Evolutionary Graph-based V+E Optimization for Protection Against Epidemics

Krzysztof Michalak krzysztof.michalak@ue.wroc.pl[0000−0003−4994−9930]

Department of Information Technologies, Faculty of Management, Wrocław University of Economics, Wrocław, Poland

Abstract. Protection against spreading threats in networks gives rise to a variety of interesting optimization problems. Among others, vertex protection problems such as the Firefighter Problem and vaccination optimization problem can be tackled. Interestingly, in some cases a networked system can be made more resilient to threats, by changing its connectivity, which motivates the study of another type of optimization problems focused on adapting graph connectivity.

In this paper the above-mentioned approaches are combined, that is both vertex and edge protection is applied in order to stop the threat from spreading. Solutions to the proposed problem are evaluated using different cost functions for protected vertices and edges, motivated by real-life observations regarding the costs of epidemics control.

Instead of making decisions for each of the vertices and edges a decision model is used (based on rules or a neural network) with parameters optimized using an evolutionary algorithm. In the experiments the model using rules was found to perform better than the one based on a neural network.

Keywords: Disease Prevention · Epidemics Control · DPEC · Combinatorial Optimization · Graph-based problems

1 Introduction

A wide variety of phenomena can be described as a spreading of a threat in a certain network. Epidemics, wildfires, floods and even bankruptcies behave in a similar way: a number of entities are affected by a threat which subsequently spreads to other entities in the system. Numerous approaches have been proposed to analyse epidemic processes in complex networks. A review of recent advancements in this area is presented in [26]. When there is a dangerous phenomenon spreading a question naturally arises how to stop this threat in a possibly effective way. This question gives rise to a number of optimization problems in which the goal is to utilize the available countermeasures to stop the threat, taking into account the costs and possible constraints.

1.1 Vertex protection

One of the abstractions that is often studied in the field of research on optimization methods is the Firefighter Problem (FFP) [14] in which spreading of fire is simulated on a graph in discrete time steps. Vertices of the graph can be in one of the states 'B', 'D', 'U' which are interpreted, respectively, as the vertex being on fire, being defended by firefighters or being in an untouched state (neither burning nor defended). In the initial state some vertices are on fire (in the 'B' state) and the remaining ones are most often left untouched (in the 'U' state). Subsequently, the fire spreads from burning vertices to the untouched ones along the edges of the graph, which determine which vertices are adjacent and can therefore catch on fire. The need for optimization in the Firefighter Problem is motivated by the limitation of resources, which is represented by a constraint stating that in any given time step at most N_f vertices can become protected against fire (i.e. be set to the 'D' state).

A very similar problem formulation can be used for tackling epidemics, bankruptcies, etc. In each of these applications neighbourhood of the vertices can be described in a different way. In the FFP and many other problems the ways through which the threat spreads are represented by edges of a graph, but other representations are also possible, for example based on geographical locations and distances such as in the Foot-and-Mouth Disease (FMD) spreading model used in the paper [2]. Also, the contagion dynamics can be different in different problems. In the classical version of the FFP the spreading of fire is deterministic: an untouched vertex is guaranteed to catch on fire in a given time step if one of its neighbours is burning. A version of the FFP with non-deterministic spreading of fire has also been studied [25]. Epidemics are often modelled using compartmental models, such as SIR (Susceptible $\rightarrow$ Infected $\rightarrow$ Recovered), SIS (Susceptible $\rightarrow$ Infected $\rightarrow$ Susceptible) or SIRV model, which, apart from the susceptible, infected and recovered states, allows the entities to be vaccinated [18]. Epidemiological models are most often probabilistic, that is the transitions from one state to another happen with certain probabilities as opposed to deterministic changes in the classical version of the FFP. Yet another threat spreading mechanism was proposed by Burkholz [5] for an economic setting in which companies on the market may go bankrupt, and because of unpaid dues incur stresses on other companies. In this model failures spread deterministically, but in order for a company to fail the total load incurred by its bankrupt cooperators has to exceed a certain threshold.

In each of the above-mentioned scenarios certain actions can be taken in order to prevent the threat from spreading. In the case of the Firefighter Problem N_f vertices can become protected against fire in each time step and the protection is 100% effective (protected vertices remain in the 'D' state until the end of simulation and fire cannot get to them). Because the order in which vertices are protected is essential in the FFP, the most common solution representation uses permutations to represent the order in which vertices should be protected. In the case of epidemiological models the most common protection mechanism is vaccination. If the individuals are vaccinated before the pathogen starts spreading (and so the time of vaccinations does not have to be taken into account) the decisions to vaccinate or not can be represented as a binary vector. Imposing a constraint on the total number of vaccinated vertices the K-Node Immunization Problem is formulated and solved using, among others, heuristic [7] and evolutionary methods [20]. For the Burkholz economic model an optimization problem was formulated [23] in which solutions are vectors of real numbers representing thresholds which can be adjusted by allowing companies to store reserves.

1.2 Edge protection and network connectivity

Another approach to network protection is to consider edges instead of vertices [8]. This approach is particularly important in computer network protection [6]. The notion of edge-failure resilience is an actively researched topic in the literature focused on networks protection [19].

It is worth noticing that, as opposed to the computer network protection problem where it is profitable to increase network connectivity, there are optimization problems where the optimum does not coincide with the highest network connectivity. Notably, financial systems can show different level of resilience to shocks of varying magnitude [16] depending on connectivity, a phenomenon that also affects optimization problems for this kind of systems [24]. In the case of counter-epidemic optimization a *lower* connectivity can be expected to produce better outcomes. This effect is the basis of epidemic control strategies based on social distancing [13].

1.3 Overview of this paper

In this paper a combination of the network protection approaches discussed above is studied in a scenario which concerns stopping an epidemic from spreading on a graph. The optimization problem studied here is the problem of optimizing a decision model that determines which action should be taken: protecting a vertex (vaccination) or limiting the number of contacts it makes in the network (isolation). The costs of infections and vaccinations are calculated taking into account the number of affected vertices. The costs of isolation are calculated as the product of the number of the removed edges and the number of time steps the isolation has lasted for. This is motivated by the fact, that contacts in the network represent vital activities undertaken by entities in the system and their removal may, for example, cause income to be lost for businesses with the loss dependent on the time the isolation lasts for.

In the following sections the optimization problem is defined (Section 2), the experiments (Section 3) and results (Section 4) are discussed. Section 5 concludes the paper.

2 Optimization problem

The optimization problem tackled in this paper is a problem of finding a counter-epidemic strategy minimizing several criteria. The epidemic is simulated on a graph $G = \langle V, E \rangle$ in which the vertices represent entities that may become infected and the edges represent contacts. The states of the vertices and the transitions between them are based on the SIVR model [27]. There are four states: 'S' - susceptible, a vertex that is not infected, but may become so; 'I' - infected; 'V' - vaccinated, and thus immune to the disease; and 'R' - a vertex recovered from the disease, which in the SIVR model cannot be infected again.

Changes in the graph occur in discrete time steps $t = 0, \ldots$ and we will denote the state of the graph at time t by $S_t \in \{$ 'S', 'I', 'V', 'R' $\}^{|V|}$ and the state of an individual vertex in the graph $v \in V$ at time t by $S_t[v] \in \{$ 'S', 'I', 'V', 'R' $\}$. The initial state is S_0 in which a fraction α_{inf} of the vertices is infected, so $\alpha_{inf}|V|$ randomly selected vertices are in the state 'I' and the remaining ones are in the state 'S'. A susceptible vertex may become infected if it is adjacent to at least one infected vertex, with the probability of transmitting the disease from each infected neighbour equal to β per a time step. Recovery occurs with the probability γ per a time step. In each time step protective actions can be taken for each susceptible vertex. If the protective action $\mathcal{P}^{(vacc)}(v)$ is applied, the vertex v is vaccinated and changes its state to 'V' in which it remains until the end of the simulation. If the protective action $\mathcal{P}_q^{(isol)}(v)$ is applied, the isolation level of the vertex v is changed, by activating or deactivating edges adjacent to v so that a fraction q of the edges adjacent to v is inactive. The number of deactivated edges for the vertex v is calculated as $\mathrm{Round}(q \cdot k(v))$, where $k(v)$ is the degree of the vertex v and the Round() function rounds the number to the nearest integer. When more, or fewer, edges need to be activated the edges that change the state are selected at random with uniform probability. An activation state of the edge e is denoted by $\mathcal{A}[e]$ with the value of $true$ representing an active edge and the value of $false$ representing an inactive edge. The disease can only spread along active edges. Therefore, applying the protective action $\mathcal{P}_q^{(isol)}(v)$ with $q > 0$ represents a situation when contacts are broken by the vertex v in order to reduce the risk of being infected. In each time step protective actions are applied before the spreading of the disease takes place (cf. Algorithm 1).

Instead of deciding which protective action to apply for each vertex separately, the optimizer adjusts parameters of a decision model Ψ which takes the information about nearby cases of the disease as inputs and returns the decision which action to perform (if any) in a given time step t for the vertex v. A solution to the optimization problem is the vector of parameters $x \in \mathbb{R}^k$ of the decision model Ψ, where the length k of the vector of parameters x depends on the type of the decision model used. The vector of inputs $\phi(v) \in \mathbb{R}^h$ representing the information about nearby cases of the disease contains fractions of infected vertices separated from v by $1, \ldots, h$ edges, where h is the horizon around the vertex v within which infected vertices are detected. The vector $\phi(v)$ is obtained by performing the Breadth-First Search (BFS) [15] around the vertex v. For example in the situation shown in Figure 1 the result is $\phi(v) = [\frac{1}{3}, \frac{2}{3}, \frac{1}{2}]$. At the distance $d = 1$ there are three vertices in the states 'S', 'I', 'V', of which only one is infected, hence $\phi(v)[1] = \frac{1}{3}$. The fourth vertex (marked (1) in the figure) is connected through an inactive edge e, so it is not counted. At the distance

$d = 2$ there is one vertex in the state 'S' and two vertices in the state 'I', hence $\phi(v)[2] = \frac{2}{3}$. The two vertices marked (2) in the figure are separated from v by a vaccinated vertex, so they are not counted. At the distance $d = 3$ the two pairs of vertices marked (3) and (4) are separated from v by infected vertices, so they are not counted. Therefore, only two vertices are taken into account, one susceptible and one infected, and $\phi(v)[3] = \frac{1}{2}$.

Algorithm 1 presents the simulation of the epidemic with selection of protective actions performed using the decision model Ψ. Inputs and outputs of this algorithm are listed in Table 1. Using this simulation procedure three objectives are calculated: N_{inf} - the number of vertices infected during the simulation, N_{vacc} - the number of vertices vaccinated during the simulation, and N_{isol} - the number of edges cut because of isolation, multiplied by the number of time steps in which the isolation was applied. Because the spreading of the epidemic is non-deterministic, the simulations are repeated N_{sim} times, each time starting from a different set of infected vertices, and the results are averaged. In this paper the number of simulations was set to $N_{sim} = 5$. Therefore, the optimization problem tackled in this paper can be formalized as follows:

$$\begin{aligned} &\text{minimize } (N_{inf}(x), N_{vacc}(x), N_{isol}(x)) = F(x) \in \mathbb{R}^3 \\ &\text{subject to } x \in \mathbb{R}^k, \end{aligned} \tag{1}$$

where:

k - the number of parameters of the decision model Ψ.

Table 1: Inputs and outputs of Algorithm 1.

Inputs:	
$G = \langle V, E \rangle$	- the graph on which the epidemic is simulated
Ψ	- the decision model used for selecting protective actions
h	- the radius of the horizon in which to count infected vertices
α_{inf}	- the fraction of initially infected vertices
x	- a solution to evaluate (a vector of parameters for the decision model Ψ), $x \in \mathbb{R}^k$
Output:	
$F(x)$	- the vector of objectives $F(x) = (N_{inf}, N_{vacc}, N_{isol})$

3 Experiments

In the experiments evolutionary multiobjective optimization was carried out on instances of the optimization problem described in Section 2. The parameters for the spread of the epidemic were: the infected fraction $\alpha_{inf} = 0.01$, the transmission probability $\beta = 0.5$ and the recovery probability $\gamma = 0.1$. A small fraction

Algorithm 1: Evaluation of a solution to the optimization problem by simulating the epidemic (inputs and outputs: see Table 1).

```
// Initialize the simulation
S := RandomlyInfected(α_inf |V|)
for e ∈ E do
    A[e] := true

// Main simulation loop
finished := false
while finished = false do
    // Isolation costs
    for e ∈ E do
        if A[e] = false then
            N_isol := N_isol + 1

    // Protective actions
    S' := S
    for v ∈ V s.t. S[v] = 'S' do
        φ(v) = BFS(G, S, v, h)
        P := Ψ(φ(v), x)

        // Vaccination
        if P is P^(vacc) then
            S'[v] := 'V'
            N_vacc := N_vacc + 1

        // Isolation
        if P is P_q^(isol) then
            SetIsolationLevel(q)

    // Spreading of the epidemic
    finished := true
    S := S'
    for v ∈ V s.t. S[v] = 'I' do
        for w ∈ V s.t. ⟨v,w⟩ ∈ E and A[⟨v,w⟩] = true do
            finished := false
            if Random(U[0,1])< β then
                S'[w] := 'I'

    // Recovery
    for v ∈ V s.t. S[v] = 'I' do
        if Random(U[0,1])< γ then
            S'[w] := 'R'

    S := S'
for v ∈ V s.t. S[v] = 'I' or S[v] = 'R' do
    N_inf := N_inf + 1;
```

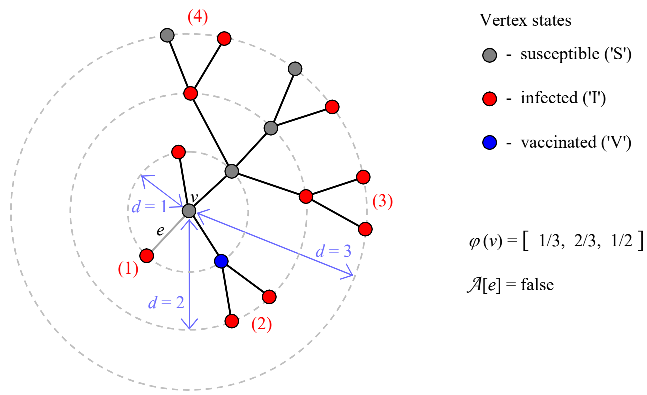

Fig. 1: An example of the calculation of $\phi(v)$ for $h = 3$. See the description in the text.

of initially infected vertices (1%) was chosen in order to simulate a typical scenario in which an epidemic starts from a small group of infected individuals. The transmission probability $\beta = 0.5$ is a value based on the literature [4]. From the properties of the geometric distribution [12] (which gives the probability that the first occurrence of a success requires t_{inf} independent trials, each with success probability γ) it follows that $\gamma = 0.1$ translates to the expected duration of the infection of $t_{inf} = 10$ time steps. Thus, the value $\gamma = 0.1$ ensures, that an infected individual remains infected long enough to spread the disease.

3.1 REDS graphs

Each optimization problem instance is based on a graph, so it is necessary to decide what type of graphs to use, and select the number of vertices. In this paper REDS graphs were used [1]. In REDS graphs the vertices are placed on the unit square $[0, 1] \times [0, 1]$ and the generation of edges is controlled by three parameters: R, E, and S. The radius R determines the maximum distance at which the addition of a new edge is possible. Social energy E imposes a limit on how many connections a vertex can make (each edge costs D which is equal to this edge's length). The cost of an edge between vertices v_i and v_j is discounted by the factor of $\frac{1}{1+Sk_{ij}}$, where k is the number of common neighbours the vertices v_i and v_j have. Because common neighbours cause the cost of creating new edges to be discounted, REDS graphs show a structure similar to a real-life social network with tightly connected groups separated by less crowded spaces. Because of varying density of edges, in REDS graphs communities are formed, even if the vertices are uniformly placed on the unit square. The instances used in the experiments described in this paper were generated using graph parameters shown in Table 2. The last column shows the average vertex degree $\overline{k}$ which is not adjustable and was calculated from the graphs that were generated using the remaining parameters.

Table 2: Parameters of graphs on which test instances were based.

N_v	R	E	S	$\bar{k}$
1000	0.1000	0.15	0.5000	7.35
1250	0.0890	0.15	0.4470	7.97
1500	0.0820	0.15	0.4080	8.11
1750	0.0760	0.15	0.3780	8.57
2000	0.0700	0.15	0.3500	9.16
2250	0.0670	0.15	0.3330	9.43
2500	0.0630	0.15	0.3160	10.09

3.2 Decision models

As described in Section 2 the optimization algorithm searches for Pareto-optimal vectors of parameters $x \in \mathbb{R}^k$ which are subsequently used by a decision model Ψ to make decisions about which protective action to apply. In this paper two different models were tested: *a rule-based model* and *a neural model*. Both models take as input the vector $\phi(v) \in \mathbb{R}^h$ which contains fractions of infected vertices around a vertex v, thereby representing the information about nearby cases of the disease. The set of protective actions used in this paper is $\mathbb{P} = \{\mathcal{P}^{(none)}, \mathcal{P}^{(vacc)}, \mathcal{P}^{(isol)}_{0.25}, \mathcal{P}^{(isol)}_{0.50}, \mathcal{P}^{(isol)}_{0.75}, \mathcal{P}^{(isol)}_{1.00}\}$. Therefore, the model can decide to take no action, vaccinate the vertex v, or to apply one of four levels of isolation, ranging from breaking $\frac{1}{4}$ of contacts, up to a total isolation. From the machine learning perspective, the model Ψ is a classifier $\Psi : \mathbb{R}^k \to \mathbb{P}$.

Rule-based model

The rule-based model consists of five rules, one for each of the actions $\mathcal{P}^{(vacc)}$, $\mathcal{P}^{(isol)}_{0.25}$, $\mathcal{P}^{(isol)}_{0.50}$, $\mathcal{P}^{(isol)}_{0.75}$, $\mathcal{P}^{(isol)}_{1.00}\}$. The conditional part of the r-th rule ($r = 1, \ldots, 5$) is:

$$w_{r,1}\phi(v)[1] + w_{r,2}\phi(v)[2] + \ldots + w_{r,h}\phi(v)[h] > \Theta_r \quad , \tag{2}$$

where:

r - the number of the rule,

$w_{r,d}$, for $d = 1, \ldots, h$ - the weight assigned to the value of $\phi(v)[d]$, which contains the fraction of infected vertices at the distance of d edges from the vertex v,

h - the maximum distance (horizon radius) from v at which the fraction of infected vertices is calculated,

Θ_r - the threshold at which the rule activates.

For the rule-based model the number of parameters for one rule is $h + 1$ (h weights and one threshold) and for all the rules it is $k = 5(h + 1)$. In the experiments the horizon was set to $h = 3$ which resulted in the number of parameters for the rule-based model equal to $k = 20$. The rule-based model is applied by calculating the left-hand sides of the rules (LHS) and comparing to the thresholds on the right-hand sides (RHS). The selected protective action is the first one for which LHS > RHS in the order of precedence: $\mathcal{P}^{(vacc)}$, $\mathcal{P}^{(isol)}_{1.00}$, $\mathcal{P}^{(isol)}_{0.75}$, $\mathcal{P}^{(isol)}_{0.50}$, $\mathcal{P}^{(isol)}_{0.25}$. If no rule activates the $\mathcal{P}^{(none)}$ action is selected.

Neural model

The neural model is a three-layer neural network [3]. The number of input neurons is $N_{in} = h$, the number of hidden neurons N_{hid} determines the size of the network, and the number of output neurons N_{out} has to be equal to the number of protective actions. The number of parameters is equal to the number of elements in weight matrices and bias vectors of the neural network $k = N_{in} \cdot N_{hid} + N_{hid} + N_{hid} \cdot N_{out} + N_{out}$. In the experiments the number of hidden neurons was set to $N_{hid} = 5$, so the number of parameters for the neural model was $k = 3 \cdot 5 + 5 + 5 \cdot 6 + 6 = 56$.

3.3 Evolutionary algorithm

For optimization of parameters of the models the MOEA/D algorithm [17] with the Tchebycheff decomposition was used in which the objectives were normalized by dividing the value of the i-th objective by the difference between the worst and the best value of this objective in the population. The population consisted of real vectors in $\mathbb{R}^k$ with $k = 20$ for the rule-based model and $k = 56$ for the neural model. The algorithm used four crossover operators: uniform, single-point, two-point and Simulated Binary Crossover (SBX) [9], and seven mutation operators: uniform, displacement, insertion, inversion, scramble, transpose and polynomial mutation [10]. The distribution index for the SBX and for the polynomial mutation was set to $\eta = 20$. For deciding which operator to use a mechanism for autoadaptation of operator probability based on success rates of the operators [22] was used. This approach was chosen following previous works on the Firefighter Problem [21, 22], which is also a graph-based problem in which vertices have to protected from a spreading threat. The population size (which has to be a triangular number for the MOEA/D working on a three-objective problem) was set to $N_{pop} = 300$ and the stopping criterion was the number of solutions evaluations $max_{FE} = 10000$. The neighbourhood size T and the probabilities of applying the operators P_{cross} and P_{mut} were tuned using the grid-search approach separately for rule-based and neural decision models, with the candidate values $T \in \{ 20, 30, 40, 50 \}$, $P_{cross} \in \{ 0.2, 0.4, 0.6, 0.8, 1.0 \}$, and $P_{mut} \in \{ 0.02, 0.04, 0.06, 0.08, 0.10 \}$. For the rule-based model the values $T = 40$, $P_{cross} = 1.0$, and $P_{mut} = 0.06$ were selected, and for the neural model the values $T = 30$, $P_{cross} = 0.6$, and $P_{mut} = 0.02$ were selected. The parameters were tuned on 30 optimization problem instances with $|V| = 1000$ vertices, which were separate from the ones used in the rest of the experiments to avoid overfitting.

4 Results

In the experiments 30 runs of the evolutionary algorithm with each of the decision models described in Section 3.2 were performed for each graph size $|V|$ ranging from 1000 to 2500. For each Pareto front produced by the optimization algorithm the value of the hypervolume indicator [28] was calculated. The hypervolume is often used in the literature to evaluate Pareto fronts, because, as shown by Fleischer [11], maximizing the hypervolume is equivalent to achieving Pareto optimality. From the 30 runs for each algorithm and each graph size $|V|$ the median value was calculated. These median results are presented in Table 3.

Table 3: Median hypervolume for the Pareto fronts produced by the tested methods obtained for $max_{FE} = 10000$. Better (larger) of the two values for a given $|V|$ is underlined.

$\|V\|$	HV: MLP	HV: Rules	p-value
1000	$6.861{\cdot}10^{10}$	$\underline{6.865{\cdot}10^{10}}$	$6.32{\cdot}10^{-5}$
1250	$1.447{\cdot}10^{11}$	$\underline{1.448{\cdot}10^{11}}$	$4.86{\cdot}10^{-5}$
1500	$2.482{\cdot}10^{11}$	$\underline{2.485{\cdot}10^{11}}$	$5.79{\cdot}10^{-5}$
1750	$4.689{\cdot}10^{11}$	$\underline{4.693{\cdot}10^{11}}$	$7.51{\cdot}10^{-5}$
2000	$7.279{\cdot}10^{11}$	$\underline{7.296{\cdot}10^{11}}$	$7.69{\cdot}10^{-6}$
2250	$1.113{\cdot}10^{12}$	$\underline{1.115{\cdot}10^{12}}$	$6.34{\cdot}10^{-6}$
2500	$1.652{\cdot}10^{12}$	$\underline{1.655{\cdot}10^{12}}$	$8.92{\cdot}10^{-5}$

For each value of $|V|$ a Wilcoxon statistical test was performed in order to verify statistical significance of the results. The null hypothesis of the Wilcoxon test states the equality of medians and thus low p-values (here, all below 10^{-4}) signify that the median hypervolume values are significantly different. The Family-Wise Error Rate (FWER) calculated as $1 - \prod_1^7 (1 - p_i)$ for the tests performed for all the graph sizes $|V|$ from the p-values shown in Table 3 is FWER = 0.00034798. Presented results show that the rule-based model outperformed the neural model for all the tested graph sizes $|V|$.

Another comparison was performed by plotting the median hypervolume calculated from the 30 runs with respect to the number of solution evaluations. Values obtained for $|V| = 1000$ are presented in Figure 2 and for $|V| = 2500$ in Figure 3. Plots presented in Figures 2 and 3 show that the rule-based model performs better than the neural model even for small number of solution evaluations.

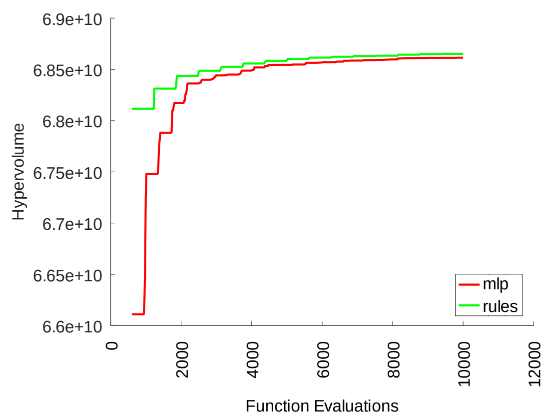

Fig. 2: The median hypervolume calculated from the 30 runs with respect to the number of solution evaluations for $|V| = 1000$.

Below, an example of the rules is given, which produced the best results with respect to the N_{inf} criterion (minimizing the number of infected vertices) for $|V| = 1000$.

if $0.966\phi(v)[1] + 0.252\phi(v)[2] + 0.242\phi(v)[3] > 0.045$ **then** $\mathcal{P}^{(vacc)}(v)$
if $0.489\phi(v)[1] + 0.829\phi(v)[2] + 0.046\phi(v)[3] > 0.202$ **then** $\mathcal{P}^{(isol)}_{0.25}(v)$
if $0.180\phi(v)[1] + 0.969\phi(v)[2] + 0.317\phi(v)[3] > 0.603$ **then** $\mathcal{P}^{(isol)}_{0.50}(v)$
if $0.671\phi(v)[1] + 0.148\phi(v)[2] + 0.313\phi(v)[3] > 0.804$ **then** $\mathcal{P}^{(isol)}_{0.75}(v)$
if $0.987\phi(v)[1] + 0.150\phi(v)[2] + 0.136\phi(v)[3] > 0.044$ **then** $\mathcal{P}^{(isol)}_{1.00}(v)$

It can be observed that for decisions concerning vaccination of vertex v the most important is the number of infected vertices adjacent to v ($d = 1$). This can be explained by the fact, that in the studied epidemic model vaccinations are immediately effective. Similarly, the closest contacts are the most important when decisions concern the introduction of the quarantine (protective action $\mathcal{P}^{(isol)}_{1.00}(v)$ which cuts off all the edges adjacent to v).

5 Conclusion

In this paper evolutionary optimization of counter-epidemic strategies was studied. The optimization problem tackled in this paper involves vaccinating vertices

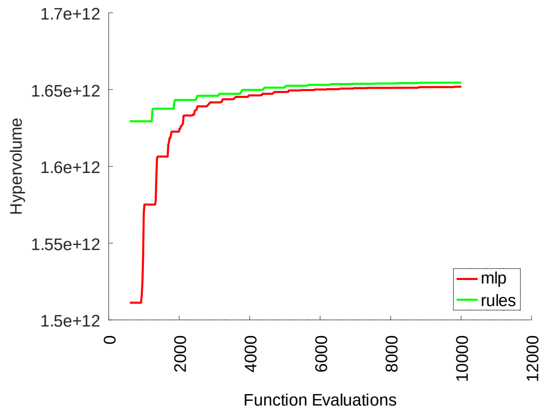

Fig. 3: The median hypervolume calculated from the 30 runs with respect to the number of solution evaluations for $|V| = 2500$.

and/or inactivating edges in the graph, thereby putting vertices in isolation. Instead of directly working on a set (V+E) of both vertices and edges and making individual decisions the counter-epidemic strategy is based on a decision model that selects protective actions. The evolutionary algorithm optimizes the parameters of this decision model which is subsequently used for deciding whether to vaccinate a vertex or to use isolation instead (and if so, what fraction of the edges adjacent to this vertex to inactivate). In the paper two decision models were tested: a rule-based one and a neural one. The rule-based model outperformed the neural one in tests on optimization problem instances based on REDS graphs with the number of vertices $|V|$ ranging from 1000 to 2500.

The optimization of decision models was studied with the models optimized for each particular problem instance. The models used in the paper are machine learning models and can be expected to show the generalization ability, that is to solve new problem instances when trained on other problem instances. Therefore, further work can be directed towards utilizing this generalization ability, for example by training the models on some problem instances and reusing them on other, possibly larger, problem instances. Another possibility could be to study the influence of the graph parameters (the R, E, and S parameters for the REDS graphs) on the quality of the results attained by the models.

Acknowledgment

This work was supported by the Polish National Science Centre under grant no. 2015/19/D/HS4/02574. Calculations have been carried out using resources provided by Wroclaw Centre for Networking and Supercomputing (http://wcss.pl), grant No. 407.

References

1. Antonioni, A., Bullock, S., Tomassini, M.: REDS: an energy-constrained spatial social network model. In: Lipson, H., Sayama, H., Rieffel, J., Risi, S., Doursat, R. (eds.) ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems. MIT Press (2014)
2. Backer, J., Hagenaars, T., Nodelijk, G., van Roermund, H.: Vaccination against foot-and-mouth disease I: Epidemiological consequences. Preventive Veterinary Medicine **107**(1), 27 – 40 (2012)
3. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
4. Britton, T., Janson, S., Martin-Lf, A.: Graphs with specified degree distributions, simple epidemics, and local vaccination strategies. Advances in Applied Probability **39**(4), 922–948 (2007)
5. Burkholz, R., Leduc, M., Garas, A., Schweitzer, F.: Systemic risk in multiplex networks with asymmetric coupling and threshold feedback. Physica D **323-324**, 64–72 (2016)
6. Chekuri, C., Gupta, A., Kumar, A., Naor, J., Raz, D.: Building edge-failure resilient networks. Algorithmica **43**(1), 17–41 (2005)
7. Chen, C., Tong, H., Prakash, B.A., Tsourakakis, C.E., Eliassi-Rad, T., Faloutsos, C., Chau, D.H.: Node immunization on large graphs: Theory and algorithms. IEEE Transactions on Knowledge and Data Engineering **28**(1), 113–126 (2016)
8. Chen, R.L.Y., Phillips, C.A.: k-edge failure resilient network design. Electronic Notes in Discrete Mathematics **41**, 375–382 (2013)
9. Deb, K., Agarwal, R.: Simulated binary crossover for continuous search space. Complex Systems **9**(2), 115–148 (1995)
10. Deb, K., Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design. Computer Science and Informatics **26**, 30–45 (1996)
11. Fleischer, M.: The measure of pareto optima. applications to multi-objective metaheuristics. In: Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003. pp. 519–533. Springer-Verlag, Berlin, Heidelberg (2003)
12. Forbes, C., Evans, M., Hastings, N., Peacock, B.: Geometric distribution. In: Statistical Distributions, chap. 23, pp. 114–116. John Wiley & Sons, Ltd (2010)
13. Glass, R.J., Glass, L.M., Beyeler, W.E., Min, H.J.: Targeted social distancing design for pandemic influenza. Emerging infectious diseases **12**(11), 1671–1681 (2006)
14. Hartnell, B.: Firefighter! An application of domination. In: 20th Conference on Numerical Mathematics and Computing (1995)
15. Kozen, D.C.: Depth-first and breadth-first search. In: The Design and Analysis of Algorithms, pp. 19–24. Springer New York, New York, NY (1992)
16. Ladley, D.: Contagion and risk-sharing on the inter-bank market. Journal of Economic Dynamics and Control **37**(7), 1384–1400 (2013)
17. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Trans. Evol. Comput. **13**(2), 284–302 (2009)
18. Martcheva, M.: Introduction to epidemic modeling. In: An Introduction to Mathematical Epidemiology, Texts in Applied Mathematics, vol. 61, pp. 9–31. Springer Science+Business Media, New York (2015)
19. Matthews, L.R., Gounaris, C.E., Kevrekidis, I.G.: Designing networks with resiliency to edge failures using two-stage robust optimization. European Journal of Operational Research **279**(3), 704–720 (2019)

20. Maulana, A., Kefalas, M., Emmerich, M.T.M.: Immunization of networks using genetic algorithms and multiobjective metaheuristics. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8. IEEE (2017)
21. Michalak, K.: Auto-adaptation of genetic operators for multi-objective optimization in the firefighter problem. In: Corchado, E., Lozano, J.A., Quintián, H., Yin, H. (eds.) Intelligent Data Engineering and Automated Learning - IDEAL 2014, LNCS, vol. 8669, pp. 484–491. Springer (2014)
22. Michalak, K.: The Sim-EA algorithm with operator autoadaptation for the multiobjective firefighter problem. In: Ochoa, G., Chicano, F. (eds.) Evolutionary Computation in Combinatorial Optimization, LNCS, vol. 9026, pp. 184–196. Springer (2015)
23. Michalak, K.: Reducing systemic risk in multiplex networks using evolutionary optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 289–290. GECCO '17, ACM, New York, NY, USA (2017)
24. Michalak, K.: Surrogate-based optimization for reduction of contagion susceptibility in financial systems. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1266–1274. GECCO '19, ACM, New York, NY, USA (2019)
25. Michalak, K., Knowles, J.D.: Simheuristics for the multiobjective nondeterministic firefighter problem in a time-constrained setting. In: Squillero, G., Burelli, P. (eds.) Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part II, pp. 248–265. Springer International Publishing (2016)
26. Pastor-Satorras, R., Castellano, C., Van Mieghem, P., Vespignani, A.: Epidemic processes in complex networks. Reviews of Modern Physics **87**, 925–979 (2015)
27. Tornatore, E., Vetro, P., Buccellato, S.M.: SIVR epidemic model with stochastic perturbation. Neural Computing and Applications **24**(2), 309–315 (2014)
28. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation **7**, 117–132 (2002)