A Mobile Disaster Management System Using the Android Technology

JOVILYN THERESE B. FAJARDO, CARLOS M. OPPUS
Department of Electronics, Computer, and Communications Engineering
Ateneo de Manila University
Katipunan Avenue, Loyola Heights, Quezon City 1108
PHILIPPINES
jovilyn.fajardo@gmail.com, coppus@ateneo.edu

Abstract: - The Philippines is one of the countries in the world vulnerable to natural hazards because of its geographic location. It also lacks an efficient disaster management system that will help in times of need. One common scenario during disasters is that the activity of rescue and relief is not well-coordinated. For this reason, there is a need for a system that will help in the efficient provision of rescue and relief to disaster-affected areas. Since the use of smart phones is gaining interest in people, the disaster management system was implemented as a smart phone application using Google's Android operating system. The disaster management system Android application known as MyDisasterDroid determines the optimum route along different geographical locations that the volunteers and rescuers need to take in order to serve the most number of people and provide maximum coverage of the area in the shortest possible time. Genetic algorithm was applied for optimization and different parameters were varied to determine the most optimum route.

Key-Words: - Android, Disaster management, Genetic algorithm, Travelling salesman problem, Smart phone application, Wireless mobile technology

1 Introduction

A report released by the United Nations' International Strategy for Disaster Reduction (UNISDR) based on a new Mortality Rate Index (MRI) ranks the Philippines no. 12 among 200 countries and territories most at risk from earthquakes, floods, tropical cyclones, and landslides [1]. Indeed, through the years, the Philippines because of its geographic location has been devastated by several natural disasters. It has claimed the lives of thousands of people and damaged billions worth of property. Even with the recurrent calamities, there is still the absence of an efficient disaster management system that will help the people in times of calamities.

2 Disaster Management

The process of disaster management involves four phases: mitigation, preparedness, response, and recovery. The mitigation phase is the attempt to reduce disaster risks by focusing on long-term measures of eliminating disasters. The preparedness phase is the development of an action plan for an upcoming disaster. The response phase is the mobilization of services and relief when disaster strikes and the recovery phase is the restoration of the affected area to its previous state [2].

2.1 SAHANA FOSS

SAHANA is a free and open source disaster management system. It is a web-based collaboration tool that addresses common coordination problems during a disaster. It is a set of pluggable, web-based disaster management solution that provides solutions to problems caused by the disaster and it is designed to help during the relief phase of a disaster. It has been already deployed in different disaster areas including the Philippines during the Southern Leyte Mudslide Disaster in 2006 [3]. At present, more tools are being developed however, the current disaster management system does not include a tool that will aid in the response phase during the disaster. The response phase includes search and rescue operations as well as the provision of emergency relief. In this phase, efficiency is important because during this kind of situation, time is of the essence. A second delay may cause someone's life. Thus, a system that determines the most optimum route for the volunteers and rescuers to take in order to serve the most number of people and provide maximum coverage of the affected area in the shortest possible time is beneficial.

2.2 Using Wireless Mobile Technologies

The use of wireless mobile technology in the Philippines is prevalent. A study showed that one Filipino in two is a subscriber to a mobile phone service [4] and more than two billion in the world are mobile users [5]. Because of the wide coverage of the mobile phone network, it may be used in the implementation of a disaster management system making it available in everyone's pocket.

A study in Bangladesh established that wireless mobile technologies can be used in disaster information management. Results showed that mobile technology may be used to disseminate pre-disaster warnings and post-disaster announcements, to receive information about relief needs, and to exchange information about health hazard [6]. Also, in disaster information management, geographic locations of those in need is important. Using their mobile phones, their locations can either be determined using the mobile network system or through the use of an integrated Global Positioning System (GPS) included in their phone.

2.2.1 Smart Phones and Disaster Response

Based on research released by Smartphone Summit, smart phones account for 10% of all cell phone sales and it is still growing and driving more interest among people [7]. One of the reasons for its continued growth is that it provides information valuable to the users.

Moreover, in times of disaster, the more people that have information with them all the time, the more they will be self-reliant allowing rescuers or responders to concentrate on those in the greatest need of help [8].

There are also a number of mobile applications available in smart phones that is beneficial in disaster response. Among these are the GPS technology, which can be used in the tracking of rescuers and resources, the translator, which can be used for communication, and the field examiner, which can be used to send information to headquarters for assessment of damages [9]. Indeed, the use of a smart phone in a disaster management system is advantageous.

2.2.2 Android Mobile Development Environment

There are a number of mobile development environments in the market. One of which is Android created by the Open Handset Alliance. Android is an open and comprehensive platform for mobile devices. It is designed to be more open than other mobile operating systems so that developers, wireless operators, and handset manufacturers will be able to make new products faster and at a much lower cost. The end result will be a more personal and more flexible mobile experience to the user [10]. For this reason, this mobile development environment was used in the implementation of the disaster management system.

3 Travelling Salesman Problem (TSP)

Determining the most optimum route along different geographic locations is similar to the travelling salesman problem wherein geographic locations represent city coordinates and the rescuers or volunteers represent the travelling salesman.

The travelling salesman problem is stated as follows: “Given a finite number of cities and the distance (or cost) of travel between each pair of them, find the shortest (or cheapest) way of visiting all the cities and returning to the starting point.”

The travelling salesman problem is formally described as a permutation problem with the objective of finding the path of the shortest length (or the minimum cost) on an undirected graph that represents cities or nodes to be visited. The travelling salesman starts at one node, visits all other nodes successively only one time each, and finally returns to the starting node. Given *n* cities, named $\{c_1, c_2, \ldots c_n\}$, and permutations $\{\sigma_1, \sigma_2, \ldots \sigma_n!\}$, the objective is to choose σ_i such that the sum of all Euclidean distances between each node and its successor is minimized. The successor of the last node in the permutation is the first one. The Euclidean distance *d*, between any two cities with coordinates (x1, y1) and (x2, y2) is calculated by:

$$d = \sqrt{|x1 - y1|^2 + |x2 - y2|^2} \qquad (1)$$

The problem with the travelling salesman problem is the rapid increase on the number of possible routes when the number of cities increase [11]. However, this can be solved using optimization techniques.

4 Genetic Algorithms

A number of algorithms have been developed to solve the TSP. Some of which are linear, dynamic, Monte Carlo, and heuristic search methods. However, results showed that genetic algorithms produced the lowest distance solution among the mentioned optimization methods [12].

Genetic algorithms are computational models inspired by evolution that provides a potential solution to a specific problem. It has a wide range of applications from optimization, test pattern generation, voice recognition, and image processing. It solves problems by mimicking the same processes Mother Nature uses. Usually, when we want to solve a particular problem, we are looking for some solution, which will be the best among others. The space of all feasible solutions is called a search space (state space). Each point in the search space represents one feasible solution. Each feasible solution can be “marked” by its value or fitness for the problem. Genetic algorithm starts with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. Its motivation is hope that the new population will be better than the old population. Solutions which are selected to form new solutions (offspring) are selected according to their fitness – the more suitable solution has a higher chance to reproduce. However, this depends on the selector operator used. The next phase is the crossover phase wherein the selected individuals are mated pair by pair to form a new offspring. Finally, some of the offspring are mutated. This is repeated until some condition (e.g. no. of populations or improvement of the best solution) is satisfied [13].

5 System Implementation

Using the travelling salesman problem as basis and using genetic algorithms to generate a solution, an Android-based disaster management system named MyDisasterDroid (MDD) was implemented. The MyDisasterDroid system is shown in Fig. 1.

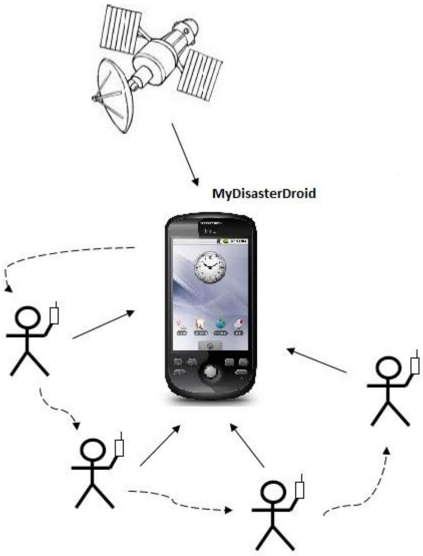


Fig. 1 MyDisasterDroid System

5.1 Geolocation

First, geographic locations of people in need are set. There are two ways that geographic locations can be set: using the application installed in MyDisasterDroid or sending location via text or short message service (SMS) to MyDisasterDroid.

Geographic locations or geolocations are described in latitude and longitude. Based on the TSP, distances between these locations were calculated. A method in the Android operating system determines the distance between geolocations and it is based on the WGS84 ellipsoid or the World Geodetic System 84, which is currently being used by the GPS at present. These calculated distances were then used in the genetic algorithm implementation in MyDisasterDroid.

5.2 MyDisasterDroid Genetic Algorithm

Fig. 2 shows a block diagram of the genetic algorithm implementation used in MyDisasterDroid. Based on the figure, after setting the geolocations and computing the distances between geolocations, initial solutions or chromosomes are generated. These chromosomes represent probable routes along the set geolocations. Then, the fitness of each solution (or chromosome) were calculated. This is equal to the summation of the distances between the different locations based on the order set by the particular solution. The shorter the distance, the fitter the solution, and a higher fitness value is achieved. Since the fitness value is very large, a linear transformation was done to reduce its value. This is reflected as the score value that is, a lower score means a higher fitness value.

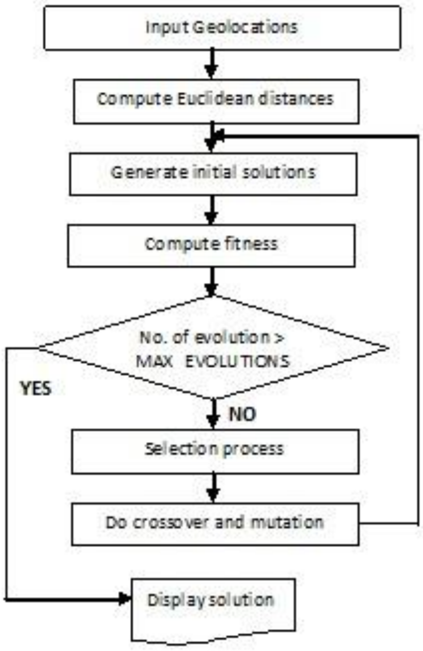

Input Geolocations
Compute Euclidean distances
Generate initial solutions
Compute fitness
No. of evolution > MAX EVOLUTIONS
YES
NO
Selection process
Do crossover and mutation
Display solution

Fig. 2 Block Diagram of the Genetic Algorithm Implementation

A new set of solutions are then selected. There are a number of selection operator algorithms: best fit chromosome selector algorithm (or elitism), threshold selector algorithm, and tournament selection algorithm to name some of them. In elitism, the best chromosomes are copied to the new population. This increases performance of the genetic algorithm because it prevents losing the best found solution. The best chromosomes are decided by evaluating their fitness value. In threshold selection, a certain level of threshold or fitness is needed for a certain chromosome to be copied to the new population. In tournament selection, tournaments are held among a few chromosomes chosen based on a certain probability. The winner of each tournament or the chromosome with the higher fitness is selected to be copied to the new population.

Crossovers and mutations were then done on the solutions. This is because solutions tend to converge faster resulting to a probability that a near optimal solution will not be achieved. Thus, it is necessary to perform crossovers and mutations in order to generate a much fitter solution.

There are a number of crossover operators: partially mapped crossover, order crossover, position based crossover, fragment reordering crossover, and single point crossover [14] to name some of them. However, in this problem, the crossover algorithm used was the greedy crossover. In this algorithm, the first location of one parent solution is selected. It compares the distances from that particular location to all locations and chooses the closer one as the next destination for its route. If the nearest location is already a part of the route, another location is chosen.

After a crossover is performed, mutation takes place. The swapping mutation operator was used wherein the genes (representing the different locations) are swapped. As end result, a new generation of solutions are produced and eventually, after successive generations, an optimum route is determined. In this genetic algorithm, a path representation was used since it is perhaps the most natural representation of a tour or route [15]. This route is then displayed on the MyDisasterDroid application.

5.3 Android Mobile Application Features

Fig. 3 shows the welcome screen of the MyDisasterDroid application. This application is based on the Android operating system.

Upon entering MyDisasterDroid, it shows two views: MapView and ListView as shown in Fig. 4. MapView shows the location map and it is based on Google Maps while the ListView shows a list of the people in need, their corresponding locations, and their distance from MyDisasterDroid as shown in Fig. 5.

MapView also provides different map images. Fig. 6 shows the location map in Satellite View. Fig. 7 shows the location map in Street View. Fig. 8 shows the location map in Traffic View.

The set geolocations are identified as Android markers on the map as shown in Fig. 9. These markers correspond to the geographical locations described in Section 5.1. Upon a click of the Show Me The Route! button, the optimum route among these geolocations is displayed as shown in Fig. 10. Dynamic recalculation of the routes can be done with just a click of the said button.

Geographic locations are entered with just a tap on a specific location on the map. A dialog box appears that asks the user to enter the location name and the number of people in need in that particular geolocation as shown in Fig. 11. These locations are stored in the phone's database and it is listed down in the ListView of MyDisasterDroid.

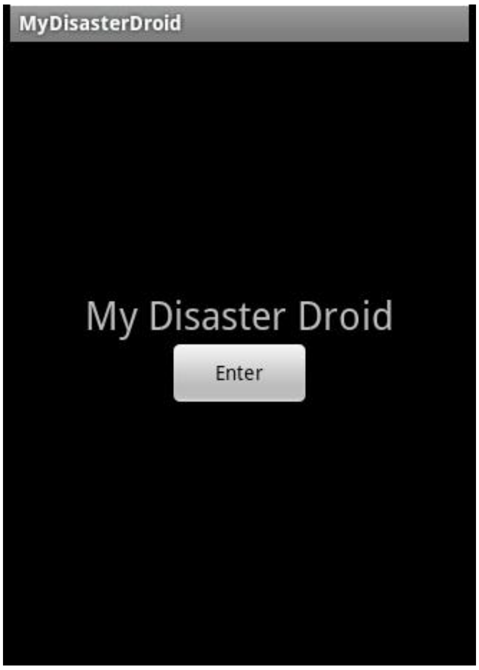


Fig. 3 MyDisasterDroid Welcome Screen

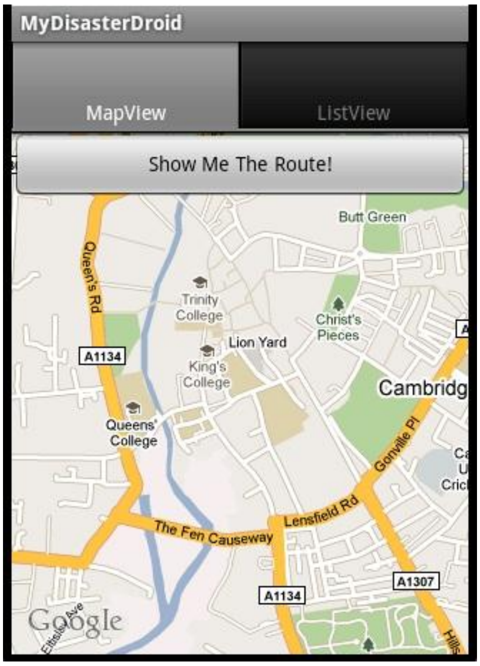
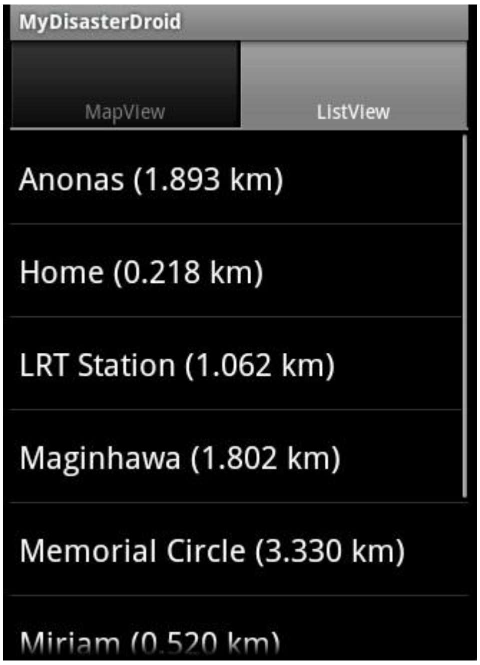


Fig. 4 MyDisasterDroid MapView



Anonas (1.893 km)

Home (0.218 km)

LRT Station (1.062 km)

Maginhawa (1.802 km)

Memorial Circle (3.330 km)

Miriam (0.520 km)

Fig. 5 MyDisasterDroid ListView

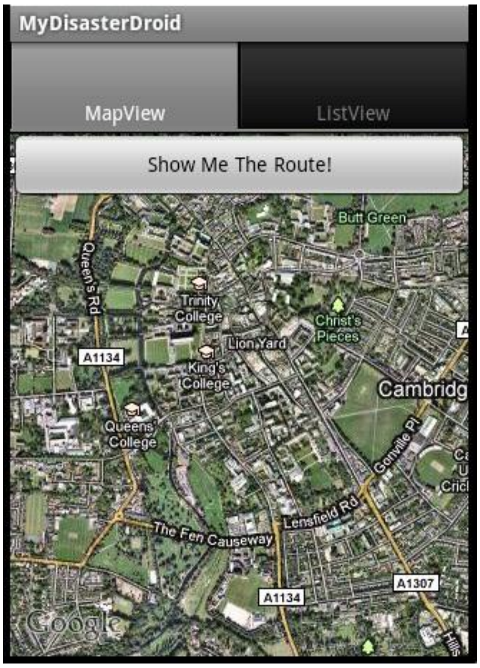


Fig. 6 MyDisasterDroid Satellite View

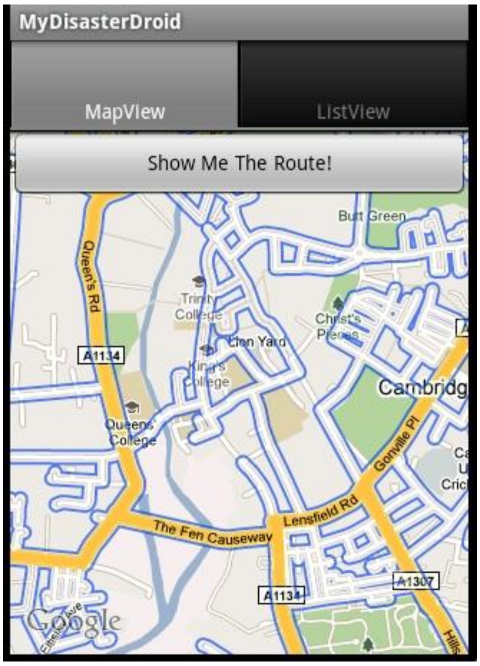


Fig. 7 MyDisasterDroid Street View

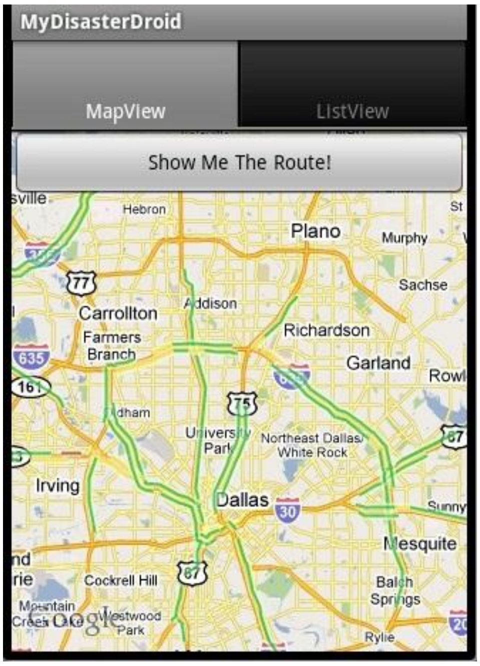


Fig. 8 MyDisasterDroid Traffic View

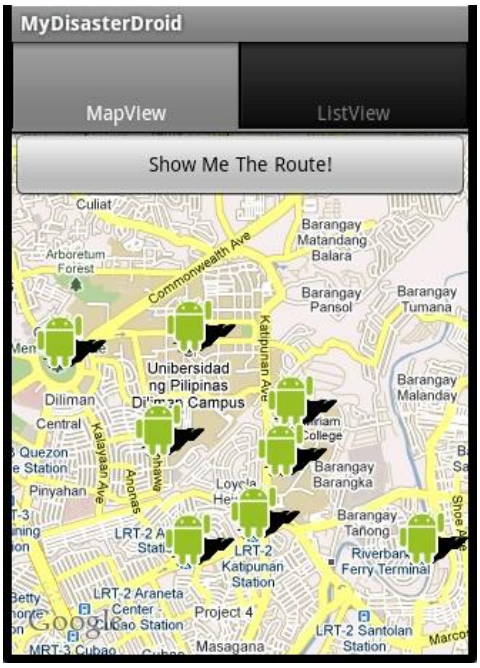


Fig. 9 MyDisasterDroid Geolocations

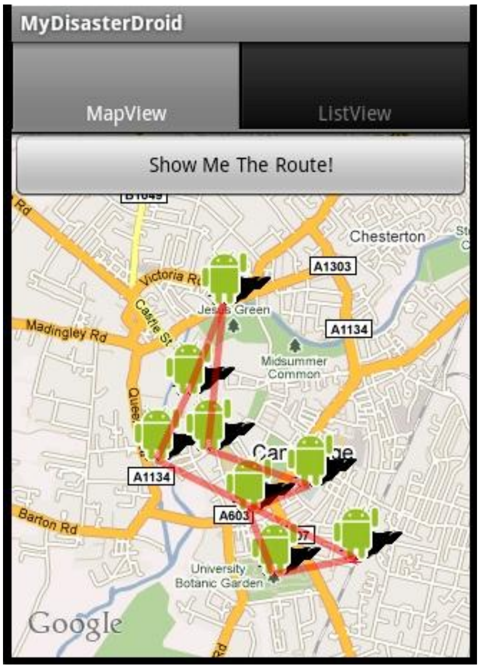


Fig. 10 MyDisasterDroid Sample Route

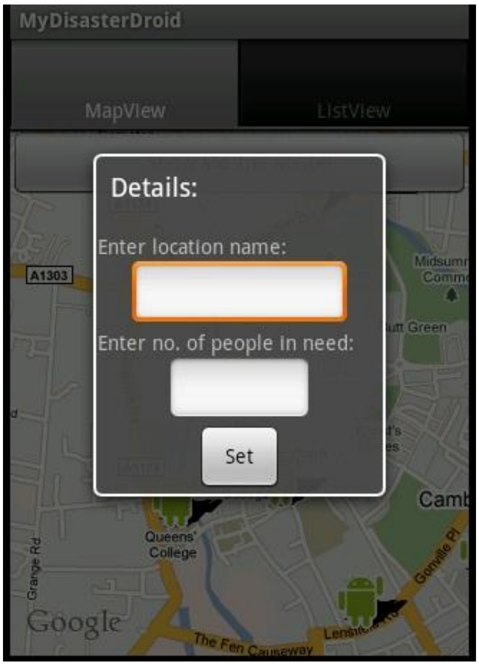


Fig. 11 MyDisasterDroid Details Dialog Box

5.4 MDD Genetic Algorithm Parameters

The MyDisasterDroid genetic algorithm was tested for various parameters such as the number of evolutions, selection operators, mutation rates, and priorities. The data used for the algorithm are shown in Table 1.

Location Name	Latitude, microdegrees	Longitude, microdegrees	No. of People In Need
Anonas	14627648	121064896	5
Home	14638115	121076119	0
LRT Station	14630887	121061291	15
Maginhawa	14640437	121061291	10
Memorial Circle	14650734	121049447	5
Miriam	14643759	121077320	5
Provident	14627814	121093046	20
UP	14652560	121065048	12

Table 1 MyDisasterDroid Sample Geolocations

First, the acceptability of using the genetic algorithm to provide a solution to the problem is based on the number of evolutions. As the number of evolutions increase, a more optimum solution is achieved. Here, the number of evolutions were varied to determine if there is a significant difference in solution. This is due to the fact that the implementation of a genetic algorithm needs processing power that grows with an increase in the number of evolutions. Since a handheld device is used, a lower number of evolutions is preferred. Based on Table 2, a higher fitness or a lower score was achieved at 40 evolutions as compared to 50 evolutions. Thus, for the next runs, the number of evolutions was set to 40.

From the computation times also shown in Table 2, it is shown that there is a marked difference in computation time of the MDD genetic algorithm as compared to the computing time of 35 seconds in an AMD 2.5GHz PC that used the following GA parameters: population size of 40, crossover probability of 0.8, mutation of 0.4, and 80 generations or evolutions [12]. However, the smart phone's processing power is continuously improving [16] and eventually, the computation time will decrease over time.

No. of Evolutions	Average Score	Average Computation Time (ms)
5	18278	4036
10	14253	23052
20	13929	73382
30	13459	93721
40	13442	127364
50	13765	203761

Table 2 Average Scores vs. Number of Evolutions (GA Parameters: Best Fit Chromosome or Elitism Selector, Greedy Crossover, Swapping Mutation Rate of 20)

With 40 evolutions, the genetic algorithm selection operator was varied. First, the best fit chromosome or elitism selector was used. Second was the threshold selector. Using this operator, the threshold or probability was set to 0.5. Last was the tournament selector. The tournament size was set to 50 and the probability was set to 0.5.

Selector Operator	Average Score	Average Computation Time (ms)
Best Fit Chromosome (Elitism)	13442	127264
Threshold	12975	237443
Tournament	12423	276435

Table 3 Average Scores vs. Selector Operator (GA Parameters: 40 Evolutions, Greedy Crossover, Swapping Mutation Rate of 20)

Based on Table 3, using the tournament selector operator achieved the lowest score at the expense of a longer execution time. However, tournament selection is increasingly being used as a genetic algorithm selection scheme because it can easily adjust the selection pressure by changing the tournament size [17]. Also, if the tournament size is larger, weak individuals have a smaller chance to be selected. Thus, using the tournament selector operator, the tournament size was varied. Table 4 shows the average scores and average computation times of the different tournament sizes. Both tournament size 20 and 50 achieved a low score but a tournament size of 50 took a longer execution time. Thus for the next runs, a tournament selector operator with a tournament size of 20 was used in the algorithm.

Tournament Size	Average Score	Average Computation Time (ms)
20	12423	273097
50	12423	276435
80	12755	279324

Table 4 Average Scores vs. Tournament Size (GA Parameters: 40 Evolutions, Tournament Selector, Greedy Crossover, Swapping Mutation Rate of 20)

Using the tournament selector operator with a tournament size of 20 and probability set to 0.5, the swapping mutation rate was varied. For the previous runs, the mutation rate was set to 20 wherein 1/20 genes are mutated on average. Table 5 shows the average scores and computation times and based on the results, a mutation rate of 20 achieved the lowest score and the fastest execution time.

Swapping Mutation Rate	Average Score	Average Computation Time (ms)
2	12746	468720
20	12423	273097
200	13528	279891

Table 5 Average Scores vs. Mutation Rate (GA Parameters: 40 Evolutions, Tournament Selector, Greedy Crossover)

Thus, for this system, the best combination of genetic algorithm operators is to use the tournament selector operator with a tournament size of 20 and a swapping mutation operator with a mutation rate of 20 for 40 evolutions. Fig. 12 – Fig. 19 show the different solutions for this problem.

Moreover, in a disaster management system, one common problem is prioritization. It is a common scenario that people need immediate help. However, not all people can be served at the same time because of limited resources. In this system, two priorities were incorporated. The first is prioritizing the closer location from the rescuer and the second is prioritizing a certain location with more people in need.

For the first priority, only the distances between the geolocations were taken into consideration. However, for the second priority, which prioritizes more people in need, a certain weight is multiplied to a particular location. This weight is equal to the inverse of the number of people in need. In effect, given a constant distance, a particular location with more people in need is prioritized since its value is lower as compared to a location with a lesser number of people in need. As a result, with this disaster management system, it is possible to incorporate different priorities as needed by the situation at hand.

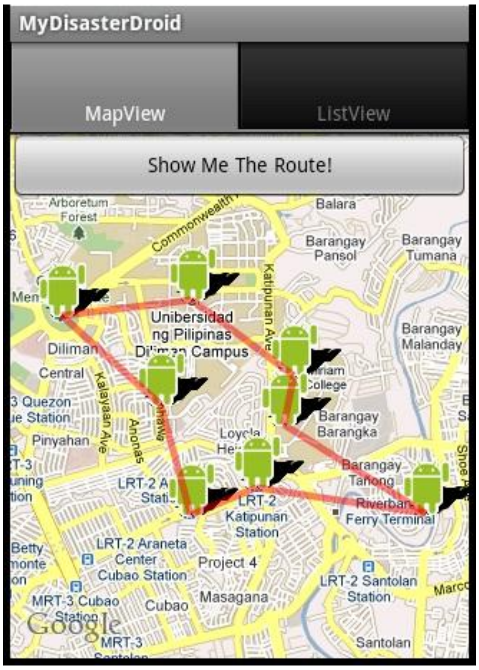


Fig. 12 Route With a Score of 12423

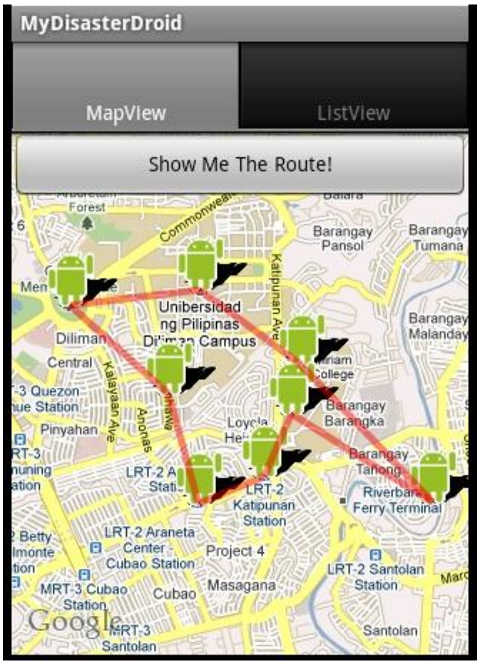


Fig. 13 Route With a Score of 12908

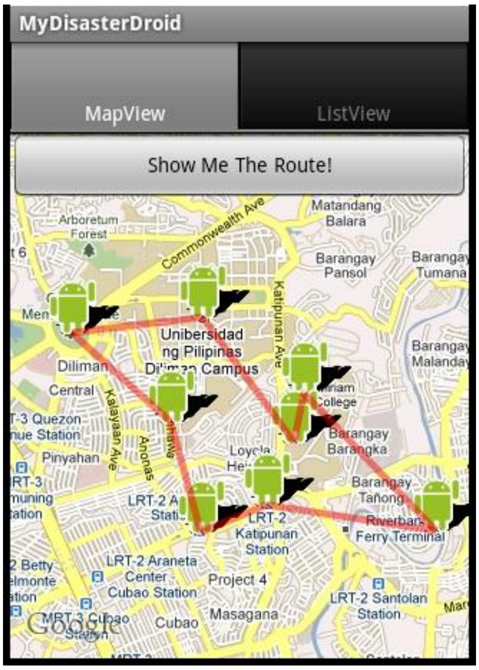


Fig. 14 Route With a Score of 13083

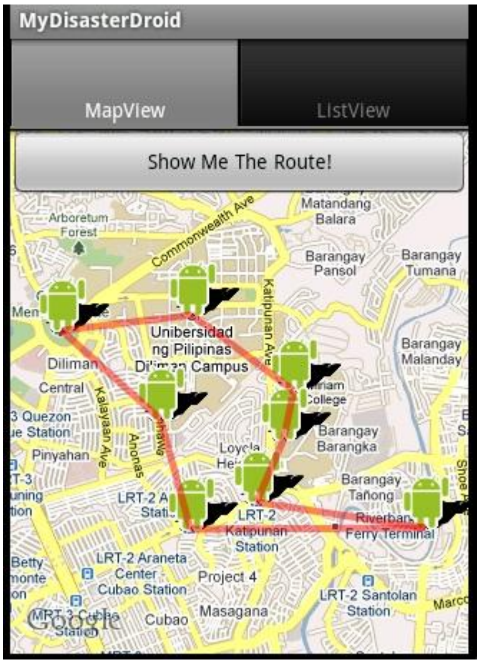


Fig. 15 Route With a Score of 13261

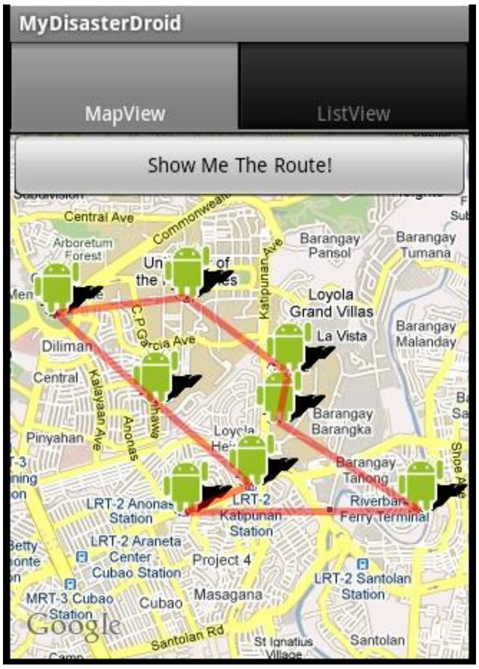


Fig. 16 Route With a Score of 13419

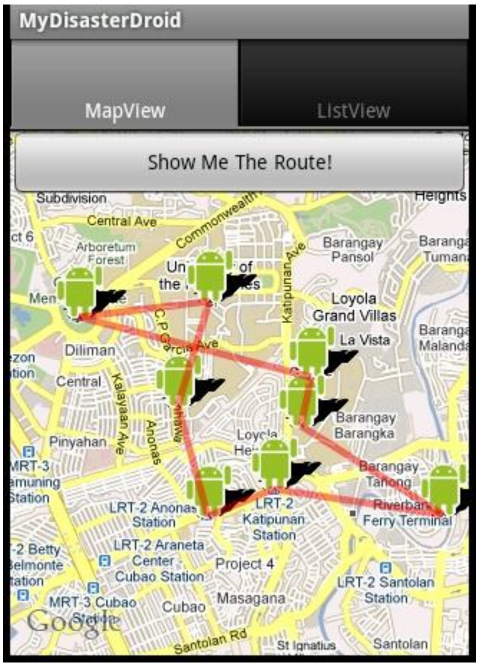


Fig. 17 Route With a Score of 13572

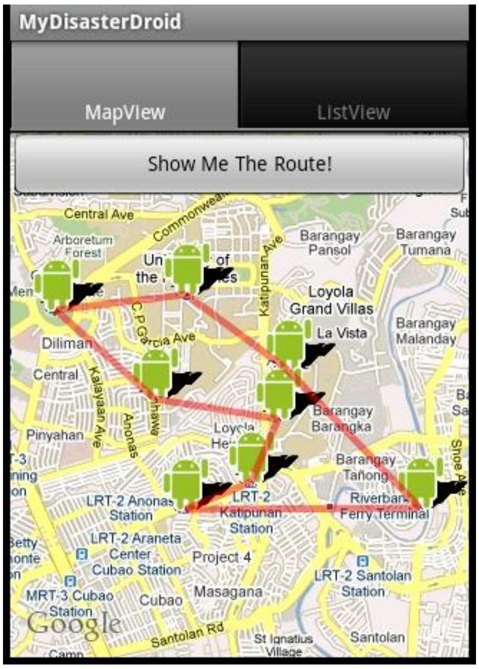


Fig. 18 Route With a Score of 13957

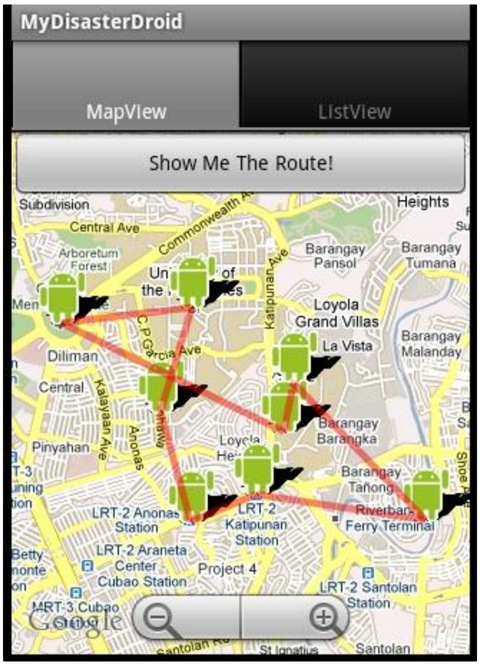


Fig. 19 Route With a Score of 13977

6 Conclusion

A disaster management system that facilitates the logistics for the rescue and relief operations during a disaster known as MyDisasterDroid was implemented in an Android- based mobile phone. Geographic locations of the people in need were sent via SMS or inputted directly to MyDisasterDroid. Determining the optimum route along the different geographic locations is similar to solving the travelling salesman problem wherein the geographic locations correspond to the cities and the rescuers or volunteers correspond to the travelling salesman. Using genetic algorithm, an optimum route along the given geographic locations was determined. Different genetic algorithm parameters were varied and based on the results, the best combination of genetic algorithm operators to use are the tournament selector operator with a tournament size of 20 and a swapping mutation operator with a mutation rate of 20 for 40 evolutions. Moreover, this system is also flexible because it allows the prioritization to be changed or extended with minimal effort. Indeed, MyDisasterDroid is an application that can be used during the response phase in a disaster especially when time is crucial.

7 Acknowledgements

This research was funded by the Department of Science and Technology – Engineering Research and Development for Technology program and the Department of Electronics, Computer, and Communications Engineering of the Ateneo de Manila University.

References:

[1] United Nations International Strategy for Disaster Reduction Secratariat (UNISDR), *Mortality Risk Index*, 15 June 2009.

[2] Damon P. Coppola, *Introduction to International Disaster Management*, Elsevier Inc., 2007.

[3] Bartel Van de Walle, Gerd Van Den Eede, and Willem Muhren, Humanitarian Information Management and Systems, *Mobile Response: Second International Workshop on Mobile Information Technology for Emergency Response 2008, Bonn, Germany, May 29-38, 2008, Revised Selected Papers*, Springer, 2009.

[4] Asia Pacific Telecom Research Ltd., *Telecommunications in the Philippines*, 1 June 2009.

[5] Kozel Tomas, Maly Filip, and Slaby Antonin, Mobile approach, trends, and technologies in modern information systems, *7th WSEAS International Conference on Applied Computer and Applied Computational Science*, Hangzhou, China, April 6-8, 2008, pp. 716-720.

[6] Chowdhury G. Hossan, Mridul Chowdhury, and Ibrahim Kushchu, Prospects of Using m-Technologies for Disaster Information Management in Bangladesh and other LDCs, *EURO mGOV 2005*, Brighton, UK, pp. 243-253.

[7] Tom Krazit, “Smartphones will soon turn computing on its head”, *CNET News*, 31 March 2008.

[8] W. David Stephenson, “Growing smartphone use will have dramatic impact on disaster response”, *FutureBlogger*, 09 April 2008.

[9] Barbara DePompa, “Smartphones & PDAs”, *Government Computer News*, http://gcn.com/Microsites/Smart-Phone-PDAs/COOP-Smartphones.aspx.

[10] Information Gatekeepers Inc., The mobile internet, *Mobile Internet*, Vol. 9, No. 11. 2007, pp. 1-19.

[11] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley & Sons Ltd., 1985.

[12] A.R. Awad, I. Von Poser, and M.T. Aboul-Ela, Advanced Intelligent Technique of Real Genetic Algorithm for Traveling Salesman Problem Optimization, *Proceedings of the 9th WSEAS International Conference on Data Networks, Communications, and Computers,* Trinidad and Tobago, November 5-7, 2007, pp. 447-453.

[13] Melanie Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1996.

[14] Minea Skok, Davor Skrlec, and Slavko Krajcar, The Genetic Algorithm Scheduling of Vehicles from Multiple Depots to a Number of Delivery Points, *Proceedings of the WSES International Conference on Evolutionary Computation*, Tenerife Playa, Canary Islands, Spain, February 12-14, 2001.

[15] Amin Mohebifar, New binary representation in Genetic Algorithms for solving TSP by mapping permutations to a list of ordered numbers, *Proceedings of the 5th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, Venice, Italy, November 22-26, 2006, pp. 363-367.

[16] Suzanne Choney, “The rise of the super-smartphone”, *msnbc*, 18 March 2010.

[17] B.L. Miller and D.E. Goldberg, Genetic Algorithms, Tournament Selection, and the Effects of Noise, *Complex Systems*, Vol. 9, 1995, pp. 193-212.