Exercises Ch3

Exercise 3.1

The problem can be modeled in a graphical way as followed:

A person has to reach from one node of a graph to another, and the current distance from the goal is not known.

Problem statement: Find a path between two nodes of a graph where the edge weights or any other kind of heuristic is unknown.

Operators: Operators in this is driving from one crossing/landmark to the next. In graphical sense, this may be said to be traversing an edge between two nodes.

Solution space: A solution in this problem can be termed as a path from his initial point to the airport. Graphically, this means all possible paths from the starting node to the goal node, irrespective of the distance.

Goal state: The goal state is the state upon reaching which, the algorithm may stop and report success. Here, the airport is the goal state.

The person has never been to the airport before, so he doesn't know how close a location is from the airport. In other words, there are no heuristic values to help the man. Thus, the man should opt for a blind search strategy (such as BFS or DFS).

Exercise 3.4

- **Define the search space:** Let S be the set of all 4x4 grids of letters, and let L be the set of all letters (in this case, $L = \{a, b, c, \dots, z\}$).
- **Define the fitness function:** Let $f : S \rightarrow R$ be the fitness function that assigns a real number to each grid in S, where higher values indicate better word squares. Let $w_{,} w_{,} \dots, w$ be the eight valid words that must appear in the word square. Then, we can define f as follows:

$$f(G) = \sum_i w_i$$

$$w_i = \begin{cases} 1, if\ w\ in\ row\ or\ column\ of\ G \\ 0, otherwise \end{cases}$$

- **Initialize the population:** Let P be the initial population of N randomly generated grids in S.
- **Evolution:** Repeat the following steps until a termination criterion is met:
    1. **Evaluate fitness:** Compute the fitness score for each grid in P using the fitness function f.
    2. **Select parents:** Choose two parents from P with probability proportional to their fitness scores.
    3. **Perform crossover:** Use a random crossover point c to create two children by combining the first part of one parent's grid with the second part of the other parent's grid. For example, if parent 1 has the grid G1 = [a, b, c, d; e, f, g, h; i, j, k, l; m, n, o, p] and parent 2 has the grid G2 = [q, r, s, t; u, v, w, x; y, z, a', b'; c', d', e', f'], and the crossover point c = 2, then the children would be G1' = [a, b, s, t; e, f, w, x; i, j, a', b'; m, n, e', f'] and G2' = [q, r, c, d; u, v, g, h; y, z, k, l; c', d', o, p].
    4. **Perform mutation:** With probability p_m, randomly change a letter in each child's grid to a new letter.
    5. **Add children to population:** Add the two children to P.
    6. **Remove least fit individuals:** Sort the individuals in P by fitness score in descending order, and keep the top N individuals.
- **Termination:** Stop when a termination criterion is met, such as reaching a maximum number of generations or finding a perfect word square (i.e., a grid with all 8 words).

Exercise 3.7

Genetic Algorithms (GA) use principles of natural evolution. There are five important features of GA:

- Fitness Function represents the main requirements of the desired solution of a problem (i.e. cheapest price, shortest route, most compact arrangement, etc). This function calculates and returns the fitness of an individual solution.
- Encoding possible solutions of a problem are considered as individuals in a population. If the solutions can be divided into a series of small steps (building blocks), then these steps are represented by genes and a series of genes (a chromosome) will encode the whole solution. This way different solutions of a problem are represented in GA as chromosomes of individuals.
- Selection operator defines the way individuals in the current population are selected for reproduction. There are many strategies for that (e.g. roulette{wheel, ranked, tournament selection, etc), but usually the individuals which are more fit are selected.
- Crossover operator defines how chromosomes of parents are mixed in order to obtain genetic codes of their offspring (e.g. one{point, two{point, uniform crossover, etc). This operator implements the inheritance property (offspring inherit genes of their parents).
- Mutation operator creates random changes in genetic codes of the offspring. This operator is needed to bring some random diversity into the genetic code. In some cases GA cannot find the optimal solution without mutation operator (local maximum problem).

Exercise 3.8

a) Evaluate the fitness of each individual, showing all your workings, and arrange them in order with the fittest first and the least fit last.

Answer:
$f(x_1) = (6 + 5) - (4 + 1) + (3 + 5) - (3 + 2) = 9$
$f(x_2) = (8 + 7) - (1 + 2) + (6 + 6) - (0 + 1) = 23$
$f(x_3) = (2 + 3) - (9 + 2) + (1 + 2) - (8 + 5) = -16$
$f(x_4) = (4 + 1) - (8 + 5) + (2 + 0) - (9 + 4) = -19$

The order is x_2, x_1, x_3 and x_4.

b) Perform the following crossover operations:
i) Cross the fittest two individuals using one{point crossover at the middle point.
Answer: One{point crossover on x_2 and x_1:

x_2 = 8 7 1 2 6 6 0 1		O_1 = 8 7 1 2 3 5 3 2
	➔	
x_1 = 6 5 4 1 3 5 3 2		O_2 = 6 5 4 1 6 6 0 1

ii) Cross the second and third fittest individuals using a two{point crossover (points b and f).
Answer: Two{point crossover on x_1 and x_3

x_1 = 6 5 4 1 3 5 3 2		O_3 = 6 5 9 2 1 2 3 2
	➔	
x_3 = 2 3 9 2 1 2 8 5		O_4 = 2 3 4 1 3 5 8 5

iii) Cross the first and third fittest individuals (ranked 1st and 3rd)

using a uniform crossover.

Answer: In the simplest case uniform crossover means just a random exchange of genes between two parents. For example, we may swap genes at positions a, d and f of parents x_2 and x_3:

x_2 = 8 7 1 2 6 6 0 1		O_5 = 2 7 1 2 6 2 0 1
	➔	
x_3 = 2 3 9 2 1 2 8 5		O_6 = 8 3 9 2 1 6 8 5

c) Suppose the new population consists of the six offspring individuals received by the crossover operations in the above question. Evaluate the fitness of the new population, showing all your workings. Has the overall fitness improved?
Answer: The new population is:
O_1 = 8 7 1 2 3 5 3 2
O_2 = 6 5 4 1 6 6 0 1
O_3 = 6 5 9 2 1 2 3 2
O_4 = 2 3 4 1 3 5 8 5
O_5 = 2 7 1 2 6 2 0 1
O_6 = 8 3 9 2 1 6 8 5
Now apply the fitness function $f(x) = (a+b)-(c+d)+(e+f)-(g+h)$:
$f(O_1) = (8 + 7) - (1 + 2) + (3 + 5) - (3 + 2) = 15$

$f(O_2) = (6 + 5) - (4 + 1) + (6 + 6) - (0 + 1) = 17$
$f(O_3) = (6 + 5) - (9 + 2) + (1 + 2) - (3 + 2) = -2$
$f(O_4) = (2 + 3) - (4 + 1) + (3 + 5) - (8 + 5) = -5$
$f(O_5) = (2 + 7) - (1 + 2) + (6 + 2) - (0 + 1) = 13$
$f(O_6) = (8 + 3) - (9 + 2) + (1 + 6) - (8 + 5) = -6$
The overall fitness has improved.
d) By looking at the fitness function and considering that genes can only be digits between 0 and 9 find the chromosome representing the optimal solution (i.e. with the maximum fitness). Find the value of the maximum fitness.
Answer: The optimal solution should have a chromosome that gives the maximum of the fitness function

$$\max f(x) = \max [(a + b) - (c + d) + (e + f) - (g + h)]$$

Because genes can only be digits from 0 to 9, the optimal solution should be:
$x_{optimal} = 9\ 9\ 0\ 0\ 9\ 9\ 0\ 0$
and the maximum fitness is
$f(x_{optimal}) = (9 + 9) - (0 + 0) + (9 + 9) - (0 + 0) = 36$

e) By looking at the initial population of the algorithm can you say whether it will be able to reach the optimal solution without the mutation operator?
Answer: No, the algorithm will never reach the optimal solution without mutation. The optimal solution is $x_{optimal} = 9\ 9\ 0\ 0\ 9\ 9\ 0\ 0$. If mutation does not occur, then the only way to change genes is by applying the crossover operator. Regardless of the way crossover is performed, its only outcome is an exchange of genes of parents at certain positions in the chromosome. This means that the first gene in the chromosomes of children can only be either 6, 8, 2 or 4 (i.e. first genes of x_1, x_2, x_3 and x_4), and because none of the individuals in the initial population begins with gene 9, the crossover operator alone will never be able to produce an offspring with gene 9 in the beginning.
One can easily check that a similar problem is present at several other positions. Thus, without mutation, this GA will not be able to reach the optimal solution.

Exercise 3.10

a) Suggest what chromosome could represent an individual in this algorithm?
Answer: On each day, a solution is a combination of 5 cabin crews assigned to 3 airplanes. Thus, we could encode different crews by different genes:
Genes (crews) : A; B; C; D; E
and then a chromosome representing a solution would be a chain of 3 genes (i.e. 3 cabin crews):

Solution 1 :	A B C
Solution 2 :	A B D
. . .	. . .
Solution 10 :	C D E

In this case, each position in the chromosome corresponds to a plane. Another encoding is possible for the same problem. For example, we could encode if a crew is assigned on a plane by number 1 (if it does not matter which plane it is) and 0 if the crew has a day off:
Genes (work, day off) : 1; 0
Then we can use a chromosome of 5 genes each corresponding to a particular crew with 0 or 1 representing if a crews is at work or has a day off:

Solution 1 :	1 1 1 0 0
Solution 2 :	1 1 0 1 0
. . .	. . .
Solution 10 :	0 0 1 1 1

As you can see, the same problem can be encoded in many different ways.

b) Suggest what could be the alphabet of this algorithm? What is its size?

Answer: This depdends on encoding used. In the first case, when genes represent the crews, the alphabet consists of 5 leters. In the second case, when binary representation is used, only two genes are required.

c) Suggest a fitness function for this problem.

Answer: You may come up with different versions, but it is important for the fitness to take into account the condition that cabin crews
cannot work more than 2 days in a row. For example, the fitness function can take into account how many days each crew has left before
a day off (e.g. 1 or 0). The fitness could be calculated as the sum of
these numbers for all drivers in the chromosome. So, if d is the number of days an employee has before a day off, and m is the number of
employees in the selected crews, then the fitness can be computed as
Fitness = $d_1 + d_2 + \cdots + d_m = \sum_{i=1}^{m} d_i$

Exercise 3.12

A genetic algorithm applies the following major steps:

Step 1: Represent the problem variable domain as a chromosome of a fixed length, choose the size of a chromosome population N, the crossover probability pc and the mutation probability pm.

Step 2: Define a fitness function to measure the performance, or fitness, of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.

Step 3: Randomly generate an initial population of chromosomes of size N:

$$x_1, x_2, \dots, x_N$$

Step 4: Calculate the fitness of each individual chromosome:

$$f(x_1), f(x_2), \dots, f(x_N)$$

Step 5: Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness. Highly fit chromosomes have a higher probability of being selected for mating than less fit chromosomes.

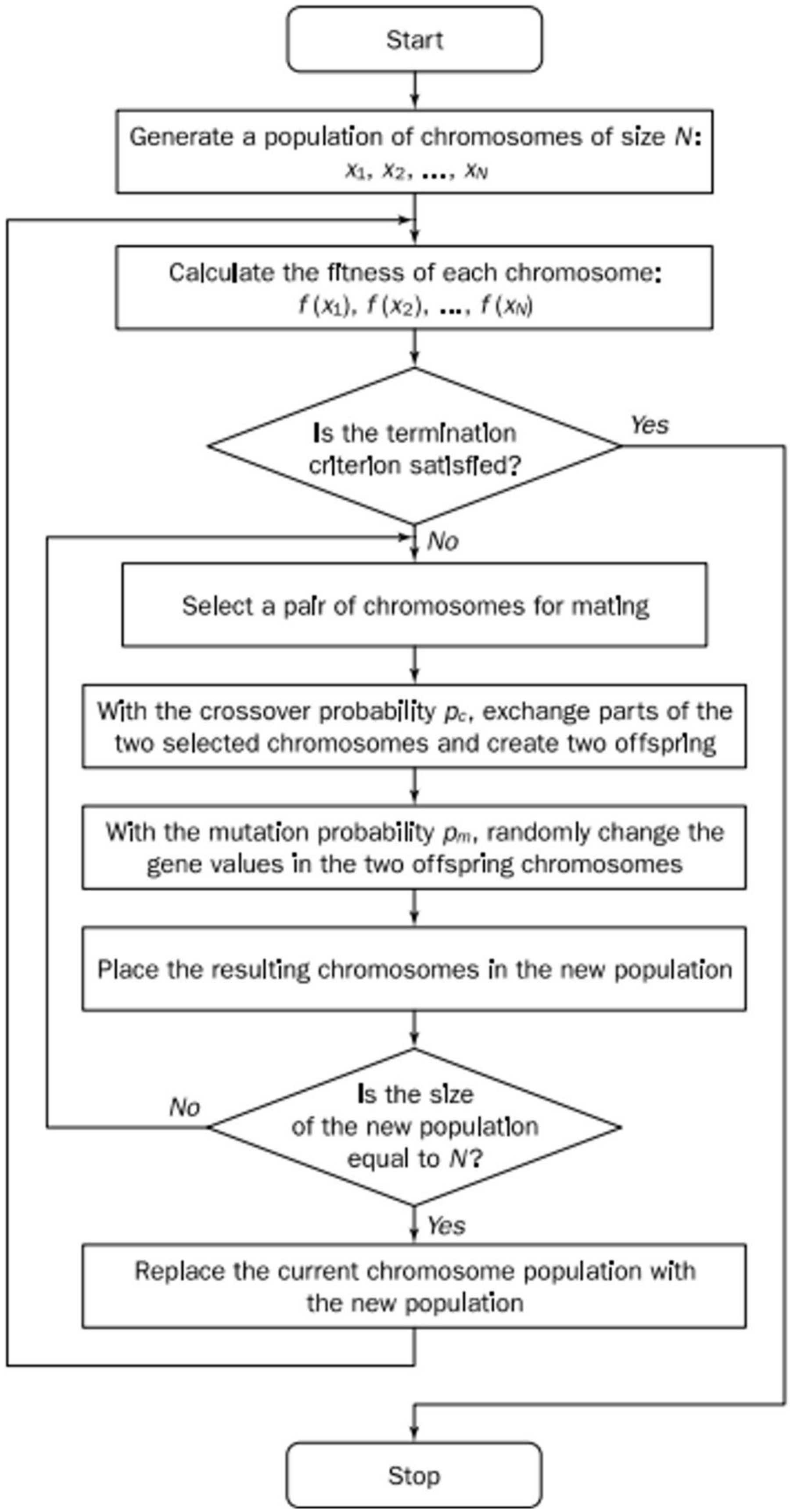
Step 6: Create a pair of offspring chromosomes by applying the genetic operators – crossover and mutation.

Step 7: Place the created offspring chromosomes in the new population.

Step 8: Repeat Step 5 until the size of the new chromosome population becomes equal to the size of the initial population, N.

Step 9: Replace the initial (parent) chromosome population with the new (offspring) population.

Step 10: Go to Step 4, and repeat the process until the termination criterion is satisfied.


Start
Generate a population of chromosomes of size N: $x_1, x_2, \ldots, x_N$
Calculate the fitness of each chromosome: $f(x_1), f(x_2), \ldots, f(x_N)$
Is the termination criterion satisfied?
Yes
No
Select a pair of chromosomes for mating
With the crossover probability p_c, exchange parts of the two selected chromosomes and create two offspring
With the mutation probability p_m, randomly change the gene values in the two offspring chromosomes
Place the resulting chromosomes in the new population
Is the size of the new population equal to N?
No
Yes
Replace the current chromosome population with the new population
Stop

Usually, we keep one of the following termination conditions:

- When there has been no improvement in the population for X iterations.
- When we reach an absolute number of generations.
- When the objective function value has reached a certain pre-defined value.

Exercise 3.20

One of the central problems in computer science is how to make computers solve problems without being explicitly programmed to do so. Genetic programming offers a solution through the evolution of computer programs by methods of natural selection.

Genetic programming searches the space of possible computer programs for a program that is highly fit for solving the problem at hand.

LISP has a highly symbol-oriented structure. Its basic data structures are atoms and lists. An atom is the smallest indivisible element of the LISP syntax. Both atoms and lists are called symbolic expressions or S-expressions. In LISP, all data and all programs are S-expressions. This gives LISP the ability to operate on programs as if they were data. In other words, LISP programs can modify themselves or even write other LISP programs. This remarkable property of LISP makes it very attractive for genetic programming.

Exercise 3.22

1. Determine the set of terminals.
2. Select the set of primitive functions.
3. Define the fitness function.
4. Decide on the parameters for controlling the run.
5. Choose the method for designating a result of the run.


Start
Generate an initial population of S-expressions of size N
Execute each S-expression and calculate its fitness
Designate the best-so-far S-expression as the result of the run
Is the termination criterion satisfied?
Yes
No
Place the best-so-far S-expression in the new population
Select generic operator
Cloning
Mutation
Crossover
Select one S-expression
Select a pair of S-expressions
Select one S-expression
Copy the S-expression in the new population
Perform crossover and place the two offspring in the new population
Perform mutation and place the mutant in the new population
Is the size of the new population equal to N?
No
Yes
Replace the current population with the new one
Stop

Genetic programming uses high-level building blocks of variable length. Their size and complexity can change during breeding. Genetic programming works well in a large number of different cases and has many potential applications.

Last Question

1. Initialization: Generate six random chromosomes, each consisting of 10 bits.
   - Chromosome 1: 1100100000
   - Chromosome 2: 1011000010
   - Chromosome 3: 0010110000
   - Chromosome 4: 1110111010
   - Chromosome 5: 1001001111
   - Chromosome 6: 0010011011
2. Evaluation: Evaluate the fitness of each chromosome by counting the number of correct answers.
   - Chromosome 1: 2 correct answers (fitness value = 2)
   - Chromosome 2: 3 correct answers (fitness value = 3)
   - Chromosome 3: 3 correct answers (fitness value = 3)
   - Chromosome 4: 7 correct answers (fitness value = 7)
   - Chromosome 5: 3 correct answers (fitness value = 3)
   - Chromosome 6: 3 correct answers (fitness value = 3)
3. Selection: Select the two chromosomes with the highest fitness to be parents for the next generation.
   - Chromosome 4 and Chromosome 2 are selected as parents.
4. Crossover: Create two new offspring by performing a one-point crossover on the selected parents.
   - Offspring 1: 1011000010 (from parent 2) + 1110111010 (from parent 4) = 1011111010
   - Offspring 2: 1110111010 (from parent 4) + 1011000010 (from parent 2) = 1110000010
5. Mutation: Perform a bit-flip mutation on the fifth bit of each chromosome in the population.
   - Chromosome 1: 1100100000 -> 1100000000
   - Chromosome 2: 1011000010 -> 1011010010
   - Chromosome 3: 0010110000 -> 0010010000
   - Chromosome 4: 1110111010 -> 1110101010
   - Chromosome 5: 1001001111 -> 1001101111
   - Chromosome 6: 0010011011 -> 0010001011
   - Offspring 1: 1011111010 -> 1011101010
   - Offspring 2: 1110000010 -> 1110010010
6. Evaluation: Evaluate the fitness of the new offspring.
   - Chromosome 1: 1 correct answer (fitness value = 1)
   - Chromosome 2: 4 correct answers (fitness value = 4)
   - Chromosome 3: 1 correct answer (fitness value = 1)

    - Chromosome 4: 6 correct answers (fitness value = 6)
    - Chromosome 5: 4 correct answers (fitness value = 4)
    - Chromosome 6: 1 correct answer (fitness value = 1)
    - Offspring 1: 6 correct answers (fitness value = 6)
    - Offspring 2: 4 correct answers (fitness value = 4)
7. Selection: Select the two offspring with the highest fitness, along with the two parents from the previous generation, to form the new population.
    - New Population: Chromosome 4, Chromosome 2, Offspring 1, and Chromosome 5
8. Repeat steps 4-7 for the second generation.
    - Selection: Chromosome 4 and Offspring 1 are selected as parents.
    - Crossover: Offspring 2 is created by one-point crossover between parent 1 and 2.
    - Mutation: Bit 5 is flipped in Offspring 2.
    - Evaluation: Offspring 2 has 5 correct answers and fitness value of 5.
    - Selection: Chromosome 4 and Offspring 2 are selected to form the new population.

At the end of two generations, we have a final population with two chromosomes that have the highest fitness values:

- Chromosome 4: 6 correct answers (fitness value = 6)
- Offspring 2: 5 correct answers (fitness value = 5)