IASCA
One-max problem with a basic Genetic Algorithm

Departamento de Automática, UAH
http://atc1.aut.uah.es

Objectives:

- Understand the structure of a basic GA
- Accustom oneself to the way EAs address problems

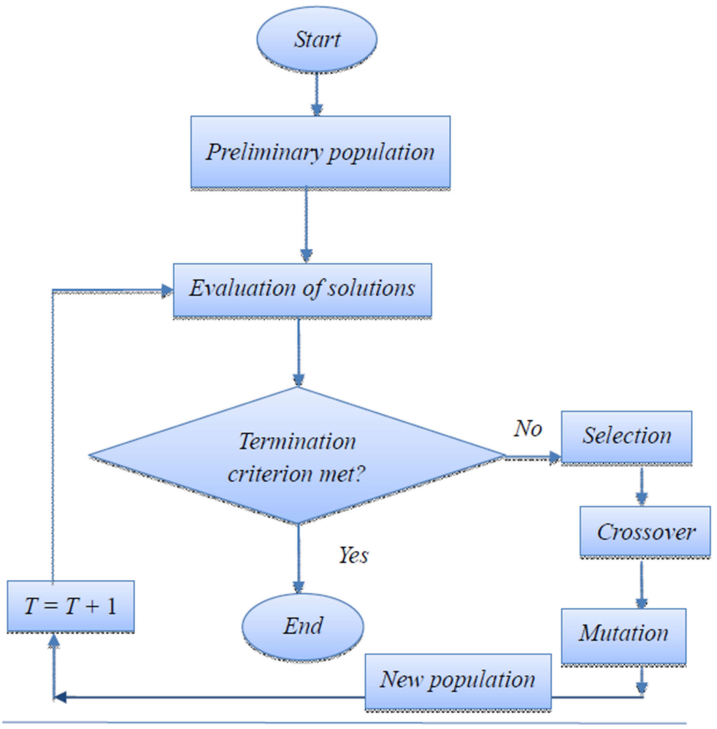
Practice

Often, when someone begins to learn a new programming language he implements the famous "Hello World" program. The equivalent to this in Evolutionary Computation is the one-max, a trivial problem often used to practice a new EC framework or algorithm. The problem is straitforward: Given a binary chromosome of size n, the goal is to maximize the number of ones, i.e., to get an all ones chromosomes. The fitness function is just the sum of ones.

Implement, with any programming language of your choice, a Genetic Algorithm (GA) that solves the one-max problem. Use the following pseudocode as guide.

```
1  n := Chromosome length
2  p := Population size
3  pm := Mutation probability
4
5  Initialize population with random values
6
7  While solution not found
8          i := 0
9          While i < p
10                 Select randomly two individuals in the population
11                 Compute their fitness
12                 Select fittest individual
13
14                 If random_number() < pm
15                         Flip a random position of the selected individual
16
17                 Store individual in the next population
```

The following figure outlines the algorithm to implement.



(Source)

Once the algorithm is implemented, perform the following tasks:

1. Show the best fitness found in each generation. Execute the algorithm. How does the best fitness evolve?
2. Compare the execution time with $n = 10$, $n = 20$ and $n = 100$. (Hint: Use the Unix command `time`).