



Faculty of Computer and Mathematical Sciences

UiTM Shah Alam

Group Project Assignment (Softcopy Submission)

Project Title: VACCINATION FACILITY ALLOCATION OPTIMIZATION

Semester	: March 2021 – August 2021
Course Code	: ISP611
Lecturer' Name	: Prof Madya Ts. Dr. Marina Yusoff

NAME	STUDENT ID	Group	Role In Project
MOHAMAD NAQUIDDIN BIN MOHD ZAINI	2020957767	CS2595A	Project Manager
MUHAMMAD HAKIM BIN MUHAMMAD AMINUDDIN	2020985189	CS2595A	Programmer 1
MOHAMAD AFIQ RADZUAN BIN MOHAMAD RADZI	2020982873	CS2595A	Programmer 2
MOHAMAD HAIKAL HAIZAD BIN AHMAD FADZLI	2020964085	CS2595A	Researcher

Deadline: Week 13, Wednesday 6 PM

Actual Date of Submission	7/7/2021	No. of Days Late:	
---------------------------	----------	-------------------	--

ABSTRACT

Facility site optimization (also known as facility placement allocation) is a typical challenge in urban infrastructure constructions (such as hospitals, schools, parks, and community services). Malaysia's level of living and household per capita income have risen dramatically. One of the keys to obtaining a high standard in healthcare is more methodical and efficient planning, which has become increasingly vital. The Location-Allocate(LA) Model was created specifically to address these issues. We look at the algorithms that were used to create the facility localization model in this study. Genetic Algorithm (GA) is one of the algorithms that has been applied to the LA model in this paper. We demonstrate the ideas first, then incorporate Genetic Algorithms into the application. Finally, we create a general example to demonstrate the algorithm's viability in facility placement optimization, as well as a summary of the algorithm's benefits and downsides.

TABLE OF CONTENT

Content	Pages
1.0 Introduction	3
2.0 Problem Statement	4
3.0 Detail Description	5
4.0 Related Work	6
5.0 Solution Representation	8
5.1 Sample Case Study	8
5.2 Data with Explanation	9
5.3 Representation with Explanation	13
5.4 Objective Function with Explanation	14
6.0 Algorithm Design	15
6.1 Algorithm Phases	15
6.2 Algorithm Coding	18
7.0 Performance of Algorithm	26
8.0 Conclusion	31
References	32

1.0 INTRODUCTION

For most urban planning and spatial decision support systems, deciding on the best site for a facility is a strategic challenge. Considering how to overcome this challenge led to the development of location theory. Alfred Weber (1909), who was investigating the challenge of locating a single warehouse and minimising the total trip distance between warehouse and a series of spatially spread customs, first explicitly established the location theory in 1909. Isard (1956) revisited this work in his study on industrial location, land use, and associated issues.

Based on the National COVID-19 Immunisation Programme in Malaysia, this project would focus on optimizing the vaccination facility as it is very crucial at this moment. According to *Jawatankuasa Khas Jaminan Akses Bekalan Vaksin Covid-19* (JKJAV), the programme targets 80% of the population in Malaysia to be vaccinated at 386 vaccines administration centres.

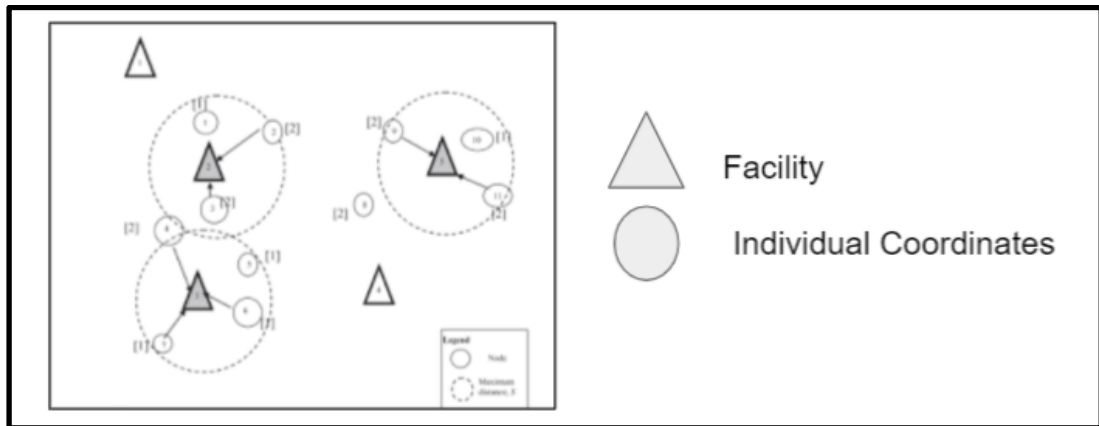
The location of vaccine administration is critical in ensuring that the chosen location network achieves the objective of minimising societal costs or, in other words, maximising people's benefits. Similarly, the allocation of demand to these facilities has a direct impact on the overall efficiency of the system. This location-allocation model is important for health-care planning because it provides a framework for analysing accessibility issues, evaluating the quality (in terms of efficiency) of prior location selections, and proposing alternative ways to change and improve the current system (Rahman & Smith, 1999).

Thus, it is vital to determine the optimal location of the vaccination facility to satisfy the demand from Malaysians who have registered for the vaccines.

2.0 PROBLEM STATEMENT

On May 29, The Ministry of Health (MoH) reported a total of 9,020 new cases, marking the fifth consecutive day of record high daily infections, which is a concerning situation. By June 13, 2021 just 43.04 percent of the population (13,772,992 persons) had registered for vaccination, according to National Immunization Program (NIP) Coordinating Minister Khairy Jamaluddin. It is still a long way from the country's goal of vaccinating more than 80% of the population (almost 27 million out of 32 million Malaysians) by the end of the year. There's growing fear that Malaysia won't be able to complete Phase Three of the NIP on time, and hence won't be able to deal with the country's growing number of Covid-19 cases. On June 9, more than 150,000 doses of Covid-19 vaccination were given out, the highest daily rate yet. The amount matches the government's goal of 150,000 daily immunisation doses by the end of this month, with the goal of 200,000 doses by next month. JKJAV announced the newest development on social media. The rise in daily vaccinations was aided by the addition of 276 vaccination centres (PPV) around the country, as well as an increase in drive-thru PPVs. Today, there are 386 PPVs across the country and we hope that the number keeps increasing as it plays a major role in daily immunisation doses in Malaysia.

The major problem that needs to be addressed is assigning individuals to the nearest vaccination centres. Thus, several assumptions have been made to complete this project. Firstly, the capacity of vaccination centres are able to accommodate the demand of all people in Selangor by adding new facilities to the current facility. Secondly, vaccines are enough according to demand.



3.0 DETAIL DESCRIPTION

Under the second phase of the National Covid-19 Immunisation Programme, Penang is considering a two-pronged plan to increase the number of vaccination centres and site vaccines in industrial regions. Dr. Norlela, the head of the state health committee, said the state was considering obtaining a larger amount of vaccinations in the second phase, citing the need for more vaccination administration centres. She stated, "I had also requested for vaccination by private clinics and hospitals, as well as vaccination at companies, so that the procedure may be done more cost-effectively and minimise public sector labour." According to Dr. Norlela, there are now five vaccination centres (PPVs) and 11 Health Ministry hospitals that are capable of administering vaccinations. "In the second phase, there are no new private hospitals to provide vaccinations, and we must wait for news from the Health Ministry whether there are any." Private hospitals that are interested in performing the immunisation can register with the ministry. "Let's wait for additional vaccinations to come at the end of May since it's about vaccine availability. We will be able to open more immunisation centres as the population grows.

4.0 RELATED WORK

Location of the facilities providing public health service is very crucial in ensuring that the chosen location network serves the purpose. Shariff et al. (2012) had proposed a solution to solve Capacitated Maximal Covering Location Problem (CMCLP) by using a modified GA (mGA) based method. The percentage of coverage of current facilities within the permissible distance targeted by the Malaysian government was examined using this genetic algorithm. The fitness function is to fulfill the population assigned to a facility within the coverage distance. The accurate the number of population that had been assigned to a facility within the coverage distance, the better it is. For computational results, with a population size of 100, mGA was run for a maximum of 100 generations. The algorithm was run ten times, and the average coverage percentage for the maximum service distance = 3 km increased by 17.2% to 86.5 percent, while it increased by 16.9% to 94.4 percent for the maximum service distance = 5 km.

Wang et al (2018) deal with the binary location decision and capacity adjustment by using a bi-level multi-objective particle swarm optimization (BLMOPSO) algorithm. Under complex continuous search spaces, the PSO method has been shown to find optimal solutions. Their fitness function is to inverse the anticipated travel distance to reach Healthcare Facility. The lesser the time to reach the Healthcare Facility, the more fit the travel distance is. For this research paper, it compares BLMOPSO with basic PSO and the computational result shows that the average distance for BLMOPSO is less with 0.0425 than basic PSO with 0.1712 average.

Other than that, Elkady & Abdelsalam (2016) also use the PSO algorithm to solve multi-objective problems(MOP) in an attempt to identify the optimal trade-off between two objectives. There are two fitness functions in this paper which is, to fulfil the demand coverage and reduce the travel distance. For computational results, this author compares MOP and single objective problem(SOP) and the results show that final results for MOP are better than SOP.

Eric Stoltz (2018) does evolution of salesmen using Genetic Algorithm(GA). Uses of Genetic Algorithm(GA) in this research to find a solution to the Traveling Salesman

Problem (TSP). The purpose of this research is to find the shortest possible route that visits each city and return to the origin city. Two important rules to always keep in mind is each city needs to be visited exactly one time and must return to the starting city, so total distance needs to be calculated accordingly. The fitness function is determined by nearest distance. In conclusion, this research result shows the shortest route from one city to another city and back to the origin city.

To minimize total distance, Sinthamrongruk et al (2017) proposed a new crossover operator for the travelling salesman problem. Fitness function for this problem is the nearest distance is, then it is counted as the best fitness. To solve this problem, a genetic algorithm was used. This method is associated with path representation, which is the most natural way to represent a legal tour. For computational results, the proposed crossover operator was compared to some traditional path representation crossover operators using MATLAB software where the proposed operator CX2 performed better than the PMX and OX operators.

Healthcare staff routing to provide healthcare service to the patients is one of the real-world scheduling problems similar to multiple travelling salesman problems (MTSP). Healthcare staff members provide daily medical services at patients' homes. The service provider authority has to schedule these staff in an effective and efficient way so that it achieves the minimum total cost. The aim of this study is to propose an Adaptive Local Search based on Genetic Algorithm (GA) to solve Healthcare Staff Routing Problem. Hussain et al (2017), The purpose of this research is to solve HSSP using three-step scheduling technique by dividing the problem into different sub-problems and then finding solutions with these steps for route scheduling, resource selection, and local improvement. Routes scheduling focuses on how to arrange effective routes for staff with minimum distance travel time and travel cost. Resource selection points to match qualified staff to each route with the minimum cost and the preferences by the scheduler and also a customer representative under feasible time constraint. The Local Improvement enhances the output solution generated by the resource selection using swapping tasks based on the cost function. The solutions gained in the experiment indicate that ADSL1 is the best type of GA compared to its rival algorithms in terms of the fitness value. In contrast, this method consumes the most time. For ADSL2, it gives

results improved slightly while the computational time increases meaninglessly. For entire experiments, Pc and Pm barely affect the mode. When Pc and Pm are set at a low level, they can provide better solutions with less computational time.

5.0 SOLUTION REPRESENTATION

5.1 SAMPLE CASE STUDY

We take two(2) vaccination centres, which are Convention Centre (PWTC), KL and Ideal Convention Centre (IDCC), Shah Alam. We made an assumption that there are 5 people from different locations who want to get vaccinated.

Table 5.1 Sample Study

	People A	People B	People C	People D	People E
Distance to PWTC	5km	10km	8km	7km	5km
Distance to IDCC	10km	5km	6km	5km	10km

As an example, the maximum number of people can be vaccinated in PWTC is 4, while in IDCC it is 5. Thus, based on the distance given, this project will satisfy the demand of people who want vaccines with the nearest location of vaccine centres.

5.2 DATA WITH EXPLANATION

The following table defines the coordinates of the centroid and the forecasted demand of the vaccine. The centroid chosen is based on Selangor district.

Table 5.2 Coordinates and demand of vaccine

Centroid	Coordinates
Gombak	(3.27,101.69)
Hulu Langat	(3.12, 101.84)
Hulu Selangor	(3.58, 101.70)
Kuala Langat	(3.12, 101.85)
Klang	(3.05,101.45)
Kuala Selangor	(3.34, 101.26)
Petaling	(3.13, 101.59)
Sabak Bernam	(3.78, 100.99)
Sepang	(2.70, 101.76)

The following table defines the coordinates of existing facilities and several facility assumptions.

Table 5.3 Coordinates of existing facility and the capacity

Centroid	Existing Facility	Coordinates
Hulu Langat	Dewan Orang Ramai Ampang	(3.13, 101.79)
	Hospital Ampang	(3.13, 101.77)
	Hospital Kajang	(2.99,101.79)
	Dewan Dato Nazir Hulu Langat	(3.11,101.82)

	Sekolah Menengah Kebangsaan Dusun Nanding	(3.08,101.79)
Gombak	Hospital Selayang	(3.24,101.65)
	Hospital Orang Asli Gombak	(3.29,101.73)
	Hospital Sungai Buloh	(3.21,101.58)
	Dewan Alwy, Jabatan Perhutanan Semenanjung Malaysia	(3.24,101.63)
	SMK Gombak Setia	(3.22,101.72)
	De' Orchid Gombak Setia	(3.22,101.71)
Klang	Dewan Kelab Kilat Pangsapuri TNB Kapar	(3.13, 101.38)
	Dewan Sukan Pandamaran	(3.01, 101.41)
	Hospital Tengku Ampuan Rahimah	(3.02, 101.44)
	Masjid Jamek Ar-Rahimiah	(3.02, 101.34)
	Sekolah Kebangsaan Kapar	(3.13, 101.38)
	Stadium Hoki MPK	(3.01, 101.39)
Hulu Selangor	Dewan Semai Bakti Felda Soeharto	(3.67, 101.40)
	Hospital Kuala Kubu Bharu	(3.54,101.65)
	Sekolah Kebangsaan KKB	(3.56, 101.65)
	MRSM Kuala Kubu Bharu	(3.54,101.68)
	Sekolah Kebangsaan Kampung Soeharto	(3.68,101.39)
Kuala Langat	Hospital Banting	(2.80,101.49)

	Kompleks Sukan Majlis Perbadanan Kuala Langat	(2.80,101.50)
	Stadium Jugra	(2.82,101.52)
	Kolej Mara Banting	(2.82,101.62)
Kuala Selangor	Hospital Tanjung Karang	(3.43, 101.18)
	Stadium Tertutup Kuala Selangor	(3.32, 101.25)
	UITM Puncak Alam	(3.20, 101.45)
	Community hall in Kampung Haji Razali	(3.34, 101.17)
Sabak Bernam	Dewan Seri Bernam Sg Besar	(3.68, 100.99)
	Hospital Tengku Ampuan Jemaah	(3.76, 100.99)
	SJK(T) Ladang Sabak Bernam	(3.75, 100.98)
Sepang	Poliklinik Medi Ihsan	(2.87, 101.70)
	Orion Clinic	(2.92, 101.64)
	Klinik Masya	(2.78, 101.75)
	Klinik Hazra Medic	(2.83, 101.69)
	Klinik Dr Nina	(2.80, 101.74)
	Hospital Serdang	(2.98, 101.72)
	Dewan Serbaguna Bandar Baru Salak Tinggi, Sepang	(2.81, 101.74)
	SJK (Tamil) West Country Barat	(2.97, 101.72)
Petaling	Hospital Shah Alam	(3.07,101.49)
	Klinik Rakan Medik	(3.11,101.60)

	Sunway Pyramid Convention Centre	(3.07,101.61)
	Columbia Asia Hospital	(3.09, 101.45)
	KPJ Selangor Specialist Hospital	(3.07, 101.54)
	Dewan Serbaguna Dato' Hormat PJS 10/34	(3.08, 101.60)

5.3 REPRESENTATION WITH EXPLANATION

The chromosome representation will be a representation of a person. In the chromosome it will contain the person coordinate, ppv coordinate, vaccine status and facility demand. Below is the example of the representation:

	Value	Discrete Representation
Person Coordinate (e.g: Gombak)	(3.27,101.69) ⁰	32710169
Initial PPV Coordinate	(0.00,000.00) ⁰	00000000
Vaccine Status	N/A	0 - Not yet received place for vaccine 1- Has received place for vaccine
Facility	N/A	Top 3 nearest facility

The longitude and latitude were converted to discrete representation with no decimal places considered. The first 3 digits represent the latitude and the last 5 digits represent the longitude that is related to this project.

Chromosome Representation:

Coordinate of Person								Initial Coordinate of PPV								Status	Facility
3	2	7	1	0	1	6	9	0	0	0	0	0	0	0	0	0	3

The vaccination centre's coordinate at first of the process will be declared as 00000000 as no vaccination centre is initialized yet to the person. Once the nearest and available vaccination centre is found, it will overwrite the

chromosome. The status also will be overwritten from 0 to 1 if the person were initialized to a vaccination centre.

5.4 OBJECTIVE FUNCTION WITH EXPLANATION

Objective Function:

To satisfy the demand of the vaccine in Selangor district by maximizing the facility allocation of the vaccine centre based on minimal distance.

$$\text{Min } Z = \sum_{f \in F} Dist_{c,f} * x_{c,f}$$

Constraints:

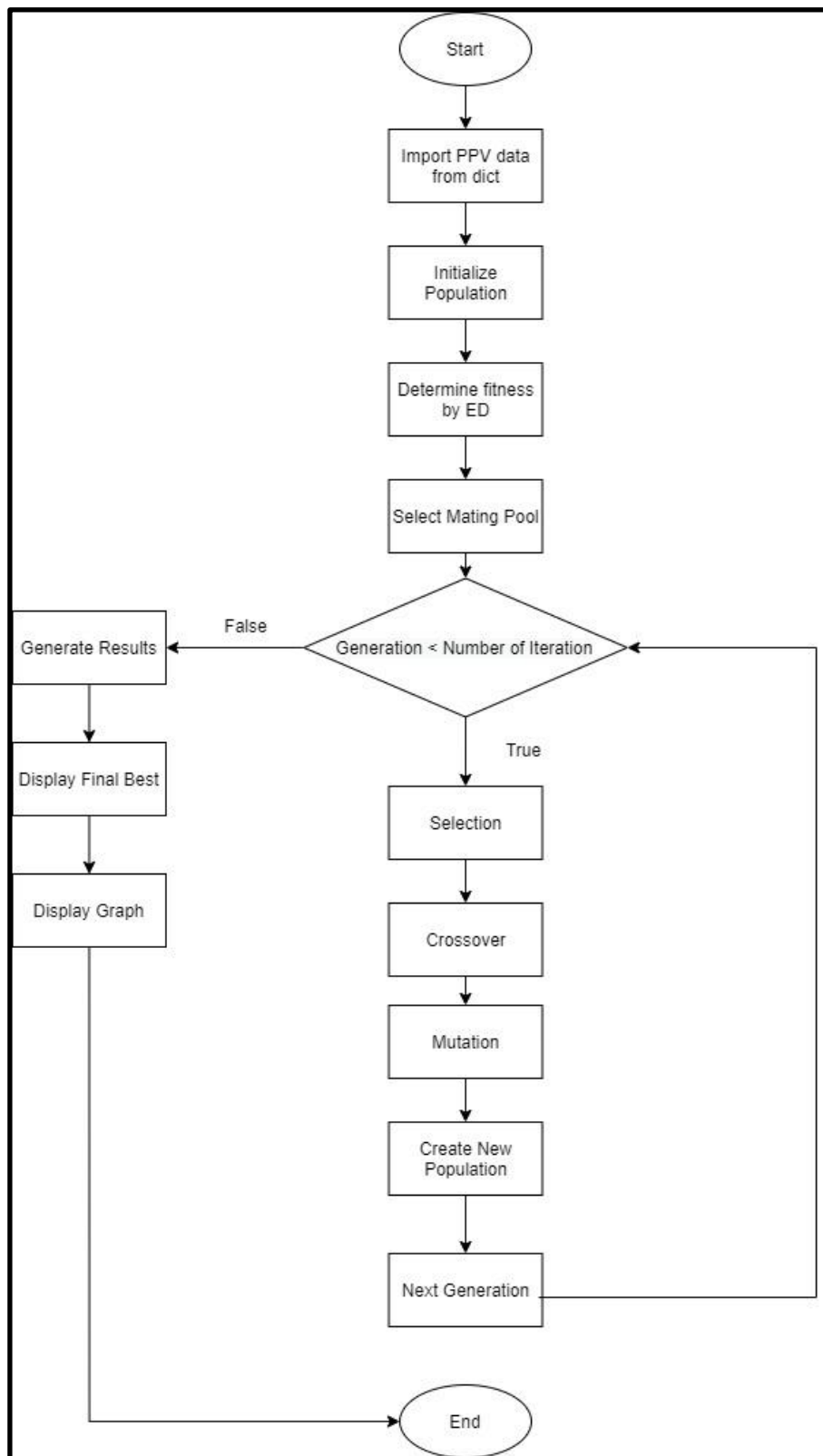
1. Existing Facilities.

Capacity of an existing location of a facility cannot exceed.

$$\sum_{c \in C} x_{c,e} \leq Cap_e \quad \forall e \in E$$

6.0 Algorithm Design

6.1 Algorithm Phases



Population

The coordinate of latitude and longitude of population were randomly generated to create chromosome for each population

Fitness

A function that tells us the nearest for each distance. In this project, the fitness is calculated by the nearest distance using Euclidean Distance (ED). Thus, the nearest the distance, the highest fitness. Thus, it has a higher chance to be chosen.

Selection

By using tournament selection, we select parents that will be used to create children. In this project we select parents based on the winning population and it will return the selected parents

Crossover

The crossover was performed using two point crossover. We swap the chromosome from index number 4 to index number 11.

Parent 1:

Coordinate of Person								Facility									
3	6	9	1	0	1	8	6	3	5	6	1	0	1	6	5	1	3

Parent 2:

Coordinate of Person								Facility									
3	6	9	1	0	1	8	6	3	3	2	1	0	1	4	5	1	3

After Crossover:

3	6	9	1	0	1	8	6	3	3	2	1	0	1	4	5	1	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

3	6	9	1	0	1	8	6	3	5	6	1	0	1	6	5	1	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Mutation

This procedure used is random resetting mutation that selects one or more genes (array indices) and replaces their values with another random value from ranges of 0 to 9. It is controlled by a low probability of the “*r_mut*” hyperparameter.

								Facility									
3	6	9	1	0	1	8	6	3	5	6	1	0	1	6	5	1	3

Example after mutation:

3	6	9	1	0	1	8	6	3	5	6	1	0	7	6	5	1	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

6.2 Algorithm Coding

(Coding: <https://colab.research.google.com/drive/17f6wT7jTAJaHQV1nkUyGFTyeUPNcGrA1?usp=sharing>)

No.	Phases	Explanation	Coding
1	Initialize coordinate	Create the coordinate for the chromosome	<pre>def popu_chromo(population): kromosom = [] coor_kromosom = [] for i in range(0, population): coor_kromosom.append(City(x=round((random.uniform(2.80, 3.80)), 2), y=round((random.uniform(100.95, 101.99)), 2)))</pre>
2	Initialize chromosome	Create full chromosome representation	<pre>latx = str(round(coor_kromosom[i].x * 100)) longy = str(round(coor_kromosom[i].y * 100)) kromosom.append(latx + longy + '00000000' + '03') return kromosom</pre>
3	Find distance	Calculate distance between coordinate of chromosome to selected PPV in Selangor	<pre>def cal_distance(places): distance = [] for i in places: kro = i latk = float(kro[8:11]) / 100 longk = float(kro[11:16]) / 100 latp = float(kro[0:3]) / 100 longp = float(kro[3:8]) / 100 x = latp - latk y = (longp - longk) * cos((latp + latk)*0.00872664626)</pre>

			<pre> dist=111.319 * sqrt(x*x + y*y) roff_dist = round(dist) distance.append(roff_dist) return distance </pre>
4	Tournament Selection	<p>Select parents that will be used to create children.</p> <p>The tournament selection procedure can be implemented as a function that takes the population and returns one selected parent.</p>	<pre> def selection(kromosom, calc_distance, k=3): selection = randint(len(kromosom)) for s in randint(0, len(kromosom), k-1): if calc_distance[s] < calc_distance[selection]: selection = s return kromosom[selection] </pre>
5	Choosing the highest and lowest fitness	<p>Choose the highest distance and the lowest distance and return the value to calculate their accuracy in the next function.</p>	<pre> def Hfitness(x, n_pop): fitness = x[0] for i in range(0,n_pop): if x[i] > fitness: fitness = x[i] return fitness def Lfitness(high,x,n_pop): fitness = high for i in range(0,n_pop): </pre>

			<pre> if x[i] < fitness: fitness = x[i] return fitness </pre>
6	Accuracy	Obtain the distance of the chromosome from the PPV and calculate the accuracy of the distance and return the result.	<pre> def accuracy(distance, lowest): accuracy = [] if distance == lowest: accuracy = 100 else: accuracy = 100 - (distance - lowest) return accuracy </pre>
7	Crossover	The function implements crossover using a draw of a random number in the range [0,1] to determine if crossover is performed, then selecting a valid split point if crossover is to be performed.	<pre> def crossover(p1, p2, r_cross): pt1 = [] pt2 = [] c1, c2 = p1, p2 if rand() < r_cross: # select crossover point that is not on the end of the string pt1 = p1[4:12] pt2 = p2[4:12] # perform crossover c1 = p1[:4] + pt2 + p1[12:] c2 = p2[:4] + pt1 + p2[12:] return [c1, c2] </pre>

8	Mutation	This procedure simply uses a random resetting mutation with a low probability controlled by the “ <i>r_mut</i> ” hyperparameter.	<pre> def mutation(bitstring,r_mut): for i in range(len(bitstring)): z=randint(0,9) if rand() < r_mut: #check for a mutation bitstring[i].replace(bitstring[i],str(z)) </pre>
9	Creating a list of children based on parents list	Loop over the list of parents and create a list of children to be used as the next generation, calling the crossover and mutation functions as needed.	<pre> selected = [selection(poplace, assign) for _ in range(n_pop)] children = list() for i in range(0, n_pop, 2): p1, p2 = selected[i], selected[i+1] for c in crossover(p1, p2, r_cross): mutation(c, r_mut) children.append(c) pop = children </pre>
10.	Genetic Algorithm Function	In this function, all processes of the genetic algorithm were called, including the rate of mutation, population size and number of iteration.	<pre> def genetic_algorithm(objective, n_iter, n_pop, r_cross, r_mut, place, popu_chromo,ppv_chromo, places, cal_distance, new_chromo, objective2, accuracy): pop = popu_chromo(n_pop) print('Chromosome for each person = %s' % (pop)) z=1 place = ppv_chromo(places,z) print('PPVs Chromosome for %s: %s' % (places[z]["Name"],place)) poplace = new_chromo(place,pop) </pre>

			<pre> print('Combine person and ppvs chromosome = %s' % (poplace)) assign = cal_distance(poplace) print('Distance from person to the choosen PPVs = %s' % (assign)) highest = objective(assign,n_pop) print('The farthest distance = %d' % (highest)) lowest = objective2(highest,assign,n_pop) print('The nearest distance = %d' % (lowest)) best, best_eval = 0, highest best_accuracy = [] generation = [] print("-----") print(" Generation Chromosome Distance Accuracy ") print("-----") print("-----") for gen in range(n_iter): for i in range(n_pop): if assign[i] < best_eval: best, best_eval = poplace[i], assign[i] accuracies = accuracy(assign[i],lowest) print(">Generation : %d, new best f(%s) = %.3f , Accuracy = %d" % (gen, poplace[i], assign[i], accuracies)) best_accuracy.append(assign[i]) generation.append(gen) z+=1 if z < 48: </pre>
--	--	--	---

			<pre> place = ppv_chromo(places,z) poplace = new_chromo(place,pop) assign = cal_distance(poplace) highest = objective(assign,n_pop) lowest = objective2(highest,assign,n_pop) # best_eval = highest selected = [selection(poplace, assign) for _ in range(n_pop)] children = list() for i in range(0, n_pop, 2): p1, p2 = selected[i], selected[i+1] for c in crossover(p1, p2, r_cross): mutation(c, r_mut) children.append(c) pop = children return [best, best_eval, assign, lowest, poplace, best_accuracy, generation] </pre>
11	Main Function	All the rate of mutation, crossover, population and number of iteration were initialized here	<pre> # define the total iterations n_iter = 47 # define the population size n_pop = 100 # crossover rate r_cross = 0.9 # mutation rate r_mut = 1.0 #calc_distance = cal_distance(place, pop) </pre>

			<pre> #print(calc_distance) # perform the genetic algorithm search best, score, assign, lowest, popu, best_accuracy, generation = genetic_algorithm(Hfitness, n_iter, n_pop, r_cross, r_mut, places, popu_chromo, ppv_chromo, places, cal_distance, new_chromo, Lfitness, accuracy) print('Done!') print('f(%s) = %f' % (best, score)) accuracy_graph = [] onebyone = [] y=1 for i in popu: onebyone.append(y) y+=1 for i in range(n_pop): accuracy_graph.append(accuracy(assign[i], lowest)) </pre>
12	Graph function	To show the generation with the accuracy and to display the distance and generation	<pre> import matplotlib.pyplot as plt %matplotlib inline #set the graph size 15 inch x 5 inch plt.figure(figsize=(15,5)) plt.subplot(1,2,1) plt.plot(onebyone,accuracy_graph) plt.xlabel('Chromosome') plt.ylabel('Accuracy') </pre>

			<pre>plt.title("Accuracy of the first generation") plt.subplot(1,2,2) plt.plot(generation,best_accuracy) plt.xlabel('Generation') plt.ylabel('Distance') plt.title("Euclidean distance in generation") plt.show()</pre>
--	--	--	---

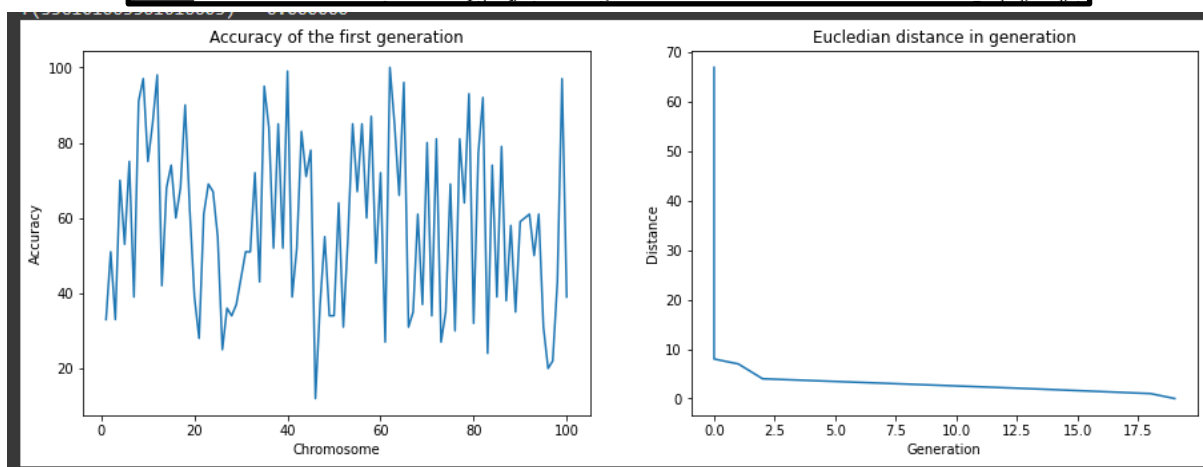
7.0 Performance of Algorithm

I. General Parameter

Population Size: **100**

Mutation rate: **1.0**

```
Chromosome for each person = ['373101710000000003', '356101700000000003', '289100970000000000']
PPVs Chromosome for Dewan Orang Ramai Ampang : ['0000000003131017900']
Combine person and ppvs chromosome = ['373101713131017903', '356101703131017903', '289100973131017903']
Distance from person to the chosen PPVs = [67, 49, 95, 32, 35, 49, 78, 17, 13, 32, 43, 1]
The farthest distance = 107
The nearest distance = 8
-----
| Generation | Chromosome | Distance | Accuracy |
-----
>Generation : 0, new best f(373101713131017903) = 67.000 , Accuracy = 41
>Generation : 0, new best f(356101703131017903) = 49.000 , Accuracy = 59
>Generation : 0, new best f(284101823131017903) = 32.000 , Accuracy = 76
>Generation : 0, new best f(300101713131017903) = 17.000 , Accuracy = 91
>Generation : 0, new best f(312101673131017903) = 13.000 , Accuracy = 95
>Generation : 0, new best f(300101893131017903) = 12.000 , Accuracy = 96
>Generation : 0, new best f(320101753131017903) = 9.000 , Accuracy = 99
>Generation : 0, new best f(320101773131017903) = 8.000 , Accuracy = 100
>Generation : 1, new best f(309101723131017703) = 7.000 , Accuracy = 100
>Generation : 2, new best f(298101762991017903) = 4.000 , Accuracy = 100
>Generation : 13, new best f(305101433031014403) = 2.000 , Accuracy = 100
>Generation : 18, new best f(356101663561016503) = 1.000 , Accuracy = 100
>Generation : 19, new best f(356101663561016603) = 0.000 , Accuracy = 100
Done!
f(356101663561016603) = 0.000000
```



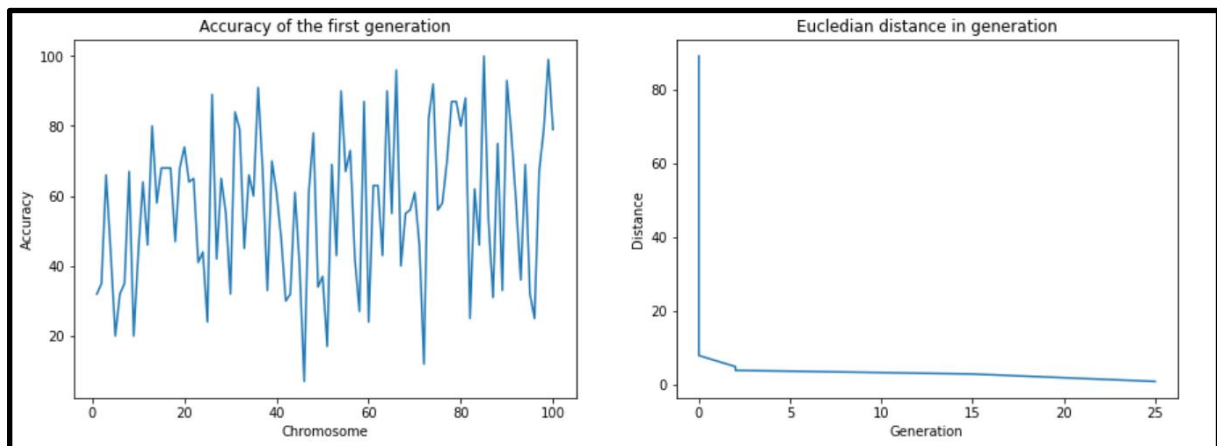
The graph on the right shows the accuracy of the first generation which included the number of populations/chromosomes (100). It will help us to identify which chromosomes have the highest accuracy based on the fitness (euclidean distance). The graph on the left shows the euclidean distance through generation. The nearer the distance, the higher the fitness score. Thus, using the graph, we can determine the fitness score (based on distance) in each generation. The generation is solved in the 19th generation. This is because it has the highest mutation rate with 0.1, thus leading to a faster solution. The best solution is obtained through this parameter tuned.

II. General Parameter

Population Size: **100**

Mutation rate: **0.7**

```
Chromosome for each person = ['348101070000000003', '366101340000000003', '310101070000000003']
PPVs Chromosome for Dewan Orang Ramai Ampang : ['000000003131017900']
Combine person and ppvs chromosome = ['348101073131017903', '366101343131017903', '310101073131017903']
Distance from person to the choosen PPVs = [89, 77, 19, 82, 94, 93, 71, 41, 71, 89]
The farthest distance = 110
The nearest distance = 8
-----
| Generation |           Chromosome           | Distance | Accuracy |
-----
>Generation : 0, new best f(348101073131017903) = 89.000 , Accuracy = 19
>Generation : 0, new best f(366101343131017903) = 77.000 , Accuracy = 31
>Generation : 0, new best f(310101073131017903) = 19.000 , Accuracy = 89
>Generation : 0, new best f(306101743131017903) = 10.000 , Accuracy = 98
>Generation : 0, new best f(310101863131017903) = 8.000 , Accuracy = 100
>Generation : 2, new best f(300101832991017903) = 5.000 , Accuracy = 99
>Generation : 2, new best f(297101822991017903) = 4.000 , Accuracy = 100
>Generation : 15, new best f(315101403131013803) = 3.000 , Accuracy = 100
>Generation : 20, new best f(353101663541016803) = 2.000 , Accuracy = 100
>Generation : 25, new best f(283101622821016203) = 1.000 , Accuracy = 100
Done!
f(283101622821016203) = 1.000000
```



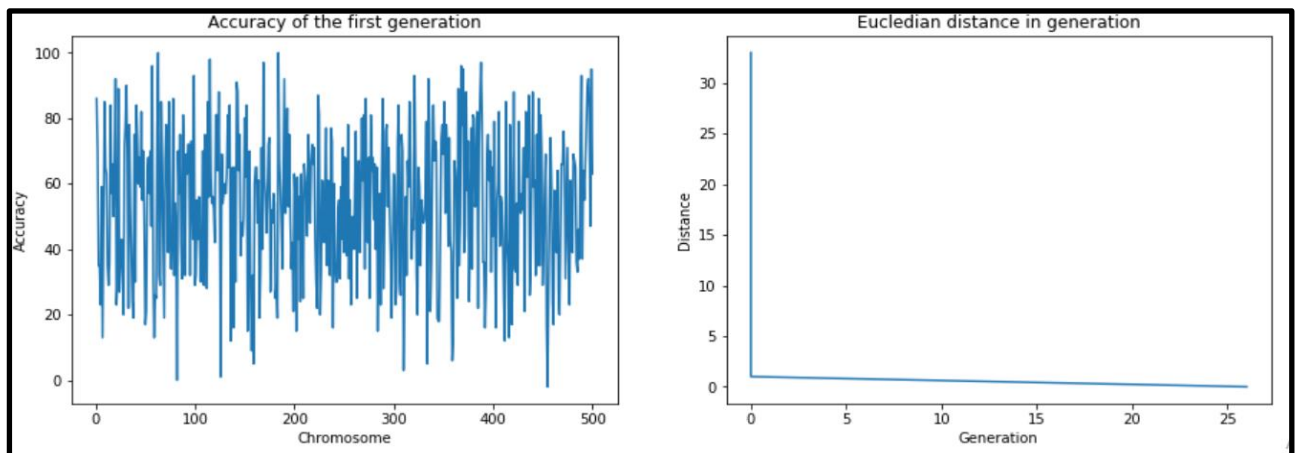
Mutation of 0.7 and population 100 with constant crossover rate and no iteration is set as the parameter of this graph. The graph is solved at the 25th generation. This shows that the lesser the mutation rate the slower the problem being solved.

III. General Parameter

Population Size: **500**

Mutation rate: **1.0**

```
Chromosome for each person = ['29510155000000003', '29910187000000003', '334
PPVs Chromosome for Dewan Orang Ramai Ampang : ['00000003131017900']
Combine person and ppvs chromosome = ['295101553131017903', '29910187313101790
Distance from person to the choosen PPVs = [33, 18, 84, 55, 96, 64, 103, 59, 2
The farthest distance = 116
The nearest distance = 1
-----
| Generation |           Chromosome           | Distance | Accuracy |
-----
>Generation : 0, new best f(295101553131017903) = 33.000 , Accuracy = 68
>Generation : 0, new best f(299101873131017903) = 18.000 , Accuracy = 83
>Generation : 0, new best f(325101883131017903) = 17.000 , Accuracy = 84
>Generation : 0, new best f(305101713131017903) = 13.000 , Accuracy = 88
>Generation : 0, new best f(315101683131017903) = 12.000 , Accuracy = 89
>Generation : 0, new best f(309101833131017903) = 6.000 , Accuracy = 95
>Generation : 0, new best f(316101783131017903) = 4.000 , Accuracy = 97
>Generation : 0, new best f(311101783131017903) = 2.000 , Accuracy = 99
>Generation : 0, new best f(313101783131017903) = 1.000 , Accuracy = 100
>Generation : 26, new best f(343101183431011803) = 0.000 , Accuracy = 100
Done!
f(343101183431011803) = 0.000000
```



The population increased into 500 with mutation rate 1.0. The problem is solved in the 26th generation. The hypothesis can be made that the more the population, the more process is needed to find the consistent solution.

IV. General Parameter

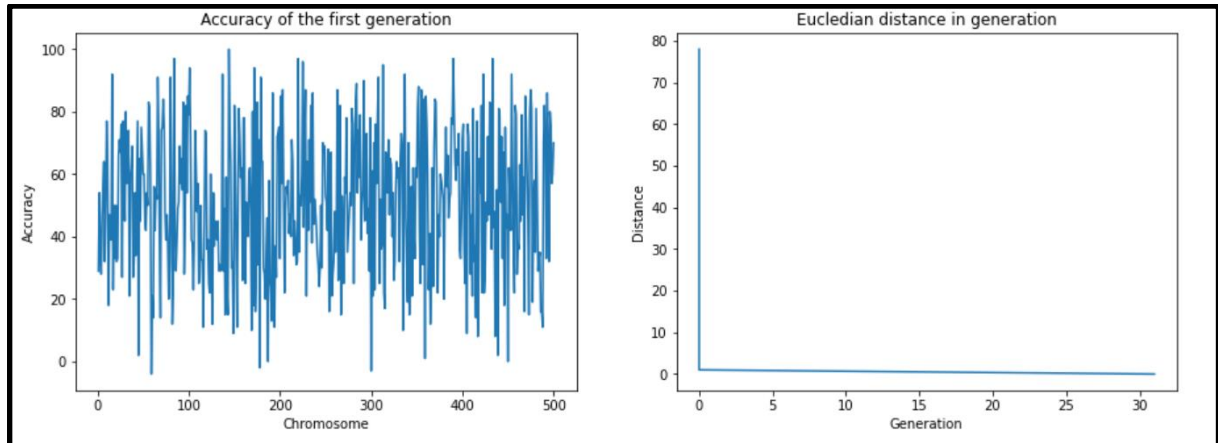
Population Size: **500**

Mutation rate: **0.7**

```
Chromosome for each person = ['369101370000000003', '298101180000000003', '369101370000000003']
PPVs Chromosome for Dewan Orang Ramai Ampang : ['000000003131017900']
Combine person and ppvs chromosome = ['369101373131017903', '298101183131017903', '369101370000000003']
Distance from person to the choosen PPVs = [78, 70, 66, 79, 61, 67, 31, 85, 61, 70]
The farthest distance = 117
The nearest distance = 1
```

Generation	Chromosome	Distance	Accuracy
>Generation : 0, new best f(369101373131017903)		= 78.000	, Accuracy = 23
>Generation : 0, new best f(298101183131017903)		= 70.000	, Accuracy = 31
>Generation : 0, new best f(368101563131017903)		= 66.000	, Accuracy = 35
>Generation : 0, new best f(358101483131017903)		= 61.000	, Accuracy = 40
>Generation : 0, new best f(340101713131017903)		= 31.000	, Accuracy = 70
>Generation : 0, new best f(300101633131017903)		= 23.000	, Accuracy = 78
>Generation : 0, new best f(332101743131017903)		= 22.000	, Accuracy = 79
>Generation : 0, new best f(299101823131017903)		= 16.000	, Accuracy = 85
>Generation : 0, new best f(300101813131017903)		= 15.000	, Accuracy = 86
>Generation : 0, new best f(311101853131017903)		= 7.000	, Accuracy = 94
>Generation : 0, new best f(313101803131017903)		= 1.000	, Accuracy = 100
>Generation : 31, new best f(376100993761009903)		= 0.000	, Accuracy = 100

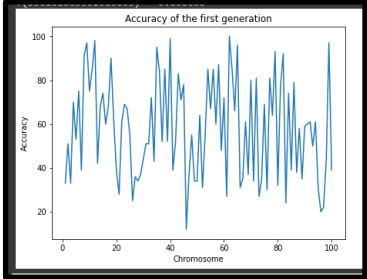
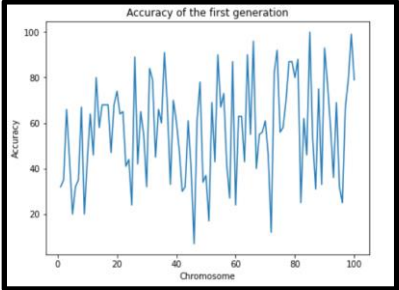
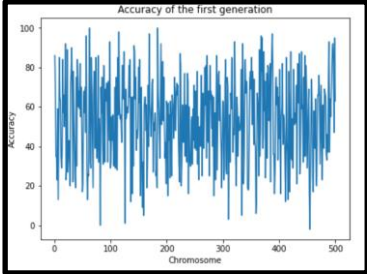
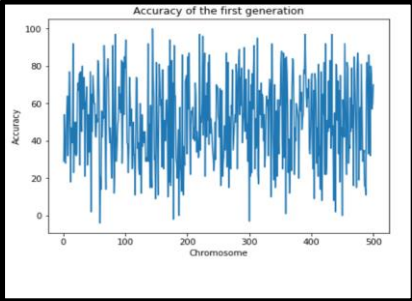
Done!
f(376100993761009903) = 0.000000



The population was set at 500 with 0.7 as the mutation rate. The generation solved at generation 31st. This is because it has more generation to be processed as more populations are initialized and the number of mutation rates is lower, thus it leads to slower mutation flow.

V. Comparison

Table 7.1 Performance Comparison

			
Parameter: Population Size: 100 Mutation rate: 1.0	Parameter: Population Size: 100 Mutation rate: 0.7	Parameter: Population Size: 500 Mutation rate: 1.0	Parameter: Population Size: 500 Mutation rate: 0.7
Constant Parameter: Crossover = 0.9 Iteration = 47			
Solved at 19th Generation	Solved at 25th Generation	Solved at 26th Generation	Solved at 31st Generation

Based on the findings above, when the population size increases, the number of generations to be solved also increases as it leads to more generations being processed. Furthermore, the higher the mutation rate, the lesser generation it takes to find the optimal value. The crossover rate remained constant with a high value of 0.9 to achieve higher quality of percentage in the results.

8.0 Conclusion

Genetic algorithm (GA) is the most common algorithm. It is faster than other algorithms and easier to implement. The only condition is that the vector representation of the individual (chromosome) has to be correct, then the solution can be gained. In this project, the chromosome is easily defined and the performance of the algorithm is easily executed as it is clear on how to play around with the chromosome. Deeper analysis and understanding need to be paid attention towards the chromosome as it is the main key that will walk through each phase of fitness, selection, crossover and mutation.

On the other hand, GA has also disadvantages that include that it does not find the optimum value as it uses the random heuristics. GA also is very sensitive in finding the right chromosome as it is very crucial.

All of the results obtained through this project with different parameter settings were observed and visualized. Different generations were produced depending on the adjustment of a few parameters such as number of population and mutation rate. It can also be concluded that by tuning the parameter, it can bring different results under the same objectives.

In conclusion, there are many things that can be learned through this project as well as widening our knowledge on Genetic Algorithms and healthcare domain. Many other methods can also be tested in this project such as, roulette wheel selection, single point crossover and many more. In addition, other algorithms also can be used to solve the same objective.

REFERENCES

1. Albadr, M. A., Tiun, S., Ayob, M., & AL-Dhief, F. (2020). Genetic Algorithm Based on Natural Selection Theory for Optimization Problems. *Symmetry*, 12(11), 1758. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/sym12111758>
2. Margaret L. Brandeau and Samuel S. Chiu, An Overview of Representative Problems in Location Research, *Management Science*, Vol. 35, No. 6 (Jun., 1989), pp. 645-674
3. *Progress Statistics For The National Covid-19 Immunisation Programme*. (n.d.). Jawatankuasa Khas Jaminan Akses Bekalan Vaksin Covid-19 (JKJAV). Retrieved June 13, 2021, from <https://www.vaksincovid.gov.my/en/statistics/P>
4. Rahman, S., & Smith, D. K. (1999). Deployment of rural health facilities in a developing country. *Journal of the Operational Research Society* (50), 892–902.
5. Shariff, S. S. R., Moin, N. H., & Omar, M. (2012). Location allocation modeling for healthcare facility planning in Malaysia. *Computers and Industrial Engineering*, 62(4), 1000–1010. <https://doi.org/10.1016/j.cie.2011.12.026>
6. Wang, L., Shi, H., & Gan, L. (2018). Healthcare facility location-allocation optimization for China's developing cities utilizing a multi-objective decision support approach. *Sustainability (Switzerland)*, 10(12). <https://doi.org/10.3390/su10124580>
7. Elkady, S., & Abdelsalam, H. (2016). *A modified multi-objective particle swarm optimisation algorithm for healthcare facility planning A modified multi-objective particle swarm optimisation algorithm for healthcare facility planning Sahar K . Elkady * and Hisham M . Abdelsalam. October.* <https://doi.org/10.1504/IJBSR.2016.073685>
8. Hussain, A., Muhammad, Y. S., Nauman Sajid, M., Hussain, I., Mohamd Shoukry, A., & Gani, S. (2017). Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator. *Computational Intelligence and Neuroscience*, 2017, 1–8. <https://doi.org/10.1155/2017/7430125>
9. Sinthamrongruk, T., Dahal, K., Satiya, O., Vudhironarit, T., & Yodmongkol, P. (2017). Healthcare Staff Routing Problem using adaptive Genetic Algorithms with Adaptive Local Search and Immigrant Scheme. 2nd Joint International Conference on Digital Arts, Media and Technology 2017: Digital Economy for Sustainable Growth, ICDAMT 2017, 120–125. <https://doi.org/10.1109/ICDAMT.2017.7904947>