2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) | 978-1-6654-0112-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/CIBCB49929.2021.9562943

Vaccinating a Population is a *Changing* Programming Problem

Sumaiya Amin
Computer Science
St. Francis Xavier University
Antigonish, NS, Canada
x2020gae@stfx.ca

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

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

***Abstract*—How best to apply vaccines to a population is an open problem. It is trivial to derive intuitive strategies, but until tested, their efficacy i s n ot k nown. T his p roblem i s particularly challenging when considering the dynamics of social contact networks and their changes over time.**

A system for automatically discovering tested vaccination strategies with evolutionary computation has been improved upon to include additional graph metrics and to generate vaccination strategies for dynamic graphs, something that is expected of real social networks within communities.

The system's ability to generate effective strategies was demonstrated along with a comparison of the strategies developed when fit t o a s tatic g raph v ersus a d ynamic g raph. I t w as observed that the additional computational resources required to generate strategies on a dynamic graph may not be necessary as strategies developed for static graphs performed similarly well; however, the authors are careful to acknowledge that results may differ significantly w hen a djusting t he s ystems m any parameters.

***Index Terms*—COVID-19; Dynamic; Epidemic; Genetic Programming; Graphs; SEIR Model; Simulation; Vaccinations.**

I. Introduction

Incorporating social contact network dynamics, and how our communities change over time, may be important factors when searching for the best strategy for applying vaccines to a population such that the impact of an infectious disease is minimized. The issue of including dynamic graphs is perhaps particularly important for low income areas and developing countries as it is typically the case that they must remain relatively mobile [1]. For example, the ability to work from home, buying bulk groceries, unemployment, and reliance on public transit all disproportionately impacts lower income individuals [2].

In previous work, a system for the automated discovery of vaccination strategies for a given social contact network using evolutionary computation was developed [3]. The system simulates epidemics with a given infectious disease model, namely, the SEIR model [4], on a provided graph representing a social contact network. With the use of genetic programming and a variety of graph metrics to use as parameters for the programs being evolved, vaccination strategies are developed with the objective of minimizing the impact of the disease. In this work, additional default metrics are included and the system's functionality is expanded to evolve strategies based on a dynamic graph.

Given the new way of approaching the problem of vaccinating a population that has been developed, minimal related and previous work exists. Most similar work involves the use of evolutionary computation to discover graphs that, when having an epidemic simulated on them, match real-world epidemic profiles [5–8].

Although the main contribution this work provides is the enhanced system, an analysis of the results generated by the system is also provided. As every community will have differences that need to be accounted for in the system's settings, the strategies developed will vary in effectiveness from one community to another. Further, although this work is inspired by COVID-19 and other epidemics, it is broadly applicable to the problem of the distribution of scarce resources.

II. Graph

Social contact networks are easily represented as graphs. Each vertex/node represents an individual, and an edge represents a well-established connection between individuals such that a disease may spread from one to the other[1]. If a community's corresponding graph, or at the very least, an approximation of a community's graph is known, it is possible to design and optimize vaccination strategies for that community.

By using NetworkX [9], the system is designed to work with an arbitrary graph by reading an adjacency list. The system also provides an easy way to generate random graphs. Although it is trivial to add additional random graphs to the system, the random graphs currently implemented within the system are *Erdős-Rényi* (ER) [10], *Newman-Watts-Strogatz* (NWS) [11], *Barabási–Albert* (BA) [12], and *Powerlaw Cluster Graphs* (PCG) (Figure 1) [13].

ER graphs were the primary focus of our previous work [3]. Here the focus is on the PCGs since they are scale-free networks, which are thought to better resemble real-world networks; however, there is evidence that this may not be the case [14]. Regardless, PCGs are an improvement upon ER graphs in this context.

Table I contains the parameters used to generate the PCGs along with some approximate summary statistics of

[1]Here, "community" is not necessarily defined by geography, but is based on well-established connections.

TABLE I: Graph and SEIR model settings. The summary statistics are rough approximations as every randomly generated graph can be quite different. Ranges are provided for some values, but note that these values can vary significantly.

Number of Vertices	500
Number of Edges	4
Triangle Probability	0.66
Average Edge Count	~ 1980
Min. Degree	4
Avg. Degree	~ 8
Max. Degree	$\sim (70 - 120)$
Min. Dist.	1
Avg. Dist.	~ 3
Max. Dist.	5
Min. Clustering Coef.	0
Avg. Clustering Coef.	$\sim (0.31 - 0.33)$
Max. Clustering Coef.	1
Add Edge prob.	0.01
Remove Edge prob.	0.01
Selection Distro.	Uniform
β (Beta)	0.09
γ (Gamma)	0.133
α (Alpha)	6.4
I_0	0.02

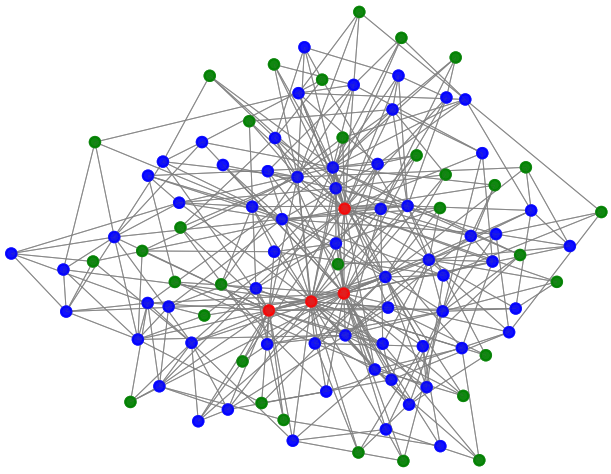

Fig. 1: Powerlaw Cluster Graph generated with 100 vertices, number of edges set to 4, and a probability of creating a triangle of 66%. Red vertices have a degree greater than 25, green have exactly 4, and blue have all other edge counts.

the stochastically generated graph. These values were chosen to create graphs that would last a reasonable duration for simulation while keeping runtimes manageable. Even though the values chosen are not arbitrary, they are not indicative of a particular community. Each community will have different features. Ultimately these are parameterized values for the system that can be tuned for the user's needs.

A. Dynamic Graph

In practice, social contact networks are not static objects. Although much of the network is likely relatively static due to routine, families, friends, public transit routes, work, clubs, etc., some amount of the interactions will change over time. One of the important questions of this work is how much these changes impact the developed vaccination strategies and their efficacy.

There are three core ideas used for making changes to the graph throughout the simulation: (1) adding an edge; (2) removing an edge; (3) selecting vertices to be modified.

For both adding and removing, a predefined value representing the proportion of vertices that will be modified in each step of the simulation is set — the add/remove edge probabilities.

The method used for adding an edge is to select a vertex v, choose a neighbour u, select one of us's neighbours, w, for which edge (v, W) does not already exist and then add the edge (v, w). If no such w exists (i.e. all of u's neighbours have an edge to v) a random vertex x is selected in the graph and the edge (v, x) is added. For ease, if we are unlucky enough for (v, x) to already exist then the add is skipped.

The method used for removing an edge is to select a vertex v, randomly choose a neighbour u, and remove the edge (u, v). For ease, if the edge removal causes the graph to become disconnected, the edge is added back, i.e. we skip that remove.

For each step of the simulation, there is a proportion of the vertices that will be selected for modification; however, not each vertex should necessarily have the same likelihood for being modified. A probability distribution is used to provide control over this vertex selection.

The parameters used for making modifications to the graph are provided in Table I. Since no two communities are the same, it is difficult to select specific values for how much the graph should change throughout a simulation of a pandemic. Therefore, the system was parameterized with these values. Here, 1% was used for the add and remove edge probabilities, but again, this should not be thought of as representative and would need to be adjusted for each situation. Additionally, in this work the vertices were chosen randomly from a uniform distribution for adding/removing edges. However, since communities will have different behaviours during and values, this too is parameterized — the user simply provides an *a priori* probability distribution to the system. This enables the user to simulate different scenarios. For example, should high degree vertices be more likely to remove edges when compared to lower degree vertices?

B. Graph Measures

Since the objective is to develop effective and human interpretable vaccination strategies, they are designed to resemble a *program* or *function*. Like many programs and functions, the vaccination strategies are in need of relevant parameters providing information about the system they operate in and are to make a decision on (e.g. $f(x_1, x_2, \ldots x_n)$), namely, information about the graph and current state of the epidemic. Some measures only need to be calculated once, while others need to be recalculated based on changes in the state of the disease within the graph. All measures are calculated from V, the set of all vertices, and E, the set of all edges defining the graph. Additionally, the simulation step and each vertex's state (susceptible, exposed, infectious, and removed — explained in more detail in Section III) is also used for calculating the graph measures.

TABLE II: Graph Measures implemented within the system. Boldface text represents new measures not present in previous work. Many of the measures are self-explanatory.

Graph Measures	
Static	
Identify Travellers	Vertices connecting clusters
Graph Avg. Degree	Degree of all vertices within the Graph
Min. Vert. Cover	Generate minimum vertex cover of the graph
Shortest Dist. All-All	Generate shortest path matrix
Avg. Short Dist. One-All	Avg. shortest distances of vertices to all others
Avg. Short Dist.	Avg. shortest distance within graph
Short Path All-All	Generate all the shortest paths
Short Path Count	Frequency of each vertex in shortest paths
Page Rank	Calculate page rank vertex weights
Clustering Coefficient	Clustering coefficient of each vertex
Global	
Vertices of a Given State	How many vertices are in a given state
Avg. Dist. Vertices in State†	Avg. distance between vertices in a given state
Local	
Vertex Status	Status of a given vertex
Vertex Degree	Degree of a given vertex
Avg. Neighbour Degree	Avg. degree of the neighbours of a given vertex
Neighbours in State‡	Number of neighbours in a given state
Is Traveler	Is a given vertex classified as a traveller
Dist. to Vert. in State†	Shortest path length to a vertex in a given state
Vertex Avg. Dist.	Avg. distance to all vertices from a given vertex
In min. Vertex Cover	Is a given vertex in the minimum vertex cover
Short Path Count	Frequency of a given vertex in shortest paths
Vertex Page Rank	Page rank weight for a given vertex
Vertex Cluster Coef.	Clustering coefficient for a given vertex
Extra	
Mitigation Count†	Number of mitigations currently available
Is Available†	Are there any remaining mitigations available
Simulation Step†	Current step of the simulation

† Included in system, but not used for generating results reported in this work.
‡ All were used except number of neighbours removed.

All graph measures implemented in the system to date, and a summary of their function, are included in Table II. A detailed description is included below for only the newly added measures. All existing measures' descriptions are available in previous work [3].

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.

A critically important note is that the static measures only stay the same for static graphs — graphs that do not change throughout the epidemic simulation. For the dynamic graphs described, although the duration of our simulations are relatively small, these static measures become outdated as soon as changes are made. Unfortunately, the computational cost of re-calculating the static measures for each change in the graph is prohibitive, thus, only the static measures for the initial state are known throughout the simulation. Further, the graph, and its changes throughout the course of the epidemic, would not be known in practice[2]

- **Minimum Vertex Cover** — The set of vertices in the minimum vertex cover of the graph. Implemented with NetworkX's `min_weighted_vertex_cover` method. This calculation is a greedy approximation using Bar-Yehuda & Evan's strategy [15] and has a computational complexity of $\mathcal{O}(|E|log|V|)$, where $|V|$ is the number of vertices and $|E|$ is the number of edges.
- **Shortest Distance All-All** — Shortest path lengths from each vertex to every other. Implemented using NetworkX's `shortest_path_length` method. Using a min-heap *Dijkstra's Algorithm* implementation, the computational complexity of calculating all shortest paths is $\Theta(|V|(|V|+|E|)log(|V|))$, where $|V|$ is the number of vertices and $|E|$ is the number of edges in the graph.
- **Average Shortest Distance One-All** — Average shortest distance each vertex has to all others. With the pre-computed *Shortest Distance All-All* for simple lookup of shortest path lengths, the computational complexity is $\mathcal{O}(|V|^2)$, where $|V|$ is the number of vertices in the graph.
- **Average Shortest Distance** — Average shortest distance between all vertices. With the pre-computed *Shortest Distance All-All* for simple lookup of shortest path lengths, the computational complexity is $\mathcal{O}(|V|^2)$, where $|V|$ is the number of vertices in the graph.
- **Shortest Path All-All** — The shortest paths from all vertices to all other vertices. Implemented with NetworkX's `shortest_path` method. Using a min-heap *Dijkstra's Algorithm* implementation, the computational complexity of calculating all shortest paths is $\Theta(|V|(|V|+|E|)log(|V|))$, where $|V|$ is the number of vertices and $|E|$ is the number of edges in the graph.
- **Shortest Path Count** — Count the number of times a given vertex exists within a shortest path between any two vertices. With the pre-computed *Shortest Path All-All* for simple lookup of shortest paths, this method has a computational complexity of $\mathcal{O}(|V|^2 * l)$ where $|V|$ is the number of vertices within the graph and l is the length of the longest shortest path within the graph (the graph diameter). The complexity is an upper bound, and assuming the use of a graph representative of a real-world community, l will typically be a small number.
- **Page Rank** — Calculates the page rank weight for each vertex in the graph. Page rank is used as a means to quantify vertex *importance*. Implemented with NetworkX's `pagerank` method. The computational complexity of calculating page rank is dependent on the algorithm used and is an iterative process that converges on a close approximation.
- **Clustering Coefficient** — Calculate the clustering coefficients for all vertices in the graph. The clustering coefficient of a given vertex is the fraction of possible triangles through that vertex that exist. This was implemented with NetworkX's `clustering` method. $\mathcal{O}(|V|^3)$ where $|V|$ is the number of vertices.

2) Local Measures: Local measures must be calculated before vaccinations are applied.

- **Distance to Vertex in State** — Shortest path length from a given vertex to the nearest vertex of a given state. Given the pre-calculated *Shortest Distance All-All*

[2]We remind the reader that the goal is to generate strategies for the duration of a pandemic/epidemic based on full simulations; the goal is *not* real-time vaccination strategy development as the strategies tell us what *would* have been effective.

static measure and the *Vertices in a Given State* whole graph measure calculated for each simulation step, this is a $\mathcal{O}(k)$ operation, where k is the current number of vertices in the graph having the given state.

- **Vertex Average Distance** — Average distance of a vertex to all others. Simple lookup based on pre-calculated *Average Shortest Distance* static measure $\mathcal{O}(1)$.
- **In Minimum Vertex Cover** — Is the given vertex within the minimum vertex cover. Simple lookup based on pre-calculated *Minimum Vertex Cover* static measure $\mathcal{O}(1)$.
- **Vertex Shortest Path Count** — Number of shortest paths between any two vertices a given vertex is in. Simple lookup based on pre-calculated *Shortest Path Count* static measure $\mathcal{O}(1)$.
- **Vertex Page Rank** — A given vertex's page rank weight. Simple lookup based on pre-calculated *Page Rank* static measure $\mathcal{O}(1)$.
- **Vertex Clustering Coefficient** — A given vertex's clustering coefficient. Simple lookup based on pre-calculated *Clustering Coefficient* static measure $\mathcal{O}(1)$.

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

- **Simulation Step** — The current step count of the simulation. Depending on the context, a step may represent different units of time. In this work, each step is treated as one day — $\mathcal{O}(1)$.

III. SEIR Model

The epidemic model used in this work is the Susceptible-Exposed-Infectious-Removed (SEIR) infectious disease model [4]. This model is similar to the classic Susceptible-Infections-Removed (SIR) model [16]; however, it includes the exposed state, a state representing an individual who currently has the disease, does not show symptoms, and is currently not infectious. With this model, individuals transition from Susceptible to Exposed ($S \rightarrow E$), Exposed to Infectious ($E \rightarrow I$), and Infectious to Removed ($I \rightarrow R$). The SEIR model has three parameters: β (probability to infect, controlling $S \rightarrow E$), α (latent period, controlling $E \rightarrow I$) and γ (probability to be removed, controlling $I \rightarrow R$). For a simulation of the epidemic, a starting infectious I_0 set of individuals is selected. In our work, the choice of 2% (10 individuals) was made to have the simulation last a reasonable amount of time, but was ultimately arbitrary.

Table I includes the parameters for the SEIR model. These values are based on those presented by Prem *et al.* [17]. These values were obtained in 2020, and although not arbitrary, would need to be tuned by the user for their needs as every community is different and the observed infection and mortality rates and incubation periods can vary significantly over space and time [18–20].

Given the extended incubation period of COVID-19, the SEIR model is a better fit when compared to the SIR model. Further, the SEIR model has been popular for modelling the spread of SARS-CoV-2 within the literature [17]. The authors make clear that they are not suggesting that the SEIR model is the best for COVID-19 modelling as it does have shortcomings, for example, presymptomatic and asymptomatic individuals can transmit SARS-CoV-2 [21]. Fortunately, like much of the system, the actual choice of model is parameterized and it is easy to use a different epidemic model within the system.

The Network Diffusion Library (NDlib) was used for implementing the epidemic models [22]. The library includes several epidemic models to choose from and makes it easy to simulate the spread of the disease over a graph.

IV. Methods

With the exception of simple parameter values, the additional graph measures, the choice of graph (all three ultimately being parameters), and the fact that the graph is now dynamic, the methods used are the same as previous work. Here we briefly summarize important details; however, one should refer to previous work for a more detailed explanation [3].

A. Genetic Programming Implementation

The evolutionary computation framework used was *Distributed Evolutionary Algorithms in Python* (DEAP) [23, 24].

Genetic programming was used as the algorithm for discovering vaccination strategies. Simply, the vaccination strategies are the programs being evolved. A strategy's fitness is calculated by testing the strategies being evolved in a simulation of an epidemic. All parameters used, including the language, are presented in Table III. The variables/parameters for the programs being evolved are the graph measures outlined in Section II-B.

We also treat the problem as having multiple objectives. Fortunately with DEAP, this is trivial and therefore it is easy to change out objectives and how many are being optimized. This is important since it is difficult to define what it means to find a vaccination strategy that minimizes the impact of an infectious disease scenario: for example, should we maximize the number of individuals never infected throughout the simulation, or should we look for a strategy that minimizes the maximum number of individuals infected at any given time (thereby "flattening the curve")? Ultimately, this ambiguity is alleviated since it is easy to change the objectives and optimize more than one at once.

B. Simulation

A whole SEIR model simulation of the epidemic is used to evaluate each vaccination strategy in each generation. A full description of the simulation, excluding the new addition of making the graph dynamic, as discussed in Section II-A, can be found in previous work [3].

The settings provided in Table IV are based on previous work with minor changes. Specifically, (a) the simulation duration is shorter, which allows the simulation to end at a more appropriate time for the SEIR model parameters, and (b) the number of vaccines available for each vaccination day has been increased from 20 to 30 (or 6% of the population).

TABLE III: GP system parameters.

Hyperparameters	
Population	1000
Generations	500
Elitism	1
Initialization	Ramped Half-and-Half
Init. Max. Depth.	3
Crossover	One Point
Crossover Rate	0.80
Mutation	New Sub-Tree Uniformly Applied
Mutation Rate	0.10
Selection	Tournament
Tournament Size	2
Depth Limit	4
Tree Node Limit	32

Language	
Arithmetic Operators	$+$ $-$ $\times$ $\div$ (protected)†
Boolean Operators	and or not
Comparison and Conditional	$>$ $<$ $==$ $\geq$† $\leq$† $\neq$† If, Then, Else: Return *Boolean* If, Then, Else: Return *Number*
Constants/Terminals	`TRUE` & `FALSE` Integers 0 – 33 Float 0 – 1000

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

Note that each step of the simulation is considered one day, and since vaccines are only applied once a week, there are only 14 vaccination days in a simulation of 98 days. This means that only a maximum of 420 vaccines may be used despite having a total of 500 vertices; however, this was by design.

Rollover and Use All are flags included in the system but not used in this work. These are included in the system since vaccination strategies may not use all available on a given vaccination day if there are no eligible vertices. For example, if the strategy says to vaccinate those that have at least 15 infected neighbours, it is possible no vertex meets this criterion. Rollover allows unused vaccines from a vaccination day to be stockpiled for future use. The Use All flag forces the system to apply all left over vaccines based on a backup strategy, such as first come first serve. Enabling Use All greatly improves performance; however, this option was not used in this work as we expect this feature to impact the static and dynamic graphs differently. Further, by not forcing every vaccine to be used, it is easier to compare how many vaccines the strategies would be using on their own.

C. Testing

In previous work, the objectives being optimized were maximizing the number of individuals left susceptible (i.e. never infected) while minimizing the number of vaccines used [3]. Both these objectives put pressure on the system to select strategies that minimize the number of vaccines used. Here, the maximum and total number of individuals infected were minimized. Note that the total number of individuals infected is the sum of the number of individuals infected over time — the area under the infected curve. For example, since individuals stay infected for variable lengths of time, if an individual was infected for 6 days, that would add 6 to the total. This should not suggest that other metrics are not important for consideration when ultimately choosing a strategy.

TABLE IV: Pandemic simulation Settings. Beta, gamma, alpha and I_0, along with the dynamic graph parameters are taken directly from Table I for ease of reference.

SEIR Model Settings	
β (Beta)	0.09
γ (Gamma)	0.133
α (Alpha)	6.4
I_0	0.02
Simulation Settings	
Iteration Time Frame	1 day
Iterations	98 *days*
Apply Vaccines Every	7 *days*
Vacs. Avail. When Applying	30
Rollover	`False`
Use All	`False`
Dynamic	`True`
Dynamic Graph Settings	
Add Edge prob.	0.01
Remove Edge prob.	0.01
Selection Distro.	Uniform

A total of 100 evolutionary searches are done for both the static and dynamic graphs to ensure a suite of high quality candidates. From each of these runs, a population of 1,000 candidates are present, therefore 100,000 candidate vaccination strategies are evaluated for both the static and dynamic graphs.

For strategy selection after evolution, each candidate is applied to 50 different PCGs and their results are recorded. These 50 PCGs are the same 50 for each of the candidates being evaluated, but will have different starting conditions. Once results are compiled, 6 of the best strategies (based on the median maximum and total infection metrics) are selected for both the static and dynamic graphs. These 12 vaccination strategies are be compared to each other and to a set of simple base strategies one may intuitively come up with.

To generate testing results for analysis, the base strategies, the 6 strategies fit to a static graph, and the 6 fit to the dynamic graphs are each applied to 100 static graph simulations and an additional 100 dynamic graph simulations.

V. Results and Discussion

Tables V and VI contain a summary of the vaccination strategy results for the strategies applied to static graphs and dynamic graphs respectively. The median value of the 100 simulations are presented along with their interquartile range. Note that removed' is the total number of individuals in the removed state ($I \rightarrow R$) plus the total number of effectively

TABLE V: Median values obtained by each strategy over the 100 simulations on a *static* graph. Interquartile range is provided within parentheses.

Strategy	Susceptible		Max Infected		Total Infected		Removed'		Mitigations		Effective		Ineffective	
None	29.0	(± 4.5)	209.0	(± 10.25)	3565.0	(± 137.88)	471.0	(± 4.5)	0.0	(± 0.0)	0.0	(± 0.0)	0.0	(± 0.0)
Random	0.0	(± 0.0)	189.5	(± 10.75)	3049.5	(± 123.62)	408.0	(± 8.5)	101.0	(± 7.12)	92.0	(± 8.5)	9.0	(± 2.0)
Traveller	34.0	(± 5.5)	184.5	(± 8.75)	3129.5	(± 131.12)	417.5	(± 9.12)	55.0	(± 6.75)	48.0	(± 6.25)	7.0	(± 1.5)
Deg. 5	42.0	(± 7.0)	183.0	(± 12.0)	3021.5	(± 129.12)	404.5	(± 9.62)	66.0	(± 4.0)	57.0	(± 4.12)	9.0	(± 2.0)
Deg. 6	47.5	(± 8.0)	183.0	(± 9.62)	3075.0	(± 112.25)	405.0	(± 9.12)	58.0	(± 5.0)	47.0	(± 4.5)	9.0	(± 2.0)
Deg. 7	55.0	(± 11.0)	178.5	(± 12.5)	3029.5	(± 149.0)	406.0	(± 15.62)	46.0	(± 5.5)	39.0	(± 5.5)	7.0	(± 2.0)
Deg. 8	53.5	(± 9.62)	180.0	(± 10.62)	3098.5	(± 123.62)	413.5	(± 12.5)	39.0	(± 4.12)	32.0	(± 4.0)	7.0	(± 1.0)
Deg. 9	60.0	(± 9.25)	172.5	(± 11.62)	3058.0	(± 114.12)	411.5	(± 11.12)	36.0	(± 4.0)	31.0	(± 3.62)	6.0	(± 1.12)
Deg. 10	60.5	(± 9.38)	174.0	(± 13.12)	3092.0	(± 138.25)	413.5	(± 10.62)	33.0	(± 2.5)	27.0	(± 3.0)	6.0	(± 1.5)
S1	0.0	(± 0.0)	171.0	(± 16.0)	2900.5	(± 133.75)	389.0	(± 10.5)	119.5	(± 9.62)	108.0	(± 9.12)	12.0	(± 2.5)
S2	58.5	(± 12.25)	168.0	(± 16.62)	3195.0	(± 173.0)	424.5	(± 16.0)	20.0	(± 3.5)	16.0	(± 4.12)	3.5	(± 1.5)
S3	62.0	(± 11.75)	165.5	(± 14.88)	3112.0	(± 143.12)	411.5	(± 14.0)	30.0	(± 2.5)	25.0	(± 3.0)	4.0	(± 1.5)
S4	29.0	(± 8.38)	166.5	(± 14.5)	2957.0	(± 149.25)	393.0	(± 12.0)	89.0	(± 4.5)	78.0	(± 5.62)	11.0	(± 2.5)
S5	0.0	(± 0.0)	163.0	(± 16.88)	2804.5	(± 156.0)	376.5	(± 19.88)	129.0	(± 17.25)	117.5	(± 17.38)	11.0	(± 2.5)
S6	0.0	(± 0.0)	168.5	(± 19.25)	2925.5	(± 201.5)	383.5	(± 22.25)	128.0	(± 22.62)	116.5	(± 22.25)	10.0	(± 2.0)
D1	0.0	(± 0.0)	172.0	(± 11.88)	2903.0	(± 143.12)	390.0	(± 13.25)	118.5	(± 12.0)	110.0	(± 12.75)	9.0	(± 2.5)
D2	54.5	(± 11.25)	172.0	(± 14.12)	2986.5	(± 167.0)	398.0	(± 12.0)	61.0	(± 3.0)	49.0	(± 3.5)	12.0	(± 2.0)
D3	0.0	(± 0.0)	179.0	(± 18.62)	2987.0	(± 180.25)	398.0	(± 16.5)	113.5	(± 15.0)	102.0	(± 16.5)	11.0	(± 2.5)
D4	0.0	(± 0.0)	183.5	(± 19.38)	3034.5	(± 210.12)	411.5	(± 22.75)	99.0	(± 21.62)	88.5	(± 22.75)	8.0	(± 2.12)
D5	0.0	(± 0.0)	176.0	(± 14.5)	2967.5	(± 201.38)	396.0	(± 14.38)	113.5	(± 15.12)	104.0	(± 14.38)	9.0	(± 2.0)
D6	27.0	(± 5.12)	167.5	(± 15.25)	2797.5	(± 146.0)	375.0	(± 17.0)	106.5	(± 11.5)	97.5	(± 14.0)	10.0	(± 3.0)

TABLE VI: Median values obtained by each strategy over the 100 simulations on a *dynamic* graph. Interquartile range is provided within parentheses.

Strategy	Susceptible		Max Infected		Total Infected		Removed'		Mitigations		Effective		Ineffective	
None	34.5	(± 4.62)	207.5	(± 9.75)	3511.5	(± 117.25)	465.5	(± 4.62)	0.0	(± 0.0)	0.0	(± 0.0)	0.0	(± 0.0)
Random	0.0	(± 0.0)	186.5	(± 9.62)	3059.5	(± 139.62)	406.5	(± 7.5)	103.0	(± 6.5)	93.5	(± 7.5)	9.0	(± 2.0)
Traveller	32.5	(± 5.0)	184.0	(± 10.0)	3202.5	(± 115.75)	418.0	(± 9.0)	56.5	(± 7.5)	49.5	(± 7.12)	7.0	(± 1.5)
Deg. 5	32.5	(± 4.62)	181.5	(± 10.5)	2982.5	(± 126.5)	397.0	(± 8.62)	79.0	(± 5.0)	69.0	(± 6.0)	9.0	(± 1.62)
Deg. 6	45.0	(± 6.62)	181.5	(± 9.0)	2974.0	(± 139.0)	399.0	(± 10.25)	66.0	(± 6.0)	57.0	(± 6.0)	9.0	(± 1.5)
Deg. 7	50.5	(± 8.62)	177.5	(± 11.75)	3033.5	(± 163.75)	405.5	(± 16.0)	53.0	(± 7.12)	45.0	(± 7.0)	8.0	(± 1.5)
Deg. 8	56.0	(± 10.62)	174.0	(± 11.12)	3020.0	(± 158.88)	406.5	(± 15.12)	44.0	(± 5.5)	37.5	(± 5.0)	7.0	(± 1.62)
Deg. 9	57.5	(± 9.62)	172.0	(± 12.25)	3088.0	(± 142.12)	412.0	(± 13.0)	38.0	(± 4.0)	32.0	(± 4.0)	6.0	(± 1.5)
Deg. 10	59.0	(± 9.12)	171.0	(± 12.62)	3088.5	(± 134.88)	411.0	(± 11.25)	34.5	(± 2.0)	29.0	(± 3.12)	6.0	(± 1.5)
S1	0.0	(± 0.0)	171.0	(± 16.0)	2928.5	(± 172.38)	389.0	(± 14.12)	121.5	(± 10.75)	108.0	(± 10.0)	12.0	(± 3.0)
S2	55.0	(± 11.0)	172.5	(± 17.12)	3205.5	(± 131.5)	430.0	(± 13.62)	19.0	(± 5.12)	15.5	(± 4.62)	3.0	(± 1.5)
S3	65.0	(± 11.5)	164.5	(± 14.38)	3067.5	(± 158.25)	409.0	(± 13.12)	30.0	(± 3.0)	25.0	(± 3.5)	5.0	(± 1.5)
S4	25.0	(± 9.62)	165.0	(± 18.12)	2911.5	(± 177.12)	389.0	(± 15.25)	99.0	(± 8.0)	88.0	(± 8.75)	11.0	(± 2.0)
S5	0.0	(± 0.0)	167.0	(± 11.5)	2864.5	(± 146.12)	380.5	(± 15.75)	128.5	(± 13.5)	118.0	(± 14.12)	10.0	(± 2.0)
S6	0.0	(± 0.0)	168.5	(± 16.62)	2870.5	(± 209.25)	384.0	(± 22.5)	127.0	(± 20.5)	116.0	(± 22.5)	9.0	(± 2.12)
D1	0.0	(± 0.0)	171.5	(± 12.75)	2878.5	(± 126.0)	389.0	(± 12.12)	121.0	(± 11.75)	111.0	(± 12.12)	10.0	(± 2.0)
D2	53.5	(± 12.12)	178.5	(± 12.62)	2972.0	(± 160.38)	399.0	(± 16.25)	60.5	(± 4.0)	48.0	(± 4.0)	12.0	(± 3.0)
D3	0.0	(± 0.0)	169.0	(± 16.12)	2900.5	(± 186.12)	393.5	(± 16.0)	120.0	(± 14.62)	106.5	(± 16.0)	12.0	(± 3.0)
D4	0.0	(± 0.0)	183.0	(± 17.12)	3061.0	(± 169.62)	410.0	(± 18.75)	99.0	(± 19.75)	90.0	(± 18.75)	9.0	(± 2.0)
D5	0.0	(± 0.0)	174.0	(± 17.0)	2983.0	(± 155.12)	396.0	(± 13.88)	113.5	(± 14.62)	104.0	(± 13.88)	9.0	(± 1.5)
D6	24.0	(± 5.5)	168.5	(± 14.25)	2835.0	(± 117.38)	376.0	(± 13.62)	109.5	(± 9.12)	99.0	(± 10.12)	10.0	(± 2.5)

vaccinated individuals. The mitigations (vaccinations) value is the total number of vaccinations applied during the simulation. This value is further broken down into effective and ineffective vaccinations. As described in previous work [3], since exposed individuals show no symptoms, we assume the distinction between susceptible and exposed is not known in real-time, therefore both cohorts are eligible for a vaccine, but only vaccines applied to susceptible individuals will be effective in protecting them from the disease.

It is difficult to find glaring differences between the results presented in Tables V and VI. The largest difference between the vaccination strategies when applied to static vs. dynamic graphs presented in these tables is the total infected metric between the vaccination strategies. Also, the base strategies do not appear as effective relative to the evolved strategies. One can also notice that, when considering max infected, the random strategy (effectively "first come, first served"), although not bad, is the second worst next to doing nothing. This demonstrates that some level of nuance in the strategies can improve results over the "first come first, first served" strategy typically employed *en masse*. Interestingly, the random strategy is more competitive with the total infected metric, but still one of the worst performing strategies in that metric.

Although the mitigation count was not directly optimized, this metric may be critically important if a very limited supply of vaccines are available. For example, Static 2 (S2) does a reasonably effective job at reducing the impact of the pandemic while using the fewest vaccines among all strategies (excluding doing nothing).

Algorithm 1 is vaccination strategy Dynamic 6 (D6). This algorithm is presented since it is arguably the best performing strategy when applied to either static or dynamic graphs in terms of the number of individuals left susceptible, max infected, total infected, and removed' metrics. Upon investigation, if the individual is within the minimal vertex cover, then d will be large if the individual exists within many of the

Algorithm 1: Dynamic 6 (D6).

if $minimal\_vertex\_cover$ **then**
  d $\leftarrow short\_path\_cnt + clus\_coef$;
else
  d $\leftarrow num\_removed \times nbr\_inf$;
g $\leftarrow 4 \times avg\_deg \times num\_susexp \times clus\_coef$;
return $d > g$;

shortest paths between individuals (the addition of clustering coefficient is likely inconsequential as it is a value between 0 and 1). If the individual is not in the minimal vertex cover, and since the number of removed grows over time, d will be large only if the individual has infected neighbours. Lastly, since the number of susceptible and exposed individuals will shrink over time, g will be large if the vertex has a sufficiently large clustering coefficient. In other words, the intuition appears to suggest: (a) vaccinate if the individual is considered important based on graph centrality measures, and (b) vaccinate more individuals indiscriminately as time progresses, but only if they are reasonably close to infected individuals.

Although not presented here, both Static 3 (S3) and Static 4 (S4) can be simplified to $deg > avg\_nbr\_degree$ (the individual's degree is greater than the average degree of their neighbours) with some differences in scaling and edge cases. This simple strategy is rather interesting as it suggests the *relatively* high degree individuals be vaccinated. This idea combines the natural and intuitive idea to vaccinate the high degree individuals (people coming into contact with many others), but putting it into a relative context. For example, a "high degree individual" may mean something very different within a dense city versus a rural town. This suggests that it is similarly important to vaccinate the higher degree individuals in less dense and connected communities despite potentially having a lower absolute degree.

Another interesting note about S3 and S4 is their difference in results in Tables V and VI. Although very similar strategies, there are some notable differences in total infected and mitigations used. This may be accounted for in the minor differences in strategies in their edge cases, or perhaps, in spite of the 100 simulations, the very stochastic nature of the simulations.

To simplify the analysis, the top 4 performing strategies from each group (base, static, and dynamic) are selected for further analysis. These strategies are highlighted within Tables V and VI.

For the purpose of understanding the distribution of results, Figures 2 and 3 show scatter plots and histograms for the maximum and total infected for the vaccination strategies applied to static and dynamic graphs respectively. Only the 100 results from the emphasized subset are included in this plot for interpretability. In both figures, octagons ranging in colours purple to cyan represent the four base strategies, triangles ranging from green to orange represent the static strategies, and squares ranging from red to pink represent the dynamic strategies. The histograms are aggregate distributions of all strategies from each group and are coloured accordingly. Within Figures 2 and 3, other than some of the evolved strategies obtaining the best results, and the green and red (static and dynamic) distributions being closer to zero, it is difficult to see any notable differences in the efficacy of the strategies.

To compare the base strategies, static graph generated strategies, and dynamic graph generated strategies, each vaccination strategy's results are aggregated with those within the same group (base, static, and dynamic) and compared — the histograms presented in Figures 2 and 3. Table VII shows each group's median value for max and total infected values along with probability values generated with a Mann-Whitney U test comparing the distribution of results from the aggregate groups. The table presents the values obtained when comparing all 21 strategies and the subset of 12 strategies described above. Note that the non-subset base strategies' group includes the none and random strategies, which will skew the results; however, one can see that the base strategies, regardless of comparing all or the subset, perform the worst by a significant amount, with the exception of the max infected comparison between the base vs. dynamic subset of functions when applied to dynamic graphs, which had a probability value of 0.2716.

In all cases for the max infected metric, the aggregated static group obtained the best results based on the median values by significant amounts; all compared distributions have probability values less than 1.519×10^{-3}. The dynamic strategies performed the best based on the median values of the total infected. Ultimately, some of the largest probability values are found when comparing distributions between static and dynamic strategies, indicating a relatively small difference between static and dynamic strategies' efficacies. This is further emphasized by the inconsistency in the metrics; while strategies developed on static graphs obtained the best results in terms of max infected, those fit to dynamic graphs did better in terms of total infected. This minimal difference confirms what is visually observed in Figures 2 and 3.

Since Table VII shows the grouped aggregate comparison, Figures 4 and 5 are provided to show the probability value comparison on the individual strategy level. These p-values are obtained by comparing the distributions with a Mann-Whitney U test. It is interesting to observe the similarities in the matrices between the static and dynamic simulations. For example, note strategy D4's similarity to the base group regardless of the simulation type. Note S3 and S4, the strategies resembling $deg > avg\_nbr\_deg$, have high similarity with respect to the max infected metric, but little similarity for total infected. Unsurprisingly, the degree based base strategies show a strong similarity to each other and there is perhaps some consistency within the groups for the max infected metric, but overall it is hard to make this conclusion in general, especially for the total infected results.

VI. Conclusions and Future Work

An updated system for developing vaccination strategies based on the topology of a graph was presented. This system

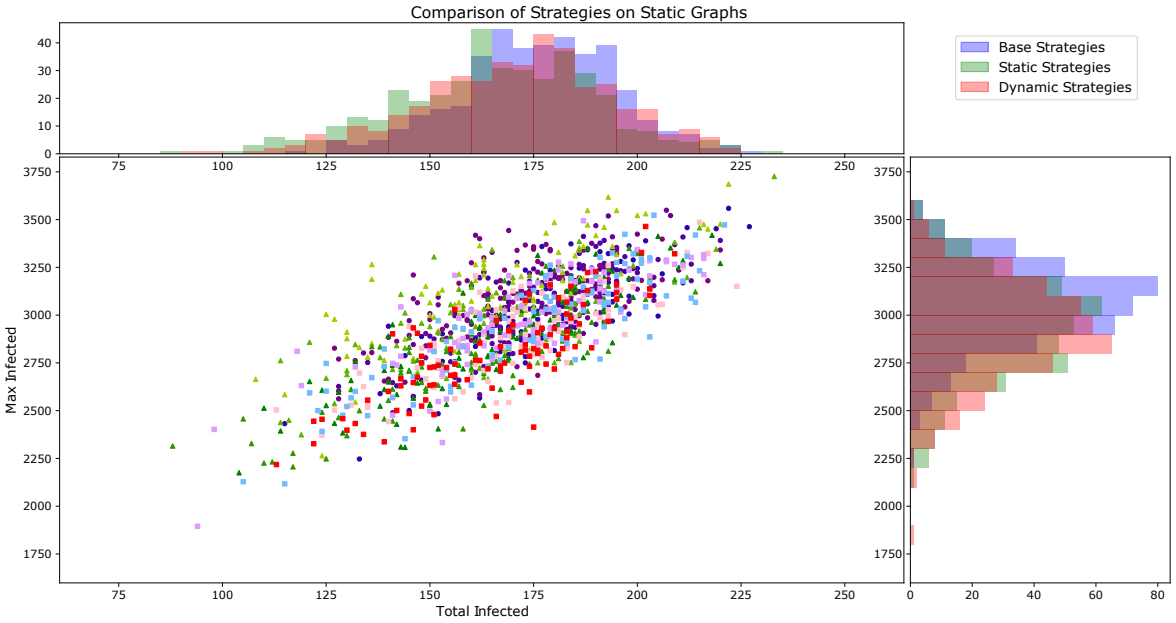

Fig. 2: Scatter plot and histograms of the max and total infected values obtained by the subset of strategies when simulated on a *static* graph.

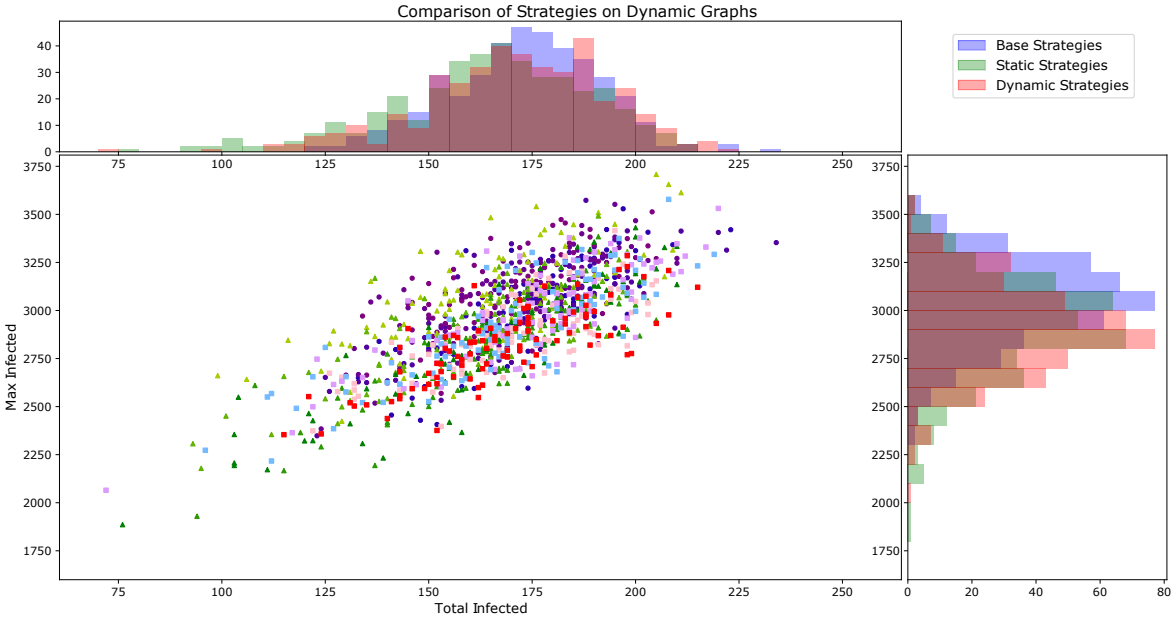

Fig. 3: Scatter plot and histograms of the max and total infected values obtained by the subset of strategies when simulated on a *dynamic* graph.

TABLE VII: Median results and probability values comparing the distribution of the results from each vaccination strategy aggregated into groups. Max and total infected values are compared for both the full set of 21 strategies and the subset of 12 (full result / subset result).

Static Graphs								
	Max Infected				Total Infected			
Function Set	Median	Base v. Static	Base v. Dynamic	Static v. Dynamic	Median	Base v. Static	Base v. Dynamic	Static v. Dynamic
Base	184.0 / 176.0	8.510×10^{-41}	1.776×10^{-17}	1.184×10^{-5}	3102.0 / 3071.5	5.258×10^{-21}	1.334×10^{-32}	1.535×10^{-2}
Static	167.0 / 166.0	/	/	/	2985.0 / 2954.0	/	/	/
Dynamic	175.0 / 173.0	1.534×10^{-10}	1.026×10^{-3}	1.519×10^{-3}	2943.5 / 2914.5	1.184×10^{-11}	7.872×10^{-19}	4.534×10^{-2}
Dynamic Graphs								
	Max Infected				Total Infected			
Function Set	Median	Base v. Static	Base v. Dynamic	Static v. Dynamic	Median	Base v. Static	Base v. Dynamic	Static v. Dynamic
Base	182.0 / 174.0	4.736×10^{-29}	2.386×10^{-11}	1.358×10^{-5}	3085.0 / 3066.0	3.085×10^{-20}	2.415×10^{-32}	1.432×10^{-2}
Static	168.0 / 166.0	/	/	/	2970.5 / 2917.5	/	/	/
Dynamic	174.0 / 172.8	4.258×10^{-6}	2.716×10^{-1}	1.627×10^{-4}	2930.0 / 2885.0	3.549×10^{-14}	3.978×10^{-20}	1.023×10^{-1}

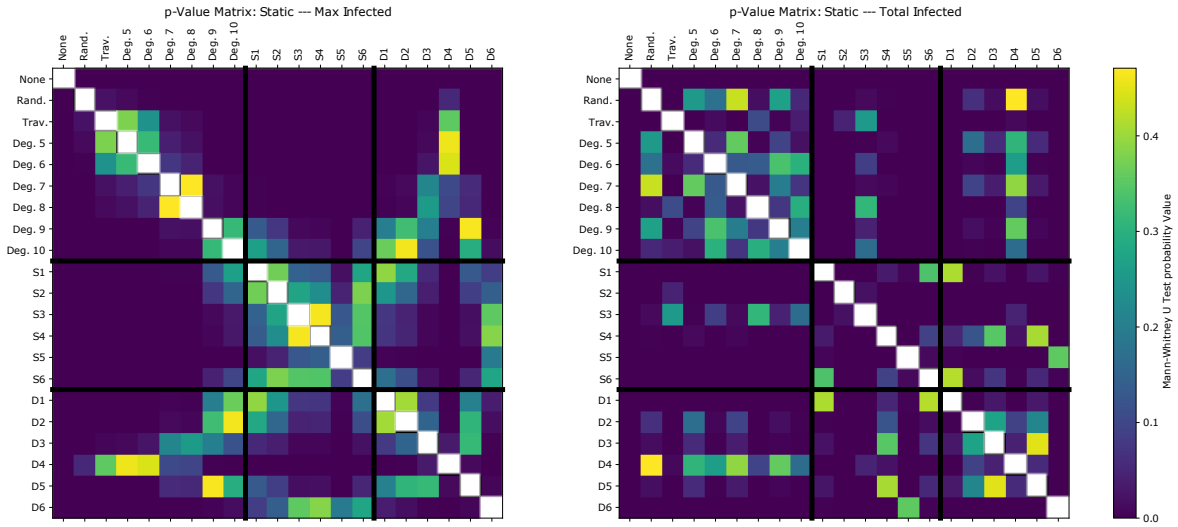

Fig. 4: Probability value matrix comparing all strategies to all other strategies with a Mann-Whitney U test. The values presented here are from the strategies being simulated and tested on *static* graphs.

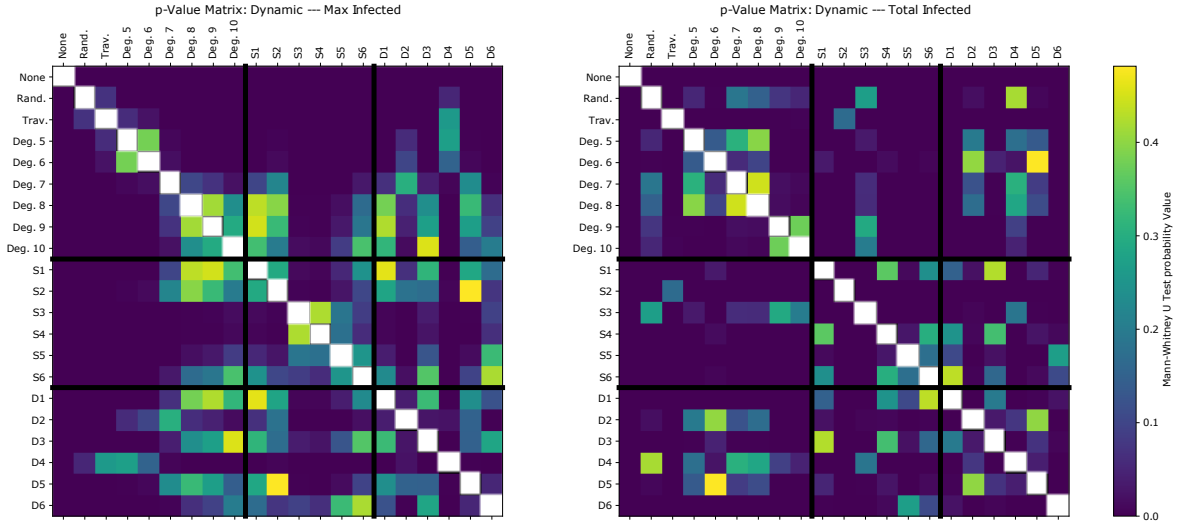

Fig. 5: Probability value matrix comparing all strategies to all other strategies with a Mann-Whitney U test. The values presented here are from the strategies being simulated and tested on *dynamic* graphs.

is built upon the one presented in [3], which simulates a pandemic on a static graph representing a community's social contact network and uses genetic programming to discover vaccination strategies that reduce the impact of the disease. The update includes additional graph measures, features, and most importantly, the ability to develop strategies based on a dynamic graph — one that changes throughout the simulation, something that would happen in a real-world scenario.

Although the major contribution is the system itself, an investigation into the effectiveness of the generated vaccination strategies was done, along with evaluating how effective strategies fit to a static graph are when applied to a dynamic graph, and *vice versa*.

The strategies developed by the system are shown to be more effective than the base, simple strategies one may intuitively develop. Despite being evolved independently, two of the vaccination strategies generated were very similar, and were based on $deg > avg\_nbr\_deg$, which can be intuitively thought of as vaccinating the *relatively* high degree nodes based on their local context.

There is a minimal difference between the results of strategies adapted to static graphs and those adapted to dynamic graphs, regardless of whether they have been tested on static or dynamic graphs. Overall, there are more noteworthy differences between the individual strategies than between the groups. This suggests that the additional computational costs associated with evolving strategies to fit dynamic graphs may not be necessary, therefore simplifying this complex problem. However, the authors are careful to not make a broad assertion at this stage as the number of parameters one may tune in the system is large and differences may arise after further investigation. Exploring a variety of parameters is of interest

to the authors, in particular, different probability distributions and larger values for the add/remove edge to simulate "shock" changes caused by events such as lockdowns and reopenings.

It is not known why the total number of infected individuals varies as much as it does. Although it is expected that it is a consequence of the amount of randomness involved in that metric specifically, this should be investigated further.

Despite being deliberately avoided, to continue evaluating the dynamic graphs, it is possible to recalculate all static measures for each iteration of the simulation, or perhaps have a series of pre-generated graphs with pre-calculated static measures. However, given the results presented here, and how the strategies developed for static graphs performed very well on dynamic graph simulations, it is unclear how much this will impact results.

One of the major areas to be investigated in the future is the impact of changing the epidemic model within the system. Initial work with a more sophisticated model was done by Chao [25], but this should be continued with multiple different infections disease models.

The authors encourage other researchers to take the system to use in their own work and contribute to the larger project. Up-to-date software is available on *GitHub* — https://github.com/convergencelab/eCov-GP.

VII. 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).

This research was enabled in part by support provided by Compute Ontario (www.computeontario.ca) and Compute Canada (www.computecanada.ca).

References

[1] J. A. Weill, M. Stigler, O. Deschenes, and M. R. Springborn, "Social distancing responses to covid-19 emergency declarations strongly differentiated by income," *Proceedings of the National Academy of Sciences*, vol. 117, no. 33, pp. 19 658–19 660, 2020.

[2] A. Y. Orhun and M. Palazzolo, "Frugality is hard to afford," *Journal of Marketing Research*, vol. 56, no. 1, pp. 1–17, 2019.

[3] J. A. Hughes, M. Dubé, S. Houghten, and D. Ashlock, "Vaccinating a population is a programming problem," in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2020, pp. 1–8.

[4] J. L. Aron and I. B. Schwartz, "Seasonality and period-doubling bifurcations in an epidemic model," *Journal of theoretical biology*, vol. 110, no. 4, pp. 665–679, 1984.

[5] W. Ashlock, "Visualizing contact networks evolved to fit epidemic profiles," in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2020, pp. 1–8.

[6] M. Dubé, S. Houghten, D. Ashlock, and J. A. Hughes, "Evolving the curve," in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2020, pp. 1–8.

[7] M. Dubé, S. Houghten, and D. Ashlock, "Modelling of vaccination strategies for epidemics using evolutionary computation," in *2020 IEEE Congress on Evolutionary Computation*. IEEE, 2020.

[8] M. Timmins and D. Ashlock, "Network induction for epidemic profiles with a novel representation," *Biosystems*, vol. 162, pp. 205–214, 2017.

[9] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[10] P. Erdős and A. Rényi, "On random graphs i," *Publ. math. debrecen*, vol. 6, no. 290-297, p. 18, 1959.

[11] M. E. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Physics Letters A*, vol. 263, no. 4-6, pp. 341–346, 1999.

[12] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[13] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Physical review E*, vol. 65, no. 2, p. 026107, 2002.

[14] A. D. Broido and A. Clauset, "Scale-free networks are rare," *Nature communications*, vol. 10, no. 1, pp. 1–10, 2019.

[15] R. Bar-Yehuda and S. Even, "A local-ratio theorm for approximating the weighted vertex cover problem," Computer Science Department, Technion, Tech. Rep., 1983.

[16] W. O. Kermack and A. 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*, vol. 115, no. 772, pp. 700–721, 1927.

[17] K. Prem, Y. Liu, T. W. Russell, A. J. Kucharski, R. M. Eggo, N. Davies, S. Flasche, S. Clifford, C. A. Pearson, J. 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] X. He, E. H. Lau, P. Wu, X. Deng, J. Wang, X. Hao, Y. C. Lau, J. Y. Wong, Y. Guan, X. Tan *et al.*, "Temporal dynamics in viral shedding and transmissibility of covid-19," *Nature medicine*, vol. 26, no. 5, pp. 672–675, 2020.

[19] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong *et al.*, "Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in wuhan, china," *Jama*, vol. 323, no. 11, pp. 1061–1069, 2020.

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

[21] N. Furukawa, J. Brooks, and J. Sobel, "Evidence supporting transmission of severe acute respiratory syndrome coronavirus 2 while presymptomatic or asymptomatic." *Emerging infectious diseases*, vol. 26, no. 7, 2020.

[22] G. Rossetti, L. Milli, S. Rinzivillo, A. Sîrbu, D. Pedreschi, and F. Giannotti, "Ndlib: a python library to model and analyze diffusion processes over complex networks," *International Journal of Data Science and Analytics*, vol. 5, no. 1, pp. 61–79, 2018.

[23] F.-M. De Rainville, F.-A. Fortin, M.-A. Gardner, M. Parizeau, and C. Gagné, "Deap: A python framework for evolutionary algorithms," in *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, 2012, pp. 85–92.

[24] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "Deap: Evolutionary algorithms made easy," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2171–2175, 2012.

[25] C. Xue, "Modeling COVID with computational learning and applying vaccination based on a scale-free network," 2021, honours Thesis, St. Francis Xavier University.