Immunization of Networks Using Genetic Algorithms and Multiobjective Metaheuristics

Asep Maulana[1], Marios Kefalas[1], and Michael T. M. Emmerich[1]

Abstract— **The immunization of complex networks can be formulated as a subset selection problem, where the goal is to select a limited number of nodes to be immunized in order to effectively prevent or decelerate the spread of an epidemic. The drop of the largest eigenvalue (eigen-drop) is a measure for the impact of an immunization strategy. It was recently shown that the problem of selecting k out of n nodes from a network such that eigenvalue drop is maximum belongs to the class of NP hard problems. Heuristic algorithms have been suggested to solve these problems approximately, most importantly the netshield algorithm, a greedy approach that approximates the eigenvalue drop by means of a submodular function, the *shield value*, and then maximizes the shield value by means of a greedy approximation algorithm. In this paper, we design a problem specific genetic algorithm and compare it to netshield plus – an improved variant of netshield – and show that on six moderate size problems from literature that their performance is competitive and often better. We also formulate the node immunization problem as a multiobjective problem, including the cost of immunization as a second objective. First results are presented on biobjective optimization using multiobjective genetic algorithms as solver. The method is demonstrated on the *USA domestic airline network* and the global city network of the *Pandemic cooperative board game*, which are augmented by immunization cost data. First insights into the reliability of solvers and the typical shapes of Pareto fronts are obtained.**

Index Terms— **K-Node Immunization, Genetic Algorithms, Netshield, Epidemiology, Multiobjective Optimization**

I. INTRODUCTION

The study of networks has received increased attention in recent years. The effective control and combating of epidemic, such as Ebola [9] or the Zika virus [7], is one major problem, where the discovery of algorithms for analyzing and controlling networks can make (or have) an impact.

This paper will focus on immunization strategies that achieve a high *eigenvalue drop*. The eigenvalue drop is the drop of the maximum eigenvalue after removal of a subset of nodes from a network, represented as an adjacency matrix. The eigenvalue drop is an effective measure for the impact of an immunization strategy, because the maximum eigenvalue is inversely proportional to the epidemic threshold which determines how fast a virus spreads in the network and how long it lingers in the network [3], [4].

The epidemiological model that is considered in this work is the susceptible-infected-susceptible model, in short SIS model. Here a node in the network can be infected via a direct neighbor and after a time it can recover and is susceptible again. See Figure 2 for different epidemiological models. Immunization of nodes can be enforced by measures outside of the network, e.g., by controlling the node or by removing the node from the network. In this work we assume that an immunized node can no longer infect other nodes, nor can it get infected itself.

Consider for instance a network of airports connected by flights, such as the one provided in the US flights dataset 1. There might be some nodes already infected and we need to make it difficult for the virus to spread by controlling some major airports, e. g., by special bio-security checks or quarantining.

A network G will be represented as a pair (V,E) where V is a set of nodes $V = \{v_1,\dots,v_n\}$ and a set of edges $E \subseteq V \times V$. Vertices and edges can have weights and edge weights will be represented by a function $w_E : E \to \mathbb{R}^+$, and node weights by a function $w_V : V \to \mathbb{R}_0^+$. Given this, for a network we can alternatively use the adjacency matrix representation $A(V,E,w_E) \in \mathbb{R}^{n\times n}$ with $a_{ij} = 0$ if $(v_i,v_j) \neq E$ and $w((v_i,v_j))$ otherwise. The first or maximum eigenvalue of the graph will be denoted λ and the corresponding eigenvector with u. The components of this eigenvector, $u_1,...,u_n$ play a special role in this work and will be called the *eigen-scores* of the matrix.

Definition 1 *Given a network G and a network G′, where G′ is a subgraph of G with some nodes and their adjacent edges removed, the eigenvalue drop $\Delta\lambda$ is defined as the difference between the maximum eigenvalue of the adjacency matrix of G and the maximum eigenvalue of the adjacency matrix of G′.*

Definition 2 *The K-Node Immunization problem, is the problem of finding a subset of k nodes to be removed from a network with n nodes, such that the eigenvalue drop is maximal.*

It has been shown in [4] that the decision problem that corresponds to the K-Node Immunization problem is NP complete, and consequently the K-Node Immunization problem is NP hard. Therefore, heuristic methods have been suggested in [4], most notably the netshield plus algorithm. This algorithm does not directly operate on the eigenvalue drop, but uses an approximation of it which is submodular and therefore lends itself for constructing an approximation

Asep Maulana gratefully acknowledges support by the Indonesian Endowment fund for Education, Lembaga Pengelola Dana Pendidikan (LPDP) Indonesia. We also acknowledge the researchers of the Leiden Complex Networks Network (LCN2) for inspiration and discussion. The authors would like to ackknowledge Aske Plaat, Iryna Yevseyeva and Frank den Hollander for pointing us to these problems in complex networks and interesting data sets.

[1] Michael Emmerich, Asep Maulana, and Marios Kefalas are with Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands.

978-1-5386-2726-6/17/$31.00 ©2017 IEEE

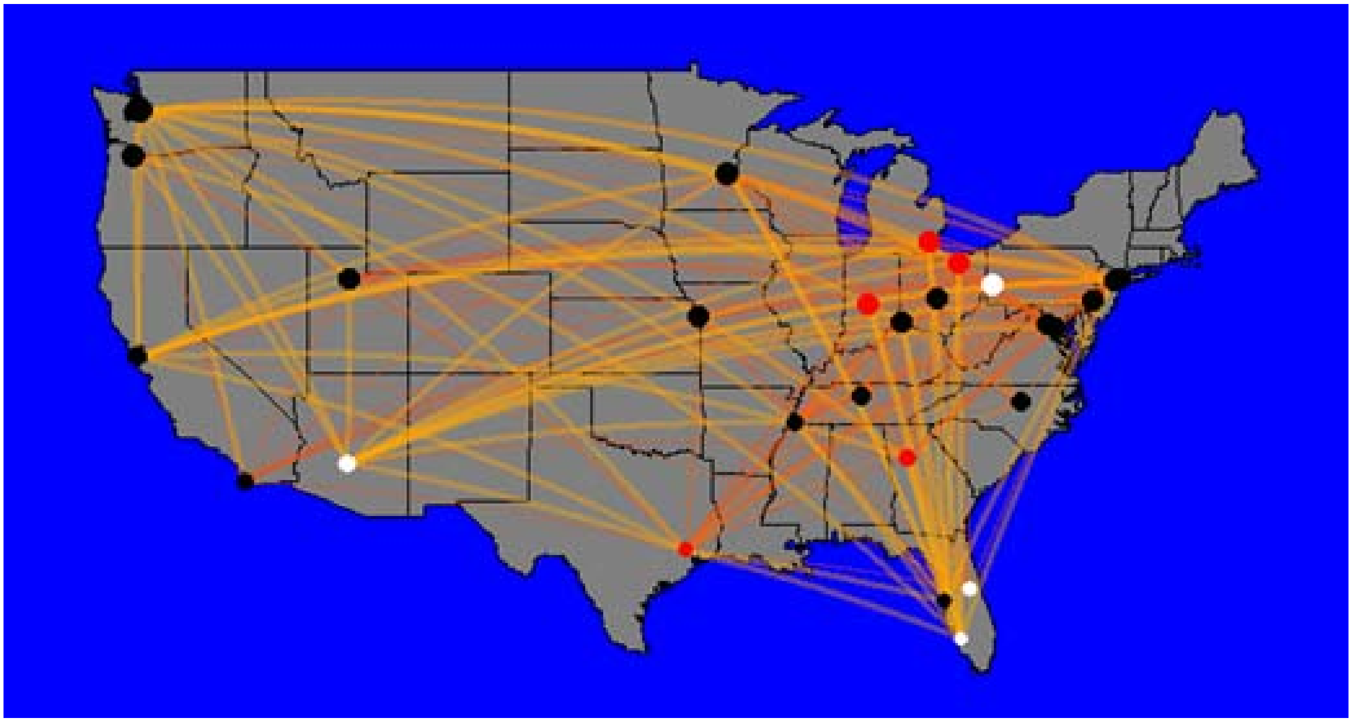


Fig. 1. US flight network of major airports. The picture shows a snapshot of the spreading of a virus. The black nodes are susceptible, the white nodes are infected, and the red nodes are immunized. `kateto.net/network-visualization`

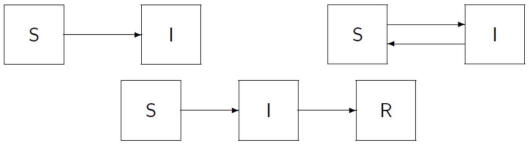


Fig. 2. Three common models in epidemiology, In the SI model, nodes stay infected, once they got infected, In the SIS model, infected nodes can return in a susceptible state, and in the SIR model nodes are immunized after having recovered and can no longer infect neighboring nodes.

algorithm. In brief, netshield seeks to maximize the following *Shield value* (Sv) function, which is closely correlated with the eigenvalue drop.

In our paper we propose an alternative approach to the k-node immunization problem based on genetic algorithms (Section III-A) and compare results to netshield plus (Section III-B). In the problem specific mutation operator some of the ideas of netshield will be adopted. Therefore, we will introduce this algorithm and the scoring function used by it briefly in Section II. Moreover, a multiobjective generalization of the k-node immunization problem is discussed. It introduces a cost term as a second objective (Section IV-A). First results on finding the Pareto front of this problem with multiobjective metaheuristics are presented (Section IV-B).

II. Netshield Algorithm

Next, we will briefly introduce the netshield algorithm. Some of the ideas of this algorithm will be useful in the design of the problem specific genetic algorithm. Moreover the netshield plus algorithm, an improved version of the netshield algorithm, will serve as a baseline algorithm in the benchmarking.

Let $G = (V, E)$ denote the original graph, and $G' = (V', E')$ the graph after some nodes have been removed, and we define $S = V \setminus V'$. Moreover, A and A' denote the corresponding adjacency matrices. Then the Shield value (Sv)of S is defined as follows.

$$Sv(S) = \sum_{i \in S} 2\lambda (u_i)^2 - \sum_{i,j \in S} a_{ij} u_i u_j$$

Here u_i denotes the i-th component of the eigenvector that corresponds to the largest eigenvalue. It is also called the i-th eigen-score. The Shield value rewards dissimilarity between nodes, that is small a_{ij}, and a high eigen-score.

As opposed to the Netshield algorithm, the Netshield plus algorithm [4], removes nodes in batches of b nodes each. After each batch the largest eigenvalue and the corresponding eigen-scores are recomputed. This way the algorithm yields more accurate results, but due to multiple eigenvalue computations the computation time increases. Netshield plus is therefore especially recommended for small or moderate size networks, as we discuss them in this paper.

III. Problem Specific Genetic Algorithm

A. *Discussion of the method*

In this work we use a standard $(\mu + \mu)$ genetic algorithm (see, e.g., [10]) with scaled proportional selection (mating selection) and truncation selection (environmental selecting). The genetic algorithm for the k subset selection uses problem specific mutation and crossover operators. The representation of solution candidates is not binary, as usual, but a problem specific representation for subset selection as it has been used in other contexts, too [11]. A solution is represented as k non duplicate integers in $[1,n] \subset \mathbb{N}$.

The mutation operator that was designed for this problem relies on two mechanisms:

- Firstly, in each mutation an integer that is in the array is replaced by an integer in $[1,n]$ that is not in the set.
- Secondly, the algorithm works with two different mutation rates. For nodes with a top-k eigen-score the probability of mutation is increased by a constant factor ≥ 0, making it more likely to be selected for the set or discarded. This way it is hypothesized that the algorithm spends more time in exploring relevant parts of the graph. The multiplication factor will be denoted with ν.

Mutation is applied to each offspring individual. First an integer in the array is selected proportionally to the mutation probabilities. Then an integer outside the array is selected proportional to the mutation probabilities. And then the node inside the array is replaced by the node outside the array. The genetic algorithm does not feature crossover, but we might consider the development of a problem specific crossover for future research.

B. *Comparison to Netshield plus*

For the empirical comparison of algorithms we will use five data sets on networks:

- Karate: social network of friendships between 34 members of a karate club at a US university in the 1970s [12].
- Dolphins: It is a social network consist of undirected network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand.[8]
- US Flights: List of the most important Airport in the United State connected one to the other based on the existing of connection flight (edge) from one port to the other ports.
- Pandemic: A cooperative board game with the goal to fight the outbreak of the virus. We used the graph that connects cities in the world as an example data set [6]. A picture of the Pandemic board is seen in Figure 3
- Conference Day 1: Social interaction of members of a conference on first day. Taken from here http://www.sociopatterns.org/datasets/infectious-sociopatterns
- Conference Day 3: From the same data set as above, but for the third day.

From the data sets US flights and Pandemic are most representative for the problem class. The other networks are added to gain more general insights into the algorithm behavior and reliability. Note that social interaction networks are also relevant in the spread of the virus, albeit control is less straightforward as compared to networks where nodes are assigned to places, such as US flights and the Pandemic board game network.

For the k-node immunization problem we used the Netshield Plus algorithm and parameters as described in [4]. For the genetic algorithm tests the following setting were applied: The number of function evaluations were 30000. Different mutation parameters were tested, with a value of $\nu \in \{1/n, 2/n, 3/n, 6/n, 1\}$, that is the mutation rate for the k components of u with the highest eigen-score. For all other nodes the mutation probability was set to $1/n$, which is a recommended rate according to Bäck [1].

Regarding the single objective genetic algorithms, they were executed 20 times each, for $k = 3,5$ and 10 on the Karate, Dolphins, US Flights, Pandemic, Conference Day 1 and Conference Day 3 networks. Table I shows results for single objective optimization of the eigenvalue drop. For assessing statistical significance we also provide box plots of our results in Figure 5 and Figure 6. We observe, that GA_5, which represents the (μ + μ) genetic algorithm that introduces a mass of 5 to the k-highest eigen-score nodes, to be the best candidate. Although there is not a unanimously best genetic algorithm for the task, we consider our genetic algorithms to be a supplementary tool to Netshield/Netshield plus, for medium sized networks ($\leq$ 200 nodes).

IV. MULTIOBJECTIVE NODE IMMUNIZATION

In real world scenarios it is likely that multiple nodes need to be controlled or immunized, but it is typically not the case that the value of k is given a priori. Rather it is the case that the immunization of a node comes with a cost, which can differ from node to node. If a larger number of nodes is immunized the total cost would be approximately proportional to the cumulated cost of immunizing the single nodes. Let S denote the set of indexes of the immunized nodes and c_i denote the cost of immunization of node i, defined a priori. Then the *immunization cost* objective function can be defined as

$$C(S) = \sum_{i \in S} c_i \rightarrow \min$$

In multiobjective optimization, problems with two or more objectives are solved. In case of node immunization problem the problem formulation reads:

$$f_1(S) := \lambda(S) \rightarrow \max \quad (1)$$

$$f_2(S) := C(S) \rightarrow \min \quad (2)$$

$$S \subseteq \{1, \dots, n\} \quad (3)$$

We are interested in the efficient set of this problem, that is the set: $\mathscr{S}_E = \{S \in \{1,...,n\} | \nexists S' \subseteq \{1,\dots,n\} : f_1(S') \geq f_1(S) \wedge f_2(S') < f_2(S) \vee f_1(S') > f_1(S) \wedge f_2(S') \leq f_2(S)\}$ and the Pareto front $\{(f_1(S), f_2(S))^T | S \in \mathscr{S}_E\}$.

A. *Multiobjective metaheuristics*

Two multiobjective evolutionary algorithms (MOEA, or EMOA) are considered as solvers: The first one is the nondominated sorting genetic algorithm (NSGA-II) [5] [5] and the second one is the S-metric selection algorithm

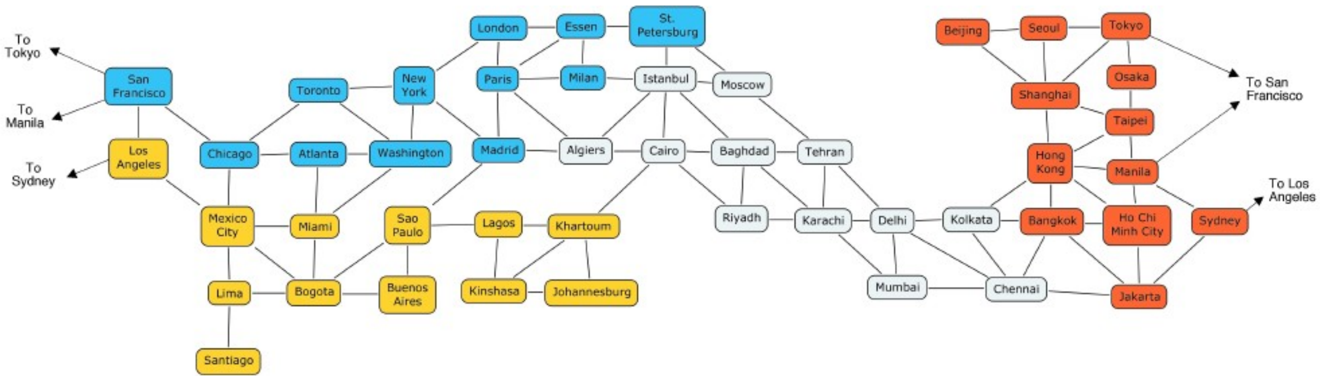
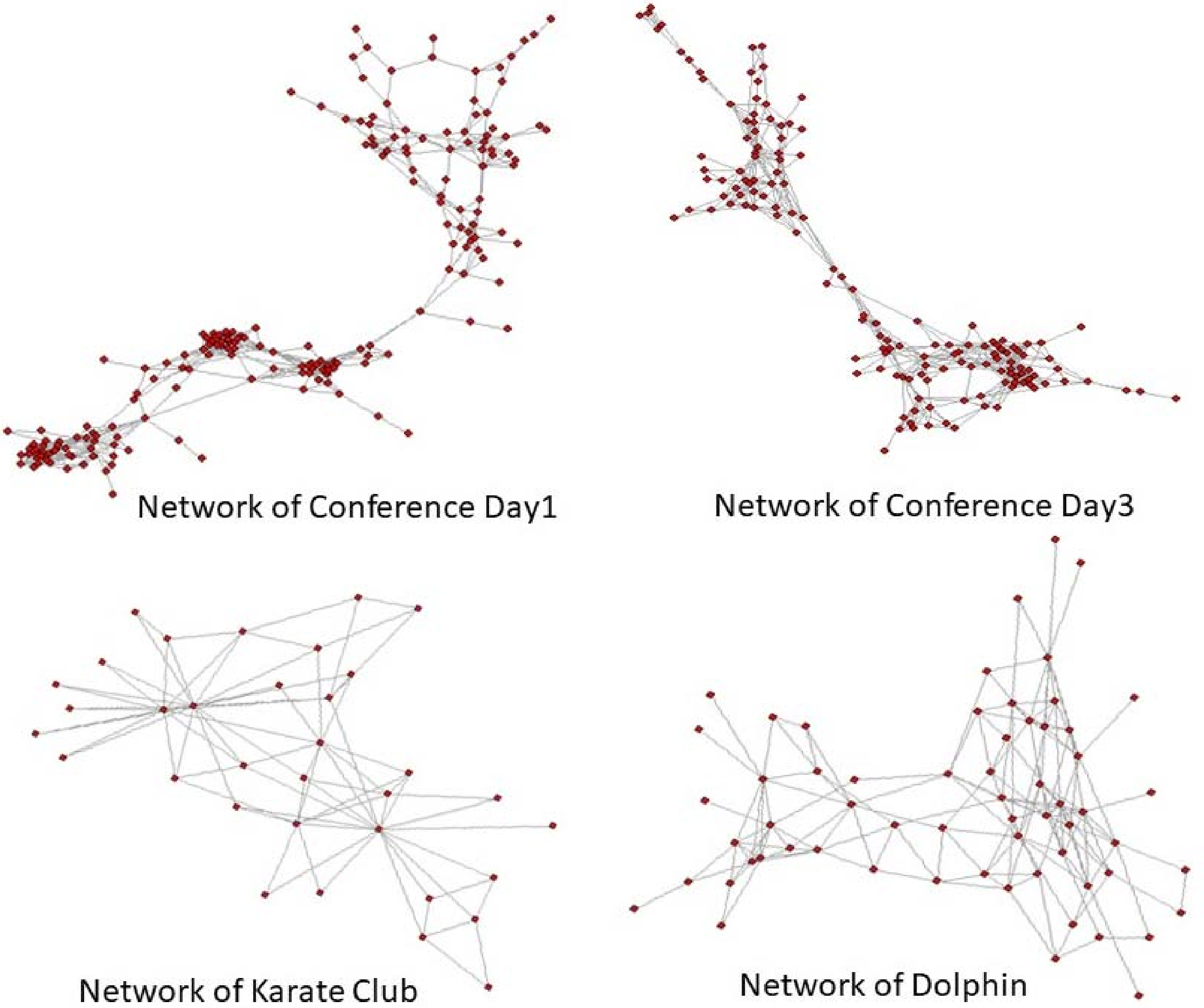


Fig. 3. Network in Pandemic game board (from: http://jhkimrpg.livejournal.com/78787.html)



Network of Conference Day1

Network of Conference Day3

Network of Karate Club

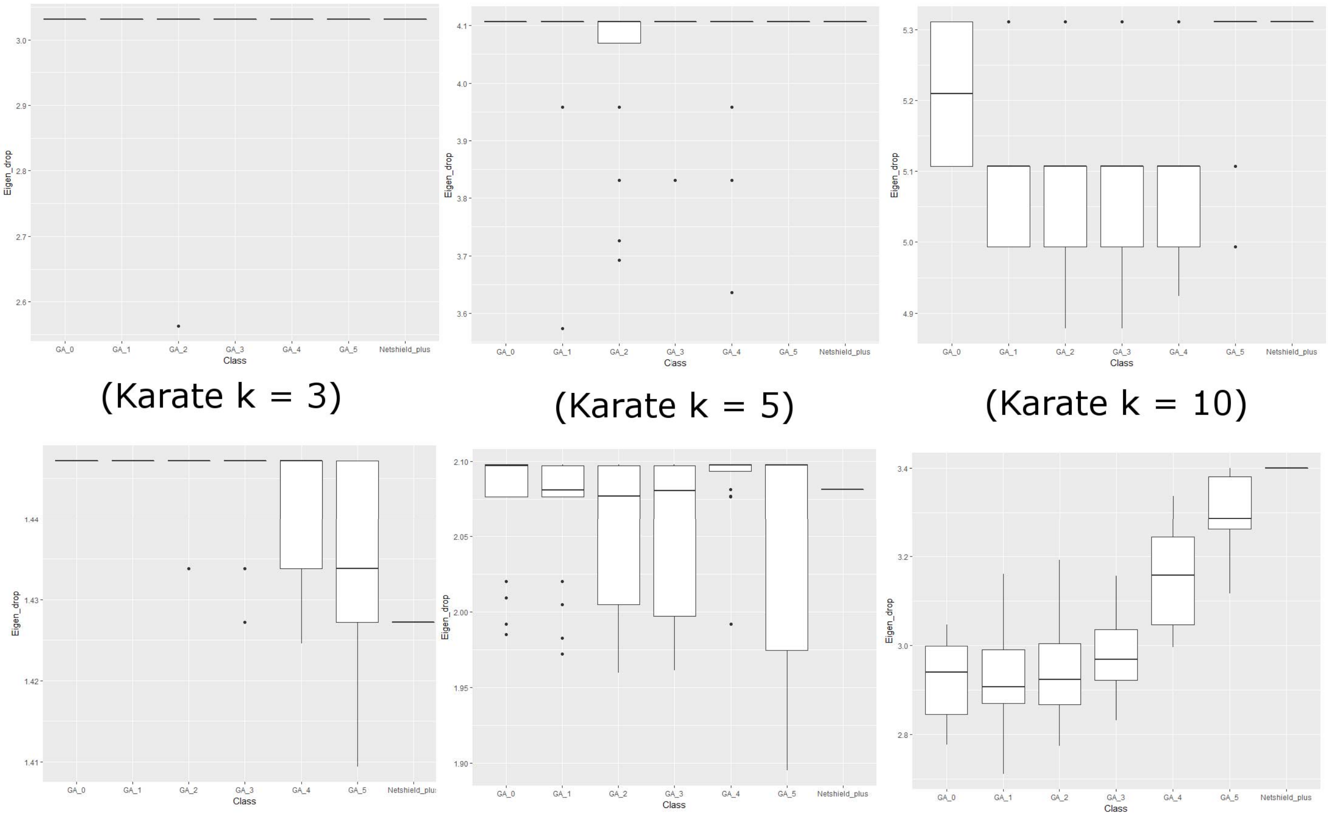
Network of Dolphin

Fig. 4. Depiction of network that we consider for our experiments consist of network of Conference Day1, Conference Day3, Karate Club and network of Dolphin

	Network	*GA_0*	*GA_1*	*GA_2*	*GA_3*	*GA_4*	*GA_5*	*NetshieldPlus*
$K = 5$	*karate*	4.106751	4.106751	4.106751	4.106751	4.106751	4.106751	4.106751
	Dolphins	2.097431	2.081171	2.076892	2.080667	2.097751	2.097751	2.081674
	USA	7.204260	7.204260	7.204260	7.204260	7.204260	7.204260	7.204260
	Pandemic	0.950231	0.924342	0.941925	0.950230	0.950231	0.913314	0.955593
	$Conf.day1$	2.958561	3.010937	2.958320	3.028930	3.045471	3.039115	3.063814
	$Conf.day3$	17.65924	17.670440	17.67136	17.669360	17.669590	17.6098	3.854225
$K = 10$	*karate*	5.209574	5.1076640	5.1076640	5.1076640	5.1076640	5.311484	5.311484
	Dolphins	2.940289	2.907666	2.923008	2.968493	3.157546	3.286219	3.399719
	USA	12.08268	11.689630	11.809470	11.92225	12.177150	12.60777	12.60777
	Pandemic	1.431002	1.420122	1.42991	1.449002	1.511400	1.521515	1.444183
	$Conf.day1$	4.252082	4.3853490	4.383103	4.420728	4.669685	19.237290	4.912077
	$Conf.day3$	17.646370	17.659040	17.664470	17.65775	17.658860	17.658000	5.648326

TABLE I

RESULTS OF GENETIC ALGORITHM AND NETSHIELD PLUS COMPARISONS



(Karate k = 3)

(Karate k = 5)

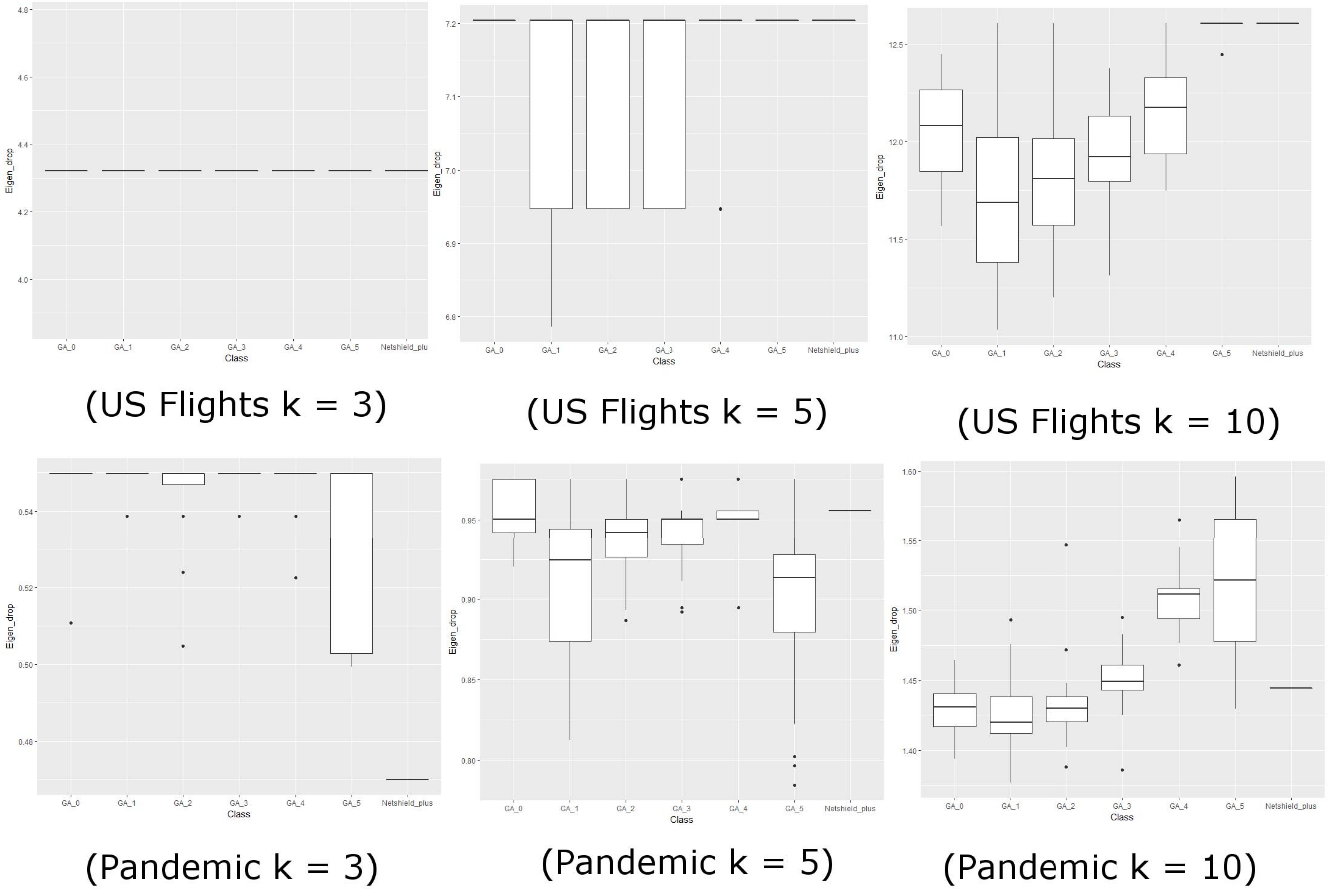
(Karate k = 10)

(Dolphins k = 3)

(Dolphins k = 5)

(Dolphins k = 10)

Fig. 5. Results of genetic algorithm and netshield Plus comparisons



(US Flights k = 3)

(US Flights k = 5)

(US Flights k = 10)

(Pandemic k = 3)

(Pandemic k = 5)

(Pandemic k = 10)

Fig. 6. Results of genetic algorithm and netshield Plus comparisons

ID	City	Population	ID	City	Population
1	*SanFrancisco*	723724	25	*Beijing*	7602069
2	*Chicago*	2830144	26	*Seoul*	9860000
3	*Montreal*	3280123	27	*Tokyo*	8372440
4	*NewYork*	8124427	28	*Shanghai*	15017783
5	*Washington*	548359	29	*HongKong*	7347000
6	*Atlanta*	424096	30	*Taipei*	2491662
7	*Madrid*	3146804	31	*Osaka*	2590815
8	*London*	7489022	32	*Bangkok*	4935988
9	*Paris*	2141839	33	*HoChiMinhCity*	3496586
10	*Essen*	596204	34	*Manila*	10546511
11	*Milan*	1316218	35	*Jakarta*	8556798
12	*St.Petersburg*	4991000	36	*Sydney*	4444513
13	*Algiers*	2029936	37	*Khartoum*	2090001
14	*Istanbul*	10034830	38	*Johannesburg*	2091491
15	*Moscow*	10472629	39	*Kinshasa*	9464000
16	*Cairo*	7836243	40	*Lagos*	9020089
17	*Baghdad*	5753612	41	*SaoPaulo*	10059502
18	*Tehran*	7160094	42	*BuenosAires*	11595183
19	*Delhi*	11215130	43	*Santiago*	4893495
20	*Karachi*	11969284	44	*Lima*	7857121
21	*Riyadh*	4328067	45	*Bogota*	7235084
22	*Mumbai*	18410000	46	*MexicoCity*	8659409
23	*Chennai*	7088000	47	*LosAngeles*	3911500
24	*Kolkata*	4497000	48	*Miami*	386740

TABLE II

COST VALUES FOR PANDEMIC NETWORK (PROPORTIONAL TO CITY SIZE).

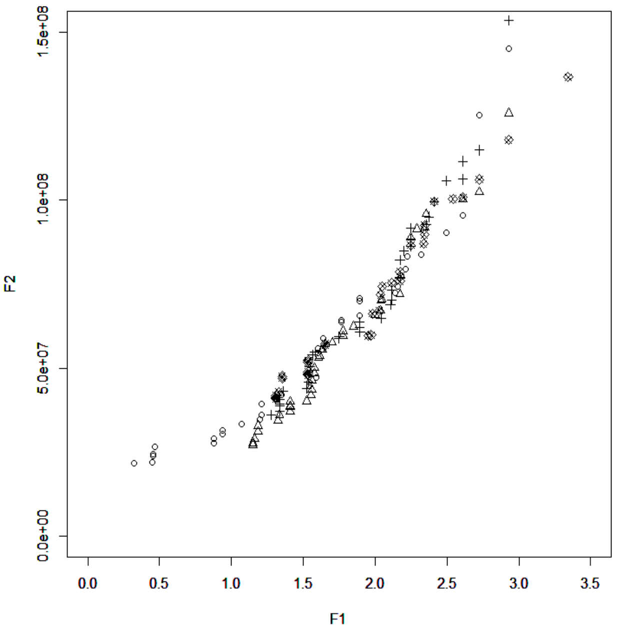


Fig. 7. Pareto Front for Pandemic Network based on NSGA-II algorithm with 5 experiments

Label	*Airport*	*Visits*
42	*Cincinnati/northernKentucky*	117
51	*DetroitMetropolitanWayneCounty*	126
71	*GeorgeBushIntercontinental*	90
81	*Hartsfield − jacksonAtlantaInternational*	102
85	*HopkinsInternational*	123
88	*IndianapolisInternational*	120
106	*KansasCityInternationalAirport*	117
110	*LaGuardia*	123
131	*MemphisInternational*	105
137	*Minneapolis − St.PaulIntl*	135
153	*NashvilleInternational*	108
155	*NewarkLibertyInternational*	123
164	*OrlandoInternational*	84
169	*PhiladelphiaInternational*	120
172	*PittsburghInternational*	120
173	*PortColumbusIntl*	120
174	*PortlandInternational*	138
177	*Raleigh − durhamInternationalAirport*	108
190	*RonaldReaganWashingtonNationalAirport*	117
193	*SaltLakeCityInternational*	123
195	*SanDiegoInternationalAirport*	99
196	*SanFranciscoInternational*	114
201	*Seattle − TacomaInternational*	141
204	*SkyHarborIntl*	99
206	*SouthwestFloridaReg*	81
214	*TampaInternational*	84
224	*WashingtonDullesInternational*	117

TABLE III

COST VALUES FOR PANDEMIC NETWORK (PROPORTIONAL TO CITY SIZE).

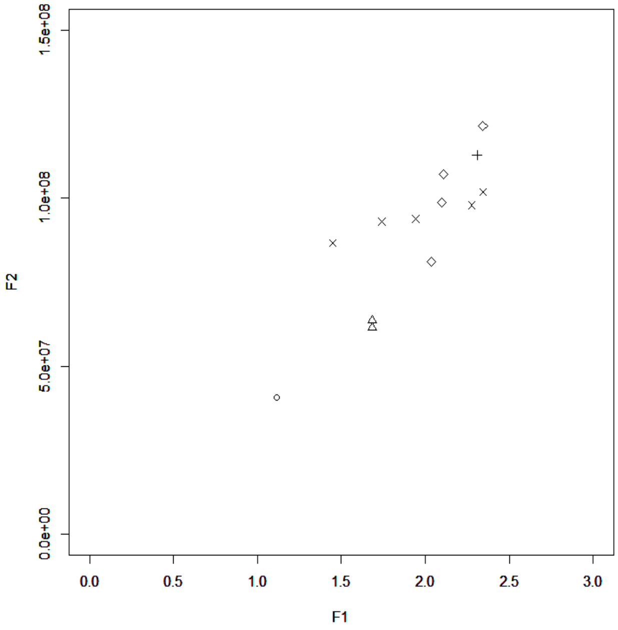


Fig. 8. Pareto Front for Pandemic Network based on SMS-EMOA algorithm with five experiments

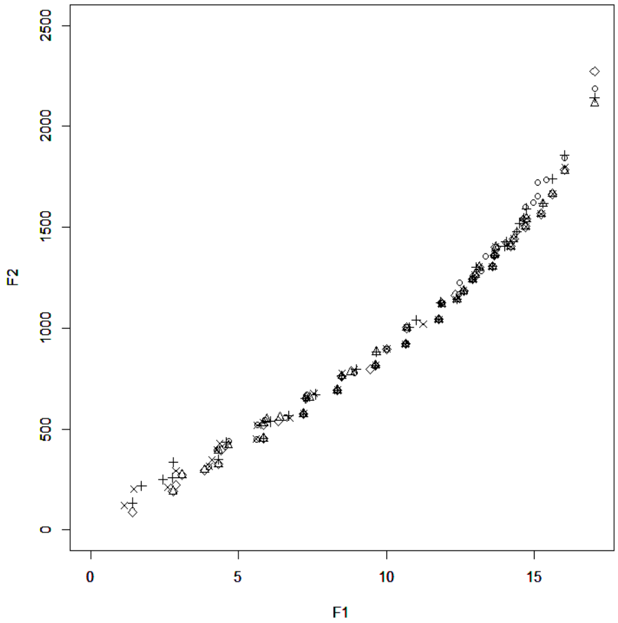


Fig. 9. Pareto Front for USA Flight Network based on NSGA-II algorithm with 5 experiments

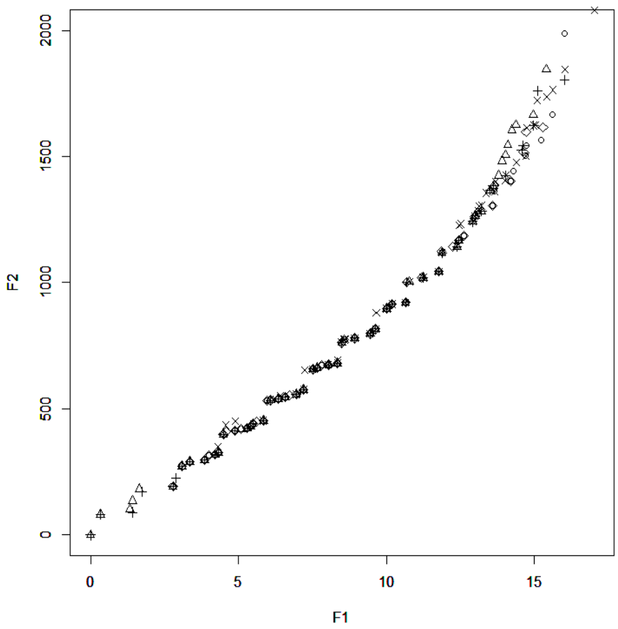


Fig. 10. Pareto Front for USA Flight Network based on SMS-EMOA algorithm with 5 experiments

(SMS-EMOA) [2]. The implementations of SMS-EMOA and NSGA-II in R featured by Bossek's `ecr` package were used in this work. The representation of a subset is chosen to be a bit vector b in $\mathbb{B}^n$, where $b_i = 1$ means the node is selected to be removed/quarantined and $b_i = 0$ means the node is not selected, for $i = 1, \dots, n$. As recombination operator, one point crossover is used. For all bits we used $p_m = 1/n$ as the mutation probability. The reason for this mutation rate is that, in contrast to the single objective genetic algorithms we discussed, here we do not know a priori the number of nodes to remove/quarantine. That is, we do not specify a subset cardinality. As a consequence the algorithm should not try to explore a particular direction of the search space (bias introduced from the mutation operator), but rather present to the decision makers a complete picture of their possible choices. For example, quarantining 10 less-important (in terms of eigen-score) airports could be more beneficial in terms of cost, than quarantining 1 important (in terms of eigen-score) airport.

B. *Empirical Results*

The Pandemic and the US flights networks serve as examples for computing the Pareto fronts and efficient sets. In case of the Pandemic network the size of the cities was used as a cost, assuming that it is more difficult to immunize larger cities. In case of the US flights network the size of the airport (number of visits) was taken into account. The cost values are tabulated in Table II (Pandemic) and Table III (US flight). While we aimed for realistic problem settings, we would like to note that in order to plan effective real-world immunization more modeling is needed, including social interactions, geographic environment, and various other factors. Here, we merely focus on the network aspects of the problem. Each algorithm for the multiobjective optimization was run 5 times, producing 5 Pareto front approximations. Results for Pandemic are shown in Figure 7 and Figure 8. Results for Pandemic are shown in Figure 9 and Figure 10. The Pareto front looks near linear. Overall the NSGA-II algorithm obtained better results and displayed a more robust performance than the indicator based SMS-EMOA on this problem. It is also interesting to node that the Pareto front looks near linear, which might be explained by the fact that big nodes (larger cities or, respectively, airports) are at the same time costly and important for immunization. For the US Flights network a knee region can be identified.

V. Discussion and Outlook

This work has shown that genetic algorithm often perform better, sometimes significantly better, in solving the k-node immunization problem. Netshield plus is a fast heuristic and produces in many cases good results. Based on our findings, a strategy could be recommended that, if time is available, uses not only Netshield Plus but also a problem specific genetic algorithm to make it more likely that the best solution for the eigendrop objective is not overlooked.

In order to achieve good results problem specific adaptations turned out to be very useful. An idea that works well is to use eigen-score values in order to adjust the mutation probabilities. This way the search is more focused on the part of the search space that is more likely to be relevant for solving the problem. We should also emphasize here that the supplementary use of a problem specific genetic algorithm has the advantage of calculating the actual eigen-drop, rather than an approximation of it. This can be useful for moderate sized networks. However, in large networks the computational cost increases, since the algorithm eigendecomposes larger adjacency matrices.

First results were also obtained on a multi-objective formulation of the node immunization problem. We discuss the formulation where the total cost of immunization is one objective and the drop of eigenvalue is a second objective. Two different metaheuristics are applied to solve this problem and they widely agree in the results and show robust performance.[1]

In the future, larger networks could be tested. However, based on our experience this would require further adaptations to the Genetic Algorithm. We believe that a promising path to accomplish this is to hybridize GA with Netshield plus, for instance by using the latter in constructing initial solutions. Moreover, the development of problem specific crossover operators could be beneficial.

References

[1] Thomas Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[2] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[3] 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, 2008.

[4] 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.

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[6] M Leacock. Pandemic.[board game]. *Z-Man Games: Mahopac, NY*, 2008.

[7] Daniel R Lucey and Lawrence O Gostin. The emerging zika pandemic: enhancing preparedness. *Jama*, 315(9):865–866, 2016.

[8] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[9] Aske Plaat. Data science and ebola. *arXiv preprint arXiv:1504.02878*, 2015.

[10] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[11] Jihoon Yang and Vasant Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications*, 13(2):44–49, 1998.

[12] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[1] We remark that the source code of the algorithms and the network data sets will be made available by the authors on the research groups webpage http://moda.liacs.nl.