CISC/CMPE 452/COGS 400

Assignment 4

Theoretical + Programming Parts

1. **Selection (3 points)** - Consider a population consisting of five individuals with the fitness values (before ranking) $F_1 = 5, F_2 = 7, F_3 = 8, F_4 = 10$ and $F_5 = 15$. Compute the probability that individual 4 will be selected (in a single selection step) with

   (a) roulette-wheel selection,

   (b) tournament selection, with tournament size equal to 2, and the probability of selecting the best individual (in a given tournament) equal to 0.75,

   (c) roulette-wheel selection, based on linearly ranked fitness values, where the lowest fitness value is set to 1 and the highest fitness value set to 10

2. **Schema Theory (3 points)** -

   (a) Given a schema $S = 0**1*1**0***$, what is the order of S and what is the defining length of H?

   (b) The fitness of bit string a with $l = 4$ is defined as the integer value represented by the binary number a (e.g., $f(0011) = 3$). What is the average fitness of the schema 1*** under f? What is the average fitness of the schema 0*** under f?

3. **Mutation (3 points)** - Consider a chromosome with m binary-valued genes and assume that it is to be mutated using a mutation rate p. Determine the probability that the chromosome will undergo

   (a) No mutation,

   (b) Exactly one mutation,

   (c) Less than three mutations.

4. **GA- Programming (4 points)** - Implement a simple GA with fitness-proportional selection, roulette-wheel sampling, population size 100, single-point crossover rate $p_c = 0.7$, and bitwise mutation rate $p_m = 0.001$. Try it on the following fitness function: $f(x) =$ number of ones in x, where x is a binary chromosome of length 20. Perform 20 runs, and measure the average generation at which the string of all ones is discovered. Perform the same experiment with crossover turned off (i.e. $p_c = 0$). Do similar experiments, varying the mutation and crossover rates, to see how the variations affect the average time required for the GA to find the optimal string. If it turns out that mutation with crossover is better than mutation alone, why is this the case?

   You may use the posted notebook on onQ as a template for your code

1. **Selection (3 points)** - Consider a population consisting of five individuals with the fitness values (before ranking) $F_1 = 5, F_2 = 7, F_3 = 8, F_4 = 10$ and $F_5 = 15$. Compute the probability that individual 4 will be selected (in a single selection step) with

   (a) roulette-wheel selection,

   (b) tournament selection, with tournament size equal to 2, and the probability of selecting the best individual (in a given tournament) equal to 0.75,

   (c) roulette-wheel selection, based on linearly ranked fitness values, where the lowest fitness value is set to 1 and the highest fitness value set to 10

(a) Total fitness: $\sum_{i=1}^{5} F_i = 45$

$\phi_1 = \frac{1}{9}, \phi_2 = \frac{4}{15} \quad \phi_3 = \frac{4}{9}, \phi_4 = \frac{2}{3}, \phi_5 = 1$

$P(F_4) = \phi_4 - \phi_3 = \frac{2}{9} \approx 0.\overline{22} = 22\%$

(b)

$P(F_4) = \frac{1}{5} \cdot \frac{3}{5} \cdot 0.75 + \frac{1}{5} \cdot \frac{1}{5} \cdot 0.25$

$= 1/10 = 0.1 = 10\%$

(c) $F_1 = 1 \quad F_2 = 3.25 \quad F_3 = 5.5 \quad F_4 = 7.45 \quad F_5 = 10$

Total sum: 27.2

$\phi_3 = \frac{5.5 + 3.25 + 1}{27.2} \approx 0.358$

$\phi_4 = \frac{43}{68} = 0.6323$

$P(F_4) = \phi_4 - \phi_3$

≈ 0.274

□

2. **Schema Theory (3 points)** -

(a) Given a schema $S = 0 * *1 * 1 * *0 * **$, what is the order of S and what is the defining length of H?

(b) The fitness of bit string a with $l = 4$ is defined as the integer value represented by the binary number a (e.g., $f(0011) = 3$). What is the average fitness of the schema 1*** under f? What is the average fitness of the schema 0*** under f?

(a) The order of S is 4 (i.e. $o(S)=4$)

The defining length of H is 8 $(9-1=8)$

b)

$$F^1_{avg} = \frac{f(1000)+f(1001)+f(1010)+f(1011)+f(1100)+\cdots+f(1111)}{8}$$

$$= \frac{8+9+\cdots+15}{8} = \frac{23}{2} = 11.5$$

$$F^2_{avg} = \frac{0+1+2+\cdots+7}{8} = \frac{7}{2} = 3.5$$

3. **Mutation (3 points)** - Consider a chromosome with m binary-valued genes and assume that it is to be mutated using a mutation rate p. Determine the probability that the chromosome will undergo

   (a) No mutation,

   (b) Exactly one mutation,

   (c) Less than three mutations.

Let $X \sim \text{Binomial}(m, p)$ where X is number of mutations in m genes

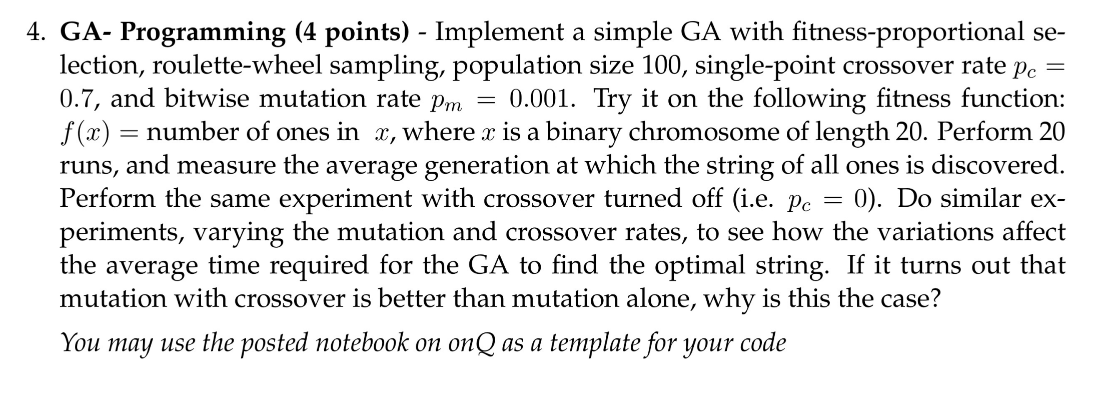
(a) $P(\text{No mutation}) = (1-p)^m$

b) $P(\text{One mutation}) = p(1) = \binom{m}{1} p (1-p)^{m-1}$

(c) $P(\text{Less than three mutations}) = P(X < 3)$

$$= \sum_{k=0}^{3} p(k)$$

$$= \sum_{k=0}^{3} \binom{m}{k} p^k (1-p)^{m-k}$$



4. **GA- Programming (4 points)** - Implement a simple GA with fitness-proportional selection, roulette-wheel sampling, population size 100, single-point crossover rate $p_c = 0.7$, and bitwise mutation rate $p_m = 0.001$. Try it on the following fitness function: $f(x) =$ number of ones in x, where x is a binary chromosome of length 20. Perform 20 runs, and measure the average generation at which the string of all ones is discovered. Perform the same experiment with crossover turned off (i.e. $p_c = 0$). Do similar experiments, varying the mutation and crossover rates, to see how the variations affect the average time required for the GA to find the optimal string. If it turns out that mutation with crossover is better than mutation alone, why is this the case?

You may use the posted notebook on onQ as a template for your code

I found that when running with crossover probability equal to zero the code ran for a very long time indicating it takes much longer without crossover than with crossover. This makes sense as our population will not vary much with no crossover so it would take longer on average to find a chromasome with optimal fitness.