Q7

Understand the schemata theorem of GA (relevance of selection/crossover/mutation with the 3 components of the schema theorem. Apply the theorem to the optimization function f(x)= x3-2x2+x. Check whether the empirical results obtained earlier in assignment-I matches with the theorem.

Introduction:

Using the established methods and genetic operators of genetic algorithms, the schema theorem states that short, low-order schemata with above-average fitness increase exponentially in successive generations. Expressed as an equation:

Here {\displaystyle m(H,t)}is the number of strings belonging to schema {\displaystyle H}H at generation {\displaystyle t}t, {\displaystyle f(H)} is the *observed* average fitness of schema {\displaystyle H}H and {\displaystyle a\_{t}} is the *observed* average fitness at generation {\displaystyle t}t. The probability of disruption {\displaystyle p}p is the probability that crossover or mutation will destroy the schema {\displaystyle H}H. It can be expressed as:

{\displaystyle p={\delta (H) \over l-1}p\_{c}+o(H)p\_{m}}where {\displaystyle o(H)} is the order of the schema, {\displaystyle l}l is the length of the code, {\displaystyle p\_{m}} is the probability of mutation and {\displaystyle p\_{c}} is the probability of crossover. So, a schema with a shorter defining length {\displaystyle \delta (H)} is less likely to be disrupted.

Implications of Schema Theorem:

Schemas, like families, need nourishment and encouragement and careful protective management

1. The more bits in your building block family the more likely one is to go off the rails (causing great frustration and heartache).
2. Genes living far apart are prone to breaking up.
3. Conducting a constructive relationship at a distance is hard.
4. Best results achieved by the family unit huddled together in consecutive positions.

Applications of Schema Theorem:

The schema theorem is more applicable at the early stages of a search rather than at the end. Schema theorem indicates that fitter than average schemas are rewarded. The fitter the schema the more it is rewarded.

1. Reward is immediate: you see it in the next generation
2. Should alert us to one danger immediately
3. Premature converge of the population

Trace:

The optimization function is,

The constraint on x is

Choose Encoding,

As x max is 31, we use 5-bit binary representation of x

Generate Schema,

The randomly generated schemas are:

[’01\*0\*’, ’000\*0’, ’0\*1\*\*’, ’\*0\*00’, ’10\*11’, ’01\*\*1’]

Calculate the average fitness of each schema,

The average fitness of a schema is the average of all the instances of that schema. In our case, the fitness of each schema is:

[1073.0, 2147.0, 1680.25, 6074.5, 20793.0, 12018.5]

Select schema for crossover,

Schemas are selected in Russian roulette fashion with probabilities of getting selected being defined as:

In our case, the probability of each schema getting selected are:

[0.025, 0.049, 0.038, 0.139, 0.475, 0.274]

And the schema chosen for crossover are:

[’10\*11’ ’000\*0’ ’10\*11’ ’10\*11’ ’10\*11’ ’10\*11’]

Crossover,

Randomly pair up the schemas selected for crossover: In our case:

10\*11 and 10\*11

000\*0 and 10\*11

10\*11 and 10\*11

We are using single point crossover and the offspring generated are:

[’01\*0\*’, ’00\*11’, ’10\*11’, ’100\*0’, ’10\*11’, ’01\*\*1’]

Mutation,

For mutation, we define a probability of mutation, pm and we go through each gene of the chromosome and mutate them.

In our case, pm = 0.02.

For schema bit with ’\*’, mutation makes no difference, so we may not see any mutation.

After mutation, our schemas are:

[’01\*0\*’, ’00\*11’, ’10\*11’, ’100\*0’, ’10\*11’, ’01\*\*1’]

Repeat,

Repeat Selection, Crossover and Mutation on the schemas until there is no improvement in the average fitness of the generation.

In our case, that is:

[’11\*\*1’, ’01\*11’, ’10\*\*1’, ’10\*11’, ’00\*11’, ’01\*\*1’]

Perform GA on the schema population,

Now that we have the best schemas, create a population with all the instances of all the schemas in the final generation and apply normal ga to this population.

In our case, after applying ga to the best schema instance population the optimal solution is: ’11111’