MONASH University
Engineering

Assessment cover sheet

Unit and student details

Unit code	TRC 4901	**Unit title**	Artificial Intelligence		
If this is a group assignment, each student must include their name and ID number and sign the student statement.					
Student ID	29392004	**Surname**	Ranjan	**Given names**	Rivyesch
Student ID		**Surname**		**Given names**	
Student ID		**Surname**		**Given names**	
Student ID		**Surname**		**Given names**	

Assessment details

Title of assignment /lab	Lab 5 - Optimization using Genetic Algorithm	**Authorised group assignment**	Yes ☐ No ☐
Lecturer/tutor	Dr. Parasuraman	**Tutorial day and time**	Wednesday 8 am
Due date	02/06/2021	**Date submitted**	26/05/2021
Has any part of this assessment been previously submitted as part of another unit/course?			Yes ☐ No ☐

Submission date and extensions

All work must be submitted by the due date. If an extension of work is required, please complete and submit a Special Consideration application (in-semester assessment task) form to your examiner/lecturer/tutor.

Plagiarism and collusion

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations.

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student statement and signature

Student Statement

I have read the university's Student Academic Integrity Policy and Procedures.
I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://www.monash.edu/legal/legislation/current-statute-regulations-and-related-resolutions
I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
No part of this assignment has been previously submitted as part of another unit/course.
I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:

i. provide to another member of faculty and any external marker; and/or
ii. submit it to a text matching software; and/or
iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.

I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Student signature	Rivyesch Ranjan	**Date**	19/05/2021
Student signature		**Date**	
Student signature		**Date**	
Student signature		**Date**	

Please note that it is your responsibility to retain a copy of your assignment/laboratory work

Privacy Statement

The information on this form is collected for the primary purpose of assessing your assignment/laboratory work and ensuring the academic integrity requirements of the University are met. Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters, and statistical analyses. If you choose not to complete all the questions on this form Monash University it may not be possible for Monash University to assess your assignment/laboratory work. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. You have a right to access personal information that Monash University holds about you, subject to any exception in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au

V1.3 October 2016

Contents

1.0 Introduction

A travelling salesperson (TSP) needs to visit 50 different locations. The aim is that the TSP should not revisit the same location more than once and at the end should return to the starting location. Hence, this will give the shortest route to visit all 50 locations. In this lab a genetic algorithm technique is used to identify the route the TSP should take to visit all 50 different locations in the shortest possible route whilst meeting all the criteria outlined above.

2.0 Literature Review

Genetic algorithms mimic the natural behavior of evolution. It is initialized with a population of chromosomes which each consists of a constant population of genes. The population of chromosomes is done through random selection. A larger population is recommended for complex optimization problems [1]. These chromosomes are then evaluated by the fitness function to determine the best chromosomes in the population. The selected genes with the 'best fitness' are then used to produce the next generation of genes. This reproduction of genes is done using crossover and mutation operations which aim to help the best fitness chromosome to survive and mate from one generation to the next. Crossover is a process where two 'fit' chromosomes exchange a selected segment of their genetic material while mutation randomly manipulates random genes in the chromosomes [2]. By doing this the population of chromosomes evolves gradually over generations through competition [3].

3.0 Methodology

3.1 Setting up Dataset and GA Parameters

For this experiment, the dataset named 'pr144' is used. The dataset consisting of the 50 different locations that the TSP needed to visit was imported into MATLAB from a comma-separated values (.csv) file. Once the data was loaded the parameters for the genetic algorithm were specified as shown below in *Table 1*.

Table 1: Baseline model's parameter settings

Parameter	Value Assigned
Mutation Probability (PM)	0.1
Crossover Probability (PC)	0.5
Number of Chromosomes (chrom)	500
Number of Genes (genes)	50

These parameters of the GA were set constant throughout the entire lab. The number of genes chosen corresponds to the number of cities the TSP has to make deliveries in. It is important to note that since the salesman needs to return back to the initial starting position at the end of the sales trip, the program takes this into account and the actual number of gene is equal to 51 (50+1). This ensured that the starting and ending point was the same. The rest of the parameters were set arbitrarily.

Besides that, *Figure 1* below illustrates all the possible routes the TSP could take to complete his or her deliveries in 50 different locations and then return back to the initial location. As can be seen there are numerous possible scenarios that could be chosen. Hence, the use of genetic algorithm to find the best possible solution is very useful as analytical calculations performed without artificial intelligence would be complex and time consuming. Since the dataset only consists of the coordinates of all the 50 cities to be visited, the distance between each city was calculated and used as one of the inputs to

the genetic algorithm so that the model was aware of the variable it was attempting to decrease. This is shown in *Figure 2* where the matrix consists of the distance between each city with the other 49 cities (only five cities shown).

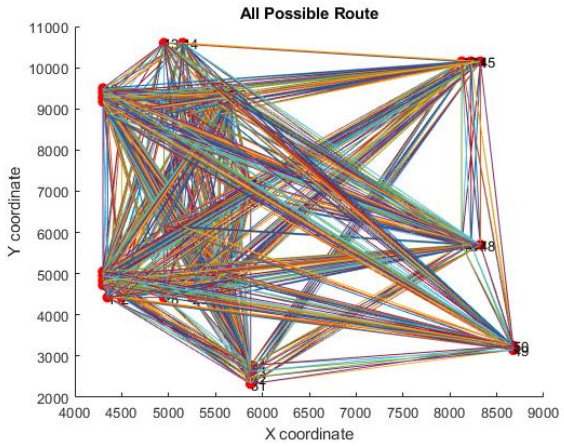


Figure 1: All possible routes that could be taken by TSP

	1	2	3	4	5
1	0	150	304.1381	403.1129	527.3756
2	150	0	360.5551	447.2136	561.8051
3	304.1381	360.5551	0	100	225
4	403.1129	447.2136	100	0	125
5	527.3756	561.8051	225	125	0

Figure 2: Distance between the each of the first five cities

3.2 Fitness Function Design

Since one of the objectives was to choose the shortest route, the fitness of each route that is a possibility was calculated. Every route will have a different distance and hence a different fitness value. Based on the fitness function used in the code, the shortest route is the one with the highest fitness.

$$Fitness = \frac{1}{Total\ Distance}$$

3.3 Selection

Similar to the natural world on which GA is based on, the routes with higher fitness are most likely to survive. What this essentially means is that the ones with higher fitness values are chosen as the wanted chromosome to be used in the crossover. The selection method used in this lab is roulette wheel where each route's fitness is converted to relative fitness and then tabulated. It is then cumulated to obtain the range for each section in the wheel.

A random number is generated to initiate the selection process. The chromosome that belongs to the particular range of the roulette wheel of the generated number will then be chosen. The routes or chromosomes with higher fitness have a larger range of values in the roulette wheel, thus increasing the likelihood of them being picked.

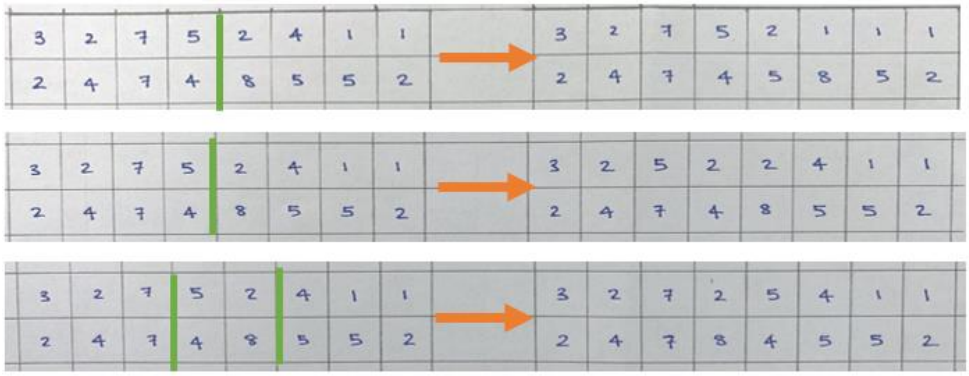
$$Relative\ Fitness_i = \frac{Fitness_i}{\sum Fitness} \times 100$$

3.4 Crossover Operator

Crossover operator is an important optimization technique that selects two chromosomes as parents to form offspring chromosomes. In this lab, three custom-built crossover operators are used. Its characteristic is that the parents swapped the sequence of their value instead of swapping their actual value. To ensure that the start and end location is the same the first and last value in both sequences remain unchanged. An example of the result of each of the cross over operation is shown on the same two 8-bit parent chromosomes below in Figure

The crossover1 and crossover2 operators take the two parent chromosome selected by the genetic algorithm and splits each parent into two sections. This essentially creates a left and right section. The point at which the split is made is done randomly so the two sections do not necessarily have to be split in half. For crossover1 the right section is selected for both the parents while for crossover2 the left section is selected for both the parents. The ascending order of the sequences are determined. The sequence of the arrays is the swapped and then rearranged based on the ascending order of the swapped sequence.

Crossover3 is similar to the other two crossovers, except that the parent chromosomes are split into three and the middle section of each split parent is selected for crossover. This is done by generating two random points within them.



3.5 Mutator Operator

Two mutator operator are also used in this lab as another form of optimization. This is rarely used in code. It represents a change in the gene of the chromosome that may or may not improve the system.

The first mutator mutation1 exchanges two random points of the offspring chromosome with each other. On the other hand, mutation2 creates a section in the offspring chromosome and inside this section the whole array is shifted to the right by one space. The values at the most right is moved to the front of the section.

4.0 Results

4.1 All Crossover and Mutator Combinations

Table 2: Various combinations of crossover and mutation type

Crossover	Mutation	Initial Distance	Best Distance	Best Fitness	Time taken
1	1	1.42E+5	4.19E+04	2.38E-05	28.6324

1	2	1.43E+5	5.08E+04	1.97E-05	28.0631
2	1	1.40E+5	5.02E+04	1.99E-05	24.887
2	2	1.30E+5	4.42E+04	2.26E-05	25.8964
3	1	1.32E+5	4.68E+04	2.14E-05	21.8655
3	2	1.40E+5	5.24E+04	1.91E-05	20.4107

4.2 Variations to PC and PM

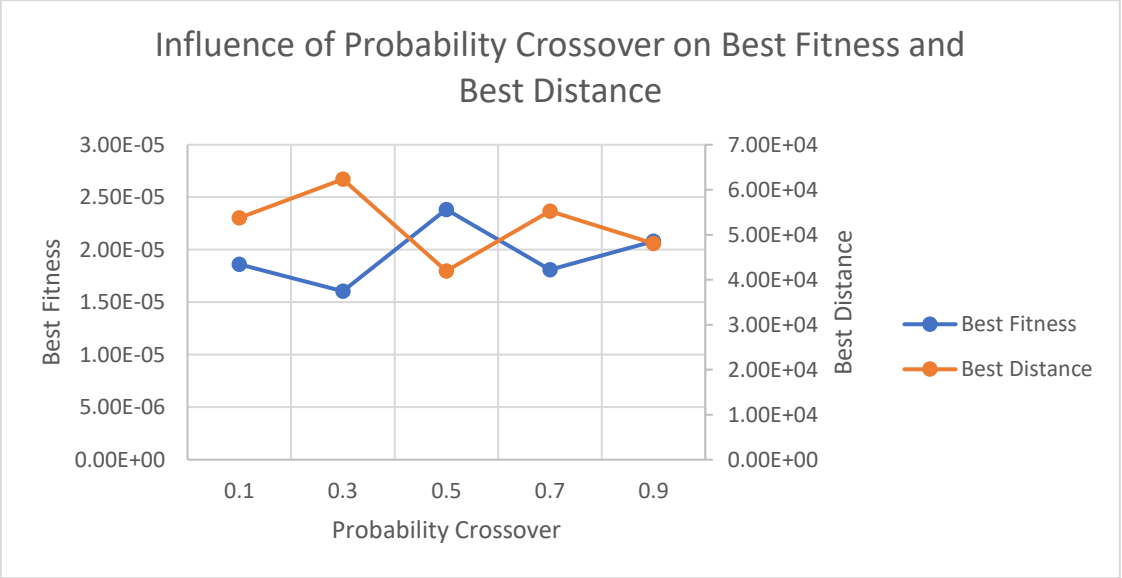


Figure 3: Study of Variations of Probability Crossover

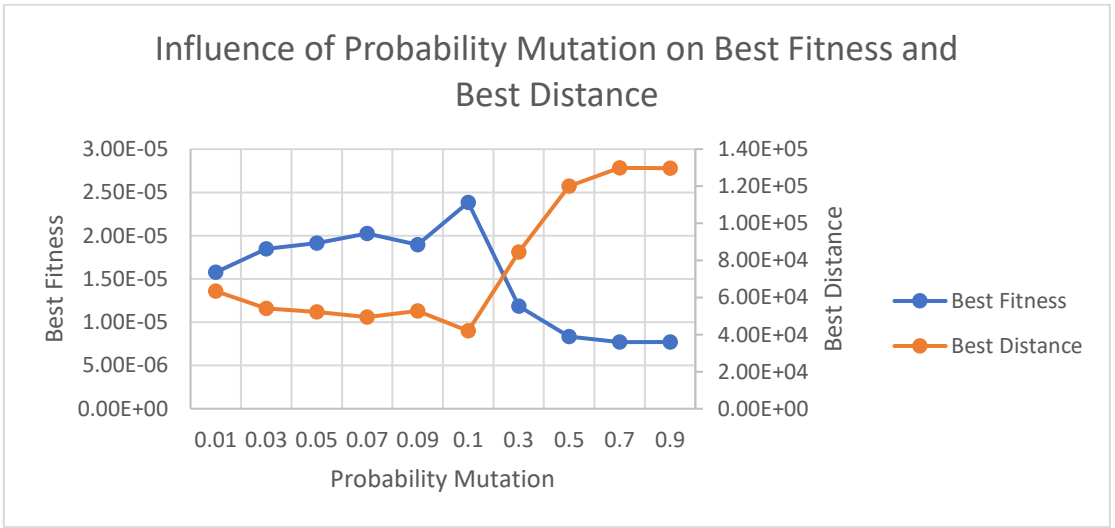


Figure 4: Study of Variations of Probability Mutation

5.0 Discussion

From the results shown in *Table 2* in Section 4.1, it is determined that crossover 1 and mutation 1 as a combination work the best. The higher fitness values were obtained when using these crossover and

mutation types respectively, thus highlighting their superior performance in finding the shortest possible route for the TSP to travel. It also means that for this particular experiment with the data being used, performing the crossover for the right section of the two parent chromosomes yields the best results. Furthermore, since mutation 1 is a part of the combinations that gives the highest fitness value and lowest best distance, this indicates that exchanging two random points belonging to the offspring chromosome is the best method to mutate the chromosome. Other methods of crossover and mutation have been proven in *Table 2* to be less effective and hence for the next two studies will be ignored. Therefore, for the experiments in Section 4.2 the crossover and mutation type will be crossover 1 and mutation 1, and these will be kept fixed.

Section 4.2 depicts two distinct figures that graphically represents the results. The tabulated and more detailed version of the results can be found in Appendix 7.1. Figure 1 illustrates the effect of changing the probability crossover on the best fitness and best distance. The graph first and foremost indicates the relationship between best fitness and best distance. As expected, based on the fitness function used, a smaller best distance value would give a larger best fitness value, and vice versa. The best situation is where the best distance is the smallest it could be, and the best fitness is the highest possible value. This scenario would satisfy the aim of the problem statement such that the TSP's travel route would be the most efficient route in terms of distance travelled. From figure 1 a probability crossover of 0.5 gives the best fitness value, indicating that the amount of crossover being done by the genetic algorithm is neither too much nor too less. The graph infers that either a smaller or larger used probability crossover would deteriorate the fitness value, resulting in a much higher distance route travelled by the TSP.

A larger range of mutation probability is investigated. The reason is that genetic algorithm similar like nature does not typically involve large amounts of mutations. However, for this study the range is extended such that very small values right up to large values of mutation are tested out using the genetic algorithm. It is common practice for the probability mutation to be much lower than the probability crossover as like nature crossover occurs way more frequently than mutation ever does. The best fitness is much better when the probability mutation is at 0.1 or under. For probability mutation higher than that the best fitness worsens significantly. For large probability mutations as the best fitness value decreases, the best distance route that the TSP will travel will be increasingly further. This finding reiterates the theoretical recommendation that the probability mutation should typically be a small value and definitely smaller than the probability crossover used. The best probability mutation that should be used is 0.1. At this value, the best fitness is by far the best which means the TSP will have to travel the shortest route distance wise to cover all 50 locations before returning back to the initial starting location.

5.0 Conclusion

The genetic algorithm was successful in identifying the shortest and most efficient route the TSP should take to make all 50 deliveries without passing any city more than once and also finally returning back to the initial location. It showed that the algorithm is capable of working within constraints defined at the beginning. A few different combinations of mutators and crossovers are used, and it is found that some combinations work better than others. The study on PM and PC shows that low PM values are ideal whereas PC should neither be too high nor too low. Overall, it is proven that choosing the right PM and PC goes a long way in achieving an optimized model which ultimately gives more accurate results for the problem defined.

6.0 References

[1] M. Seyedmahmoudian et al., "State of the art artificial intelligence-based MPPT techniques for mitigating partial shading effects on PV systems – A review", 2021.

[2] A. Youssef, M. El-Telbany and A. Zekry, "The role of artificial intelligence in photo-voltaic systems design and control: A review", *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 72-79, 2017. Available: 10.1016/j.rser.2017.04.046.

[3] A. Mellit and S. Kalogirou, "Artificial intelligence techniques for photovoltaic applications: A review", *Progress in Energy and Combustion Science*, vol. 34, no. 5, pp. 574-632, 2008. Available: 10.1016/j.pecs.2008.01.001.

7.0 Appendix

7.1 Results

Table 3: Variation of crossover probability for crossover 1 and mutation 1 with $P_M = 0.1$

Crossover Probability (P_C)	Best Distance	Best Fitness	Time taken
0.1	5.38E+04	1.86E-05	14.7808
0.3	6.23E+04	1.60E-05	21.2303
0.5	4.19E+04	2.38E-05	28.6324
0.7	5.52E+04	1.81E-05	33.3368
0.9	4.81E+04	2.08E-05	38.7069

Table 4: Variation of mutation probability for crossover 1 and mutation 1 with $P_C = 0.5$

Mutation Probability (P_M)	Best Distance	Best Fitness	Time taken
0.01	6.34E+04	1.58E-05	25.3691
0.03	5.41E+04	1.85E-05	26.5473
0.05	5.23E+04	1.91E-05	26.8197
0.07	4.94E+04	2.02E-05	25.5229
0.09	5.28E+04	1.89E-05	26.5048
0.10	4.19E+04	2.38E-05	28.6324
0.30	8.45E+04	1.18E-05	27.4753
0.50	1.20E+05	8.33E-06	27.6431
0.70	1.30E+05	7.70E-06	28.273
0.90	1.30E+05	7.71E-06	29.0701

7.2 Crossover and Mutation type combinations

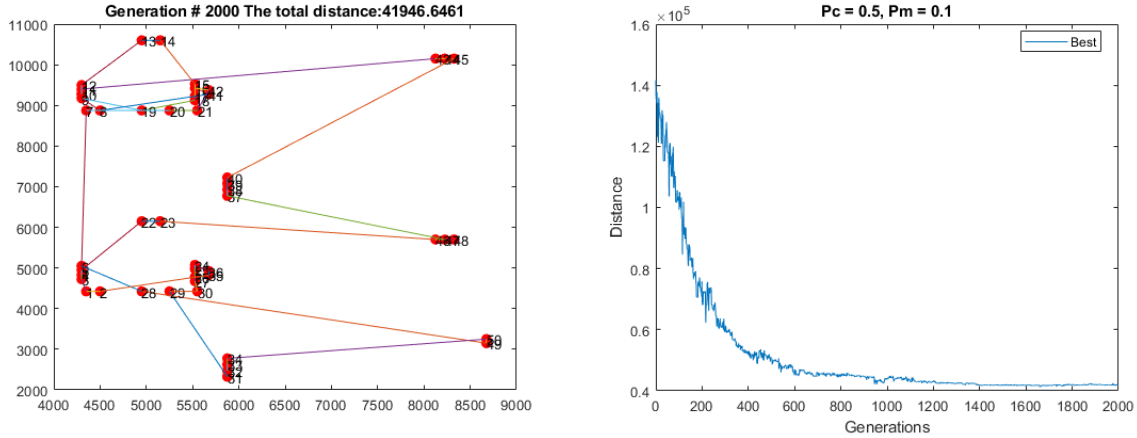


Figure 5: Results for Crossover 1 and Mutation 1 for $P_C = 0.5$ and $P_M = 0.1$

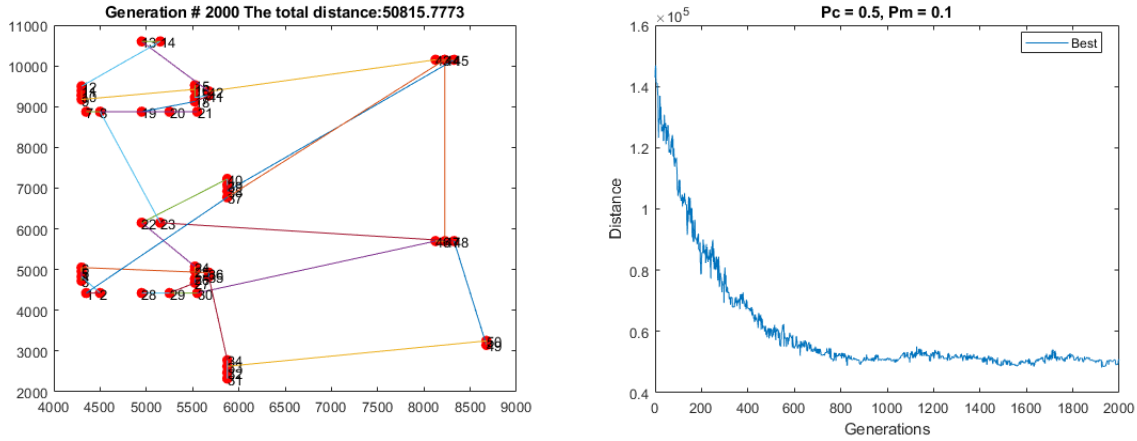


Figure 6: Results for Crossover 1 and Mutation 2 for $P_C = 0.5$ and $P_M = 0.1$

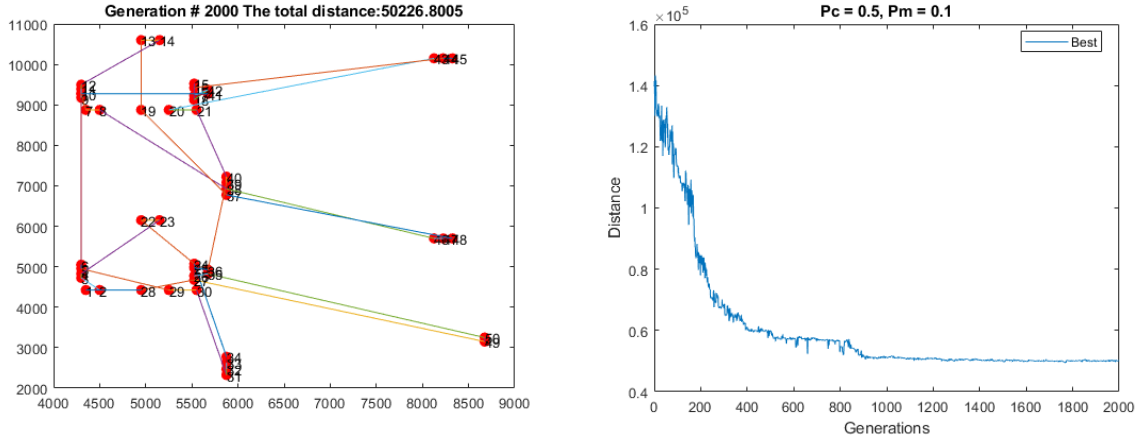


Figure 7: Results for Crossover 2 and Mutation 1 for $P_C = 0.5$ and $P_M = 0.1$

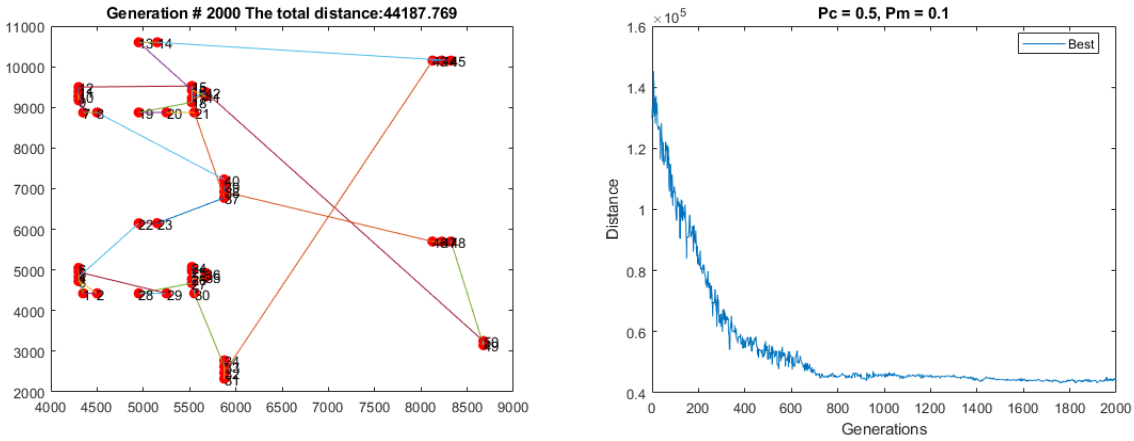


Figure 8: Results for Crossover 2 and Mutation 2 for $P_C = 0.5$ and $P_M = 0.1$

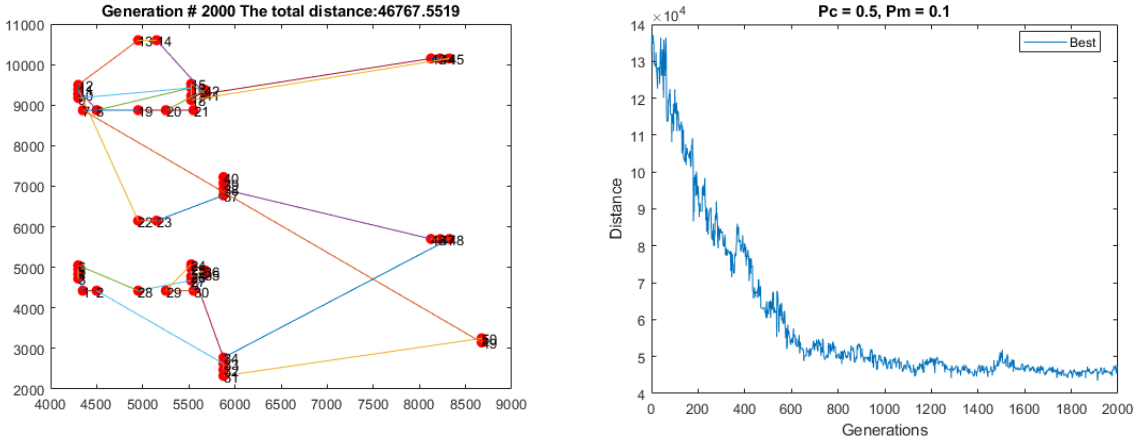


Figure 9: Results for Crossover 3 and Mutation 1 for $P_C = 0.5$ and $P_M = 0.1$

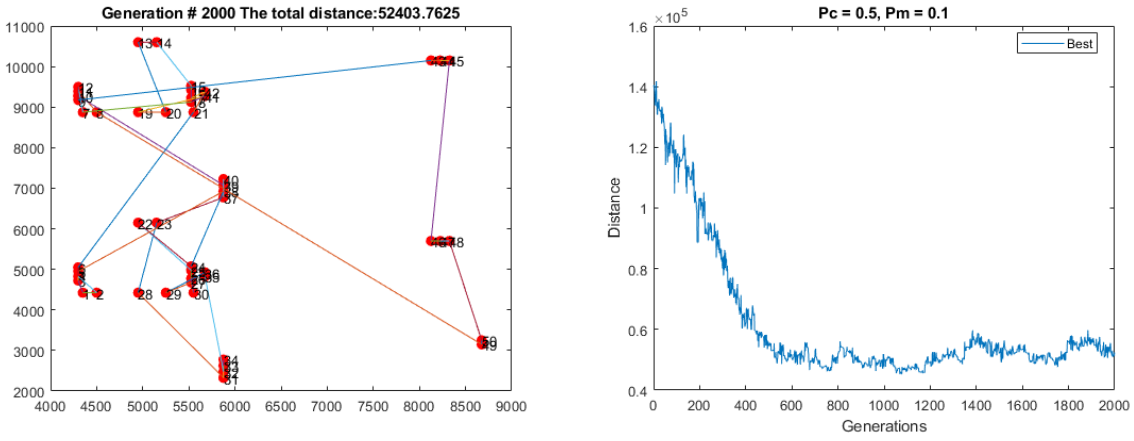


Figure 10: Results for Crossover 3 and Mutation 2 for $P_C = 0.5$ and $P_M = 0.1$

7.3 Variations of Crossover Probability

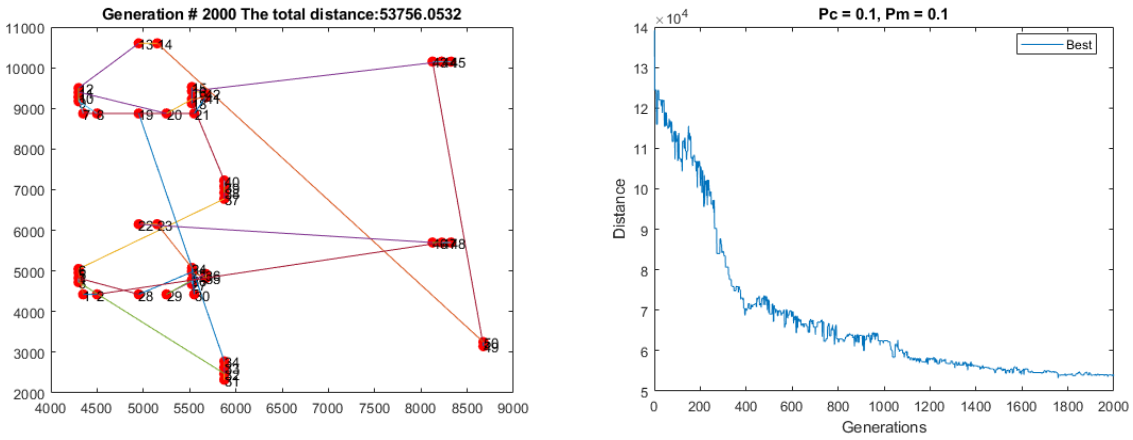


Figure 11: Results for Crossover 1 and Mutation 1 for $P_C = 0.1$ and $P_M = 0.1$

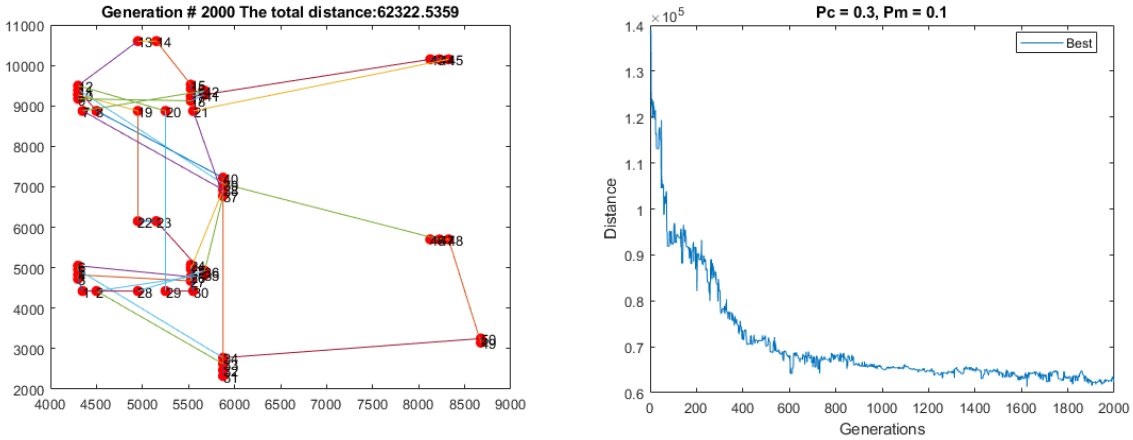


Figure 12: Results for Crossover 1 and Mutation 1 for $P_C = 0.3$ and $P_M = 0.1$

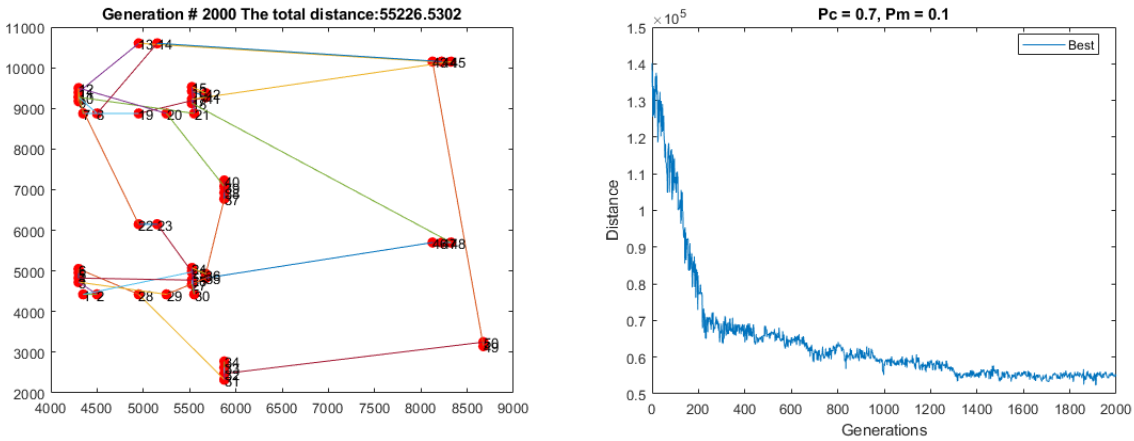


Figure 13: Results for Crossover 1 and Mutation 1 for $P_C = 0.7$ and $P_M = 0.1$

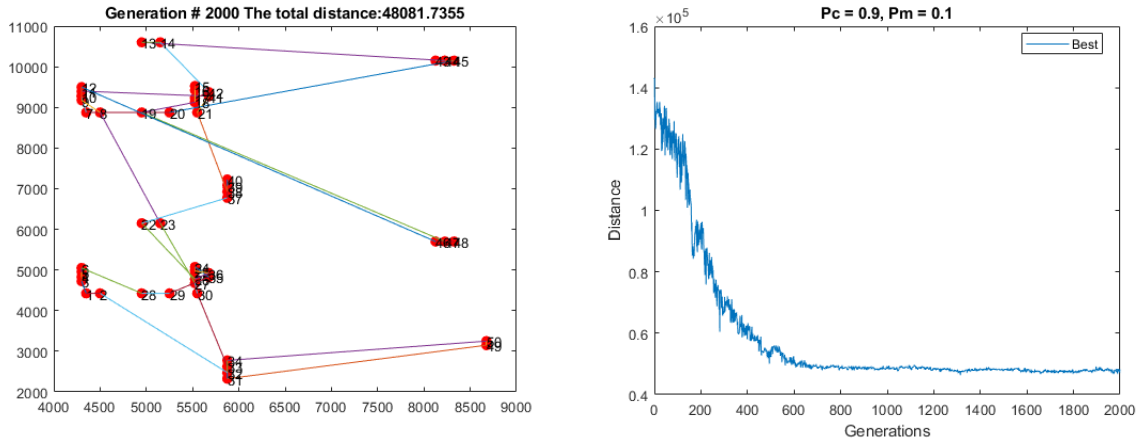


Figure 14: Results for Crossover 1 and Mutation 1 for $P_C = 0.9$ and $P_M = 0.1$

7.4 Variations of Mutation Probability

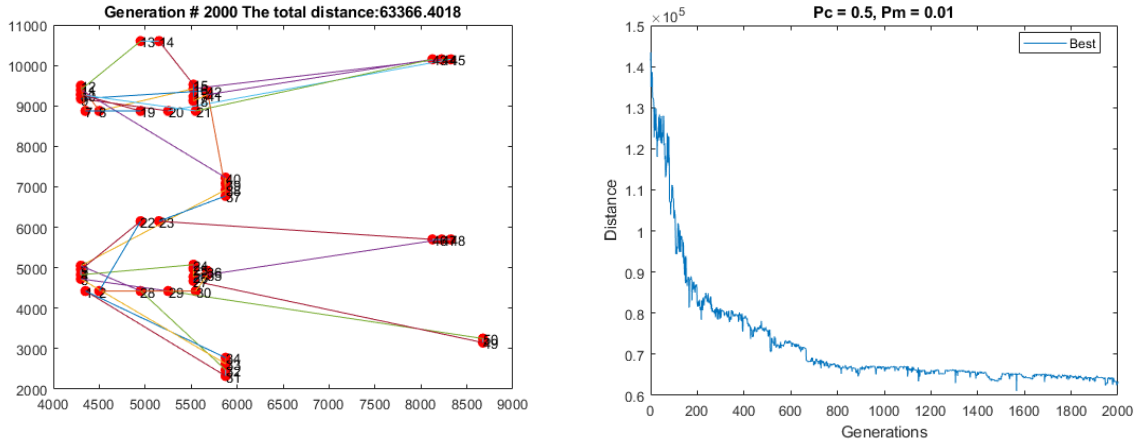


Figure 15: Results for Crossover 1 and Mutation 1 for $P_C = 0.5$ and $P_M = 0.01$

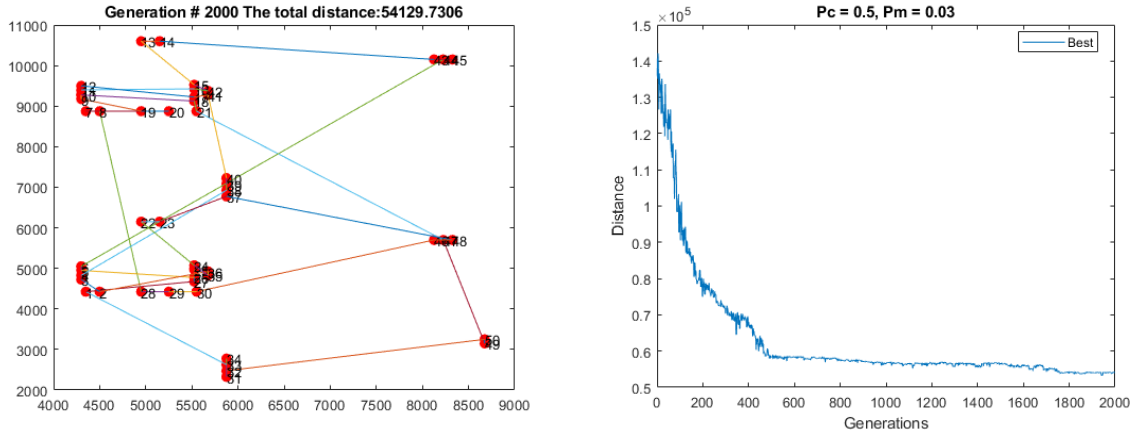


Figure 16: Results for Crossover 1 and Mutation 1 for $P_C = 0.5$ and $P_M = 0.03$

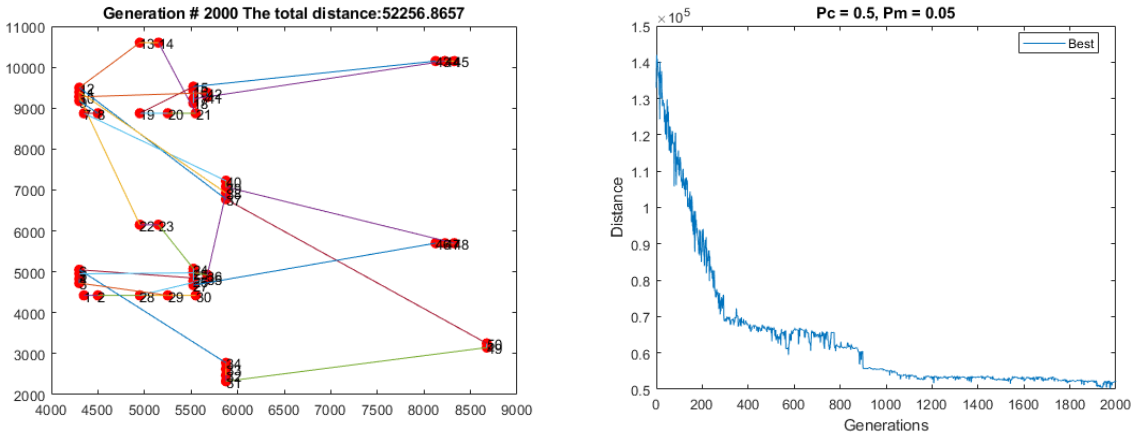


Figure 17: Results for Crossover 1 and Mutation 1 for $P_C = 0.5$ and $P_M = 0.05$

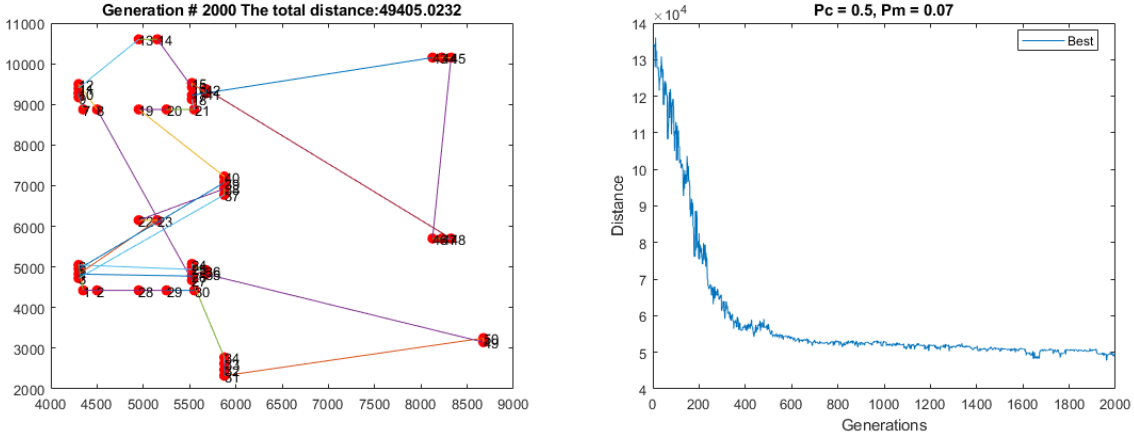


Figure 18: Results for Crossover 1 and Mutation 1 for $P_C = 0.5$ and $P_M = 0.07$

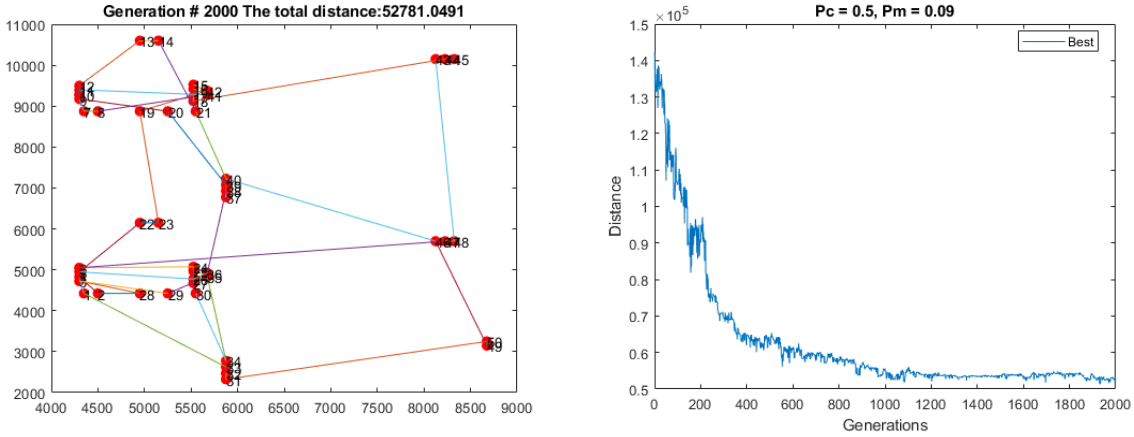


Figure 19: Results for Crossover 1 and Mutation 1 for $P_C = 0.5$ and $P_M = 0.09$