

# Optimizing the Efficiency, Vulnerability and Robustness of Road-based Para-transit Networks using Genetic Algorithm

Briane Paul V. Samson<sup>1,2</sup>[0000-0002-0253-452X], Gio Anton T. Velez<sup>1</sup>, Joseph Ryan Nobleza<sup>1</sup>, David Sanchez<sup>1</sup>, and Jan Tristan Milan<sup>1</sup>

<sup>1</sup> De La Salle University, Manila, Philippines  
`{gio.velez, joseph.ryan.nobleza, david.sanchez, jan.milan}@dlsu.edu.ph`  
<sup>2</sup> Future University Hakodate, Hakodate, Hokkaido, Japan  
`b-samson@sumilab.org`

**Abstract.** In the developing world, majority of people usually take para-transit services for their everyday commutes. However, their informal and demand-driven operation, like making arbitrary stops to pick up and drop off passengers, has been inefficient and poses challenges to efforts in integrating such services to more organized train and bus networks. In this study, we devised a methodology to design and optimize a road-based para-transit network using a genetic algorithm to optimize efficiency, robustness, and invulnerability. We first generated stops following certain geospatial distributions and connected them to build networks of routes. From them, we selected an initial population to be optimized and applied the genetic algorithm. Overall, our modified genetic algorithm with 20 evolutions optimized the 20% worst performing networks by 84% on average. For one network, we were able to significantly increase its fitness score by 223%. The highest fitness score the algorithm was able to produce through optimization was 0.532 from a score of 0.303.

**Keywords:** Complex Networks · Network Optimization · Genetic Algorithm.

## 1 Introduction

As cities grow with the influx of urban migration, the capacity and efficiency of their transportation systems have consistently been challenged with the consequential boost in travel demands [3]. And before our transportation systems reach their limits, governments must focus more on improving transportation capabilities instead of focusing on adding more infrastructure (Braess's Paradox [5, 6]).

Dense urban areas like Metro Manila, Philippines have had little success addressing transportation capabilities. Due to the limited operations of high-capacity transport systems, passengers have mostly preferred road-based para-

transit services like jeepneys<sup>3</sup>, express shuttles, tricycles<sup>4</sup>, and pedicabs<sup>5</sup> [2]. This is also the case in other developing world cities, with diverse vehicle categories (i.e. tuk-tuk, minivans, etc.), because they can pass through smaller roads, and are more demand-responsive and affordable [13]. However, such transportation services do not have formal stops and were arbitrarily planned to serve the entrepreneurial interests of its private operators. In particular, they have been observed to stop anywhere to pick up a hailing passenger and to drop them off. As a response, local governments have attempted to regulate, integrate and rationalize their services [3] by re-assessing and formalizing the route network, and consolidating operators.

Inasmuch as we want to improve the efficiency of public transportation networks, some topological and geospatial optimizations bear negative consequences to their operations [12]. One can reduce the spacing between designated stops, which can result to an increased number of stops available in an area. With more stops, public utility vehicles will stop more often, resulting to longer waiting times in subsequent stops. Thus, there has to be a balance between accessibility and efficiency by assessing how the geospatial layout and spacing between designated stops could affect the overall performance of a road-based transportation network. In addition, since road networks in the developing world are more prone to disruptions (i.e. flooding, poor road conditions, etc.) [7], we also have to consider the robustness and vulnerability of any mode of transportation that will use them.

This study aims to use network centrality measures and genetic algorithm to optimally design efficient, robust and invulnerable para-transit networks, that governments can use in planning the integration of para-transit services to higher-capacity transportation networks. As a case study, we applied this methodology to four (4) cities in Metro Manila and generated intra-city jeepney networks.

The next section discusses some significant related works in designing transportation networks. Section 3 discusses our methodology for network generation and optimization. In section 4, we share our results and analysis on the performance of our methodology. Lastly, we provide our conclusion and plans for future work in sections 5 & 6.

## 2 Related Works

Planning transportation systems require stakeholders to look at different aspects like travel demand, land use and urban form, and transportation coordination and scheduling, among others. Focused on minimizing the number of transfers, [8] developed an optimization model using an operations research approach. However, its mixed integer programming implementation can be limiting given the large search space and multiple conflicting constraints involved.

---

<sup>3</sup> A popular public utility vehicle with a capacity of 20-22 passengers

<sup>4</sup> An auto rickshaw consisting of a motorbike and a sidecar

<sup>5</sup> A cycle rickshaw consisting of a bicycle and a sidecar

In a survey of works from 1998 to 2012 on designing transport networks by [9], they concluded that genetic algorithms are more suited to handle the complex and non-linear nature of this problem. And in almost all works surveyed, they optimized irregular grid network structures to have the least cost for passengers and operators. In addition, [14] considered the location of school dormitories to optimize the location of bus stops to be installed. The resulting bus network design reduced the number of bus stops, total route length, and travel distance between dormitories. On the other hand, [11] focused on maximizing the number of satisfied passengers while minimizing transfers and travel time. Taking a different direction, [12] designed transport networks based on a city's geospatial distribution and used genetic algorithm to generate bus routes with the least travel time for passengers. After which, they assessed how each type of geospatial distribution affected the robustness of the network. However, the networks were not recreated to address negative effects on their robustness.

In relation to para-transit operations, [1] minimized the operation costs of a jeepney route in Taft Avenue, Manila by considering the waiting time before operation, length of the time when ignoring stops, the number of passengers and amount of time to board the jeepney, waiting cost of commuters for the jeepneys to arrive, length of travel time, length of time to accelerate and decelerate, and length of time a jeepney dwells in a particular stop. In China, [10] optimized the networks of customized buses in terms of maximum passengers served, passenger travel time and arrival delays, and line revenues.

While many have considered operational factors in optimizing transportation networks, which represent regular occurrences, very few have regarded unprecedented scenarios and anticipated interruptions in planning their networks. Thus, we intend to simulate random failures and targeted attacks on the designed transport network and iterate using genetic algorithm to minimize their effects.

### 3 Methodology

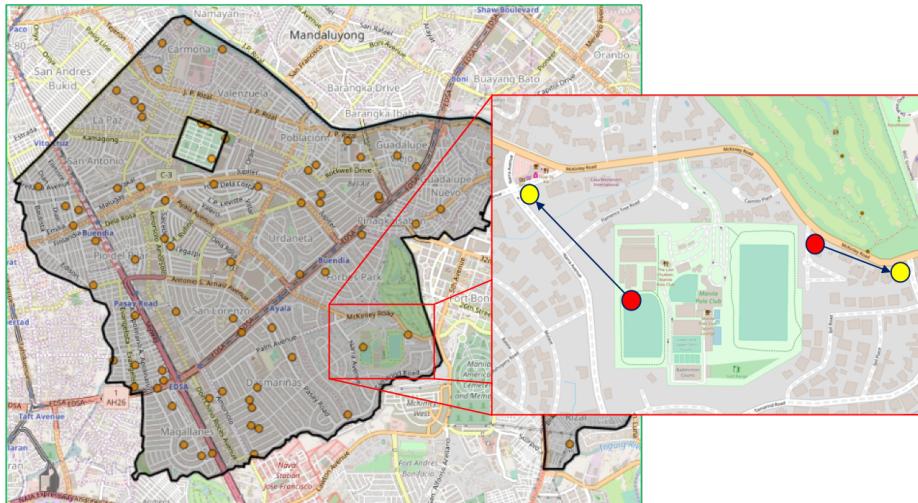
This section discusses our methodology which starts with the collection and pre-processing of road network data, followed by the generation of an initial set of candidate stops. Then, the stops are connected to form routes. Finally, the generated networks are optimized for its efficiency, robustness and invulnerability.

#### 3.1 Data

We collected the road networks and boundaries of four (4) Metro Manila cities, namely, Manila, Makati, Paranaque, and Quezon City using OSMnx [4]. To avoid redundancies, we used a network that only contains nodes for road intersections and joints.

### 3.2 Stop Generation

First, we generated stops by following the lattice, random and N-hub geospatial layouts [12]. In a lattice layout, stops are generated equidistant from each other, while in a random layout, the location coordinates were uniformly distributed within the city boundaries. Lastly, the stop locations in an N-hub layout is generated by first defining the location of the N hubs representing areas of high trip generation, such as central business districts. We used a covariance coefficient of 65 to provide an ideal shape of a circle revolving around the specified hub. The stops were generated by getting random samples from a multivariate normal distribution. After generating the stops, some were positioned in unusual and unrealistic locations, like in the middle of parks. Thus for poorly placed stops, we adjusted their locations to nearest road segments (Figure 1).



**Fig. 1.** Two stops (red circles) were initially generated far from intersections of major roads. The yellow circles indicate where their new locations are.

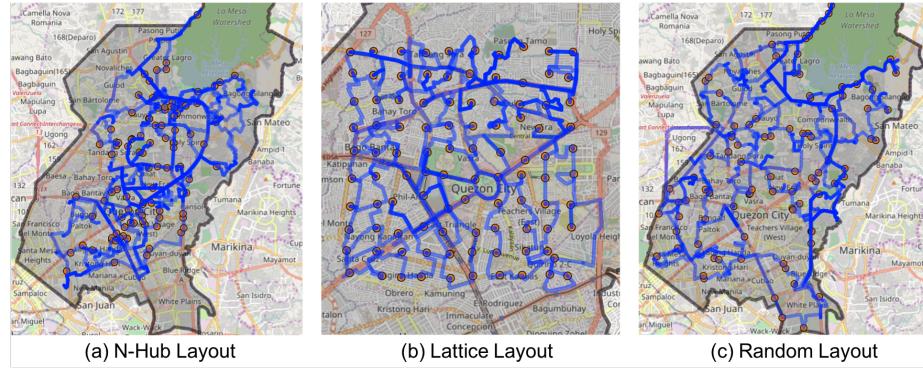
### 3.3 Route Network Generation

In connecting the stops, we used a maximum allowable walking distance ( $d_{max}$ ) value as a threshold for choosing candidate stops to connect to. This is based on a person's tolerance of how far they can walk from a stop to their destination.

In creating a route  $R_0$ , a starting stop  $S_{(0,0)}$  is chosen in random. Then, we used  $d_{max}$  as a discriminating factor for nearby stops, so they cannot be selected anymore until the creation of another route. The stops beyond  $d_{max}$  are now candidates and are assigned exponentially decreasing probabilities based on their Euclidean distance from the stop to connect to:

$$p(n_d) = \frac{e^{-d(n_s, n_d)}}{\lambda} \quad (1)$$

where  $n_s$  is the starting stop and  $n_d$  is the destination stop,  $d$  is the Euclidean distance function and  $\lambda$  is a normalization constant. From these candidates, we select  $S_{(0,1)}$  and connect it to  $S_{(0,0)}$ . We repeated the same steps until there are no more stops to discriminate or to connect to. With this algorithm, we made sure that there are no cycles, but there will be instances wherein not all stops will be part of the network. In such cases, they will be removed. In this study, we created 25 routes to build the networks.



**Fig. 2.** Generated route networks for Quezon City with 100 stops following the (a) N-Hub, (b) Lattice, and (c) Random geospatial layouts.

**Generated Route Networks.** We generated 960 undirected route networks using a combination of layout and maximum allowable walking distance values to observe the consistency of our algorithms. For each of the four (4) cities in our scope, we used the lattice, random, and 1-hub and 2-hub layouts (Figure 2). For each of these layouts, we generated route networks using 300m, 550m, and 800m as  $d_{max}$  values. Each edge has a corresponding weight representing distances in meters.

### 3.4 Network Optimization

After initially generating route networks, we optimized them in terms of efficiency, robustness, and vulnerability using a genetic algorithm.

**Network Metrics.** In optimizing the generated route networks, we used the following network metrics in evaluating their fitness scores. First, we computed

for the efficiency of a route network that gives us an idea of how straightforward a simulated trip is from a source to a destination. For each network, we randomly selected 1,000 source and destination stop pairs and computed their weighted radius of gyration by (a) creating an adjacency matrix, (b) computing for the shortest path using Dijkstra's algorithm, and (c) incorporating a penalty for possible transfers between routes. The weighted adjacency matrix for all selected pairs  $(i,j)$  in the network is represented by:

$$A_{i^*j} = \begin{cases} d(n_i, n_j), & \text{if a route directly connects from } n_i \text{ to } n_j \\ w * d(n_i, n_j), & \text{if there is more than 1 transfer from } n_i \text{ to } n_j \\ 0, & \text{if } i = j. \end{cases}$$

It has all the edges in the graph, including the walking edges between stops connecting different routes. We assigned a weight of 10 to these edges because walking is roughly 10 times slower than riding a jeepney. Lastly, we normalized the efficiency scores by getting the ratio of its score to the number of randomly selected source-destination pairs (Equation 2).

$$E = \frac{\sum_{i=1}^{1000} \frac{d_p}{\sum_{p \in \text{Paths}} w_i d_p + 20T}}{1000} \quad (2)$$

Next, we assessed their robustness by randomly removing stops from the network until there's nothing to remove. This simulates the possibility of stops suddenly becoming inaccessible because of unexpected closures, floods, and congestion. Aside from network robustness, we also measured how vulnerable the network is to targeted attacks by removing stops in order of their degrees, starting from the highest degree until nothing is left. As we removed stops from the network for both robustness and vulnerability simulations, we computed for the average path length and network diameter. The average path length is the mean of the lengths of all shortest paths. This factor tells us the average number of stops needed to traverse the network from any stop. The network diameter is the longest shortest path of a network which gives us an idea of the maximum number of stops needed to traverse without repetition or cycles.

Using these simulations and metrics, we computed for the ratio of the average path length over the network diameter at every 5% interval, which indicates how close the network is to being destroyed. For this ratio, higher values indicate low robustness and high vulnerability.

After running these metrics for all 960 generated route networks, we observed that most route networks got disconnected or have reached their peak at 15% and 30% node removals, after simulating vulnerability and robustness, respectively. Thus for computational efficiency, we used the ratio of the average path length over the network diameter at those thresholds for the fitness function.

**Fitness Function.** Considering the efficiency, robustness, and vulnerability scores, we derived the fitness function as:

$$F(G) = \frac{\alpha E - \beta R - \delta V}{\alpha + \beta + \delta} \quad (3)$$

where  $E$  is the efficiency score,  $R$  is the robustness score, and  $V$  is the vulnerability score, with  $\alpha$ ,  $\beta$ , and  $\delta$  as their respective weights. A higher weight of 80% significance ( $\alpha$ ) was used for efficiency since the straightforwardness from source to destination affects daily operations. Straightforwardness aside, it was also observed that a minimum of 70% is required to produce positive fitness scores for better optimized networks. On the other hand, robustness and vulnerability were given the weights of 10% each ( $\beta$  and  $\delta$ ) since they simulate seasonal events.

Higher scores are better for efficiency while lower scores are better for robustness and vulnerability. Thus, we subtracted the vulnerability and robustness scores from the efficiency of the network. Lastly, everything is then divided by the sum of the weights for normalization.

**Genetic Algorithm.** In this study, we optimized the 20% worst performing networks from the pool of 960 generated networks. We ranked them based on their current fitness scores. For each network to be optimized,  $N_A$ , we randomly selected a network with a good fitness score from the 20% best performing networks as its pair,  $N_O$ .

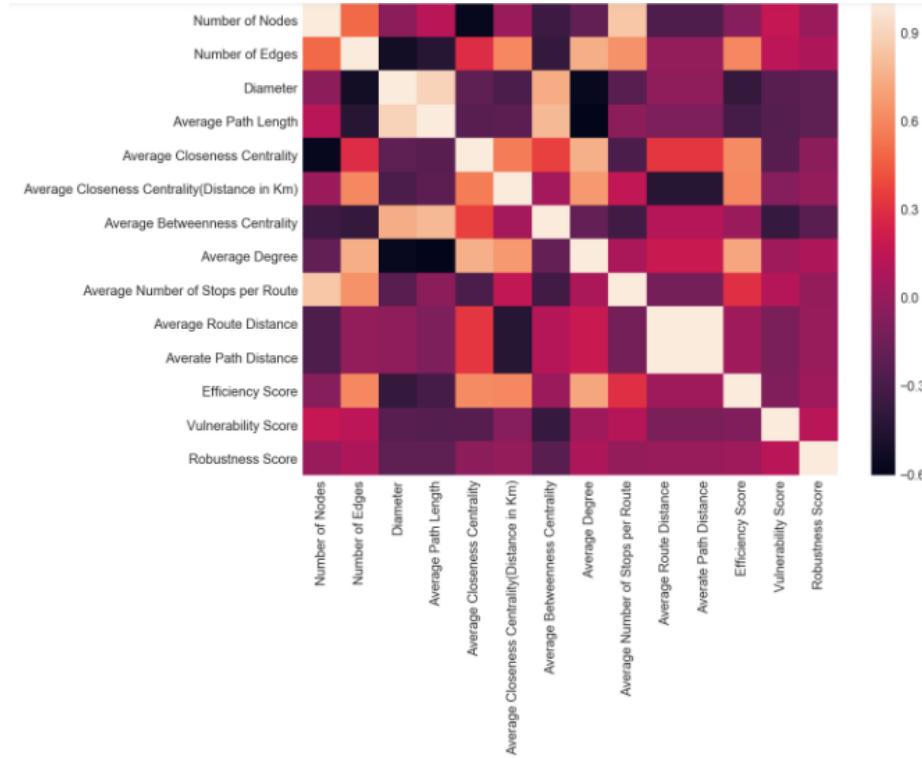
At the beginning of each evolution, we randomly selected the number of routes to be mutated:  $M \in \{0, \dots, 3\}$  where  $P(M = 0) = 0.7$ ,  $P(M = 1) = 0.2$  and  $P(M = 2, 3) = 0.05$ . To produce a child network, we randomly selected route IDs from both parents and replaced the routes from  $N_A$  with the routes of the same route IDs from  $N_O$ . In each evolution, we produced a generation of 10 child networks, ranked by their fitness scores. If the highest ranked child network had a higher fitness score than one of the parents, then that child is selected as a parent in the next evolution, replacing the current parent with the lower fitness score. If no child had a higher fitness score than both parents, the current parents are selected again for generating children in the next evolution. We stopped the genetic algorithm at the 20th evolution because it was observed to have reached a plateau.

## 4 Results and Analysis

In evaluating the generated route networks, we first looked into the effects of an initial stop layout and different values for the maximum allowable walking distance on the efficiency, robustness, and vulnerability of route networks. After optimizing the worst performing networks, we looked at how much the networks improve in terms of efficiency, robustness, and vulnerability.

**Effects of Layout and Maximum Allowable Walking Distance** For each city, we generated 20 networks per combination of geospatial layouts (Lattice,

Random, 1 Hub, 2 Hubs) and  $d_{max}$  (300m, 550m, 800m). Each network had 100 stops and 25 routes. We had a total of 960 generated routes. We first analyzed how our defined network characteristics correlate with our metrics from the fitness score.

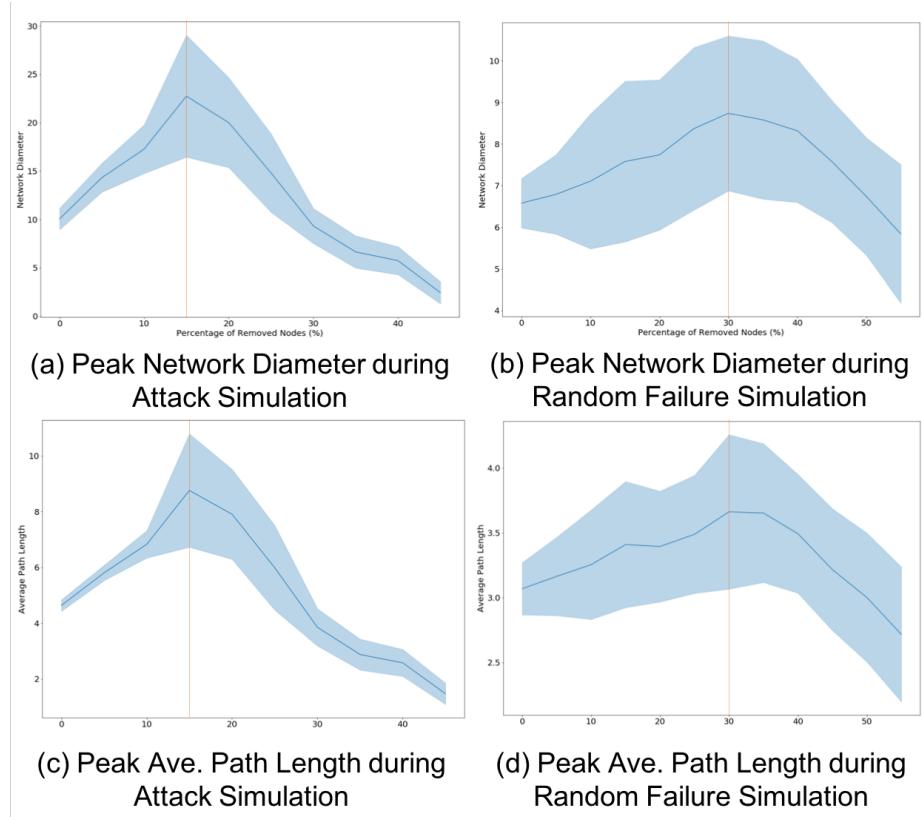


**Fig. 3.** Correlation matrix of all network characteristics with the derived fitness function metrics.

In Figure 3, we focused on the correlation of the network characteristics towards our fitness function. Based on the matrix, we observed that among the three fitness function metrics, the efficiency of a network had strong positive correlation to five (5) route network characteristics.

First, a high number of edges meant that there will be more possible paths from source to destination. It lessens the possibility of selecting paths that use routes that are either not directly connected at a stop or have many route transfers. Second, as more stops become close to each other in terms of number of hops in between, the straightforwardness of the route network becomes better. This reduces the need to transfer between routes even in longer distances but this would also mean that distances between stops are much greater.

Aside from closeness in terms of the number of stops in between, the efficiency scores also showed a strong positive correlation with the closeness centrality using spatial distance. This time, as the spatial distance of a path (in km) becomes shorter for most source-destination pairs, the faster it is to travel because of shorter trips. At the same time, as more stops get used by multiple routes and become hubs, it allows shorter commutes between stops. A passenger could reach different destinations provided from a stop with a high degree.



**Fig. 4.** Network Diameters and Average Path Lengths during the simulation of targeted attacks and random failures after every removal of additional 5% stops from the network for Manila having a Lattice Layout with a  $d_{max}$  of 300m.

Lastly, a high average number of stops per route means that there are many stops which can be connected to connect shorter paths with other straightforward routes. Having more available paths will contribute to the lessening of transfers between routes, thus, improving the efficiency score of a network.

**Threshold of Robustness and Vulnerability** After simulating random failures and targeted attacks, we then observed at which point a route network's diameter and average path length reached their peaks. The peak indicates the destruction or disconnection of network. We got the average network diameter and average path length at every 5% stop removals among the 20 route networks for each configuration. Once this has been achieved, we recorded the stop removal with the highest average value. Finally, we got the most frequent occurring percentage removal among all the configurations.

After simulating targeted attacks to the networks, we observed that 44% of them started to disconnect after removing at least 15% of the stops (Figure 4a & 4c). At the same time, 31% of the networks had their longest diameters after removing at least 30% stops (Figure 4b & 4d).

#### 4.1 Optimization

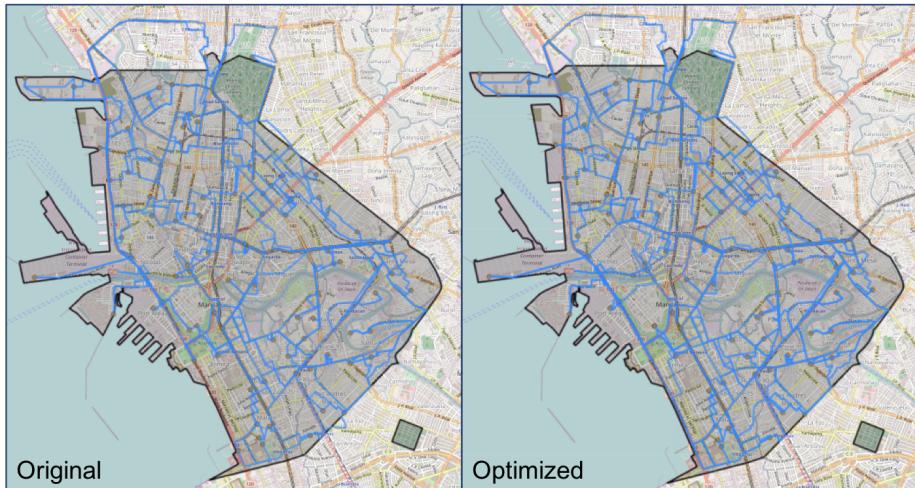
We then compared the network metrics of the 20% worst performing networks before and after optimization. Having a positive difference for efficiency means that the efficiency score improved. Having a negative difference for the robustness and vulnerability means that the robustness and vulnerability scores improved.

**Table 1.** Average Network Characteristics of the 20% Worst Performing Route Networks

Metric	Unoptimized	Optimized
Number of Edges	162.037	166.22
Diameter	8.667	10.185
Average Path Length	3.814	4.017
Average Closeness Centrality	0.238	0.259
Average Closeness Centrality(Distance in Km)	0.0001	0.0001
Average Betweenness Centrality	0.03	0.034
Average Degree	3.638	3.741
Average Number of Stops per Route	76.951	73.412
Average Route Distance	14.855	24.846
Average Path Distance	2.277	2.213
Efficiency Score	0.33	0.552
Vulnerability Score	0.42	0.415
Robustness Score	0.42	0.423

Table 1 shows the average network characteristics of 27 route networks sampled from the 20% worst performing networks, before and after their optimization. It can be observed that the number of edges, network diameter, average path length, and average route distance increased as the efficiency score increased for the optimized network as compared to the unoptimized network scores. This is because having longer routes in the network means that more stops are reachable in one route and if more stops are reachable in one route, it reduces the

possibility of more transfers, making a network more straightforward and efficient. It can also be observed that the vulnerability score decreased a little while the robustness score almost maintained its score. A decrease in the vulnerability and robustness scores means that the scores got better, even by a little margin. It makes sense that there is little improvement for the scores since they have lower weights as compared to the weight of the total efficiency or total weighted radius of gyration. However, the average scores for the characteristics tell us that the efficiency, vulnerability, and robustness scores improved in general for the optimized versions.



**Fig. 5.** A sample optimized network for Manila with a random layout and a  $d_{max}$  of 300m.

As an example, Figure 5 shows an original and optimized network for Manila. Though the changes are not so visually apparent, it has gained a lot in terms of number of edges (from 175 to 228) and average distance per route (from 14km to 30km). At the same time, we were able to decrease its diameter (from 9 to 6) and average path length (from 4 to 3).

Overall after the optimization, 18 out 27 networks improved its efficiency, 13 out of 27 networks improved its robustness, and 9 out of 27 networks improved in terms of vulnerability. In terms of the overall fitness scores, the optimized route networks showed an 84% improvement over the unoptimized networks on average. However, there are 2 instances, among the 27 networks, that had lower optimized scores.

## 5 Conclusion and Future Work

In this study, we devised a methodology to design transportation networks for para-transit services. In generating the networks, we initially followed a geospatial layout for distributing stops and used a maximum allowable walking distance to connect the routes. We then optimized the networks in terms of efficiency, robustness and invulnerability. To test our methodology, we generated 960 routes for four (4) Metro Manila cities. Our genetic algorithm was able to improve the network metrics of the worst performing networks.

There are still some aspects of the research that we were not able to consider. For example, instead of following a geospatial layout, a database of amenities or travel demand data could improve the positioning of the stops. Another recommendation is considering the width of a road in deciding how many routes can use it.

## References

1. Abad, R.P.B., Fillone, A.M., Dadios, E.P., Roquel, K.I.Z.: An application of genetic algorithm in optimizing Jeepney operations along Taft Avenue, Manila. In: 8th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2015 (2016). <https://doi.org/10.1109/HNICEM.2015.7393243>
2. Asian Development Bank: Philippines: Transport Sector Assessment, Strategy, and Road Map p. 18 (2012)
3. Attard, M., Haklay, M., Capineri, C.: The Potential of Volunteered Geographic Information (VGI) in Future Transport Systems. *Urban Planning* **1**(4), 6–19 (2016). <https://doi.org/10.17645/up.v1i4.612>
4. Boeing, G.: OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems* **65**, 126–139 (2017). <https://doi.org/10.1016/j.compenvurbsys.2017.05.004>
5. Braess, D., Nagurney, A., Wakolbinger, T.: On a Paradox of Traffic Planning. *Transportation Science Unternehmensforschung* **39**(12), 446–450 (2005). <https://doi.org/10.1287/trsc.1050.0127>
6. Di, X., He, X., Guo, X., Liu, H.X.: Braess paradox under the boundedly rational user equilibria. *Transportation Research Part B: Methodological* **67**, 86–108 (2014). <https://doi.org/10.1016/j.trb.2014.04.005>
7. Jain, V., Sharma, A., Subramanian, L.: Road Traffic Congestion in the Developing World. In: Proceedings of the 2nd ACM Symposium on Computing for Development. pp. 1–10. ACM, New York, NY (2012)
8. Jaramillo-Alvarez, P., González-Calderón, C., González-Calderón, G.: Route Optimization of Urban Public Transportation. *Dyna* **80**(180), 41–49 (2013), <http://www.scielo.org.co/pdf/dyna/v80n180/v80n180a06.pdf>
9. Johar, A., Jain, S.S., Garg, P.K.: Transit network design and scheduling using genetic algorithm a review. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)* **6**(1), 9–22 (2016). <https://doi.org/10.11121/ijocta.01.2016.00258>, <http://ijocta.balikesir.edu.tr/index.php/article/view/258>

10. Ma, J., Zhao, Y., Yang, Y., Liu, T., Guan, W., Wang, J., Song, C.: A model for the stop planning and timetables of customized buses. *PLoS ONE* **12**(1), 1–28 (2017). <https://doi.org/10.1371/journal.pone.0168762>
11. Nayeem, M.A., Rahman, M.K., Rahman, M.S.: Transit network design by genetic algorithm with elitism. *Transportation Research Part C: Emerging Technologies* **46**, 30–45 (2014). <https://doi.org/10.1016/j.trc.2014.05.002> <http://dx.doi.org/10.1016/j.trc.2014.05.002>
12. Pang, J.Z.F., Bin Othman, N., Ng, K.M., Monterola, C.: Efficiency and robustness of different bus network designs. *International Journal of Modern Physics C* **26**(03), 1–15 (2015). <https://doi.org/10.1142/S0129183115500242>
13. Schalekamp, H., Behrens, R.: An International Review of Paratransit Regulation and Integration Experiences: Lessons for Public Transport System Rationalisation and Improvement in South African Cities. In: *Proceedings of the 28th Southern African Transport Conference*. pp. 442–450. Pretoria, South Africa (2009)
14. Takakura, M., Furuta, T., Tanaka, M.S.: Urban Bus Network Design Using Genetic Algorithm and Map Information. In: *Proceedings of the Eastern Asia Society for Transportation Studies 2015*. pp. 1–13 (2015)