



Assignment 4  
Computational Intelligence  
Dr. S. Hajipour

Mohammad Hossein Shafizadegan  
99104781

December 17, 2023

## Contents

1	Question 1	2
2	Question 2	3
3	Question 3	3
4	Question 4	4
5	Question 5	5
6	Question 6	5
7	Question 7	5
8	Question 8	6
9	Question 9	6

## 1 Question 1

**a**

$$\begin{aligned}
 x_1 = 65413532 &\Rightarrow f(x_1) = (6+5) - (4+1) + (3+5) - (3+2) = 9 \\
 x_2 = 87126601 &\Rightarrow f(x_2) = (8+7) - (1+2) + (6+6) - (0+1) = 23 \\
 x_3 = 23921285 &\Rightarrow f(x_3) = (2+3) - (9+2) + (1+2) - (8+5) = -16 \\
 x_4 = 41852094 &\Rightarrow f(x_4) = (4+1) - (8+5) + (2+0) - (9+4) = -19
 \end{aligned}$$

$$f(x_4) < f(x_3) < f(x_1) < f(x_2)$$

**b**

$$\begin{aligned}
 (x_2, x_4) &\rightarrow f(x_2) > f(x_4) \Rightarrow \text{winner: } x_2 \\
 (x_1, x_3) &\rightarrow f(x_1) > f(x_3) \Rightarrow \text{winner: } x_1 \\
 (x_1, x_4) &\rightarrow f(x_1) > f(x_4) \Rightarrow \text{winner: } x_1 \\
 (x_3, x_4) &\rightarrow f(x_3) > f(x_4) \Rightarrow \text{winner: } x_3
 \end{aligned}$$

**c**

$$\text{Two point crossover for } x_2, x_1 \rightarrow \begin{cases} 8712 | 660 | 1 \\ 6541 | 353 | 2 \end{cases} \Rightarrow \begin{cases} 8712 | 353 | 1 \\ 6541 | 660 | 2 \end{cases} \Rightarrow \begin{cases} 87123531 & f = 16 \\ 65416602 & f = 16 \end{cases}$$

$$\text{Two point crossover for } x_1, x_3 \rightarrow \begin{cases} 65 | 4135 | 32 \\ 23 | 9212 | 85 \end{cases} \Rightarrow \begin{cases} 65 | 9212 | 32 \\ 23 | 4135 | 85 \end{cases} \Rightarrow \begin{cases} 65921232 & f = -2 \\ 23413585 & f = -5 \end{cases}$$

**d**

Old Generation

$$average = \frac{1}{4} (9 + 23 - 16 - 19) = -\frac{3}{4}, \quad max = 23$$

New Generation

$$average = \frac{1}{4} (16 + 16 - 2 - 5) = \frac{25}{4}, \quad max = 16$$

The average value of fitness has increased but the maximum value of fitness has increased in the new generation in comparison to the old one. Having a greater average value means we are searching better zones but the bad news is that the maximum fitness is not located in this zone.

**e**

$$\text{optimum genotype: } 99009900, \quad f = 36$$

**f**

Remember that the optimum genotype consists of 9 and 0. In the given initial population some chromosomes like  $x_1$  and  $x_2$  which have relatively high fitness values don't contain 9. Therefore, crossover for these chromosomes will not lead to the optimal answer.

**g**

In the following, "+" means mutation will be performed and "-" is the opposite.

$$\begin{bmatrix} - & + & - & - & - & - & - & + \\ - & - & + & - & - & - & - & - \\ - & - & - & - & + & + & - & - \\ - & - & - & - & - & + & - & - \end{bmatrix} \Rightarrow \begin{cases} 89123530 \\ 65016602 \\ 65929932 \\ 23413985 \end{cases}$$

## 2 Question 2

**a**

The parameter  $\mathbf{T}$  in this algorithm determines the probability of accepting worse solutions. It plays a crucial role in controlling the search behavior and exploration-exploitation trade-off during the optimization process.

At higher temperatures, the algorithm allows more exploration by accepting solutions that are worse than the current solution. In such way, algorithm can escape local optima.

For lower temperatures, algorithm shifts towards exploitation, focusing more on improving the current solution by accepting only better solutions or solutions that are slightly worse.

**b**

$$\text{if } T \rightarrow 0 \Rightarrow e^{\frac{-\Delta f}{kT}} \rightarrow 0 \Rightarrow \text{Hill Climbing algorithm}$$

**c**

$$\text{if } T \rightarrow \infty \Rightarrow e^{\frac{-\Delta f}{kT}} \rightarrow 1 \Rightarrow \text{Random searching algorithm}$$

## 3 Question 3

**a**

This approach at the first step of ranked-based selection can be reasonable, since we have no prior knowledge. It provides an initial exploration of the search space and allows for diverse solutions to be sampled.

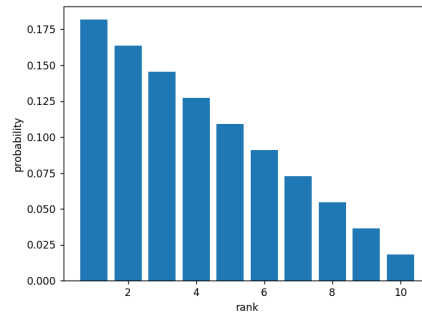
**b**

Remember that in ranked-based selection, we choose first ranks with more probability and the ones with lower rank will be chosen less often with lower probability. We should provide a method to implement this approach.

one commonly used method is the Fitness Proportionate Selection, also known as Roulette Wheel Selection. This method assigns selection probabilities to individuals based on their fitness values relative to the entire population. Here is a step-by-step algorithm for Fitness Proportionate Selection:

1. Calculate the total fitness of the entire population.
2. Calculate the selection probability (P) for each individual by dividing its fitness value by the total fitness

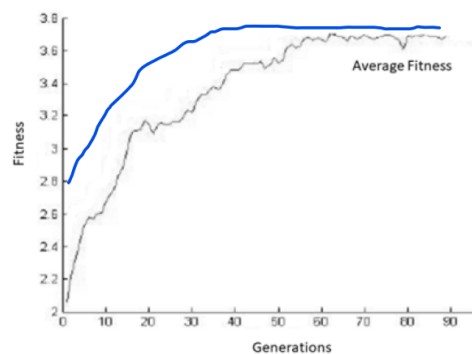
The resulting values represent the probability distribution for each rank. Since in ranked-based algorithm we sort ranks decreasingly the probability distribution will finally be something like this:



## 4 Question 4

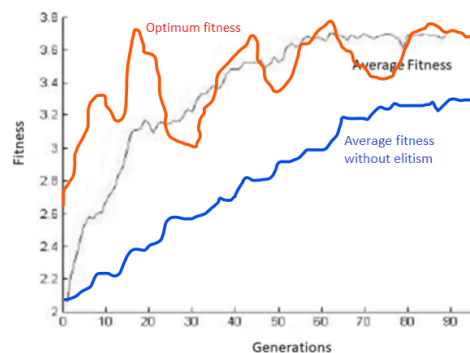
a

The trend for optimum fitness in each generation will be non-decreasing. This is because elitism ensures that the best individuals from each generation are carried over to the next, so the optimum fitness should either improve or remain constant in subsequent generations. It is also expected that the curve of optimum fitness in each generation is above the provided curve.



b

There is no guarantee that the best individuals from each generation are preserved in the next generation. This means that the optimum fitness may decrease or fluctuate over time, depending on the selection, crossover, and mutation operators. The average fitness may also converge slower or get stuck in local optima, as the population may lose diversity and beneficial genes.



## 5 Question 5

If the fitness values have a small range and are close together, choose a smaller value for  $\theta_i$  to reduce selection pressure and prevent excessive bias.

If the fitness values have a large range, choose a larger value for  $\theta_i$  to increase selection pressure and focus more on exploitation.

$$\begin{aligned} f_5 : \quad range &= 7.9 - 1.8 = 6.1 & \Rightarrow & \theta_5 = 3 \\ f_{20} : \quad range &= 9.8 - 7.8 = 2 & \Rightarrow & \theta_{20} = 1 \\ f_{40} : \quad range &= 9.9 - 9.0 = 0.9 & \Rightarrow & \theta_{40} = 0.3 \end{aligned}$$

## 6 Question 6

**b**

For this problem, a suitable crossover operator would be the **One-Point Crossover**. Here we can see an example of this crossover for our problem:

$$\begin{cases} 111 + 1 | 1 + +1111 \\ 11 + 11 | 1 + 1 + 11 + 11 \end{cases} \Rightarrow \begin{cases} 11 + 11 | 1 + +1111 \\ 111 + 1 | 1 + 1 + 11 + 11 \end{cases}$$

In general case, n-point crossover will work well for this problem.

**c**

The suitable mutation operation for this problem is swap operation. We can randomly choose two genes and perform the swap operation. Here is an solid example:

$$111 + 11 + +11111 \Rightarrow 111 + 11 + 111 + 11$$

Shift, arbitrary permutation and inversion can be suitable for this problem too.

## 7 Question 7

### Encryption

We designate an array with the length of  $N \times M$  as the chromosome. The first N genes are associated to the first worker, the second N genes are associated to the second worker and so on. The alleles of genes are float numbers ranging from 0 to 1, indicating the percentage of the job each worker is assigned to.

### Corrigendum

As a correction phenomena that must be considered in this problem is to note that the summation of the alleles for each worker must be 1. Therefore, a normalization approach must be considered in the process of mutation and crossover.

### Mutation

Since our alleles are float numbers ranging from 0 to 1, we will utilize the gaussian mutation. As of the correction term we should divide the alleles of the corresponding worker by summation of the alleles after the mutation process.

## Crossover

As of the crossover we can utilize the n-point crossover. We should remember to perform the correction procedure too

## Initial population

Using a random process we generate numbers between 0 to 1 as allele for each gene of the chromosome. Don't forget the normalization correction process.

## Fitness function

The following formula is considered for the fitness function:

$$-K \left( \sum_{i=1}^M w_i v_i > V_{max} \right) - \max_i w_i T_i$$

## 8 Question 8

### Encryption

We designate an array with the length of  $3N$  as the chromosome. The first  $N$  genes are associated to the first (red) color, the second  $N$  genes are associated to the green color and the third  $N$  genes encode yellow colors. The alleles of genes are binary digits where 0 denotes not having that color and 1 denotes having that color

### Corrigendum

Along the chromosome, there are three genes for each cell correspondingly as of the three colors. We should note that for these three cells, there must be only one gene with value of 1 (one hot coding). So for the correction process we should check and correct the these specific genes to ensure this one hot encoding.

### Mutation

The bit mutation is highly recommended for this problem. Pair swap, perhaps is supposed to be an appropriate approach too. We shouldn't forget the correction process.

### Crossover

n-point crossover can be used for this problem. We should be aware of the correction process too.

### Initial population

Using a random process we generate numbers between 0 to 1 as allele for each gene of the chromosome. Don't forget the correction process.

## 9 Question 9

### Encryption

We designate an array with the length of  $N$  as the chromosome. each gene corresponds to a request. The alleles of the genes are numbers demonstrating the number of the server responsible for the task.

## Crossover

We can utilize the n-point crossover e.g. one point crossover.

## Mutation

Using standard mutation we can randomly change the allele of a gene with a number ranging from 1 to N. Pair swap can be also beneficial as it will ensure that no new server will be involved in the chromosome.

## Fitness function

The following formula is considered for the fitness function:

$$-\left(\sum_{i=1}^N ismemeber(i, A)\right) - B \left( \max_i \sum_{j=find(A==i)} w_j > M \right)$$