1 Problem 2.1

The length of the shortest path from AS program is 98.55011. The path is shown in Figure 1.

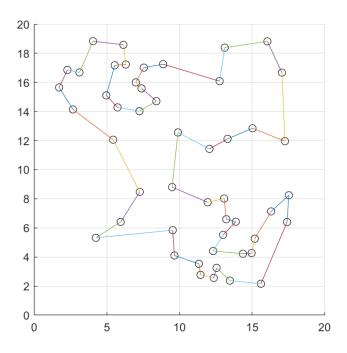


Figure 1: The shortest path.

2 Problem 2.2

The contour plots are shown in Figure 2 on the following page. In this figure, it is clear that there are four blue areas where the function values are much lower than the yellow areas, which means this function has 4 local minima over the range $(x_1, x_2) \in [-5, 5]$.

x_1	x_2	$f(x_1, x_2)$
3.00000	2.00000	0.00000
-2.80512	3.13131	0.00000
-3.77931	-3.28319	0.00000
3.58443	-1.84813	0.00000

Table 1: Locations and function values of local minima

Table 1 lists the locations of these four minima and their function values obtained through the PSO program.

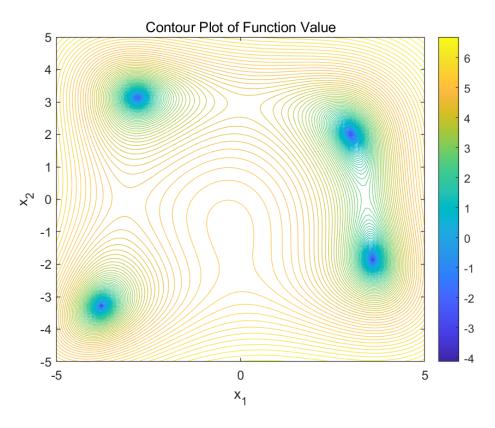


Figure 2: The contour plot.

3 Problem 2.3

The fitness is measured by $F_i = average(v_i) \times d_i$, v_i is the speed over the evaluation and d_i is the distance travelled before termination.

Figure 3 below shows the fitness values of the training slopes and validation slopes. The best chromosome is chosen from the highest points which the difference between the two fitness values is not greater than 1000. This is to avoid overfitting.

And the fitness value in the Figure 3 is the average value of 10 training slopes and 5 validation slopes, so it is not too large.

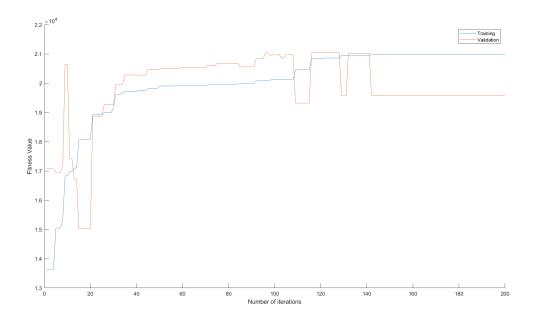


Figure 3: The fitness values of the training and validaation.

4 Problem 2.4

1) Describe the program: Compared with other GAs, this program needs to take into account the significance of each bit of the instructions in the initialize chromosome and decode chromosome functions, so I use the switch case statement to deal with some problems. In the process of crossover, since it is a two-point crossover, it uses array splicing. Also because of the two-point crossover and the 4 bits of instructions, the minimum length of the chromosome is set to 12 bits, mainly because two-point crossover is more difficult when the chromosome is shorter.

For how to reduce the likelihood of the algorithm falling into a local optimum at an early stage, first the mutation rate of the chromosome is related to the length of the chromosome. What I set directly in the setup is the number of genes mutated per chromosome, i.e. mutation-Num.Secondly there are also a couple of determinations set up, which are: 1. When 1000 iterations do not produce a better genome, the mutation rate is multiplied by a factor greater than 1 at each subsequent iteration until a set maximum is reached. 2. With error between 0.55 and 0.35, mutationNum = 10/error to prevent falling into a local optimum. 3. Retain half of the previous generation's chromosomes at a time.

2) Parameter: 10 Variable Registers, 8 Constant Registers. Constants

are [0.1, -1, -4, 3, 15, 0.7, 1, 7].

3) Results: The error at last is 0.04752. The fitness value is 21.04163. And the function g(x) which I found is as below. The figure shows the data points and the best fit.

$$g(x) = \frac{-55575x^3 - 90405x^2 + 2488906x + 1920800}{280(6175x^2 + 10045x + 6566)}$$

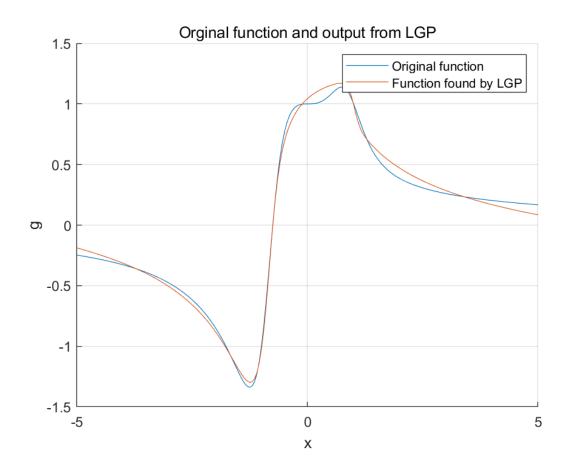


Figure 4: Data points and the best fit.