connectivity in the simulation is limited by the number of close contacts per activity N_{close}. However, as course sessions are generally repeated on a weekly basis, students may still become a close contact to many of the other students in the class over the semester as they change seating over these different sessions.

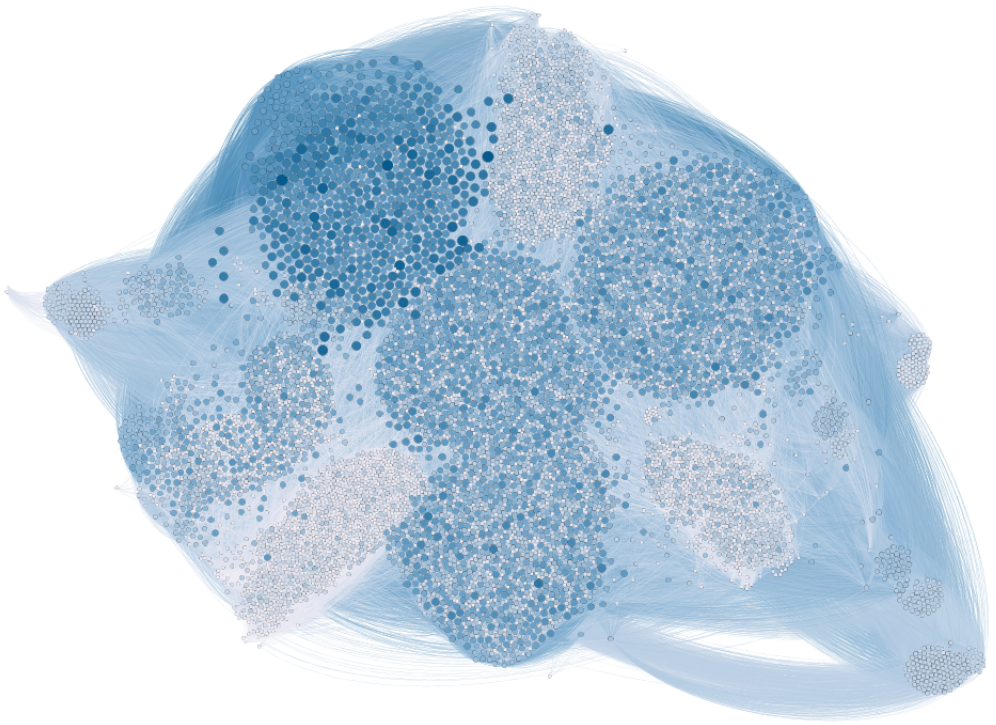


Figure 5: Visualization of the DTU contact graph G', where each node indicates a student and an arc connects two students if they participate at least once in the same course. The node size and color intensity indicate the connectivity of a node, where more connected nodes are larger and darker in color.

As the visualization of G' in Figure 5 seems to indicate the presence of some highly connected communities, we additionally evaluated the community structure of the graph G'. We used the Louvain method [Blondel et al., 2008] for this, which was found to be efficient for both synthetic and real-world networks [Lancichinetti and Fortunato, 2009, Yang et al., 2016]. The results of the Louvain method on the DTU contact graph are given in Table 2. Here, we also estimate a mixing coefficient μ, which is the ratio of a node's external neighbors, i.e., neighbors that are in a different community, to the total degree of the node [Lancichinetti and Fortunato, 2009]. The modularity score of 0.558 shows that G' has a moderate to strong community structure, where the Louvain method identifies 9 different communities. The mixing parameter μ further confirms these findings, as it shows that, on average, less than 25% of the edges connected to a node are to a node outside of the node's own community.

Table 1: Graph structure properties of the full DTU contact graph G' and the smaller department graphs, where V denotes the number of nodes and E the number of edges. $\bar{\delta}$ represents the average degree, D the network's diameter, L the average shortest path length, and C the average clustering coefficient. L_{random} and C_{random} are calculated for an Erdős–Rényi random graph with the same amount of nodes and edges.

Graph	$\|V\|$	$\|E\|$	$\bar{\delta}$	D	L	C	L_{random}	C_{random}
DTU Full	9,602	1,668,884	347.61	5	2.44	0.64	1.96	0.036
DTU Management	2,415	341,511	282.82	4	2.09	0.84	1.88	0.117
DTU Eng. Technology	1,933	134,703	139.37	7	2.76	0.88	1.93	0.072

Table 2: Topological community properties of the DTU contact graph G'

Topological Property	Value
Number of communities C	9
Modularity Q	0.558
Estimated mixing coefficient μ	0.242

5.1 Department Subgraphs

As we would like to evaluate our methods on graphs with varying degrees of connectivity, we created additional (smaller) instances by considering subgraphs that cover single departments at DTU: DTU Management and DTU Engineering Technology. We then created a contact graph by considering all students taking courses at this department and only those classes that are offered by this department. Note that DTU students generally take courses from multiple departments but often follow the majority of their courses at one to two departments.

Summary statistics for these two additional contact graphs are given in the lower part of Table 1. It can be seen that both graphs are significantly smaller than the full DTU contact graph, having about 20–25% of the number of nodes and about 8–20% of the edges of the full contact graph. It can also be seen that the average number of neighbors is slightly lower, which is not surprising considering that we are looking at (edge-induced) subgraphs. Instead, one can see that the average clustering coefficient is higher for these subgraphs, indicating that the nodes tend to be more clustered together. The two new graphs differ in the diameter and average path length, where the graph for DTU Management has a slightly lower path length and diameter than the full graph and the DTU Engineering Technology graph a higher one. Based on the above, it can be concluded that these graphs are more clustered than the full graph, where the DTU Management graph additionally shows to be very connected.

5.2 Disease Characteristics

The disease characteristics that we consider are based on those of COVID-19. In particular, the chosen values are mostly based on the data provided for the Danish society by the *Statens Serum Institute (SSI)*, which is under the auspices of the Danish Ministry of Health. An overview of the

used parameter values is given in Table 3.

Table 3: The SEIR model parameters used for simulations. Some parameters take a fixed value over all simulations, while others are varied between a minimum and maximum value.

Parameter	Minimum	Maximum	Value
p^Q_{self}	–	–	0.5
$p^Q_{neighbor}$	–	–	0.4
μ	–	–	$\frac{1}{4}$
γ	–	–	$\frac{1}{6}$
t^{max}_{trace}	–	–	14
t^{delay}_{notify}	–	–	2
N_{close}	–	–	10
β_{spon}	–	–	0.0003
β_{con}	0.15	0.35	–

The values of all parameters except β_{con} are the same over the different experiments. The values of $\mu, \gamma, t^{max}_{trace}, t^{delay}_{notify}$ and N_{close} equal those used in Bagger et al. [2022], which further motivates their choice. Moreover, the values for the probability of self-quarantining p^Q_{self} and $p^Q_{neighbor}$ are based on earlier experiments in Bagger et al. [2022], who showed that these are values in which quarantining has a clear impact and which are also not unrealistically high. Note in particular that values close to 1 have shown to be unrealistic in practice, see, e.g., Davis et al. [2021]. In addition, the value of β_{spon} is chosen as the middle value of the range for this parameter used in Bagger et al. [2022]. The value of β_{con} differs over the runs. The chosen range for this parameter is roughly based on the values used within the national models developed by SSI [Statens Serum Institut, 2020]. Note that β_{con} will, in practice, depend strongly on the characteristics of both the activity and the room in which the activity takes place, making it hard to estimate a single value beforehand.

6 Numerical Study

In this section, we evaluate the proposed solution methods numerically for the DTU contact graph by looking at the resulting number of infections. Our aim is two-fold. On the one hand, we would like to investigate the benefit that the stochastic programming approach and genetic algorithm provide compared to using existing graph-based measures. On the other hand, we would like to evaluate how strong the effect of network immunization is for the given contact graphs at different immunization rates. In the remainder of this section, we first introduce the benchmark methods and the setup of our experiments. Afterwards, we discuss the performance of the immunization methods and analyze the extent to which their solutions coincide. Moreover, we analyze the impact of network immunization by comparing its results to that of the scheduling policy introduced in Bagger et al. [2022] that minimizes the number of distinct contacts.

6.1 Benchmark Methods

As illustrated by our literature review, a large number of graph-based methods have been proposed for solving network immunization problems. We will use these to benchmark our proposed solution methods, as well as within the genetic algorithm to determine the search space and the initial solution. We consider the following benchmark methods in our numerical study:

M1 *Random*: The nodes to immunize are chosen uniformly at random. We will consider the best (in-sample) solution out of 10 randomly generated solutions.

M2 *Degree centrality (Degree)*: Nodes are ranked according to the number of nodes adjacent to them, i.e., their number of neighbors. Nodes with more neighbors are then prioritized for immunization [Pastor-Satorras and Vespignani, 2002].

M3 *Harmonic centrality*: A centrality measure that looks at the path distance of a node to the other nodes in the graph [Rochat, 2009]. Nodes with a shorter distance to the other nodes are seen as more central and thus prioritized for immunization. This measure is similar to closeness centrality [Freeman, 1978], but unlike closeness centrality, it also applies to disconnected graphs.

M4 *Eigenvector centrality*: A centrality measure proposed by Bonacich [1972] based on the idea that a node's importance is related to its neighbors' importance. It can be computed by determining the principal eigenvector of the adjacency matrix, i.e., the eigenvector corresponding to the largest eigenvalue.

M5 *Betweenness centrality*: A centrality measure that was proposed by Freeman [1977], which looks at the number of times a node lies on the shortest path between any other nodes.

M6 *Community Bridge Finder (CBF)*: A community-based algorithm proposed by Salathé and Jones [2010] that aims to find bridge nodes, i.e., nodes that connect different communities. The found bridge nodes are prioritized for immunization.

It should be noted that the above methods are defined for general graphs and not for the activity-based contact hypergraph G considered in this study. Therefore, we apply these methods for the graph G' considered before, in which each hyperarc in the hypergraph is replaced by the complete graph between all nodes that are part of the activity.

An additional consideration for the centrality-based measures (M2 – M5) is that they can be applied both in a *static* and *dynamic* way. In the static approach, the score for all nodes is computed once and the k nodes with the highest nodes are then selected to be immunized. In the dynamic approach, the centrality score of all remaining nodes is re-computed after each node removal, and only the (remaining) node with the highest score is removed in each iteration. Due to the computation time of the different methods, we will use a dynamic approach for the Degree centrality (M2) measure in our experiments and a static approach for the other centrality measures.

6.2 Experimental Setup

To evaluate the solutions of our methods and the benchmark methods, we will look at the total number of contact infections (CI). We have chosen here to evaluate the number of contact infections

over the number of total infections, as the immunization strategy only has a direct effect on the contact infections. Instead, spontaneous infections can only be prevented if people are in quarantine, meaning that a policy leading to many infections might actually lead to a lower number of spontaneous infections due to more people being in quarantine. All evaluations were made out-of sample, i.e., using a different stream of random numbers for the simulation than used within the simulation-optimization methods themselves, and are based on 200 simulation runs.

The parameters used for the genetic algorithm in our experiments are given in Table 4. Here, the number of highest ranking nodes t selected in the solution representation and the mutation rate ρ are chosen depending on the number of nodes in the graph and the number of nodes to be immunized, respectively. Fixed values are chosen for the other parameters. Note that we choose the number of simulation runs σ_s to identify promising solutions significantly lower than the number of simulation runs σ_l to evaluate the best-found solutions. Moreover, a relatively small tournament size ϵ is selected to ensure that also slightly lower-scoring parents are chosen for crossover, especially as nodes are already chosen based on their centrality score within the solution representation. In addition, note that we use multiple of the centrality-based measures discussed in Section 6.1 to define the solution representation and select some of the initial solutions. Furthermore, we use one additional centrality measure that is specific to our setting of university education:

M7 *Neighboring weights*: Considers a weighted network in which the edge weights are computed based on the number of courses a pair of students attend together. The importance of a node is then determined by the sum of the neighboring edge weights, where nodes with higher importance are prioritized for immunization.

The only parameter that needs to be specified for the stochastic programming approach is the parameter σ_p that determines the number of infection forests. Here, we use $\sigma_p = 300$ simulation runs for the full contact graph to generate the infection forests to balance the quality of the solutions with the computation time of the solved IP problems. We use a larger number of runs $\sigma_p = 400$ for the department contact graphs, considering their smaller size and the better solution quality that is to be expected for a larger number of samples. The effect of the parameter σ_p will be further studied in Section 6.5.

All experiments were performed on an Intel Xeon Gold 6142 processor, utilizing 8 CPU cores and 32GB of internal memory. Each immunization method was given a maximum computation time of three hours, translating to a time-based cut-off of three hours for the genetic algorithm and a maximum IP computation time of three hours for the stochastic programming approach. The IP model in the stochastic programming approach was solved using the Gurobi 10.0 solver. The simulation model for determining the disease spread and all immunization methods were programmed in the Java programming language.

Parameter	Description	Value
N	Size of solution population	50
M	Used centrality measures	$\{\text{M2}, \text{M3}, \text{M4}, \text{M7}\}$
t	Number of highest-ranking nodes per centrality measure	$\frac{n}{2}$
τ	Tournament size of the tournament selection	4
ϵ	Number of solutions chosen in elitist selection	5
ρ	Average mutation rate for each individual	0.05
σ_s	Number of simulation runs in small simulations	25
σ_l	Number of simulation runs in large simulations	150
λ	Number of promising solutions evaluated per iteration	3

Table 4: Overview of parameters used in the GA

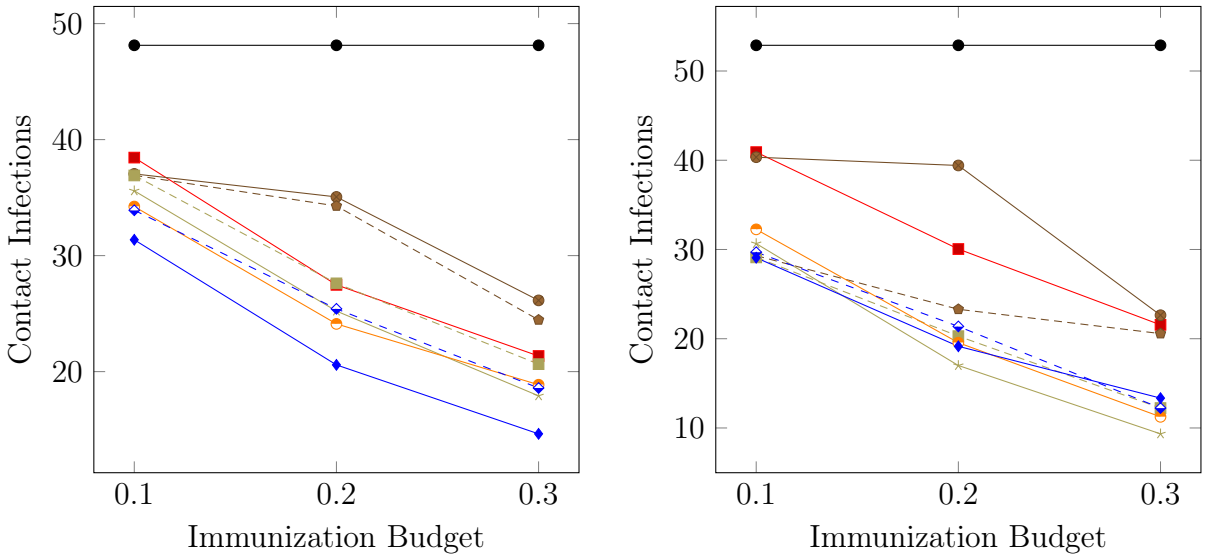
6.3 Comparison of Immunization Approaches

We now compare the stochastic programming approach and genetic algorithm to the benchmark methods (M1) – (M6). Here, we explored the performance of the immunization methods both for different immunization rates, i.e., values of k, and contact infection probabilities β_{con}. Figure 6 shows the results of the immunization methods for varying immunization rates (10%, 20% and 30% of the population size) for a fixed contact infection probability of $\beta_{con} = 0.25$. Figure 7 shows the results of the methods for varying contact infection probabilities ($\beta_{con} \in \{0.15, 0.25, 0.35\}$) for a fixed immunization rate of 20%. In both of these figures, the results correspond to the average number of contact infections after immunization, obtained by means of the simulation model discussed in Section 3.

The immunization results in Figure 6 show that the stochastic programming approach performs well over the different contact graphs and immunization rates. This is especially the case for the DTU Engineering Technology and full DTU graphs, for which the number of contact infections resulting from this method is the lowest among all methods for each of the immunization rates. The results of this method are more mixed for the DTU Management graph, but here the method is still among the best performing methods for each immunization rate. The genetic algorithm is also consistently among the best immunization methods. However, unlike the stochastic programming approach, it never obtains the best result of all immunization methods.

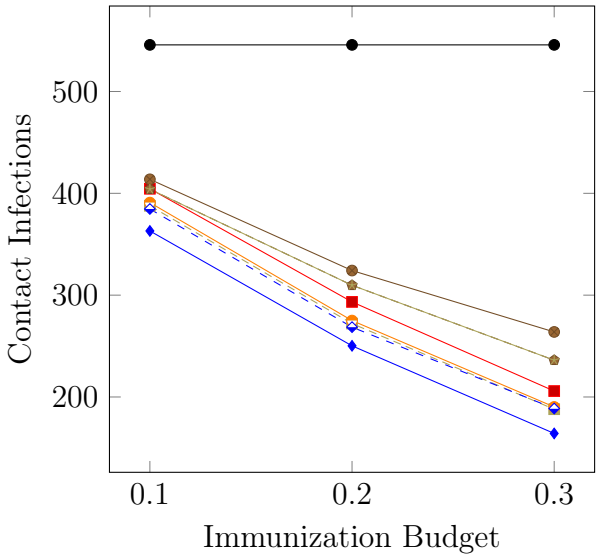
These results on the performance of the stochastic programming approach and genetic algorithm are mostly confirmed by Figure 7 that considers different contact infection probabilities. Again, the stochastic programming approach peforms best on the DTU Engineering Technology and full DTU graph, but more mixed on the DTU management graph. A potential explanation for this might be the very connected nature of the DTU Management graph, which might imply a better performance of the centrality-based measures and, specifically, the betweenness centrality method. It can again be seen that the genetic algorithm is consistently among the best methods but never exceeds all others. This result is somewhat surprising, especially as the genetic algorithm considers an initial population consisting of some solutions that are based on those of the considered centrality measures.

When considering the results of the benchmark methods, it can be seen that these are also relatively stable over the different instances and parameter settings. Betweenness centrality particularly performs well over all instances and obtains the best result for many of the parameter settings of

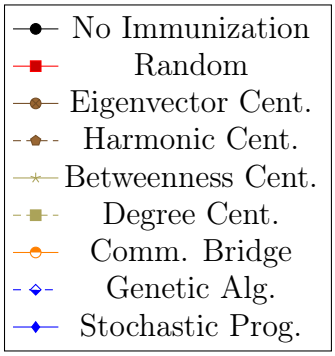


(a) DTU Engineering Technology

(b) DTU Management

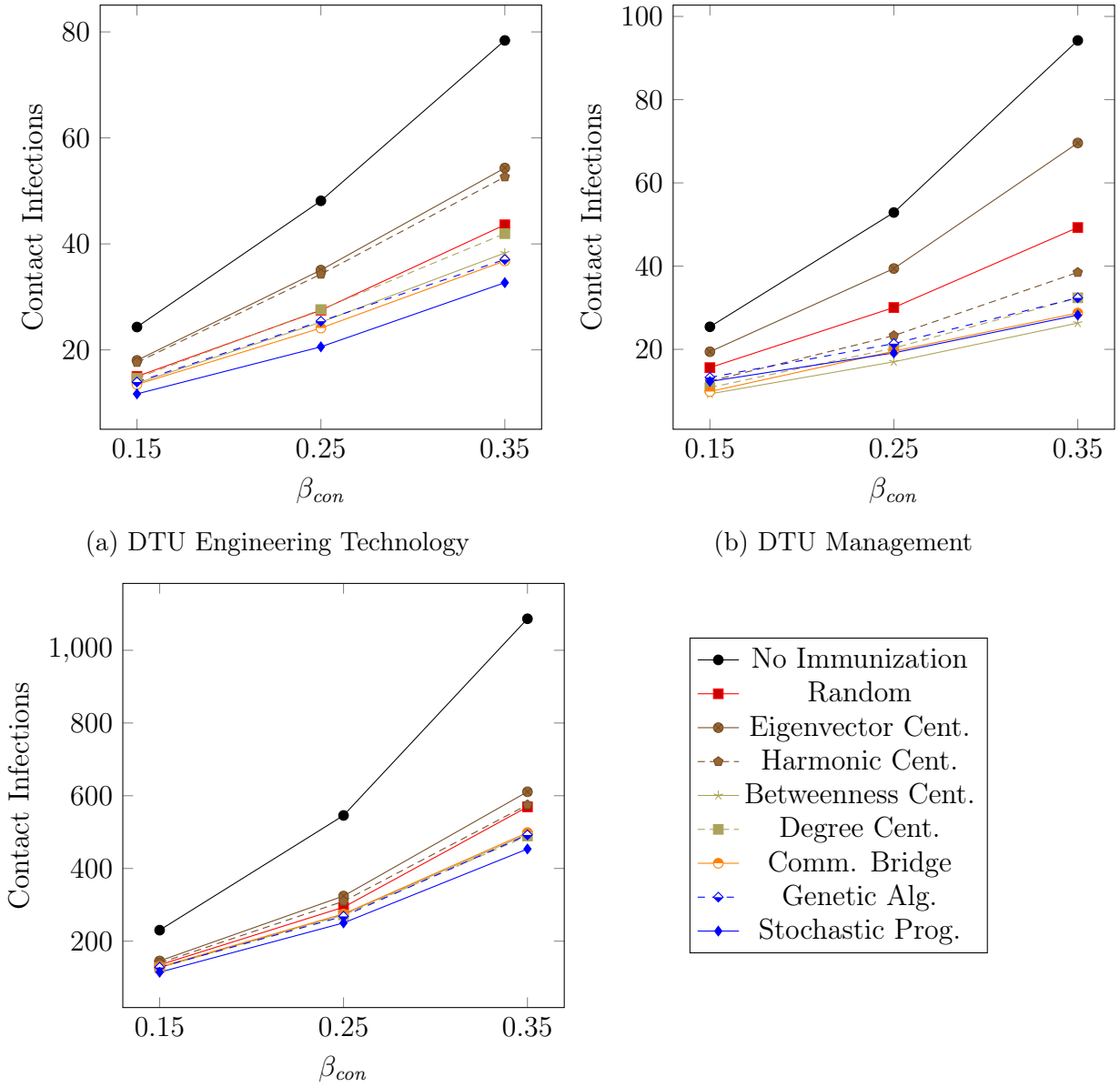


(c) Full DTU



(d) Legend

Figure 6: Results of the immunization methods for different immunization rates at a fixed contact infection probability $\beta_{con} = 0.25$.



(a) DTU Engineering Technology

(b) DTU Management

(c) Full DTU

(d) Legend

Figure 7: Results of the immunization methods for different contact infection probabilities at a fixed immunization rate of 20%.

the DTU Management graph. The Degree centrality and the Community Bridge Finder method also perform well, where the latter shows particularly good performance on the smaller department contact graphs. Eigenvector centrality and Harmonic centrality perform less well and are often unable to outperform the random choice of nodes to be immunized. It should be noted here, though, that the best out of 10 random solutions is considered in the latter method.

6.4 Comparing the Immunized Nodes

To obtain insight into how the solutions from the different immunization methods differ, we use the Jaccard similarity measure. Given any two sets of immunized nodes $X, Y \subseteq V$, this measure can be computed as

$$\text{Jaccard}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}.$$

The Jaccard similarity measure thus computes the fraction of common nodes in relation to the total number of unique nodes over both solutions. We have computed the Jaccard similarity measure between any two solutions obtained in the experiments in Figure 7 for the full contact graph. Note that these solutions of the immunization methods can differ over the contact infection probabilities, meaning that we compute the average over the similarity scores. The resulting scores are given in Table 5.

Table 5: Jaccard similarity score between the solutions of any two immunization methods, averaged over the three instances in Figure 7.

Algorithm	Random	Eigenvector	Harmonic	Betweenness	Degree	Comm. Bridge	GA	Stoch.
Random	1	0.11	0.11	0.12	0.12	0.11	0.11	0.11
Eigenvector	0.11	1	0.65	0.35	0.52	0.17	0.18	0.13
Harmonic	0.11	0.65	1	0.44	0.6	0.2	0.18	0.13
Betweenness	0.12	0.35	0.44	1	0.45	0.25	0.18	0.15
Degree	0.12	0.52	0.6	0.45	1	0.21	0.19	0.16
Comm. Bridge	0.11	0.17	0.2	0.25	0.21	1	0.15	0.13
GA	0.11	0.18	0.18	0.18	0.19	0.15	1	0.13
Stoch.	0.11	0.13	0.13	0.15	0.16	0.13	0.13	1

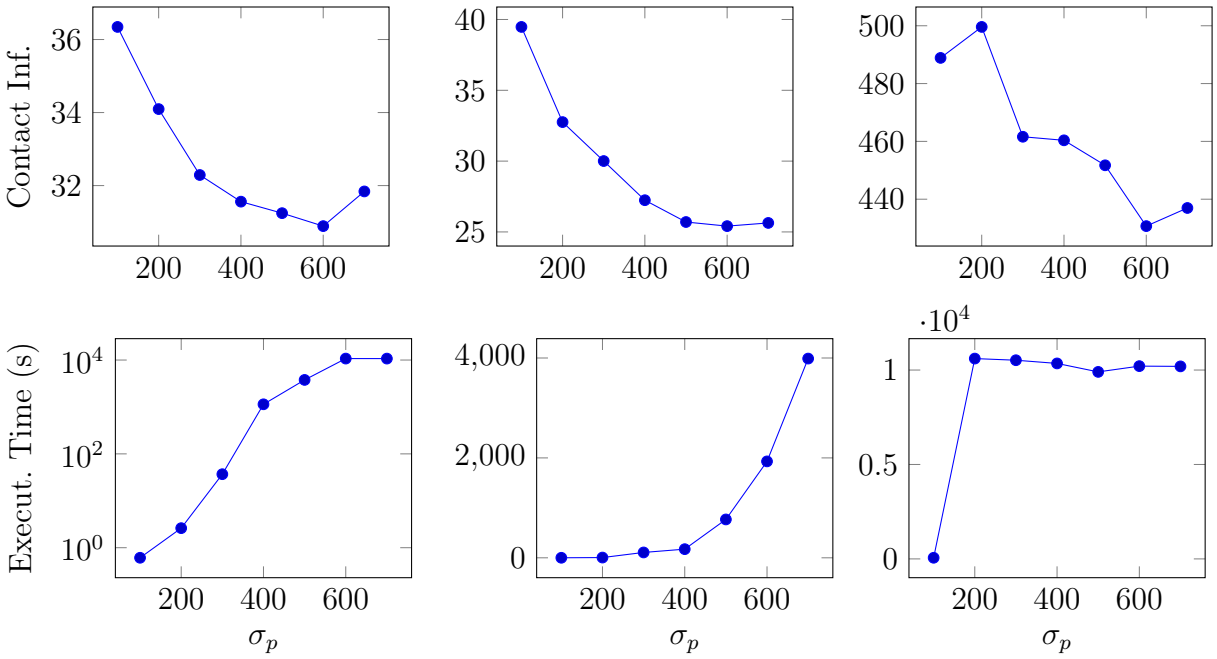
The table shows that the similarity scores overall are relatively low. The highest scores are obtained for solutions of the Harmonic centrality measure, for which about 2/3 of the immunized nodes collide with the Eigenvector centrality measure. Similarly, about 60% and 45% of the immunized nodes for the Harmonic centrality measure overlap with the Degree and Betweenness centrality measures, respectively. Unsurprisingly, the lowest similarity scores are obtained for the random selection of nodes, where, on average, about 11% of the nodes are common with any other solution method.

When looking at our newly proposed immunization methods, it can be seen that the GA solutions have the highest similarity score to the Eigenvector, Harmonic, Betweenness, and Degree centrality solutions. This can likely be explained by some of these centrality measures being used to generate initial solutions within the GA. However, even for these measures, on average, only about one in five nodes is in common with the solutions of the GA. The similarity scores for the stochastic programming approach are even a bit lower. Even compared to the best-performing centrality

measures, which often obtain scores not very far away from those of the stochastic programming approach, no more than 16% of the nodes is shared to these measures. Hence, these results suggest that there is a large number of relatively similar nodes in the DTU contact graph, which can be exchanged in solutions without very strongly impacting the immunization result.

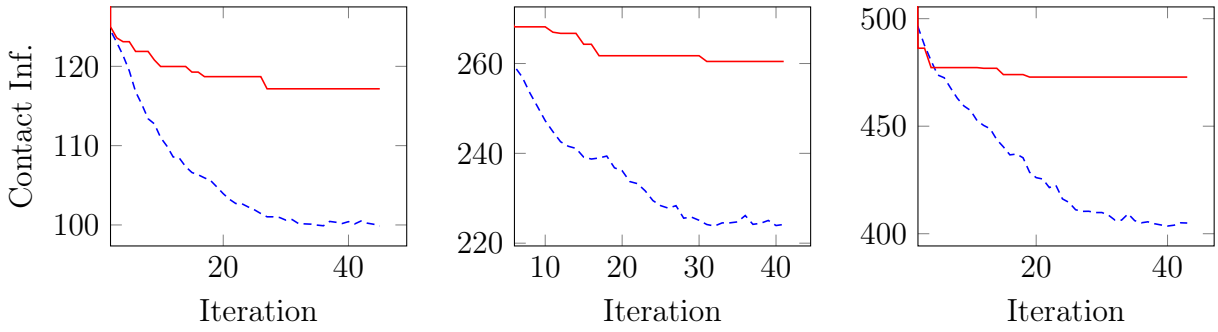
6.5 Performance of the Stochastic Programming Approach and Genetic Algorithm

To get a better insight into the performance of the stochastic programming approach and genetic algorithm, we further zoom in on the performance of these methods. Here, we analyze the impact of the number of sampled infection forests σ_p on the performance of the stochastic programming approach. The results of these experiments are given in Figure 8, where both the number of obtained contact infections and execution time for different values of σ_p are given for all three contact graphs. These results are based on a contact infection probability of $\beta_{con} = 0.35$ and immunization rate of 20%. Moreover, we look at the performance of the GA over its iterations for the experiments for the full contact graph in Figure 7. These results are displayed in Figure 9. In these plots, both the average score over all solutions in the population and the score of the best solution so far are given at each iteration. Note that the former is based on the evaluation in the small simulations, while the latter is based on the evaluation of the large simulation. Hence, the score of the best solution can be higher than the average score of the solutions.



(a) DTU Eng. Tech. (b) DTU Management (c) Full DTU graph

Figure 8: Performance of the stochastic programming approach, in terms of the number of contact infections and execution time, for different values of σ_p.



(a) $\beta_{con} = 0.15$ (b) $\beta_{con} = 0.25$ (c) $\beta_{con} = 0.35$

Figure 9: Evolution of the contact infections over the iterations of the genetic algorithm, showing both the average contact infections for each generation (blue dashed line) as evaluated in the small simulations and the current best solution (red solid line) as evaluated in the large simulation.

The results in Figure 8 show that the performance of the stochastic programming approach overall improves as the number of sampled infection forests increases. In particular, it can be seen that the lowest number of contact infections is obtained for each contact graph at 600 sampled infection forests. Moreover, the number of contact infections almost uniformly decreases until 600 infection forests are sampled and then shows a first sign of increase hereafter. However, it can be seen that the execution time, comprised mainly of the solving time of the IP model, grows quickly. This effect is especially clear for the full DTU contact graph, where the time limit of three hours is already reached at 200 sampled infection forests. While this leads to the method finding non-optimal IP solutions, often with significant optimality gaps, this does not prevent the overall result from improving until a significantly larger number of infection forests is used. Hence, these results show that a high number of samples is required to gain an adequate overview of the typical flow of infections and, thus, of the most important nodes to immunize.

When looking at the progression of the genetic algorithm over the iterations in Figure 9, it can be seen that the best score clearly lags behind the average score. Moreover, the improvement in the number of infections of the best solution is significantly less than the improvement in the average score. Both can likely be explained by the larger number of simulation runs used in evaluating the best solution, which makes it harder to find improving solutions. When looking at the progression over the iterations, it can also be seen that the improvement clearly levels off for the average score but is not fully flat yet after the chosen computation time. Similarly, the frequency of finding a better solution decreases but some improving solutions are still found in the later iterations of the genetic algorithm. Hence, it can be seen that full convergence cannot be obtained within the set solution time of three hours and the clearly limited number of iterations that can be executed in this time.

6.6 Comparison to Minimizing Distinct Contacts

Lastly, we zoom in on the effect of immunization instead of focusing on the performance of the individual immunization methods. Here, we compare the effect of immunization with the effect of

limiting group sizes, in which students are distributed over multiple smaller groups for large courses in such a way that the number of distinct contacts between students is minimized. We use the algorithm proposed in Bagger et al. [2022] to limit the group size for the complete DTU dataset to at most 50 and 30 students, resulting in two new contact graphs. Moreover, we evaluate the effect of combining the measures, where immunization is applied after first assigning students to these smaller course groups such to minimize the number of distinct contacts. The corresponding disease spread for these different approaches is shown in Figure 10, where the immunization results were obtained by using the stochastic programming approach and an immunization rate of 10%, 20%, or 30%.

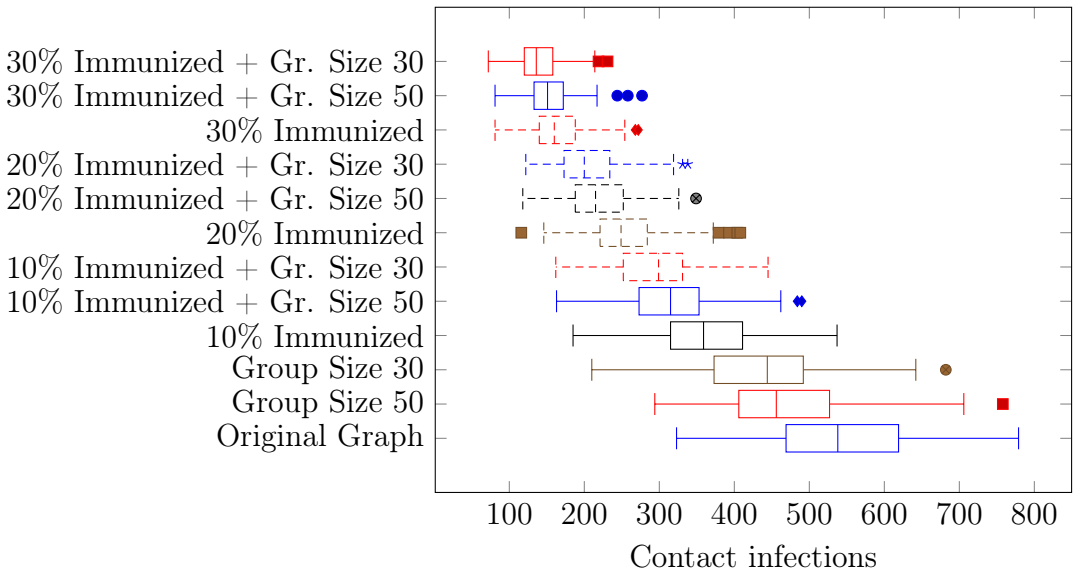


Figure 10: Comparison between network immunization and minimizing the number of distinct contacts. The number of contact infections obtained through simulation is given as a boxplot, where the centre line in each box gives the median number of infections.

The results in Figure 10 show that both the minimization of distinct contacts and network immunization clearly reduce the median number of contact infections. Reducing the group size to 50 or 30 students leads to a reduction of contact infections from about 550 infections in the original graph to about 475, and 450 infections, respectively. Immunization has an even larger effect, since the number of contact infections reduces to about 375, 275 and 225 for the different immunization rates, respectively. It can be concluded that immunization has the largest effect for the chosen parameters, as immunizing just 10% of the students leads to a lower median number of infections than reducing the group size to a maximum of 30 students. Moreover, one can see that the number of contact infections decreases sharply at higher immunization rates.

The results in Figure 10 additionally show that the combination of minimizing distinct contacts and immunization can lead to a further reduction of infections. A clear jump in the median number of infections can especially be seen when moving from immunization only to combining immunization and a maximum group size of 50 students. In comparison, the jump is smaller when the group size is further reduced to 30 students. This is in line with the results we saw for reducing group sizes only, where the jump from the original graph to a maximum of 50 students is also larger than the

jump of moving from 50 to 30 students in a group. It should be noted, though, that all results come with a considerable amount of variation, meaning that many comparisons do not achieve statistical significance.

7 Conclusion

In this paper, we looked at the Network Immunization Problem in the context of an epidemic disease. This problem focuses on choosing individuals to immunize, given a maximum immunization rate, such that the spread of the disease through the population is minimized. Compared to the existing literature on this problem, we consider a richer epidemiological setting that includes the quarantining of infected individuals and their close contacts, and a limited willingness to test and quarantine. As a result, a simulation approach is used to evaluate the effect of an immunization strategy, where we focus on the number of infections that follow from contacts in the population.

We proposed two simulation-optimization approaches for the Network Immunization Problem: a stochastic programming approach and a genetic algorithm. The stochastic programming approach is based on sample average approximation, where we sample infection forests through simulation. After sampling these infection forests, we solve an optimization problem to choose the immunized nodes so that the number of infections that would occur in these infection forests is minimized. In the genetic algorithm, we use the results from existing centrality measures to choose an initial population and combine simulation runs of small and large size to balance the time needed to find promising solutions with the uncertainty that results from simulation. Moreover, we parallelized the simulation runs to increase the number of iterations that can be run.

We applied the proposed algorithms to a contact graph based on students' course assignments for a major university in Denmark. This contact graph shows both small-world properties and a community structure. Our results show that our proposed methods are competitive with the best centrality measures and that the stochastic programming approach is able to outperform these immunization measures for a considerable number of these instances. We also compared the corresponding solutions of the immunization methods, showing that the solutions of the stochastic programming method, in particular, tend to be different from those of the centrality-based measures. Furthermore, we looked at the effectiveness of immunization by comparing it to the strategy proposed in Bagger et al. [2022] to minimize the number of distinct contacts when assigning students to course groups. Our experiments show that immunization can lead to a relatively large reduction in infections and that the number of contact infections is reduced quickly when the immunization rate increases. Moreover, we showed that combining immunization and the minimization of distinct contacts can lead to a further reduction in infections.

Our paper has thus shown that simulation-optimization approaches form a promising direction for method development in network immunization problems, especially when the desire is to evaluate strategies under a rich set of conditions, such as the limited quarantining of individuals. Future research could focus on researching robust strategies for individual selection, e.g., under the assumption that only a limited percentage of invited individuals will actually show up for vaccination. Moreover, the problem of sub-group selection rather than the selection of individuals could be interesting in the considered context of contact graphs that result from planned activities. Finally, future research could investigate the best generation of infection forests and representation of possible infection chains in the proposed stochastic programming approach.

Acknowledgement

This research was funded by DFF (Independent Research Fund Denmark) as part of the project *FIND: Finding the "new normal", the power of distinct contacts (Grant number 0213-00040B)*. Moreover, we would like to thank Guðmundur Óskar Halldórsson and Rakel Guðrún Óladóttir, whose Master thesis provided a starting point for this paper.

References

J.M. Anthonisse. The rush in a directed graph. Technical Report BN 9/71, Stichting Mathematisch Centrum, 1971.

Niels-Christian Fink Bagger, Evelien van der Hurk, Rowan Hoogervorst, and David Pisinger. Reducing disease spread through optimization: Limiting mixture of the population is more important than limiting group sizes. *Computers & Operations Research*, 142:105718, 2022. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2022.105718.

Simon Bauer, Sebastian Contreras, Jonas Dehning, Matthias Linden, Emil Iftekhar, Sebastian B. Mohr, Alvaro Olivera-Nappa, and Viola Priesemann. Relaxing restrictions at the pace of vaccination increases freedom and guards against further COVID-19 waves. *PLOS Computational Biology*, 17(9):1–37, 09 2021. doi: 10.1371/journal.pcbi.1009288. URL `https://doi.org/10.1371/journal.pcbi.1009288`.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008 (10):P10008, oct 2008. doi: 10.1088/1742-5468/2008/10/p10008.

Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology*, 2(1):113–120, 1972. doi: 10.1080/0022250X.1972.9989806. URL `https://doi.org/10.1080/0022250X.1972.9989806`.

Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC)*, 10(4):1–26, 2008. URL `https://doi.org/10.1145/1284680.1284681`.

Chen Chen, Hanghang Tong, B. Aditya Prakash, Charalampos E. Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. Node immunization on large graphs: Theory and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):113–126, 2016. doi: 10.1109/TKDE.2015.2465378.

Emma L. Davis, Tim C. D. Lucas, Anna Borlase, Timothy M. Pollington, Sam Abbott, Diepreye Ayabina, Thomas Crellen, Joel Hellewell, Li Pi, Rachel Lowe, Akira Endo, Nicholas Davies, Georgia R. Gore-Langton, Timothy W. Russell, Nikos I. Bosse, Matthew Quaife, Adam J. Kucharski, Emily S. Nightingale, Carl A. B. Pearson, Hamish Gibbs, Kathleen O'Reilly, Thibaut Jombart, Eleanor M. Rees, Arminder K. Deol, Stéphane Hué, Megan Auzenbergs, Rein M. G. J. Houben, Sebastian Funk, Yang Li, Fiona Sun, Kiesha Prem, Billy J. Quilty, Julian Villabona-Arenas,

Rosanna C. Barnard, David Hodgson, Anna Foss, Christopher I. Jarvis, Sophie R. Meakin, Rosalind M. Eggo, Kaja Abbas, Kevin van Zandvoort, Jon C. Emery, Damien C. Tully, Frank G. Sandmann, W. John Edmunds, Amy Gimma, Gwen Knight, James D. Munday, Charlie Diamond, Mark Jit, Quentin Leclerc, Alicia Rosello, Yung-Wai Desmond Chan, David Simons, Sam Clifford, Stefan Flasche, Simon R. Procter, Katherine E. Atkins, Graham F. Medley, T. Déirdre Hollingsworth, Petra Klepac, and CMMID COVID-19 Working Group. Contact tracing is an imperfect tool for controlling COVID-19 transmission and relies on population adherence. *Nature Communications*, 12(1):5412, Sep 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-25531-5. URL `https://doi.org/10.1038/s41467-021-25531-5`.

Ye Deng, Jun Wu, and Yue jin Tan. Optimal attack strategy of complex networks based on tabu search. *Physica A: Statistical Mechanics and its Applications*, 442:74–81, 2016. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2015.08.043.

Michael Emmerich, Joost Nibbeling, Marios Kefalas, and Aske Plaat. Multiple node immunisation for preventing epidemics on networks by exact multiobjective optimisation of cost and shield-value, 2020.

Shakiba Enayati and Osman Y. Özaltın. Optimal influenza vaccine distribution with equity. *European Journal of Operational Research*, 283(2):714–725, 2020. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2019.11.025.

Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3): 215–239, 1978. ISSN 0378-8733. doi: https://doi.org/10.1016/0378-8733(78)90021-7.

Chao Gao, Jiming Liu, and Ning Zhong. Network immunization and virus propagation in email networks: experimental evaluation and analysis. *Knowledge and Information Systems*, 27(2): 253–279, May 2011. ISSN 0219-3116. doi: 10.1007/s10115-010-0321-0. URL `https://doi.org/10.1007/s10115-010-0321-0`.

Sujin Kim, Raghu Pasupathy, and Shane G. Henderson. *A Guide to Sample Average Approximation*, pages 207–243. Springer New York, New York, NY, 2015. ISBN 978-1-4939-1384-8. doi: 10.1007/978-1-4939-1384-8_8. URL `https://doi.org/10.1007/978-1-4939-1384-8_8`.

Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80:056117, Nov 2009. doi: 10.1103/PhysRevE.80.056117. URL `https://link.aps.org/doi/10.1103/PhysRevE.80.056117`.

Kaihui Liu and Yijun Lou. Optimizing COVID-19 vaccination programs during vaccine shortages. *Infectious Disease Modelling*, 7(1):286–298, 2022. ISSN 2468-0427. doi: https://doi.org/10.1016/j.idm.2022.02.002.

Asep Maulana, Marios Kefalas, and Michael T. M. Emmerich. Immunization of networks using genetic algorithms and multiobjective metaheuristics. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2017. doi: 10.1109/SSCI.2017.8285368.

Lucy A McNamara, Ryan E Wiegand, Rachel M Burke, Andrea J Sharma, Michael Sheppard, Jennifer Adjemian, Farida B Ahmad, Robert N Anderson, Kamil E Barbour, Alison M Binder, Sharoda Dasgupta, Deborah L Dee, Emma S Jones, Jennifer L Kriss, B Casey Lyons, Meredith McMorrow, Daniel C Payne, Hannah E Reses, Loren E Rodgers, David Walker, Jennifer R Verani, and Stephanie J Schrag. Estimating the early impact of the US COVID-19 vaccination programme on COVID-19 cases, emergency department visits, hospital admissions, and deaths among adults aged 65 years and older: an ecological analysis of national surveillance data. *The Lancet*, 399 (10320):152–160, 2022. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(21)02226-1.

Jan Medlock and Alison P. Galvani. Optimizing influenza vaccine distribution. *Science*, 325(5948): 1705–1708, 2009. doi: 10.1126/science.1175570. URL `https://www.science.org/doi/abs/10.1126/science.1175570`.

Apurba K. Nandi and Hugh R. Medal. Methods for removing links in a network to minimize the spread of infections. *Computers & Operations Research*, 69:10–24, 2016. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2015.11.001.

M. E. J. Newman. Spread of epidemic disease on networks. *Phys. Rev. E*, 66:016128, Jul 2002. doi: 10.1103/PhysRevE.66.016128. URL `https://link.aps.org/doi/10.1103/PhysRevE.66.016128`.

Daniela Olivera Mesa, Alexandra B. Hogan, Oliver J. Watson, Giovanni D. Charles, Katharina Hauck, Azra C. Ghani, and Peter Winskill. Modelling the impact of vaccine hesitancy in prolonging the need for non-pharmaceutical interventions to control the COVID-19 pandemic. *Communications Medicine*, 2(1):14, Feb 2022. ISSN 2730-664X. doi: 10.1038/s43856-022-00075-x. URL `https://doi.org/10.1038/s43856-022-00075-x`.

Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86:3200–3203, Apr 2001. doi: 10.1103/PhysRevLett.86.3200. URL `https://link.aps.org/doi/10.1103/PhysRevLett.86.3200`.

Romualdo Pastor-Satorras and Alessandro Vespignani. Immunization of complex networks. *Phys. Rev. E*, 65:036104, Feb 2002. doi: 10.1103/PhysRevE.65.036104. URL `https://link.aps.org/doi/10.1103/PhysRevE.65.036104`.

Sancheng Peng, Guojun Wang, Yongmei Zhou, Cong Wan, Cong Wang, Shui Yu, and Jianwei Niu. An immunization framework for social networks through big data based influence modeling. *IEEE Transactions on Dependable and Secure Computing*, 16(6):984–995, 2019. doi: 10.1109/TDSC.2017.2731844.

Mahendra Piraveenan, Mikhail Prokopenko, and Liaquat Hossain. Percolation centrality: Quantifying graph-theoretic impact of nodes during percolation in networks. *PLOS ONE*, 8(1):1–14, 01 2013. doi: 10.1371/journal.pone.0053095. URL `https://doi.org/10.1371/journal.pone.0053095`.

Yannick Rochat. Closeness centrality extended to unconnected graphs: The harmonic centrality index. Technical report, 2009.

Sudip Saha, Abhijin Adiga, B. Aditya Prakash, and Anil Kumar S. Vullikanti. Approximation algorithms for reducing the spectral radius to control epidemic spread. In *Proceedings of the 2015 SIAM International Conference on Data Mining (SDM)*, pages 568–576, 2015. doi: 10.1137/1.9781611974010.64.

Marcel Salathé and James H. Jones. Dynamics and control of diseases in networks with community structure. *PLOS Computational Biology*, 6(4):1–11, 04 2010. doi: 10.1371/journal.pcbi.1000736. URL https://doi.org/10.1371/journal.pcbi.1000736.

Christian M. Schneider, Tamara Mihaljev, Shlomo Havlin, and Hans J. Herrmann. Suppressing epidemics with a limited amount of immunization units. *Phys. Rev. E*, 84:061911, Dec 2011. doi: 10.1103/PhysRevE.84.061911. URL https://link.aps.org/doi/10.1103/PhysRevE.84.061911.

S.N. Sivanandam and S.N. Deepa. *Genetic Algorithms*, pages 15–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-73190-0. doi: 10.1007/978-3-540-73190-0_2. URL https://doi.org/10.1007/978-3-540-73190-0_2.

Statens Serum Institut. Teknisk gennemgang af modellerne. https://files.ssi.dk/teknisk-gennemgang-af-modellerne-10062020 (last accessed 10-03-2022), 06 2020. version 1.0.

Sara Y Tartof, Jeff M Slezak, Heidi Fischer, Vennis Hong, Bradley K Ackerson, Omesh N Ranasinghe, Timothy B Frankland, Oluwaseye A Ogun, Joann M Zamparo, Sharon Gray, Srinivas R Valluri, Kaije Pan, Frederick J Angulo, Luis Jodar, and John M McLaughlin. Effectiveness of mRNA BNT162b2 COVID-19 vaccine up to 6 months in a large integrated health system in the USA: a retrospective cohort study. *The Lancet*, 398(10309):1407–1416, 2021. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(21)02183-8.

Piet Van Mieghem, Dragan Stevanović, Fernando Kuipers, Cong Li, Ruud van de Bovenkamp, Daijie Liu, and Huijuan Wang. Decreasing the spectral radius of a graph by link removals. *Phys. Rev. E*, 84:016101, Jul 2011. doi: 10.1103/PhysRevE.84.016101. URL https://link.aps.org/doi/10.1103/PhysRevE.84.016101.

Eleftheria Vasileiou, Colin R Simpson, Ting Shi, Steven Kerr, Utkarsh Agrawal, Ashley Akbari, Stuart Bedston, Jillian Beggs, Declan Bradley, Antony Chuter, Simon de Lusignan, Annemarie B Docherty, David Ford, FD Richard Hobbs, Mark Joy, Srinivasa Vittal Katikireddi, James Marple, Colin McCowan, Dylan McGagh, Jim McMenamin, Emily Moore, Josephine LK Murray, Jiafeng Pan, Lewis Ritchie, Syed Ahmar Shah, Sarah Stock, Fatemeh Torabi, Ruby SM Tsang, Rachael Wood, Mark Woolhouse, Chris Robertson, and Aziz Sheikh. Interim findings from first-dose mass COVID-19 vaccination roll-out and COVID-19 hospital admissions in Scotland: a national prospective cohort study. *The Lancet*, 397(10285):1646–1657, 2021. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(21)00677-2.

M. Ventresca and D. Aleman. A randomized algorithm with local search for containment of pandemic disease spread. *Computers & Operations Research*, 48:11–19, 2014. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2014.02.003. URL https://www.sciencedirect.com/science/article/pii/S030505481400029X.

Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, Jun 1998. ISSN 1476-4687. doi: 10.1038/30918. URL https://doi.org/10.1038/30918.

Zhao Yang, René Algesheimer, and Claudio J. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6(1):30750, Aug 2016. ISSN 2045-2322. doi: 10.1038/srep30750. URL https://doi.org/10.1038/srep30750.