# **ECE 457a Assignment 3 – Group 1**

# **Question 1**

## Part (a)

Hooke’s law in physics states that the **length** of a uniform spring is a linear function of the **force** applied to it. The function can be represented as shown below where is a constant and represents stiffness of the spring.

We would like to find an optimal roots (solution) to the above polynomial. This is a real-valued problem and the solution can be represented by a vector of parameters where and are real values. Thus the solution can be represent as:

## Part (b)

Crossover operators for real numbers can be:

1. Single Arithmetic
2. Simple Arithmetic
3. Whole Arithmetic

The most commonly selected one is whole arithmetic, so we will use this operator. The operator is of the following form:

Where α is an arbitrary floating point number between 0 and 1 (exclusive). For this problem, we will use a value of α = 0.2.

## Part (c)

For real-valued problems a mutation scheme is to add random noise, for each gene:

Where N(0,σ) is a random Gaussian number with zero mean and standard deviation σ.

## Part (d)

Objective/fitness function can be least squares of errors. Given that exist on the polynomial, a potential solution vector can be evaluated with the following objective function

Where is the value calculated from the corresponding using the equation .

## Part (e)

**Iteration 1**

The parents are

Using and from

If the fitness function is applied for , we get the following results.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Square of Errors** |
| 0 | 6.1 | -68.9 | 4747.21 |
| 2 | 7.6 | -86.1 | 7761.61 |
| 4 | 8.7 | -98.7 | 10547.29 |
| 6 | 10.4 | -118.2 | 15425.64 |

Using and from

If the fitness function is applied for , we get the following results.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Square of Errors** |
| 0 | 6.1 | -33.6 | 1128.96 |
| 2 | 7.6 | -42.3 | 1962.49 |
| 4 | 8.7 | -48.7 | 2777.29 |
| 6 | 10.4 | -58.6 | 4173.16 |

Using α = 0.2 in the crossover operator the children are:

Using and from

If the fitness function is applied for , we get the following results.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Square of Errors** |
| 0 | 6.1 | -40.9 | 1672.81 |
| 2 | 7.6 | -51.4 | 2851.56 |
| 4 | 8.7 | -59.1 | 3981.61 |
| 6 | 10.4 | -70.9 | 5913.61 |

Using and from

If the fitness function is applied for , we get the following results.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Square of Errors** |
| 0 | 6.1 | -61.9 | 3831.61 |
| 2 | 7.6 | -77.4 | 6304.36 |
| 4 | 8.7 | -88.8 | 8611.84 |
| 6 | 10.4 | -106.4 | 12633.76 |

**Iteration 2**

From the calculations we see that and have the smallest values for the fitness function. Thus, the parent solution for the next generation will be from Iteration 1 and . Hence:

The second generation parents are

Using α = 0.2 in the crossover operator the children are:

Using and from

If the fitness function is applied for , we get the following results.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Square of Errors** |
| 0 | 6.1 | -35.3 | 1246.09 |
| 2 | 7.6 | -44.4 | 2152.96 |
| 4 | 8.7 | -51.1 | 3036.01 |
| 6 | 10.4 | -61.4 | 4542.76 |

Using and from

If the fitness function is applied for , we get the following results.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Square of Errors** |
| 0 | 6.1 | -39.5 | 1560.25 |
| 2 | 7.6 | -49.7 | 2672.89 |
| 4 | 8.7 | -57.1 | 3733.21 |
| 6 | 10.4 | -68.6 | 5565.16 |

From the calculations we see that and have the smallest values for the fitness function. Thus, the parent solution for the next generation will be from Iteration 2 and .

# Question 2

1. a) The length of the binary encoding chromosome for a variable xi with a range, Li and a precision mi is given by

ni = ⌈log2(Li · 10mi )⌉

This problem specifies two variables, x and y, each with a range of [0, 10), so Li = 10 and precision mi = 4. Therefore, the length of chromosomes should be

nx + ny = 2·⌈log2(10·104)⌉= 34

Four individuals of this length are

A 0011111111111111101110000111100001

B 0101000101101110010000110110111101

C 1000110111011101101011100111101111

D 1001001000100010000000000000000000

b) In terms of the crossover operator, we can use 1-point crossover over two parents to produce two children. A and B crossover to produce A1 and B1. Similarly, C and D crossover to produce C1 and D1. The point of cross over is after 10 bits. After 10 bits, the tails of the parents get exchanged to produce the two new children.

Following are the resultant chromosomes after crossover

A1 0011111111 101110010000110110111101

B1 0101000101 111111101110000111100001

C1 1000110111 100010000000000000000000

D1 1001001000 011101101011100111101111

c) The mutation operator takes in 1 chromosome and produces 1 child. Assume a mutation threshold r in the range [0,1]. For every bit in the chromosome, generate a random probability, p, in the range [0, 1]. If p > r, then flip the bit. Here, chromosome A2 is a mutation of chromosome A, B2 of B and so on.

A2 000111111110111110011100000101000010

B2 010100000110101001100001101011111010

C2 100110111101110111010111001011011110

D2 100100100011001001000100000000110001

2d) **Repeat a, b, c for Decimal Real Number Floating Point Encoding**

a) The length of the chromosome is the sum of the number of bits needed to represent the integer portion and the number of bits needed to represent the decimal portion. For a real number with integer part range of αi and decimal part range of βi is calculated as

ni = ⌊log2(αi) + 1⌋ + ⌊log2(βi) + 1⌋

In the case described by this question, we have two variables, x and y, both with αi = 10 and βi = 9999. Therefore the number of bits required to represent our chromosome is

nx +ny =2·⌊log2(10)+1⌋+⌊log2(9999)+1⌋=36

Four individuals of this length are

A 0011 11111111111111 0111 00001111000011

B 0101 00010110111001 1000 01101101111011

C 1000 11011101110111 0101 11001111011111

D 1001 00100010001001 0000 00000000000001

b) In terms of the crossover operator, we can use 1-point crossover over two parents to produce two children. A and B crossover to produce A1 and B1. Similarly, C and D crossover to produce C1 and D1. The point of cross over is after 18 bits. After 18 bits, the tails of the parents get exchanged to produce the two new children.

A1 0011 11111111111111 1000 01101101111011

B1 0101 00010110111001 0111 00001111000011

C1 1000 00100010001001 0000 00000000000001

D1 1001 11011101110111 0101 11001111011111

c) The mutation operator takes in 1 chromosome and produces 1 child. Assume a mutation threshold r in the range [0,1]. For every bit in the chromosome, generate a random probability, p, in the range [0, 1]. If p > r, then flip the bit. Here, chromosome A2 is a mutation of chromosome A, B2 of B and so on.

A2 0001 11111110111110 0111 00000101000010

B2 0101 00000110101001 1000 01101011111010

C2 1001 10111101110111 0101 11001011011110

D2 1001 00100011001001 0001 00000000110001