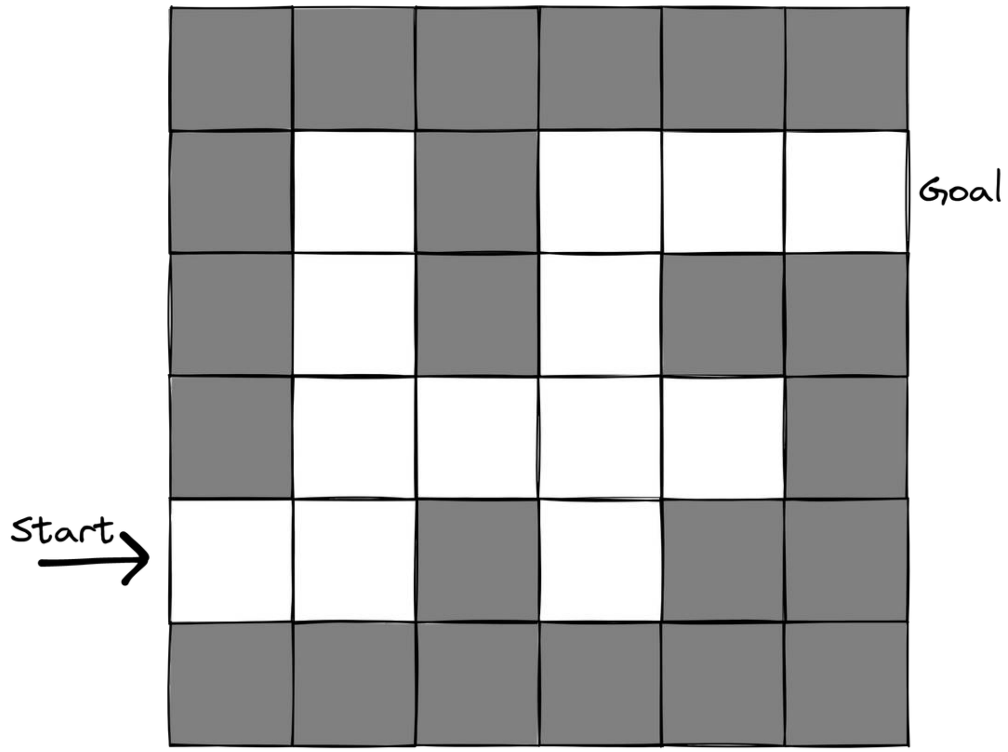
Reinforcement Learning

Genetic Algorithms



Week 21

Middlesex University Dubai; CST4050 Fall21;
Instructor: Dr. Ivan Reznikov

v1.1

Genetic Algorithms (GA)

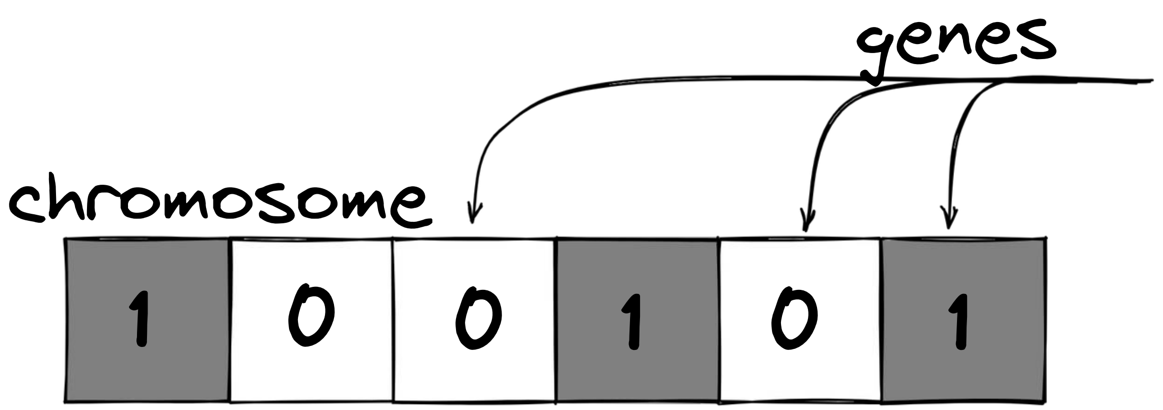
Natural Selection is a very successful organizing principle for optimizing individuals and populations of individuals.

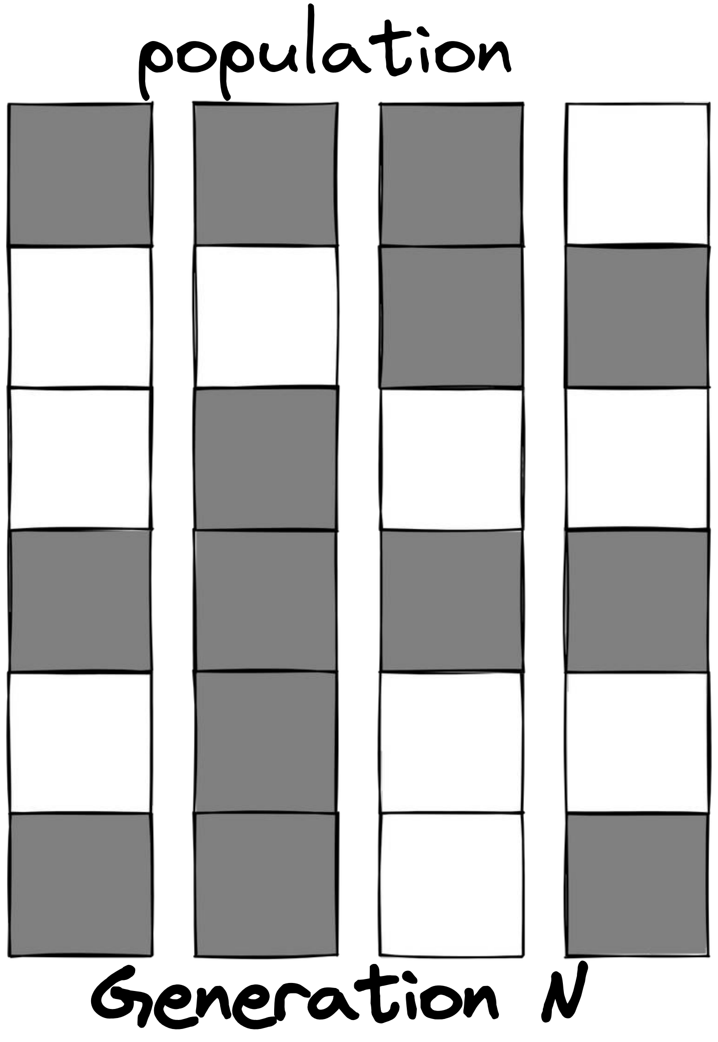
If we can mimic natural selection, then we will be able to optimize more successfully.

A possible design of a system – as represented by its design **vector X [Model].**
It can be considered individual fighting for survival within a larger population.

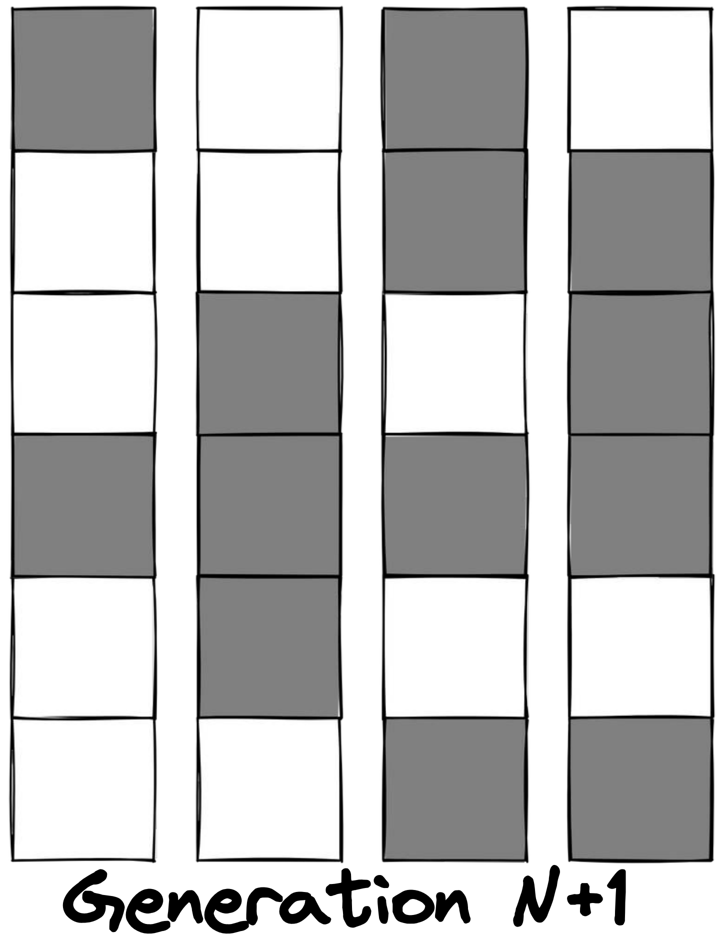
Only the fittest survive. Fitness is assessed via objective **function J [Policy]**.

GA: Terminology



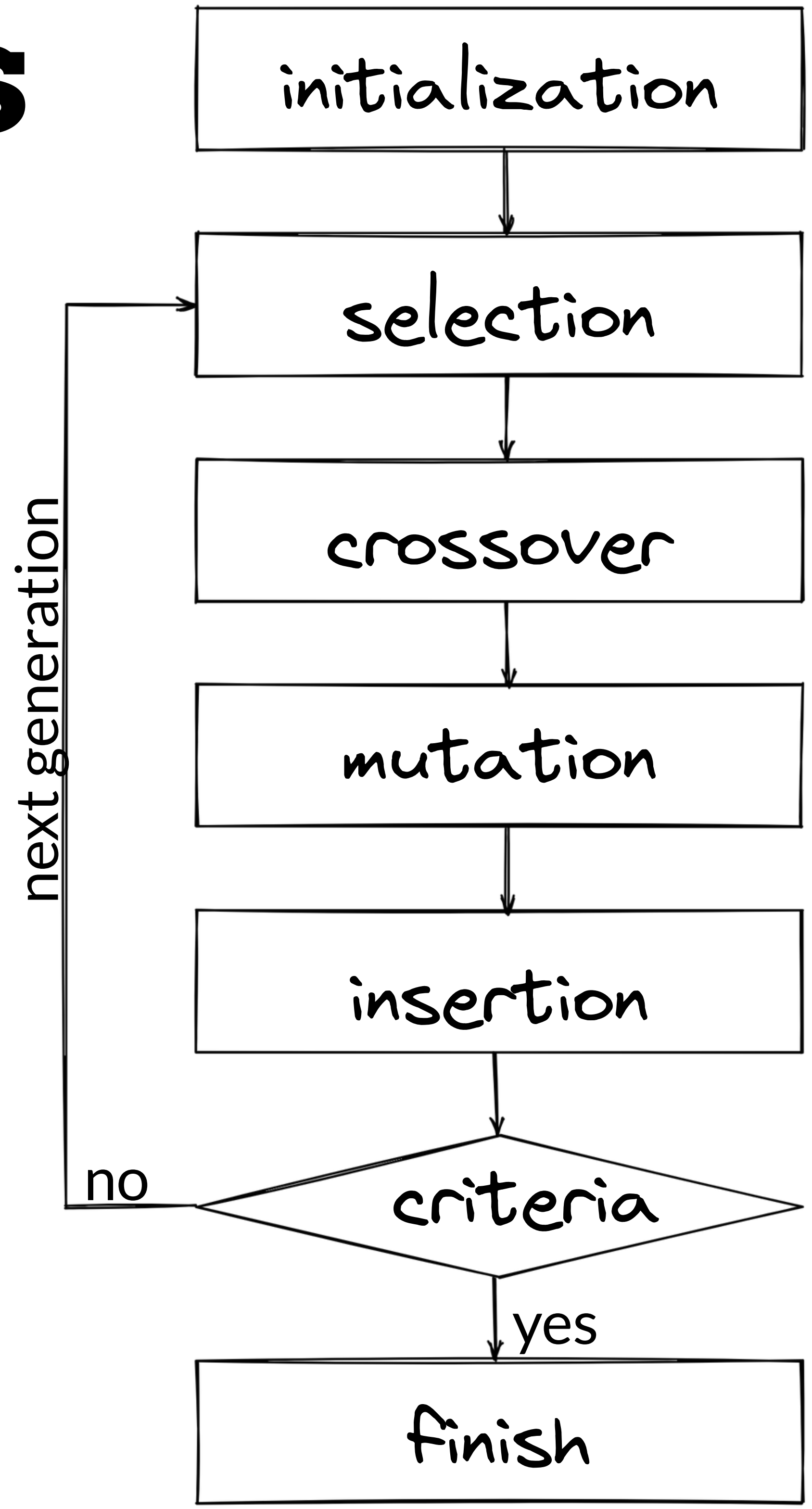


Genetic operators:
selection
crossover
insertion
mutation



GA over generations

1. Initialization – initialize population
2. Selection – select individual for mating
3. Crossover – mate individuals and produce children
4. Mutation – mutate children
5. Insertion – insert children into population

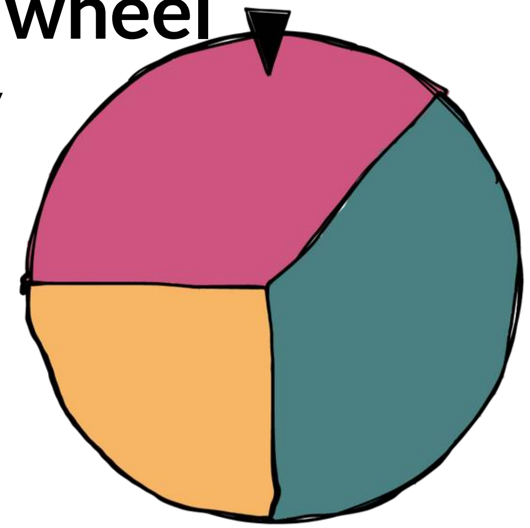


Goldberg, D.E.," Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, 1989

Selection

After calculating the fitness of the individuals of the whole population, we should pick some of them for reproduction. This process is called selection.

There are several strategies for selection - for example, a **roulette wheel selection** when individuals are selected randomly with probability proportional to their fitness.



Ranked selection is when we select only a few fittest individuals.

If the size of the population remains the same in all generations, then not all individuals will be allowed to reproduce, as, usually, the fitter reproduces more.

Selection

Fitness function is needed to compare different solutions (individuals) and choose the better ones.

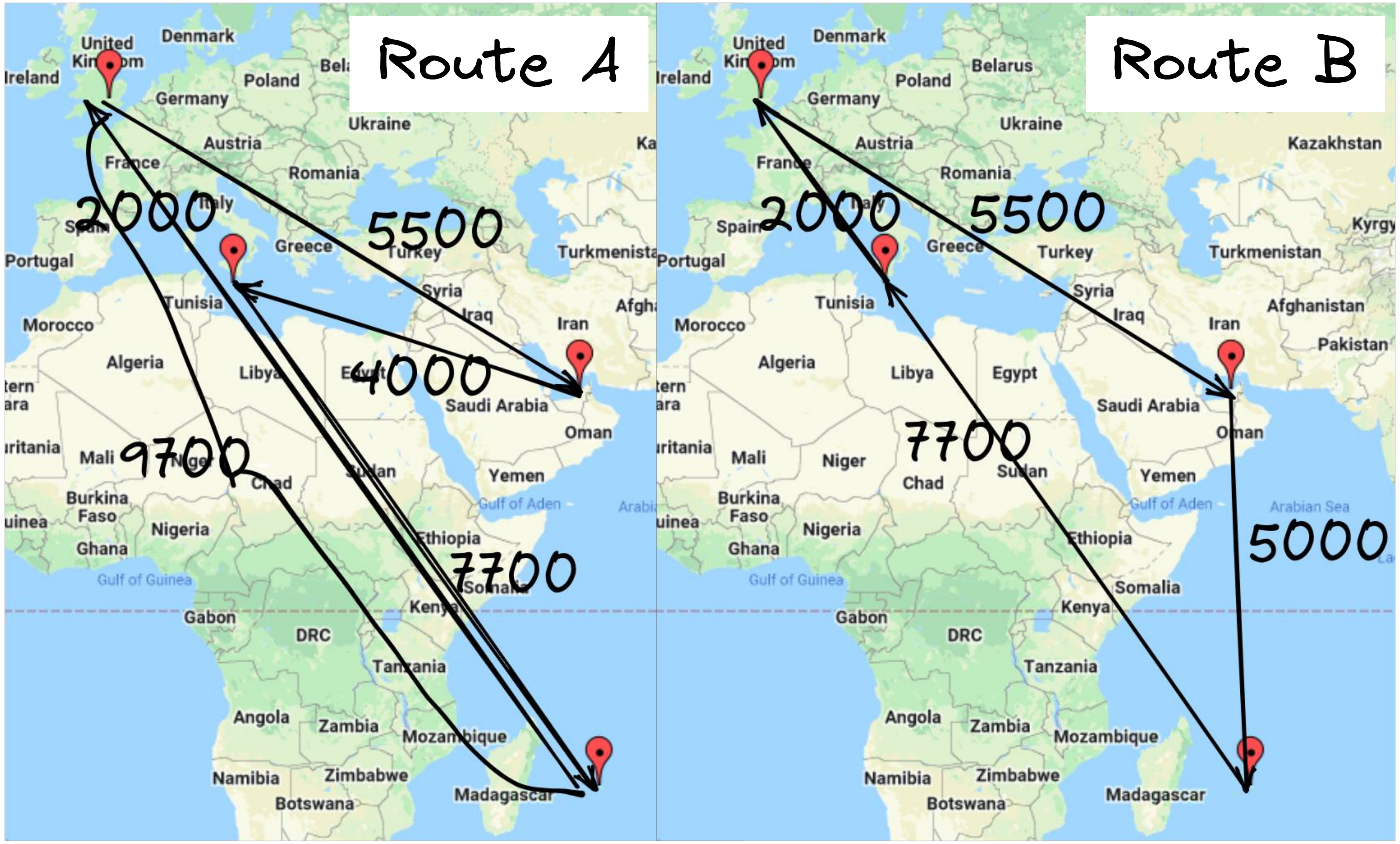
In our case fitness is

$$f(x) = -\,\mathrm{length}(x)$$

$$\mathrm{length}(R_A) > \mathrm{length}(R_B)$$

$$\Rightarrow f(R_A) < f(R_B)$$

So, we choose Route B

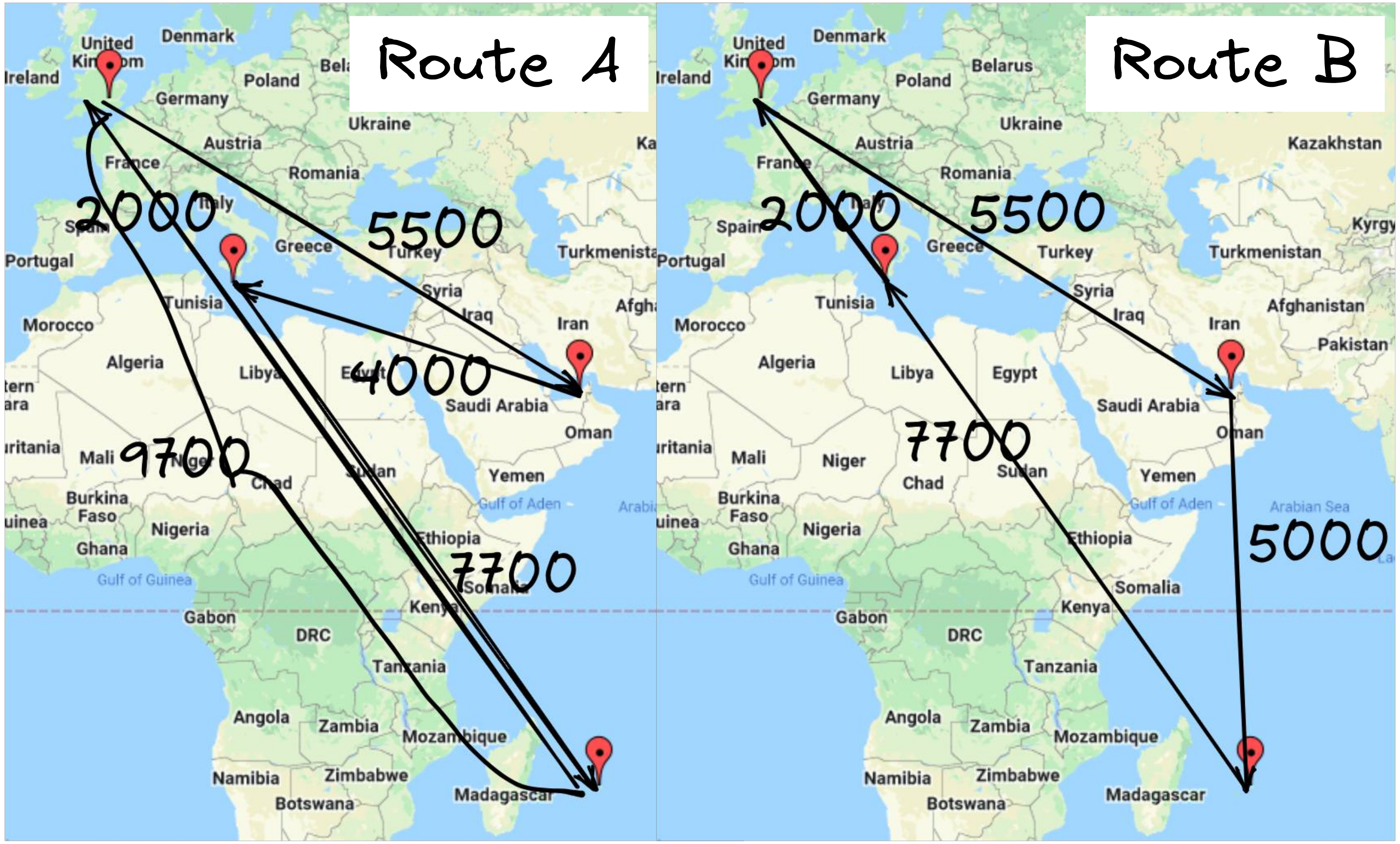


Encoding genes

For the Travelling Salesman Problem, the genes may represent different cities, and chromosomes represent different routes:

L – London
D – Dubai
Ma – Malta
Ms – Mauritius

RouteA: DMaMsLD
RouteB: DMsMaLD



Crossover

The selected parents are grouped in pairs to produce their offspring (children).
Crossover replaces some genes in one parent with the corresponding genes of the other. Thus, each child becomes an offspring of two parents.
There are many ways to do the crossover, such as the one-point crossover.
For example, one–point crossover at point 3:

$$
\begin{array}{l} P_1 = 1\ 0\ 1 \mid 1\ 0\ 1\ 1 \\ P_2 = 0\ 1\ 1 \mid 1\ 0\ 0\ 0 \end{array}
\longrightarrow
\begin{array}{l} O_1 = 1\ 0\ 1 \mid 1\ 0\ 0\ 0 \\ O_2 = 0\ 1\ 1 \mid 1\ 0\ 1\ 1 \end{array}
$$

Mutation

A randomly chosen gene (or several genes) is changed to some other gene (mutate). For example, mutation of genes 3 and 6 in the offspring O_1 will give:

$$O_1 = 1\ 0\ \underline{1}\ 1\ 0\ \underline{1}\ 1 \longrightarrow O_1 = 1\ 0\ \underline{0}\ 1\ 0\ \underline{0}\ 1$$

Mutation helps to add diversity to the population. It works as random experimentation.
Mutation can help the population to avoid local maximum.
The process of mutation may occur at a variable rate.
Some problems cannot be solved at all without mutation.

Why GA works? Variations of GA

- In each generation, we check several solutions at once. Thus GA is a kind of a parallel search.
- Fitness and selection filter out bad solutions from good ones.
- Offspring inherit properties of mostly good solutions.

Variations of GA:

- Different selection strategies: roulette wheel, ranked, tournament selection.
- Other forms of the crossover operator: two-point, uniform.
- Static or variable population size
- Variable mutation rates
- Some "good" parts of the chromosomes may be kept intact. They are usually called schemas.
- Some outstanding individuals may survive for several generations (elitism).

GA: Case 1

Consider the following maximization problem:

$$\max f(x) = (x-16)^2 + (x-32)^2 - x^2 - 1000/(x+1) + 1200$$

where x is an integer between 0 and 63.

Maximum value 758.67 at x = 3

Gene Representation:

We'll use binary integer of length 6:

000000 => 0
111111 => 63
011001 = $0^5 + 2^4 + 2^3 + 0^2 + 0^1 + 2^0$ = 16 + 8 + 1 = 25

GA: Case 1. Initialization

Initial populations are randomly generated

Suppose that our population size is 4

An example initial population:

id	G1	G2	G3	G4	G5	G6	x
1	0	0	1	1	1	1	15
2	1	1	0	1	0	0	52
3	0	1	1	1	0	0	28
4	0	1	0	1	0	1	21

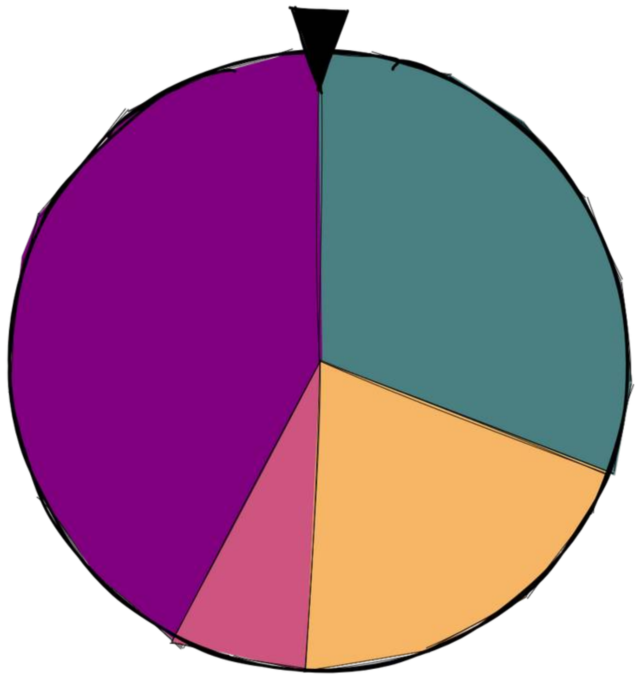
GA: Case 1. Fitness Evaluation

Let's evaluate the fitness of the initial population. The objective function f(x) is used as the fitness function. Each individual is decoded to integer, and the fitness function value is calculated.

id	G1	G2	G3	G4	G5	G6	x	f(x)	$f_i(x)/sum(f(x))$
1	0	0	1	1	1	1	15	1202.50	0.43
2	1	1	0	1	0	0	52	173.13	0.06
3	0	1	1	1	0	0	28	541.52	0.20
4	0	1	0	1	0	1	21	859.55	0.31

GA: Case 1. Selection

Let's select individuals for the mating pool. Selection probability is proportional to the fitness value of the individual. In our case, we'll use the roulette wheel selection method.

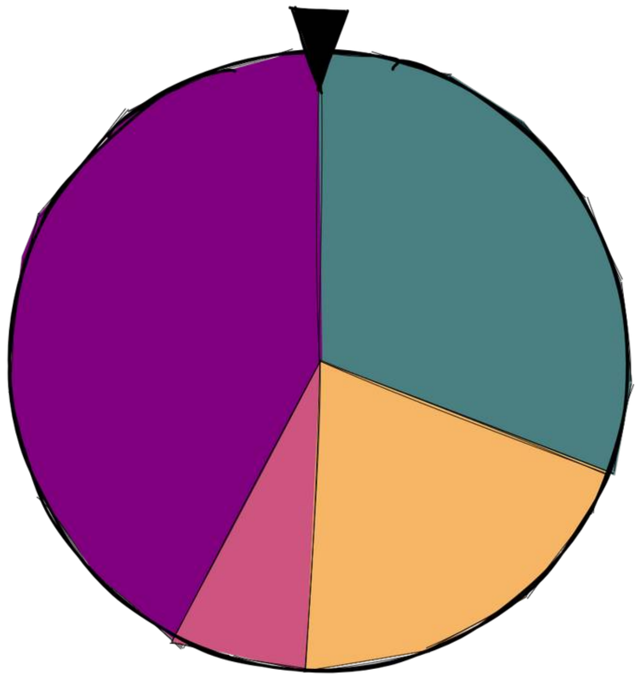


id	G1	G2	G3	G4	G5	G6	x	f(x)	$f_i(x)/sum(f(x))$	Expected number
1	0	0	1	1	1	1	15	1202.50	0.43	1.73
2	1	1	0	1	0	0	52	173.13	0.06	0.25
3	0	1	1	1	0	0	28	541.52	0.20	0.78
4	0	1	0	1	0	1	21	859.55	0.31	1.24

GA: Case 1. Selection

Outcome of Roulette wheel is 1, 1, 3, and 4.

Our final table will look the following way:



id	G1	G2	G3	G4	G5	G6	x	f(x)	$f_i(x)/sum(f(x))$	Expected number
1	0	0	1	1	1	1	15	1202.50	0.43	1.73
1	0	0	1	1	1	1	15	1202.50	0.43	1.73
3	0	1	1	1	0	0	28	541.52	0.20	0.78
4	0	1	0	1	0	1	21	859.55	0.31	1.24

GA: Case 1. Crossover

Our crossover rules:

- Two individuals are randomly chosen from mating pool
- Crossover occurs with the probability of $p_c = 1$
- Crossover point is chosen randomly

id	Matting pool	Crossover partner	Crossover point	New population	x	f(x)
1	001111	3	3	001100	12	1395.08
1	001111	4	2	000101	5	1858.33
3	011100	1	3	011111	31	433.75
4	010101	1	2	011111	31	433.75

GA: Case 1. Mutation

Mutation is applied on a bit-by-bit basis. Each gene mutated with probability of $p_m = 0.001$. That makes mutation a rare event.

id	Before mutation	After mutation	x	f(x)
1	001100	001100	12	1395.08
1	000101	000101	5	1858.33
3	011111	010111	23	759.33
4	011111	011111	31	433.75

GA: Case 1. Generation2

Fitness values of the new population are calculated. Our new generation looks more promising than the previous population.

Old population	x	f(x)	New population	x	f(x)
001111	15	1202.50	001100	12	1395.08
110100	52	173.13	000101	5	1858.33
011100	28	541.52	010111	31	433.75
010101	21	859.55	011111	23	759.33
	sum	2776.69		sum	4446.49
	avg	694.17		avg	1111.62
	median	700.53		median	1077.21
	max	1202.50		max	1858.33

GA: Case 1. Generation3

Fitness values of the new population are calculated. Our new generation looks more promising than the previous population.

Old population	x	f(x)	New population	x	f(x)
001100	12	1395.08	000010	3	1951.00
000101	5	1858.33	001101	13	1329.57
010111	31	433.75	001110	14	1265.33
011111	23	759.33	000111	7	1732.00
	sum	4446.49		sum	6277.90
	avg	1111.62		avg	1569.48
	median	1077.21		median	1530.79

GA Application

Problems solved by GA (e.g., TSP, timetable, etc.) are the choice problems. (Indeed, we prefer different solutions, there is a fitness function, which acts as a utility).

- The GA problems, however, usually have astronomically large choice sets.
- It is unclear how to find the optimal solution other than searching through all possible combinations.
- GA is a very sophisticated parallel search algorithm that uses a particular representation of solutions (encoding), a mixture of stochastic (random mutation), and deterministic (crossover and inheritance) variations of solutions.

Result: it saves time.