Lecture 9:

Genetic Algorithms

Dr. Roman V Belavkin

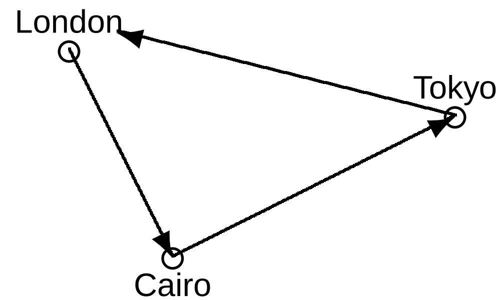
BIS4435

OVERVIEW

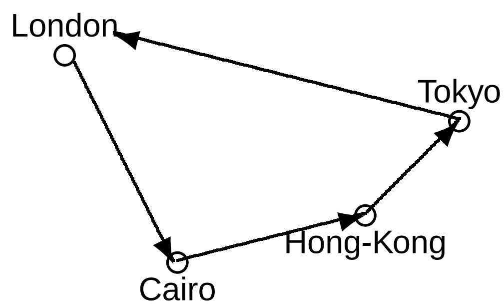
1. Combinatorial Problems
2. Natural Selection
3. Individuals and Population
4. Fitness Functions
5. Encoding Genes of an Individual
6. Selection, Crossover and Mutation
7. GA Algorithm
8. Why GA work?
9. Problem Example

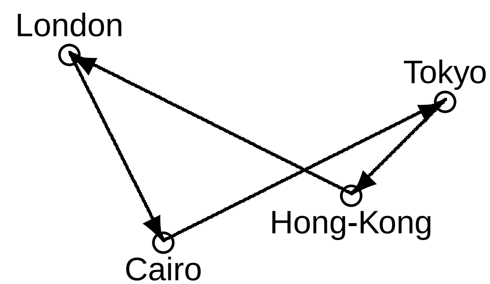
THE TRAVELLING SALESMAN PROBLEM

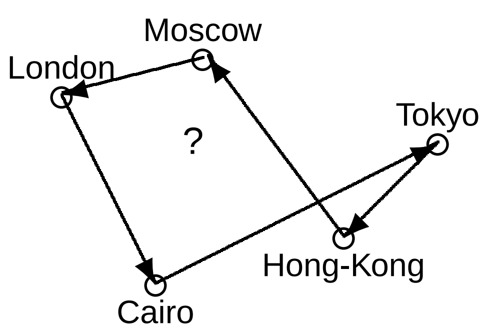
A salesman goes on a tour around several cities. Each city has to be visited only once. What is the shortest tour?



The problem becomes harder if we increase the number of cities ($n = 4, 5, \ldots$).







Mathematicians calculated that for n cities the number of possible tours is:

$$(n-1)!$$

(e.g. for $n = 10$, $(10-1)! = 9! = 362,880$)

THE TIMETABLING PROBLEM

Problem of balancing the resources (teachers, rooms) so that all teaching is done and students can take sensible combinations of courses. The input variables are teachers, rooms and time slots. There are many more objectives:

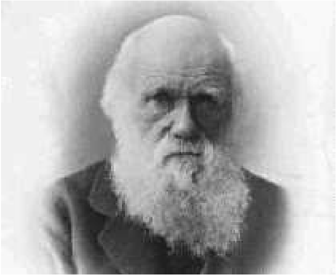
- Teachers can only be at one place at one time
- Rooms can only be used by one teacher
- Each teacher can only teach certain subjects
- Subject clashes must be minimal
- Rooms must be big enough for all students
- All subjects must be taught to all registered students

“Neural networks are second best way to solve just about anything... and genetic algorithms are the third”

- There are problems for which there are no clear methods of finding an optimal (best) solution.
- Some problems seem to have only one way of finding a solution: by random search. These are so–called **combinatorial** problems.
- The difficulty is that many tasks have so many possible combinations, that it is simply impossible to try all of them.
- Various search methods and heuristics have been developed to tackle such problems. **Genetic Algorithms** is one of such techniques that was inspired by the laws of natural evolution.

NATURAL SELECTION

- Living organisms are fighting the forces of nature to survive. Those who are the **fittest** (i.e. strongest, fastest, biggest) are more likely to survive.



Charles Darwin

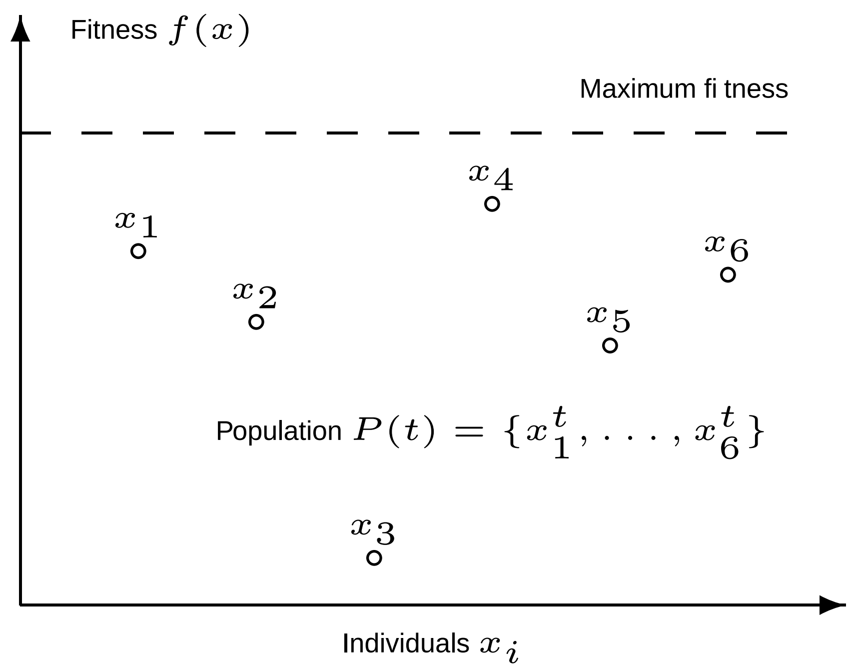
- Those who survive (the fittest) mate and reproduce (**selection**).
- Children are similar (**inheritance**), but not exactly like the parents because of **cross–fertilisation** and **mutation**. Thus, children can be either less or more fit than their parents.
- Children repeat the path of their parents. After several generations the organisms become much fitter than at the beginning.

SOLUTIONS AS INDIVIDUALS IN POPULATION

- Suppose that there are many possible solutions for the problem: x_1, x_2,...,x_n. The main idea of GA is to view each solution x_i of the problem as an **individual** living organism
- The number of all possible solutions for the problem can be incredibly large (more than atoms in the universe). So, we consider only a relatively small number $m \ll n$ of them. We shall call a collection of m solutions at the current time moment a **population** $P(t) = \{x_1^t, \ldots, x_m^t\}$.
- With time the individuals (like organisms) and the whole population will be evolving. So, time t here simply describes the generation of the population and its individuals.

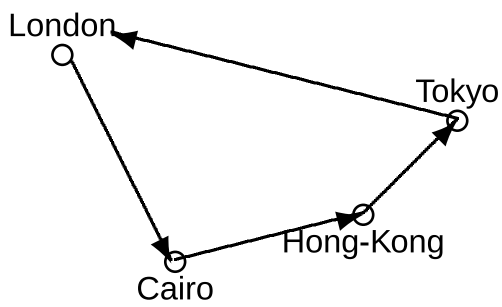
FITNESS FUNCTION

Fitness function is simply another word for the **utility** function. It is needed to compare different solutions (individuals) and choose the better ones. As with utility, it returns a number such that if $f(x_1) > f(x_2)$, then x_1 is preferable to x_2 (x_1 is 'fitter').

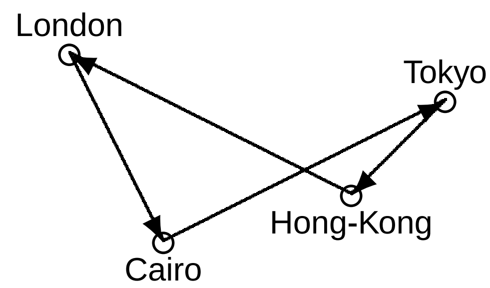


EXAMPLE

We can choose the tours by comparing their lengths



x_1



x_2

Thus, fitness is $f(x) = -\text{length}(x)$ and

$$\begin{aligned} \text{length}(x_1) &< \text{length}(x_2) \\ f(x_1) &> f(x_2) \end{aligned}$$

So, we choose x_1

GA and THE CHOICE PROBLEM

- Problems, solved by GA (e.g. TSP, timetable, etc), are the choice problems (Indeed, we have preference between different solutions, there is a fitness function, which acts as a utility).
- The GA problems, however, usually have astronomically large choice sets.
- It is not clear how to find the optimal solution other then by searching through all possible combinations.
- GA is a very sophisticated parallel search algorithm that uses a special representation of solutions (encoding), a mixture of stochastic (random mutation) and deterministic (crosover and inheritance) variations of solutions. Result: it saves time.

ENCODING GENES OF AN INDIVIDUAL

Before we can start GA we need to represent each individual by its **genes**. This is probably the biggest challenge in GA.

- Identify the simplest elements (building blocks) of which a solution may consist, and assign a letter (or a number) to each element. We shall call each letter a **gene**.
- The number of these simplest elements (and hence the genes) is finite and not very large. For example, in DNA there are only four kinds of genes: A, G, T and C. We shall call the collection of all these letters an **alphabet** of the algorithm.
- This way a solution x_i is represented by a string of genes. We shall call this string a **chromosome** of an individual.

EXAMPLE: TSP

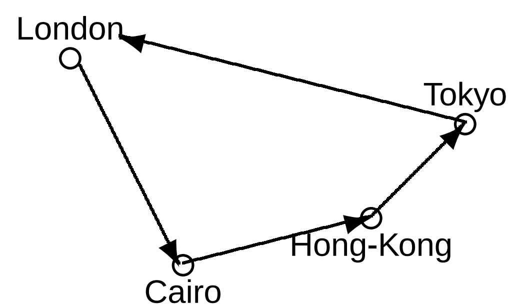
For the Travelling Salesman Problem the genes may represent different cities, and the chromosomes represent different routes:

L London

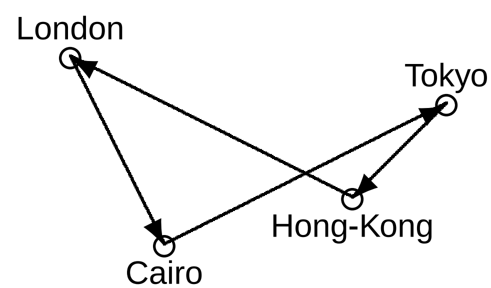
C Cairo

H Hong–Kong

T Tokyo



$$x_1 = LCHTL$$



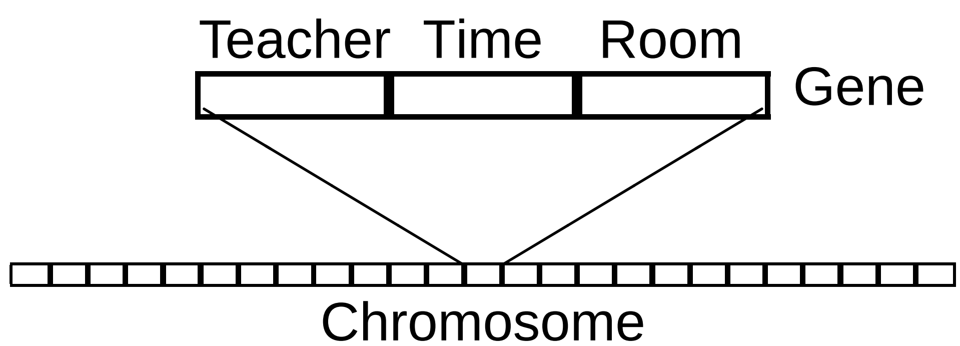
$$x_2 = LCTHL$$

EXAMPLE: TIMETABLING

We can assign numbers to a teacher, to a time slot and to a room:

Number	Teacher	Time	Room
1	Prof. Newell	9:00am	Room B48
2	Dr. Ritter	11:00am	Room C52
3	Prof. Wood	1:00pm	
⋮	⋮	⋮	

Each gene will be a sequence of three numbers, such as 132 means “Prof. Newell teaches at 1:00pm in Room C52”.



SELECTION

- After the fitness of the individuals of the whole population has been calculated, we should pick some of them for reproduction. This process is called **selection**.
- There are several strategies for selction. For example a **roulette wheel** selection, when individuals are selected randomly with probability proportional to their fitness.
- **Ranked** selction is when we select only the few fittest individuals.
- If the size of the population remains the same in all generations, then not all individuals will be allowed to reproduce. This is because usually the fitter reproduce more.

CROSSOVER

- The selcted parents are grouped in pairs in order to produce their **offspring** (children).
- **Crossover** is an operation of replacing some genes in one parent by the corresponding genes of the other. Thus, each child becomes an offsring 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:

$$\begin{array}{rr|l} P_1 = & \mathbf{1\,0\,1} & \mathbf{0\,0\,1\,0} \\ P_2 = & 0\,1\,1 & 1\,0\,0\,1 \end{array} \Rightarrow \begin{array}{rr|l} O_1 = & \mathbf{1\,0\,1} & 1\,0\,0\,1 \\ O_2 = & 0\,1\,1 & \mathbf{0\,0\,1\,0} \end{array}$$

MUTATION

A randomly chosen gene (or several genes) is changed to some other gene (**mutate**). For example, mutation of genes 3 and 5 in the offspring O_1 will give:

$$O_1 = 1\,0\,\underline{1}\,1\,\underline{0}\,0\,1 \quad \Rightarrow \quad O_1 = 1\,0\,\underline{0}\,1\,\underline{1}\,0\,1$$

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

SUMMARY OF GENETIC ALGORITHM

After the crossover and mutation operations the new generation may have individuals which are even fitter than their parents (i.e. better solutions of the problem). The process is repeated for several generations untill a good enough solutions is found.

1. Defi ne fi tness function;
2. Encode solutions;
3. Choose initial (random) population;
4. Repeat
   - Evaluate fi tnesses of all the individuals in the population;
   - Select the best individuals for reproduction;
   - Create new generation by mutation and crossover;
5. Until the optimal or a suffi ciently good solution is found.

WHY GA WORK?

- 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 IN GA

- Different selection strategies: **roulette wheel**, **ranked**, **tournament** selection.
- Different 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 very good individuals may survive for several generations (**elitism**).

APPLICATION OF GA

- Transport scheduling
- Timetabling
- Circuit design
- Route optimisation (TSP)

For some classes of problems there are other specific methods that can perform better than GA on these problems. However, when no special method is known for your problem, then try GA and see if it can give you a good result. GA are easily programmable and can be applied to a wide range of problems.

PROBLEM EXAMPLE

Consider a GA with chromosomes consisting of six genes $x_i = abcdef$, and each gene is a number between 0 and 9. Suppose we have the following population of four chromosomes:

$$x_1 = 4\,3\,5\,2\,1\,6 \qquad x_2 = 1\,7\,3\,9\,6\,5$$

$$x_3 = 2\,4\,8\,0\,1\,2 \qquad x_4 = 9\,0\,8\,1\,2\,3$$

and let the fitness function be $f(x) = (a + c + e) - (b + d + f)$.

1. Sort the chromosomes by their fitness
2. Do one–point crossover in the middle between the 1st and 2nd fittest, and two–points crossover (points 2, 4) for the 2nd and 3rd.
3. Calculate the fitness of all the offspring

SOLUTION 1

Apply the fitness function $f(x) = (a + c + e) - (b + d + f)$:

$$
\begin{aligned}
f(x_1) &= (4 + 5 + 1) - (3 + 2 + 6) = -1 \\
f(x_2) &= (1 + 3 + 6) - (7 + 9 + 5) = -11 \\
f(x_3) &= (2 + 8 + 1) - (4 + 0 + 2) = 5 \\
f(x_4) &= (9 + 8 + 2) - (0 + 1 + 3) = 15
\end{aligned}
$$

So, the order is x_4, x_3, x_1 and x_2.

SOLUTION 2

One–point crossover in the middle between x_4 and x_3:

$$\begin{array}{rl|l} x_4 = & \mathbf{9\,0\,8} & \mathbf{1\,2\,3} \\ x_3 = & 2\,4\,8 & 0\,1\,2 \end{array} \Rightarrow \begin{array}{rl} O_1 = & \mathbf{9\,0\,8}\,0\,1\,2 \\ O_2 = & 2\,4\,8\,\mathbf{1\,2\,3} \end{array}$$

Two–point crossover at points 2 and 4 between x_3 and x_1:

$$\begin{array}{rl|l|l} x_3 = & \mathbf{2\,4} & \mathbf{8\,0} & \mathbf{1\,2} \\ x_1 = & 4\,3 & 5\,2 & 1\,6 \end{array} \Rightarrow \begin{array}{rl} O_3 = & \mathbf{2\,4}\,5\,2\,\mathbf{1\,2} \\ O_4 = & 4\,3\,\mathbf{8\,0}\,1\,6 \end{array}$$

SOLUTION 3

Again, apply the fitness function $f(x) = (a + c + e) - (b + d + f)$:

$$
\begin{aligned}
f(O_1) &= (9 + 8 + 1) - (0 + 0 + 2) = 16 \\
f(O_2) &= (2 + 8 + 2) - (4 + 1 + 3) = 4 \\
f(O_3) &= (2 + 5 + 1) - (4 + 2 + 2) = 0 \\
f(O_4) &= (4 + 8 + 1) - (3 + 0 + 6) = 4
\end{aligned}
$$

Has the fitness of the new population improved?