Vaccinating a Population is a Programming Problem

James Alexander Hughes
St. Francis Xavier University
Computer Science
Antigonish, NS, Canada
jhughes@stfx.ca

Michael Dubé
Brock University
Computer Science
St. Catharines, ON, Canada
md12ol@brocku.ca

Sheridan Houghten
Brock University
Computer Science
St. Catharines, ON, Canada
shoughten@brocku.ca

Daniel Ashlock
University of Guelph
Mathematics and Statistics
Guelph, ON, Canada
dashlock@uoguelph.ca

***Abstract*—It is important to understand how best to apply a limited number of vaccines to a population such that the spread of a disease, like SARS-CoV-2, is minimized. Although intuition provides a number of mitigation strategies that may be effective, they remain largely untested.**

A system was developed to test a given disease mitigation strategy. It is designed to work with a graph representing a real social network. A Genetic Programming system was used to discover novel mitigation strategies that are easily interpretable by a public health decision maker.

Effective strategies were developed by the GP system. The strategies are easily explainable and intuitive. Novel mitigation strategies were compared to simple baseline strategies with varying success using a number of different metrics. Many of these strategies proved effective in general, however the topology of the graph influences the effectiveness of a strategy.

The system has been made publicly available and the authors call on the research community to contribute their own mitigation strategies and measure their efficacy.

***Index Terms*—COVID-19; Epidemic; Genetic Programming; Optimization; Pandemic; SARS-CoV-2; SEIR Model; Simulation; Vaccinations.**

I. Introduction

If a limited supply of vaccines becomes available during a global pandemic, how best are they applied to a population such that the spread of the disease is minimized?

In the US, the Centers for Disease Control and Prevention (CDC) has procedures for determining how and who to apply vaccinations to within a population [1], [2]. Risk analysis, health economics, implementation issues, and the values of a population are considered, as is demographic information that indicatess relative risks, such as age and pregnancy.

Healthcare workers are considered key individuals to vaccinate as they may be under a larger virus load or they may be working with patients with health conditions that put those patients at higher risk. The identification of other *hot spots* for disease spread are done to protect other larger communities.

While acknowledging the importance of these considerations, aside from the hot spot identification, exceptionally little data-driven work has been done to determine how best vaccines, or other mitigation strategies, could be applied to minimize the spread of a disease given the topology of a community's social network.

Although vaccination is the primary focus, we use the term *mitigation strategy* as the strategies developed may be generalized and inform other forms of mitigation (e.g. public policy, self-isolation).

Zhant *et al.* [23] and Thakare & Mathurkar [19] studied vaccination strategies using information from the structure of personal contact networks and compared them to a random vaccination strategy. This was done to evaluate the impact of the strategies when selecting individuals for vaccination *prior to the start of the epidemic*.

Dubé, Houghten & Ashlock studied random vaccinations, ring vaccinations (vaccinate individuals around a known infectious individual), and high degree vaccination strategies [8]. They evaluated the strategy's effectiveness in reducing the length and spread of a disease throughout a network. The strategies were applied during the time that the epidemic was spreading, for two different scenarios: when the personal contact networks were known ahead of time and static, and when they evolved over time.

A system was developed to test a given mitigation strategy on a social network for the purpose of minimizing the spread of a disease. The epidemic model used to simulate the spread of the disease is the SEIR model [4] as it more accurately fits how COVID-19 manifests and spreads when compared to the SIR model [15]. Genetic programming (GP), a form of artificial intelligence that searches for *programs* via a process inspired by natural evolution [16], was used to discover novel mitigation strategies. Here, the programs being evolved by GP are the mitigation strategies.

Although the system was designed with SARS-CoV-2 virus in mind, it is generalizable and can be used to understand an arbitrary disease, if the relevant disease parameters are known.

The system was developed in Python and uses a number of external libraries. Details of the system are presented throughout the next sections with much of the description found in Section V. Up-to-date software is available on *GitHub*[1].

An explanation of how graphs are used to represent social networks and a number of graph measures used by the system to inform mitigation strategies can be found in Section II. Information on the SEIR model is found in Section III. Details on the GP system implementation, settings, encoding, and language are found in Section IV, with additional details on how the GP system fits into the overall system found in Section V-B.

978-1-7281-9468-4/20/$31.00

[1]https://github.com/convergencelab/eCov-GP

TABLE I: Graph and SEIR Model Settings.

Number of Nodes	500
Edge Probability	0.04
β (Beta)	0.025
γ (Gamma)	0.133
α (Alpha)	0.175
I_0	0.01

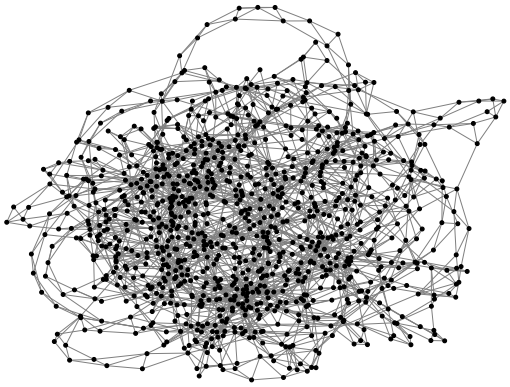

Fig. 1: Example Erdős-Rényi graph with 500 vertices and an edge connection probability of 4%. This provides context for the results presented within this article as they were generated using these types of graphs.

In addition to the creation of the evaluation system, a number of novel mitigation strategies were found. Although preliminary, they proved effective, and despite being perhaps complex, they are explainable and intuitive. Further, it was noted that topology of the graph is important to consider when designing an effective mitigation strategy.

II. Graph

Most of the graph portions of the system are implemented using the *networkx* Python package, software for creating, manipulating, and studying complex graphs [12]. The system is designed to work with an arbitrary graph as it can read an adjacency list. This makes it easy to create or test mitigation strategies for specific social network topologies. For example, if one has access to a graph representing their community, it becomes possible to design tailor-made strategies for that specific graph. This is important as it seems that the spread of a disease through a graph depends on the specific topology [14], [6].

If a graph is not provided to the system, an *Erdős-Rényi* (ER) [9] graph will be created based on provided parameters. Random ER graphs are helpful given their ubiquity and generality, however the choice of ER graphs was somewhat arbitrary. The system can easily be manipulated to create any other form of graph, and future work will include the option to create the perhaps more relevant *Watts-Strogatz* graphs as they better fit small-world properties found in real social networks (low average shortest path lengths and large clustering coefficients) [21].

Due to resource constraints, a random ER graph of size 500 with edge probability of 4% was used (see Table I). This resulted in the vertices having, on average, 20 edges. Figure 1 shows an example ER graph created with these parameters. Larger graphs were explored, but 500 was selected to reduce runtimes for these preliminary results.

A. Graph Measures

Graph measures and information about the pandemic are the primary parameters considered by the mitigation strategy to determine if an individual should be vaccinated. In other words, the mitigation strategy is some function ($f(x_1, x_2, \ldots x_n)$) of these measures that simply returns a Boolean value. Although all the measures listed are included in the system, not all were used when generating the mitigation strategies presented in this article. Those measures that were used are emphasized in boldface. Further, additional measures can easily be added to the system. Computational complexity is presented in terms of the set of vertices V and set of edges E.

1) Static Graph Measures: Static graph measures only need to be calculated once before the execution of an evolutionary search or any simulation of the epidemic/pandemic.

- **Identify Travelers** — *Travelers* are not well defined, but they are intended to represent individuals that connect clusters/communities within the graph. The *Clauset-Newman-Moore Greedy Modularity Maximization* algorithm [5], as implemented within the networkx package, was used to identify clusters. Computational complexity of Clauset-Newman-Moore is $\mathcal{O}(d|E|log|V|)$, where d is the depth of the dendrogram describing the community structure; however, in typical real-world graphs, $\mathcal{O}(|V|log^2|V|)$. Once clusters are identified, arbitrary vertices from within each cluster are identified and minimal cuts (using the *Preflow-Push Algorithm*, $\mathcal{O}(|V|^2\sqrt{|E|})$) are used to identify critical vertices connecting communities. The authors acknowledge that the algorithm, as described, could likely be improved. Note that this measure is only implemented to facilitate the *Is Traveler* measure (described below).
- **Average Degree of Vertices** — Calculate the average degree of all vertices within the graph: $\mathcal{O}(|V|)$. Although this could be approximated for random ER graphs in constant time, this is not the case for arbitrary graphs.

2) Whole Graph Measures: Whole graph measures are required to be run before mitigations are applied.

- **Number of Vertices of a Given State** — Only the number of vertices having the *infected* state are used for the results presented in this article, $\mathcal{O}(1)$ as the simulation keeps track of the vertices in each state.
- Average Distance Between Vertices of a Given State — Implemented with *Dijkstra's Algorithm* between all vertex pairs: $\mathcal{O}(|V|^2|E|+|V|^3log|V|)$. Runtimes are reduced by only considering a small sample of all vertices. Future implementations could use *Floyd-Warshall* to improve

computational complexity. This measure, although implemented, was not used due to the runtime limitations.

3) Local Measures: Local measures must be calculated before mitigations are applied.

- Current Node State — $\mathcal{O}(1)$. Important for future versions of the software as it will have different vaccination applications.
- **Current Node Degree** — $\mathcal{O}(1)$.
- Average Neighbour Degree — $\mathcal{O}(|V|)$, but in practice, $\mathcal{O}(1)$ given that the number of edges each vertex has will be much smaller than the number of vertices in the whole graph.
- **Number of Neighbours of a Given State** — Only the number of neighbours having the *infected* state is used for the results presented. $\mathcal{O}(|V|)$, but in practice, $\mathcal{O}(1)$.
- **Is Traveler** — Return a Boolean value indicating if the vertex is in the traveler set (as described above under *Identify Travelers*): $\mathcal{O}(1)$.

4) Extra-Graph Measures: The extra-graph measures are those that are not directly related to the graph, but provide additional information for the mitigation strategies.

- **Number of Mitigations Currently Available** — $\mathcal{O}(1)$.
- Is Mitigation Available — $\mathcal{O}(1)$. Returns a Boolean Value. Important for future versions of the software.

III. SEIR Model

The Susceptible, Exposed, Infectious, Removed (SEIR) epidemic model was chosen for the system [4]. Unlike the Susceptible, Infectious, Removed (SIR) model [15], SEIR allows for an extended incubation period where an individual has contracted the disease, but is not infectious and has yet to be identified as they have not started showing symptoms. Given the lengthy incubation period of COVID-19, the SEIR model is more appropriate; however, the authors acknowledge that this model is not a perfect match as there is currently strong evidence that presymptomatic and asymptomatic individuals can transmit SARS-CoV-2 [11]. Still, the SEIR model is well defined within the literature and has been a popular model for studying the spread of SARS-CoV-2 [17].

The implementation of the SEIR model found in the Python *Network Diffusion Library* (NDlib) [18] was used, which is built on top of the *networkx* package [12].

The SEIR model has three key parameters having real values $[0, 1]$: β — the probability of transmission at contact of an infectious and susceptible individual, causing the susceptible individual to transition into the exposed state, α — latent period to move from the exposed state to infectious, and γ — probability that an infected individual will transition from the infectious state to the removed state.

The values for these parameters can be found in Table I and are based on those presented by Prem *et al.* [17], however these values are easily changed within the system. The value of β (0.025) is based on early evidence, however this value has different reported values depending in the source. γ is based on a mean duration of 7 days [22]. The α value is based on a mean incubation period of 5.2 days [20], although He *et al.* reported 4 days [13] and Li *et al.* reported roughly 3. The day values were converted to probabilities with $1 - exp(-1/t)$, where t is the number of days.

The last important value for the pandemic model is the number of initial infected individuals (I_0) within the graph. This value was arbitrarily set to 1% (5 individuals in our tests).

TABLE II: GP system parameters.

Hyperparameters	
Population	25
Generations	50
Initialization	Ramped Half-and-Half
Init. Max Depth	4
Crossover	One Point
Crossover Rate	0.75
Mutation	New Sub-Tree Uniformly Applied
Mutation Rate	0.1
Selection	Tournament
Tournament Size	2
Depth Limit	5
Tree Node Limit	32
Language	
Arithmetic Operators	$+$ $-$ $\times^{\dagger}$ $\div$ (protected)†
Boolean Operators	and or not
Comparison and Conditional	$>$ $<$ $==$ If, Then, Else
Constants/Terminals	TRUE/FALSE Integers 0 – 30

† Operators included in the system, but *not* used when generating results presented in this article.

IV. Genetic Programming Implementation

The evolutionary computation framework used for generating the functions to determine if an individual should be vaccinated was *Distributed Evolutionary Algorithms in Python* (DEAP) [7], [10]. Only the GP functionality was used here, however the framework can easily be used for many other forms of evolutionary computation.

The system was used out of the box and the relevant GP system hyperparameter settings can be found in Table II. The authors acknowledge the small population size and low number of generations and these values will be increased for future runs on large computing clusters. Despite such unusually small values, we present our results as interesting mitigation strategies and insights are obtained.

The chromosomes being evolved are functions representing mitigation strategies. The language operators and constants are included in Table II. The variables/function parameters are the graph measures outlined in Section II-A. Therefore, the function tells us, given an individual and the current graph measures, do we vaccinate the individual? The ultimate goal is to find a strategy that, when applied to all admissible vertices (individuals that are not known to have or have had the disease)

throughout the course of the simulation, reduces the spread and/or severity of the disease. See Figure 2 for an example mitigation strategy function (chromosome).

What it means to *reduce the spread and/or severity of the disease* is open to interpretation. Is it best to maximize the number of individuals left as susceptible? Reduce the maximum number infected at any given time, thereby flattening the curve? Minimize the total infected? This problem is naturally multi-objective as there are a number of considerations. Additionally, one may want to minimize the number of vaccinations used, or, reduce the number of ineffective vaccines (discussed further in Section V). Fortunately DEAP makes it easy to play with the objectives used, which gives the user the flexibility to find a strategy suited for their needs. For the results presented below, we maximized the number of individuals left as susceptible at the end of the simulation while also minimizing the number of vaccines used.

The depth and number of nodes within the trees is kept relatively low to maintain function interpretability. This is critical as any strategy would need to be intuitively understood by relevant public health decision makers and stakeholders.

V. Methods

All up-to-date code has been made available online via GitHub. This includes both those used to generate the mitigation strategies and those used to test the strategies.

A. Simulation

The simulation specific parameters that were included in the system at this stage are (a) the number of iterations the simulation lasts (these can be thought of as days), (b) how frequently mitigations can be applied and how often the population can be measured (how often the graph measures can be obtained), and (c) how many mitigations (in our case, vaccinations) are available each time mitigations can be applied. All these values are parameterized and can be changed.

The number of iterations was set to 140 as this would result in the simulation stabilizing for the given graph and SEIR settings (Table I).

Mitigations could be applied to the population once every 7 days. This means that the graphs' measures only needed to be calculated once for every 7 days. This value was chosen as it was deemed reasonable that a population could be evaluated, and new vaccines could be obtained once a week. Note that the first day mitigations could be applied would be on day 7, and not day 0.

Given that the graph the simulations were performed on was of size 500, the number of mitigations (vaccinations) made available was 20, or 4% of the population. Since the simulation lasted 140 days, this resulted in 20 vaccination periods, for a total of 400 vaccines being made available throughout the simulation. If any vaccines were not used during a mitigation period then they could be stockpiled for the next period (a Boolean flag can turn this feature on/off). A restrictive number of vaccines was chosen for these initial tests since there is a real possibility that only a few may be procured, and it was thought that it may lead to particularly interesting results.

Algorithm 1: High-level epidemic/pandemic simulation pseudocode.

```
   input: f:      Function defining a mitigation strategy.
          ITERS: Number of days the simulation lasts.
          MEAS:  Graph measure and mitigation frequency.
          MITS:  Number of mitigations available.
1  for i ← 0 to ITERS do
       ▷ If mitigation day
2      if i != 0 and i % MEAS == 0 then
3          mits_used ← 0
4          susexp ← get_susceptible_exposed()
5          shuffle(susexp)
6          m_global ← global_measures()
7          foreach v ∈ susexp do
               ▷ Mitigations remaining
8              if mits_used < MITS then
9                  m_local ← local_measures()
10                 m_extra ← extra_measures()
                   ▷ If function says mitigate
11                 if f(m_global, m_local, m_extra) then
                       ▷ If effective
12                     if status(v) is susceptible then
13                         set_removed(v)
14                     mits_used ← mits_used + 1
```

Algorithm 1 presents a high-level overview of the simulation. This algorithm assumes that some number of self explanatory functions exist. The algorithm also receives some function f representing the mitigation strategy and the parameters described above.

A number of important observations should be made.

Although the SEIR model does know the difference between susceptible and exposed individuals, we must assume that the simulation does not know the difference. This is done since many jurisdictions will only test individuals if they present symptoms, and exposed individuals are presymptomatic. This is why both susceptible and exposed individuals are considered (line 4). Further, these individuals are shuffled such that the ordering of nodes does not affect the simulation (line 5).

If the mitigation strategy function being used indicates that an individual v should be vaccinated (line 12), then susceptible individuals' states will be changed to removed. However, it is possible that the vaccine is wasted since vaccines are only effective when applied to a susceptible individual, and therefore exposed individuals will remain exposed. Regardless of the state of the individual, if a vaccine is used, it is counted (line 14).

B. Evolution

The simulation, as outlined in Section V-A, is the fitness evaluation for the functions being evolved by the GP system.

After each simulation the graph and SEIR model is reset with different, randomly selected vertices being set to the infectious state. The topology of the graph remains the same because, in addition to random graphs, the system is designed to work on arbitrary graph topologies that are defined by an

adjacency list. Further, mitigation strategies developed should be robust to different starting conditions (initial infectious set) with the same topology.

Since the starting conditions change for each simulation/fitness evaluation, the environment evolution is occurring in changes slightly. Thus, every candidate solution must be reevaluated during each generation, even if the candidate solution was propagated to the next generation without any genetic operators being applied.

C. Testing

Since the results generated for this article used an ER graph, a given strategy was tested in two ways.

First, generate a random graph and perform 100 simulations of the epidemic/pandemic, each with a different starting set of infectious individuals. This is referred to as the *static* tests/results. Note that the graph topology used for static testing was not the same topology used for evolution as those used for evolution were not saved.

Second, generate 100 random ER graphs, each with a different topology and starting set of infectious individuals, and perform a simulation on each. This is referred to as the *non-static* tests/results. This test provides insight into how good a given strategy is in general, regardless of the topology, while also indicating how much the topology of the graph matters for a given strategy.

In addition to the generated mitigation strategies, a number of simple strategies were also included in the tests to provide baseline results for comparison. These included:

- Do nothing
- Random
- Traveler — Intuition being that this would prevent spread between clusters/communities
- Degree 15 or higher
- Degree 20 or higher — The average degree of a node in the graphs used
- Degree 25 or higher

The above list should not be considered comprehensive or exhaustive; future work should include a larger number of baseline strategies for a more effective comparison.

VI. Results and Discussion

A. Noteworthy Strategies Discovered

Hundreds of strategies were developed via evolution, but only a few that obtained particularly effective results were selected for testing, interpretation, and simplification — given the nature of GP, the functions were rather complex but would often be reduced to something simpler (see Section VI-A1).

It was common for the simple baseline strategies to be found with GP. Programs could be reduced to IS_TRAV, $DEG > C$ (some constant), or $TRUE$, which is effectively the random strategy as the individuals are shuffled before the application of any mitigation. Although this is interesting, we will focus on five of the more nuanced functions produced:

- **F1:** IS_TRAV or $((14 - NB\_INF) < (DEG - 8))$

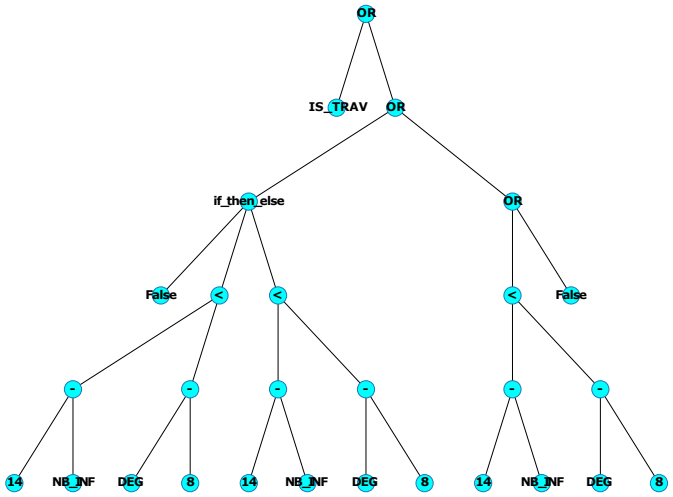

Fig. 2: F1 as derived by GP. This tree can be simplified significantly.

- **F2:** $DEG > 10$ and $NB\_INF > 3$
- **F3:** $NB\_INF > 1$
- **F4:** IS_TRAV or $NB\_INF > 5$
- **F5:** $NUM\_MIT < NB\_INF$

These results suggest that the traveler Boolean, vertex degree, and the number of neighbours that are currently infected are important measures. Any of the integer values present in these functions should probably be interpreted as some constant C that could be further optimized. For simplicity, the values found with evolution will be left alone for analysis.

With the exception of F1, which is investigated in Section VI-A1, the intuition is rather obvious for these functions. Although F5 may seem puzzling at first, considering that the number of mitigations currently available (NUM_MIT) is a relatively small number that will continue to decrease towards 0, it seems unlikely that the number of mitigations is what's actually important. Rather, it is a more sophisticated form of F3 that mitigates individuals with many infected neighbours, but the number of infected neighbours required before a mitigation could be used decreases as more and more mitigations are applied.

1) F1: The most interesting strategy found by GP was F1, which also obtained very high-quality results for the specific ER graph being fit to. The final results had 109 subjects left as susceptible and 215 mitigations used, meaning only a total of 176 people had contracted the disease. Although this is a single result obtained on a single set of starting infectious individuals, this function is likely to be robust to different starting sets since it survived evolution — a new starting set was selected for each fitness evaluation.

Figure 2 shows the final tree structure, and the simplified version of the function can be found in the previous section. This function can be further simplified to IS_TRAV or $NB\_INF > 22 - DEG$. Additionally, the number of infected neighbours is bound by the degree ($NB\_INF \leq DEG$). As stated previously, the value of 22 should be considered

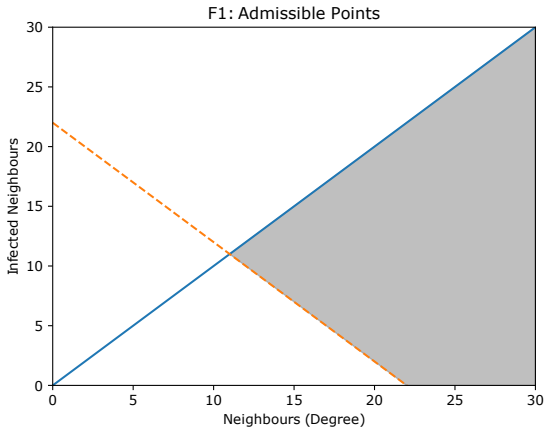

Fig. 3: The admissible values of NB_INF and DEG for F1 are found in the shaded area bound by $NB\_INF > 22 - DEG$ and $NB\_INF \leq DEG$. These lines intersect at $DEG = 11$.

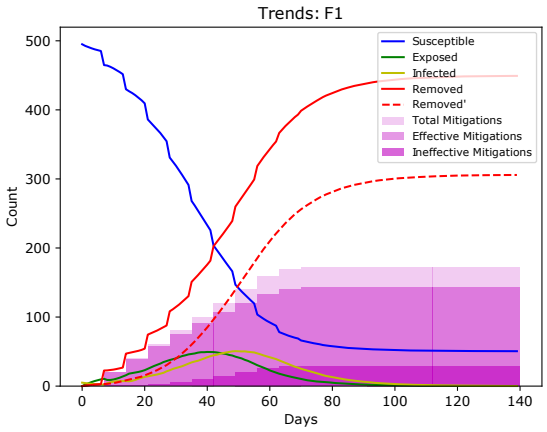

Fig. 4: Example trends curve generated by the system. Unlike typical SEIR trend curves, this includes the number of mitigations used. This particular example is the average results over the 100 non-static tests for F1.

simply as some constant C, however 22 will be used for the interpretation here. Figure 3 shows the admissible values for NB_INF and DEG in the shaded area.

Notice that past degree 12 the number of infected neighbours required before an individual is vaccinated decreases as the degree increases. To gain the intuition, consider that the average degree of a vertex in our graphs was roughly 20. Given this, 12 is a rather small value, therefore it is only worth applying a mitigation to that individual if a high number of those neighbours are infected. As the degree increases, the number of infected neighbours required decreases as the risk of likelihood of an individual spreading the disease grows with the degree of the node. There is even a point where the degree is large enough that no neighbours need be infected ($DEG > 22$).

Also consider the case where, in this example, an individual had degree 10 and all 10 are infected. One may be tempted to vaccinate this individual as they appear likely to contract the disease, however this individual has a high probability of already having been exposed, and even if they are susceptible, it would protect at most one individual. It seems as if this strategy may have developed a way to not waste vaccines by applying them to individuals likely to already be exposed. It is still likely the case that the individual is exposed when $DEG == NB\_INF$ when $DEG > 12$, however, perhaps once the degree is high enough, it is worth the risk of wasting a vaccine. Additionally, as the degree increases, the likelihood of all neighbours being infected would seem low.

B. Comparing Strategies

Table III presents the results obtained from the static tests, and Table IV contains the results from the non-static tests. In these tables, *Total Infected* refers to the area under the infected curve. *Removed'* refers to the number of individuals removed and not vaccinated. *Effective* refers to the number of vaccines that were applied to susceptible individuals and *Ineffective* refers to those applied to those that were already exposed. None of the graph topologies used for testing were those that any evolved strategy were fit to. The emphasized rows correspond to the strategies that are particularly effective on multiple metrics and are the focus of further analysis. Probability values obtained with a *Mann-Whitney U test* comparing the static and non-static results can be found in Table V.

When investigating the results in Table III and IV, it is clear that no one strategy dominates across the multiple objectives, although the simple strategy of $DEG > 20$ performed very well and did dominate in the non-static tests. It is also interesting that 20 happens to be the average degree of the vertices in the graphs used here. Although causality is difficult to confirm with the level of analysis done at this stage, it was the success of this simple strategy that led to the addition of the *average degree* global graph measure to the system as discussed in Section II-A.

There are some observable differences when comparing the static and non-static results; however, most strategies, if they improved when applied to the non-static graphs in some metrics, significantly or not, would perform worse in others, and *vice versa*. For example, the traveler strategy did not improve when applied to different topologies, but the number of mitigations needed decreased.

The GP derived strategies also performed better and worse in different metrics between the static and non-static tests. This shows that the topology of the graph matters a lot. Remember, none of these test results were generated with the graphs the strategies were was actually fit to, where it obtained even better results.

Figure 4 shows the epidemic/pandemic trend curves for the F1 mitigation strategy averaged over all 100 non-static tests. The jagged decrease and increase in the susceptible and removed curves are a result of the mitigations being applied. Removed' is the total number of individuals that had the disease and removed is the number of individuals that had the

TABLE III: Summary statistics of mitigation strategy performance on the *static* graphs with 100 different starting sets of infectious individuals. Results presented are the median and interquartile range.

Strategy	Susceptible		Max Infected		Total Infected		Removed'		Mitigations		Effective		Ineffective	
Nothing	23.0	(± 5.0)	98.0	(± 6.5)	3566.0	(± 108.75)	477.0	(± 5.0)	0.0	(± 0.0)	0.0	(± 0.0)	0.0	(± 0.0)
Random	0.0	(± 0.0)	64.0	(± 8.5)	2414.0	(± 170.88)	321.0	(± 18.38)	215.0	(± 16.5)	179.0	(± 18.38)	35.0	(± 3.5)
Traveler	31.0	(± 6.0)	71.0	(± 6.62)	2953.0	(± 127.12)	394.0	(± 7.0)	79.0	(± 1.5)	76.0	(± 2.5)	3.0	(± 1.12)
Degree 15	17.0	(± 4.5)	62.0	(± 8.0)	2336.0	(± 181.12)	317.5	(± 18.38)	198.0	(± 12.62)	164.0	(± 15.12)	33.0	(± 4.12)
Degree 20	63.0	(± 11.5)	55.5	(± 7.12)	2374.0	(± 208.62)	312.0	(± 17.88)	139.0	(± 6.62)	121.5	(± 10.25)	17.0	(± 3.12)
Degree 25	36.0	(± 6.5)	75.0	(± 7.5)	3114.0	(± 120.0)	416.0	(± 7.5)	51.0	(± 1.0)	49.0	(± 1.5)	2.0	(± 1.0)
F1	53.0	(± 10.62)	58.0	(± 7.5)	2334.5	(± 202.38)	306.5	(± 18.25)	168.0	(± 8.0)	140.0	(± 9.12)	28.0	(± 4.12)
F2	27.0	(± 4.5)	83.0	(± 4.75)	2797.0	(± 122.38)	373.0	(± 8.25)	161.5	(± 6.62)	98.0	(± 5.5)	64.0	(± 4.5)
F3	26.0	(± 7.0)	64.0	(± 8.12)	2343.5	(± 182.5)	321.5	(± 18.62)	196.0	(± 13.62)	150.0	(± 15.0)	46.5	(± 3.62)
F4	37.0	(± 5.62)	69.0	(± 5.62)	2895.0	(± 113.5)	385.0	(± 6.12)	88.0	(± 5.5)	77.5	(± 2.62)	10.0	(± 4.0)
F5	26.0	(± 5.5)	63.0	(± 7.12)	2423.0	(± 165.62)	322.5	(± 13.62)	200.0	(± 10.62)	151.5	(± 11.25)	47.5	(± 5.0)

TABLE IV: Summary statistics of mitigation strategy performance on the *non-static* graphs with 100 different ER graphs. Results presented are the median and interquartile range.

Strategy	Susceptible		Max Infected		Total Infected		Removed'		Mitigations		Effective		Ineffective	
Nothing	22.0	(± 4.5)	98.0	(± 5.5)	3613.5	(± 127.75)	478.0	(± 4.5)	0.0	(± 0.0)	0.0	(± 0.0)	0.0	(± 0.0)
Random	0.0	(± 0.0)	65.0	(± 6.5)	2392.0	(± 156.25)	319.0	(± 14.62)	214.0	(± 13.62)	181.0	(± 14.62)	35.0	(± 3.5)
Traveler	30.0	(± 5.5)	76.0	(± 6.12)	3053.5	(± 120.12)	403.5	(± 11.62)	68.0	(± 8.0)	65.0	(± 7.62)	3.0	(± 1.5)
Degree 15	18.0	(± 5.12)	64.0	(± 7.12)	2363.0	(± 184.88)	313.0	(± 16.5)	199.5	(± 13.12)	165.0	(± 13.75)	34.0	(± 4.5)
Degree 20	53.0	(± 12.62)	57.5	(± 7.62)	2347.5	(± 183.38)	313.0	(± 17.5)	153.0	(± 9.12)	132.5	(± 10.75)	22.0	(± 4.0)
Degree 25	37.0	(± 4.62)	73.5	(± 6.5)	3116.0	(± 123.62)	415.0	(± 6.88)	50.5	(± 4.62)	48.0	(± 4.5)	2.0	(± 1.0)
F1	45.5	(± 9.12)	61.0	(± 9.62)	2390.0	(± 197.75)	313.0	(± 22.38)	170.0	(± 10.0)	140.5	(± 14.5)	29.0	(± 4.5)
F2	28.0	(± 4.62)	80.0	(± 6.62)	2807.0	(± 95.12)	373.5	(± 7.5)	160.0	(± 6.5)	99.0	(± 5.62)	61.0	(± 5.62)
F3	26.0	(± 6.12)	61.0	(± 8.75)	2393.5	(± 205.62)	323.0	(± 16.75)	201.0	(± 11.5)	151.0	(± 11.25)	47.0	(± 5.0)
F4	31.5	(± 6.5)	73.0	(± 6.5)	2919.0	(± 127.88)	390.0	(± 7.62)	94.0	(± 8.62)	78.5	(± 7.12)	14.5	(± 4.5)
F5	25.0	(± 7.75)	64.5	(± 8.0)	2453.0	(± 171.38)	321.0	(± 17.75)	198.0	(± 11.12)	151.5	(± 12.62)	46.5	(± 4.0)

TABLE V: Mann-Whitney U test p-values obtained when comparing the static (Table III) and non-static (Table IV) tests on the various mitigation strategy functions. Values less than 0.05 are emphasized.

Strategy	Susceptible	Max Infected	Total Infected	Removed'	Mitigations	Effective	Ineffective
Nothing	4.40e-01	3.33e-01	2.16e-01	4.52e-01	—	—	—
Random	—	3.92e-01	4.45e-01	2.34e-01	3.00e-01	2.32e-01	2.00e-01
Traveler	5.16e-02	**3.89e-04**	**4.61e-04**	**1.64e-07**	**1.41e-11**	**3.68e-11**	4.15e-01
Degree 15	**3.22e-02**	2.04e-01	2.35e-01	1.73e-01	3.32e-01	3.21e-01	4.47e-01
Degree 20	**2.23e-04**	**3.27e-02**	3.84e-01	4.63e-01	**3.51e-13**	**1.11e-04**	**6.83e-10**
Degree 25	2.25e-01	4.47e-01	4.84e-01	4.94e-01	2.19e-01	2.07e-01	3.71e-01
F1	**1.33e-02**	1.78e-01	4.24e-01	1.26e-01	1.14e-01	3.12e-01	1.33e-01
F2	3.51e-01	**2.67e-02**	2.23e-01	3.07e-01	1.38e-01	2.80e-01	**2.21e-02**
F3	4.33e-01	3.74e-01	2.24e-01	4.51e-01	4.12e-01	4.68e-01	1.97e-01
F4	**8.70e-05**	**2.23e-03**	1.48e-01	**7.39e-03**	**1.05e-04**	2.64e-01	**5.63e-06**
F5	2.68e-01	4.12e-01	4.63e-01	3.52e-01	4.04e-01	4.12e-01	2.99e-01

disease plus the number of individuals that were vaccinated.

Based on these results, it seems that the simpler strategies do well in general, but a more complex function like F1 performs well on certain topologies. This suggests that tailor-made strategies for certain populations may reduce the spread of a disease most effectively for that specific population.

Since the non-static results represent effectiveness more generally, these results will be further analyzed to compare strategies. Figure 5 presents the probability values obtained when comparing each of the four strategies of interest's results. The degree 20 strategy had the most susceptible left by a significant amount and used the least number of mitigations, however it did non significantly outperform F1 for max and total infected.

VII. Conclusions and Future Work

A system to test mitigation strategies for an epidemic or pandemic over a given social network graph was developed and has been made available online via GitHub. Additionally, the system also includes a GP based method to develop novel mitigation strategies.

The GP system was used to discover mitigation strategies for the SARS-CoV-2 virus, and although the authors feel that the results are preliminary, results are reported as they provide interesting insight. Further, given that GP was used to develop the strategies, they were interpretable and intuitive.

Mitigation strategies were developed for a random ER graph and tested in multiple ways. Most generated results proved to work well in general, however it was noted that the success of a given strategy may depend on the topology of the graph, which confirms observations within the literature. In the end, no *silver bullet* strategy was discovered as the more effective methods had varying success across different metrics.

Currently the strategies developed and tested did not use all the mitigations available. Although saving resources (vaccines) is beneficial, using all mitigations may prove to be more important. Having the system apply the remaining mitigations

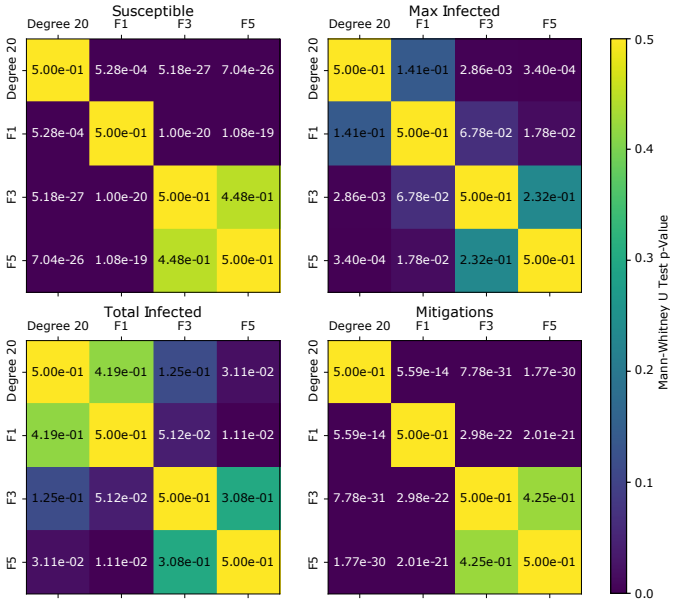

Fig. 5: Probability value matrices comparing various mitigation strategies over four different metrics. Mann-Whitney U tests were used to generate the p-values.

after a mitigation period to the population in some way, perhaps randomly, would likely improve results.

The system is currently designed to easily include additional graph measures. For example, measures such as a minimal vertex cover may be particularly effective for this problem.

Further results will be obtained on static graphs derived from data representing real social networks. This will allow for more accurate results and the ability to design custom tailored strategies for a given community. Although ER graphs were used as a random graph for testing, a more appropriate random graph having better small-world characteristics, like a Watts-Strogatz graph [21], or the scale free *Barabási–Albert* model [3] will be used.

Since the SEIR model is not perfect for SARS-CoV-2, a more sophisticated model may be developed that incorporates infections individuals that are presymptomatic (SEE'IR). Further, including disease parameters for specific demographics, such as recovery or infections rates for older individuals, would increase the accuracy of the model.

A website to host the system and current results for graphs and pandemics will be created. Researchers will be able to submit their functions to the website for automatic evaluation. Results will be stored on the system for future reference.

VIII. Acknowledgments

This research was supported in part by ResearchNS' Nova Scotia COVID-19 Health Coalition.

This research was also supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] ACIP. Advisory Committee on Immunization Practices Policies and Procedures, 2018.

[2] Faruque Ahmed. U.S. Advisory Committee on Immunization Practices Handbook for Developing Evidence-based Recommendations, 2013.

[3] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[4] Joan L Aron and Ira B Schwartz. Seasonality and period-doubling bifurcations in an epidemic model. *Journal of theoretical biology*, 110(4):665–679, 1984.

[5] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[6] MM Telo Da Gama and A Nunes. Epidemics in small world networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 50(1-2):205–208, 2006.

[7] François-Michel De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau, and Christian Gagné. Deap: A python framework for evolutionary algorithms. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 85–92, 2012.

[8] Michael Dubé, Sheridan Houghten, and Daniel Ashlock. Modelling of vaccination strategies for epidemics using evolutionary computation. In *2020 IEEE Congress on Evolutionary Computation*. IEEE, 2020.

[9] P Erdős and A Rényi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

[10] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner Gardner, Marc Parizeau, and Christian Gagné. Deap: Evolutionary algorithms made easy. *The Journal of Machine Learning Research*, 13(1):2171–2175, 2012.

[11] NW Furukawa, JT Brooks, and J Sobel. Evidence supporting transmission of severe acute respiratory syndrome coronavirus 2 while presymptomatic or asymptomatic. *Emerging infectious diseases*, 26(7), 2020.

[12] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[13] Xi He, Eric HY Lau, Peng Wu, Xilong Deng, Jian Wang, Xinxin Hao, Yiu Chung Lau, Jessica Y Wong, Yujuan Guan, Xinghua Tan, et al. Temporal dynamics in viral shedding and transmissibility of covid-19. *Nature medicine*, 26(5):672–675, 2020.

[14] Matt J Keeling and Ken TD Eames. Networks and epidemic models. *Journal of the Royal Society Interface*, 2(4):295–307, 2005.

[15] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[16] John R Koza and John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[17] Kiesha Prem, Yang Liu, Timothy W Russell, Adam J Kucharski, Rosalind M Eggo, Nicholas Davies, Stefan Flasche, Samuel Clifford, Carl AB Pearson, James D Munday, et al. The effect of control strategies to reduce social mixing on outcomes of the covid-19 epidemic in wuhan, china: a modelling study. *The Lancet Public Health*, 2020.

[18] Giulio Rossetti, Letizia Milli, Salvatore Rinzivillo, Alina Sîrbu, Dino Pedreschi, and Fosca Giannotti. Ndlib: a python library to model and analyze diffusion processes over complex networks. *International Journal of Data Science and Analytics*, 5(1):61–79, 2018.

[19] Punam R Thakare and SS Mathurkar. Modeling of epidemic spread by social interactions. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 1320–1324. IEEE, 2016.

[20] Dawei Wang, Bo Hu, Chang Hu, Fangfang Zhu, Xing Liu, Jing Zhang, Binbin Wang, Hui Xiang, Zhenshun Cheng, Yong Xiong, et al. Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in wuhan, china. *Jama*, 323(11):1061–1069, 2020.

[21] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[22] Roman Woelfel et al. Clinical presentation and virological assessment of hospitalized cases of coronavirus disease 2019 in a travel-associated transmission cluster. *MedRxiv*, 2020.

[23] Zhaoyang Zhang, Honggang Wang, Chonggang Wang, and Hua Fang. Modeling epidemics spreading on social contact networks. *IEEE transactions on emerging topics in computing*, 3(3):410–419, 2015.