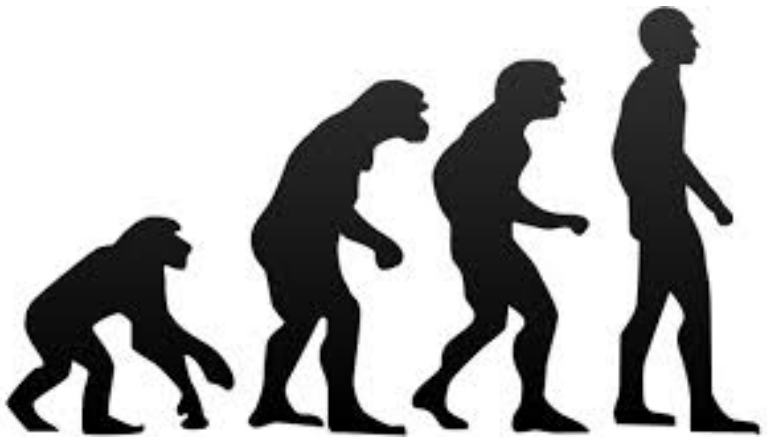
Genetic Algorithm

Alexandre Bergel
http://bergel.eu
05/10/2020



Premise of Genetic Algorithm

Natural selection is pioneered by *Charles Darwin* (1809-1882), biologist who worked on *natural evolution*

"On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life", 1859

> *"One general low, leading to the advancement of all organic beings, namely, vary, let the strongest live and the weakest die"*
>
> *— Charles Darwin*

“Guess the 3-letter word I have in mind”

Secret word: *cat*

“Guess the 3-letter word I have in mind”

“For each try, I tell you the number of correct letters”

Secret word: *cat*

Secret word: *cat*

gaz

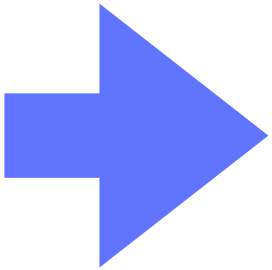
cow

poz

gaz

cow

poz



Using genetic operators, the words are combined and some letters are randomly modified

gow

caz

pow


gaz
cow
poz
gow
caz
pow
goz
cat
poz

1st generation

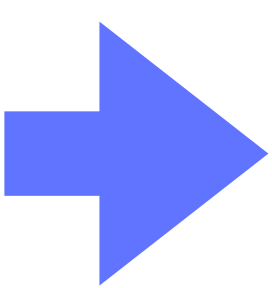
2nd generation

3rd generation

gaz

cow

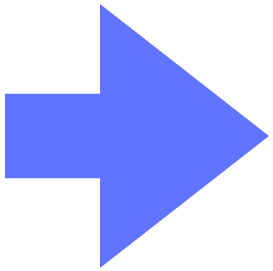
poz



gow

caz

pow

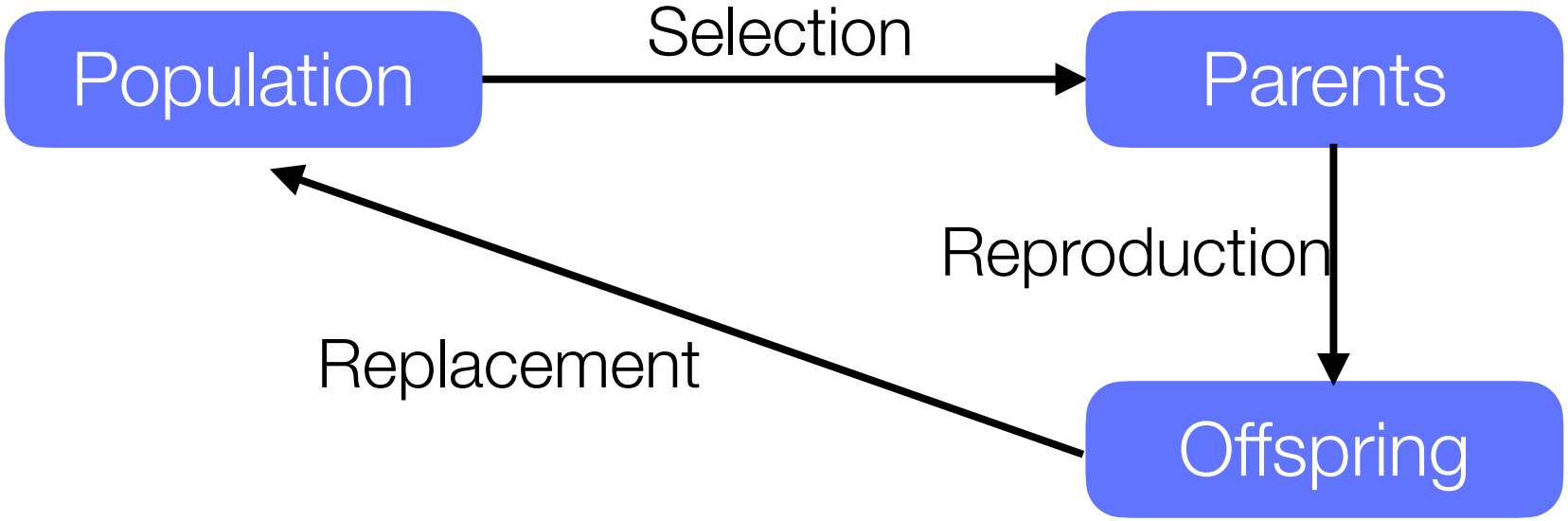


goz

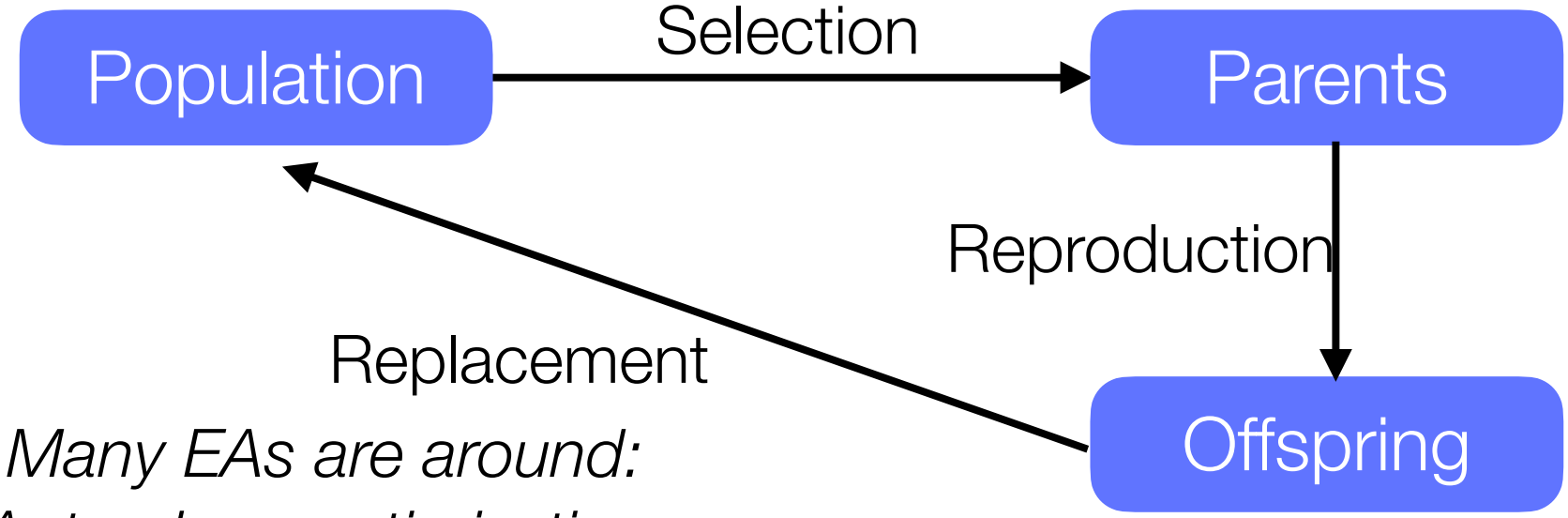
cat

poz

Flow chart of an evolution algorithm

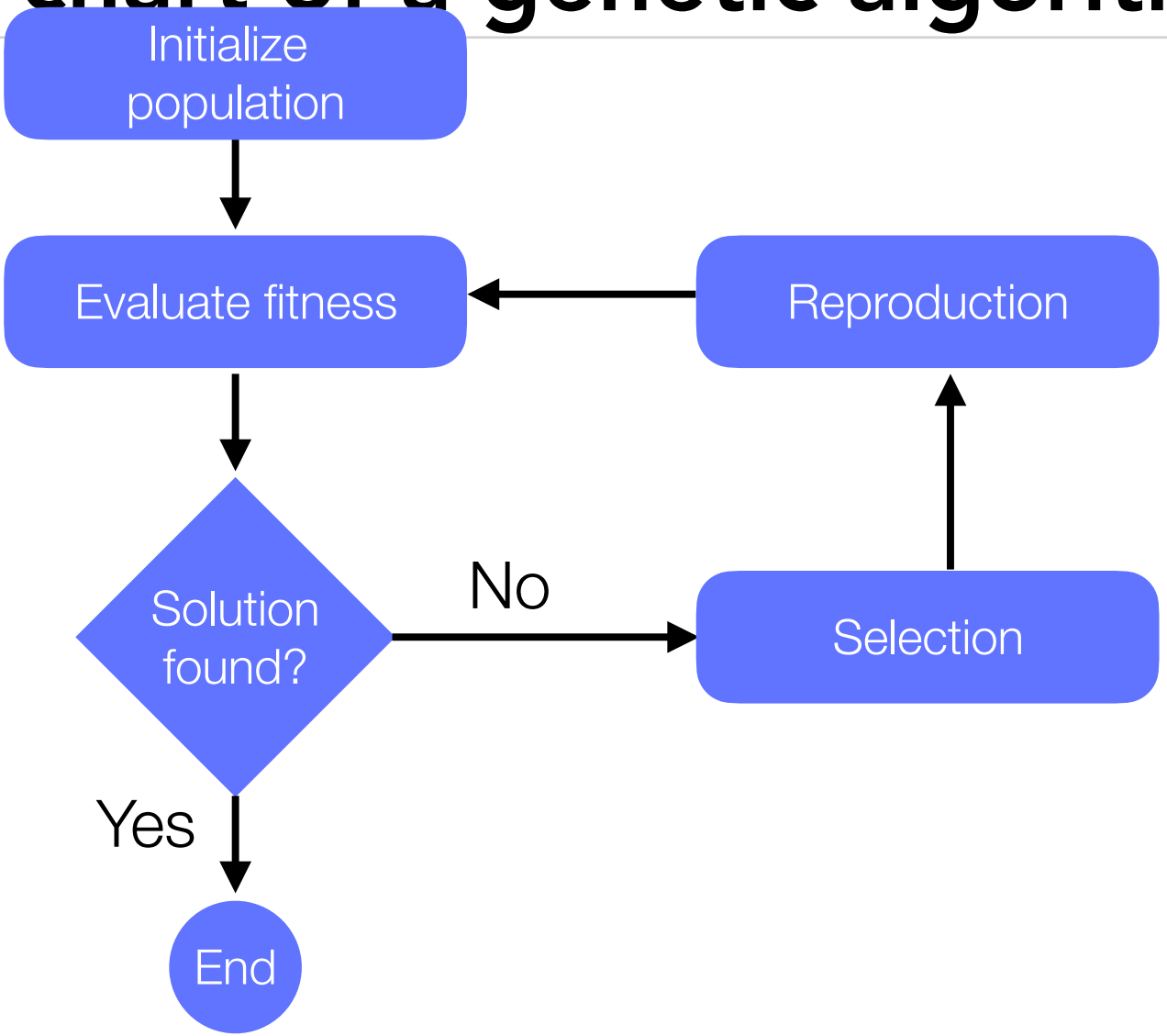


Flow chart of an evolution algorithm

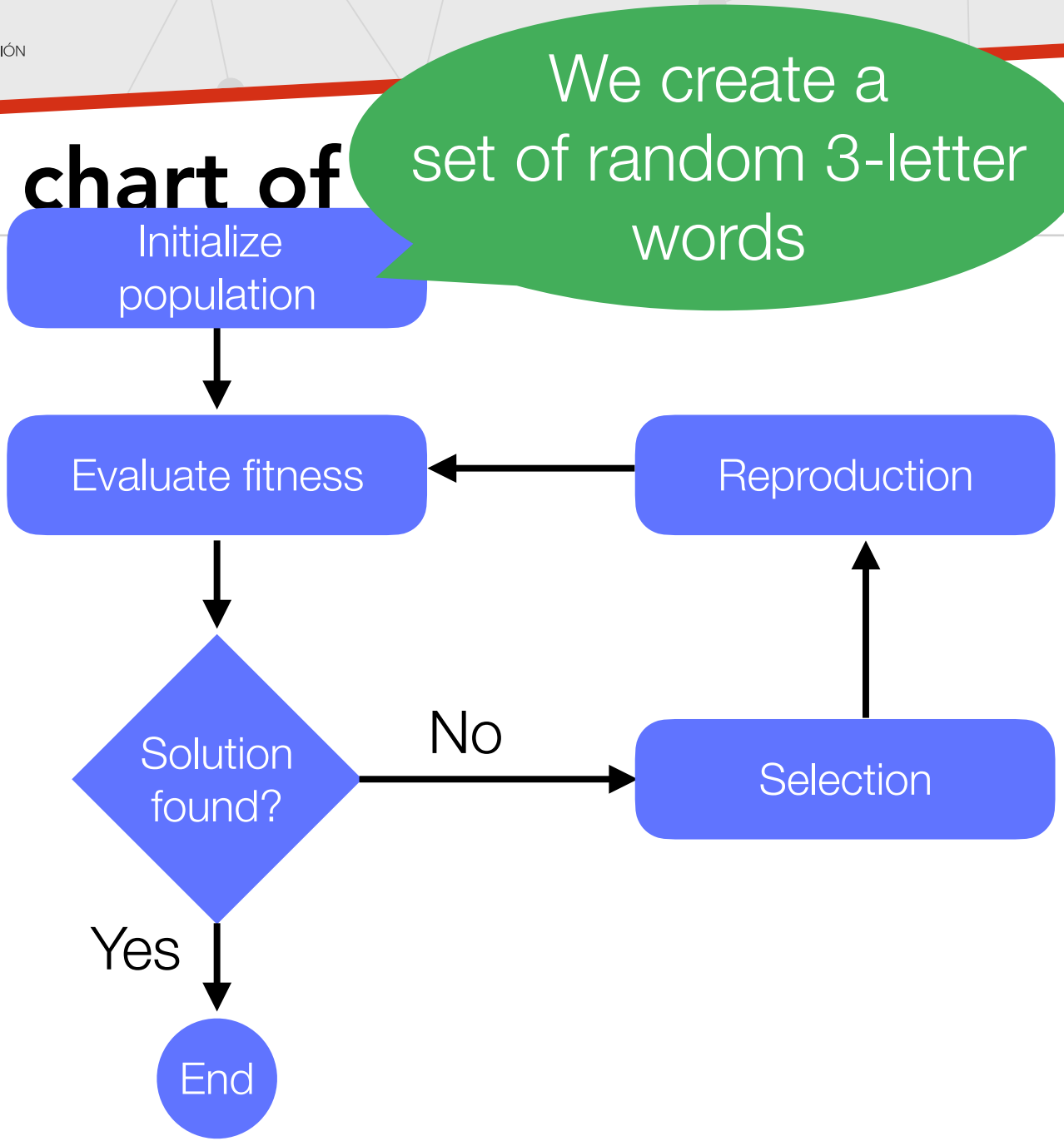


Many EAs are around:
Ant colony optimization,
Artificial immune system,
Cultural algorithms,
Genetic Algorithm,
Genetic Programming,
Grammatical evolution,
…

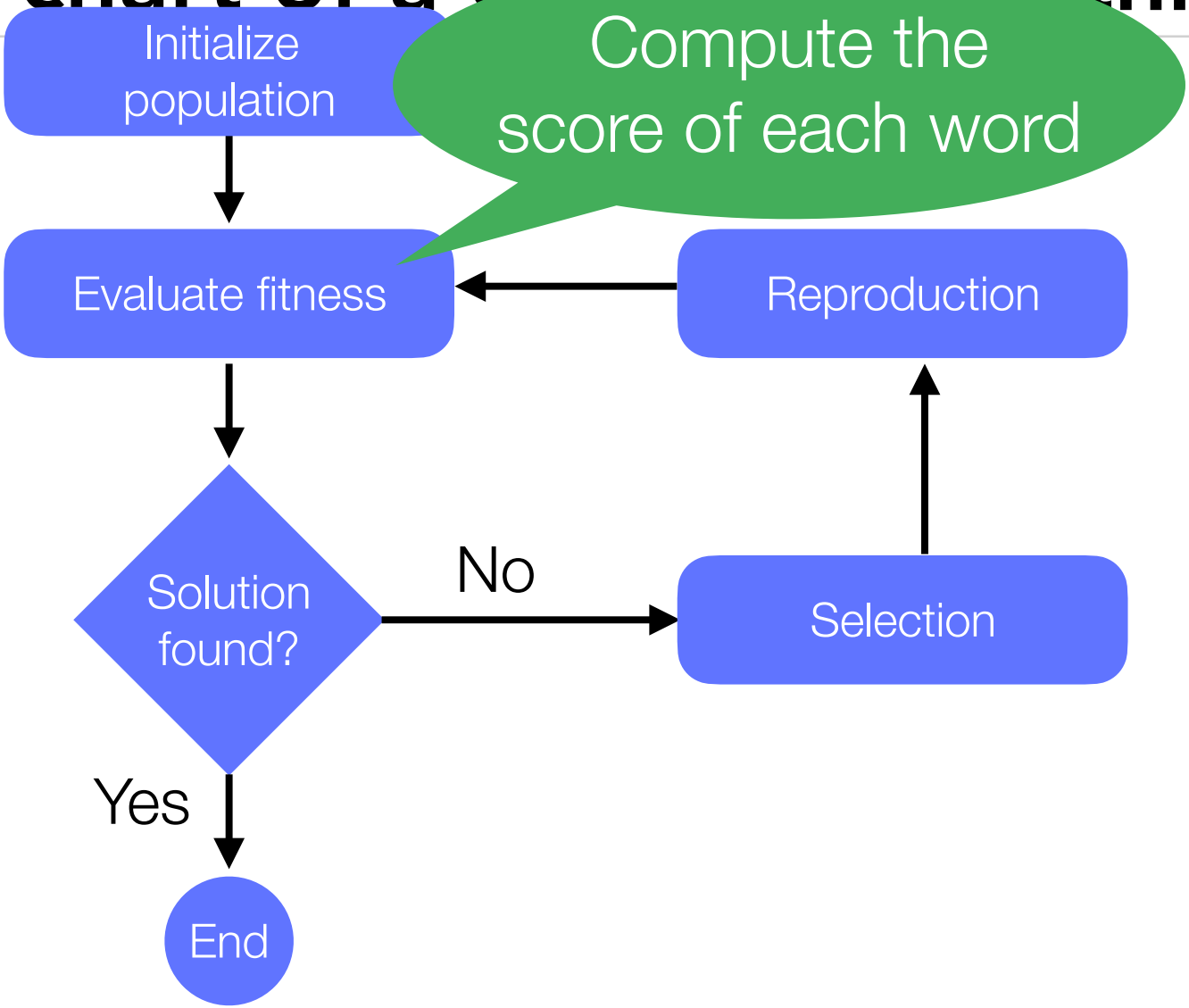
Flow chart of a genetic algorithm



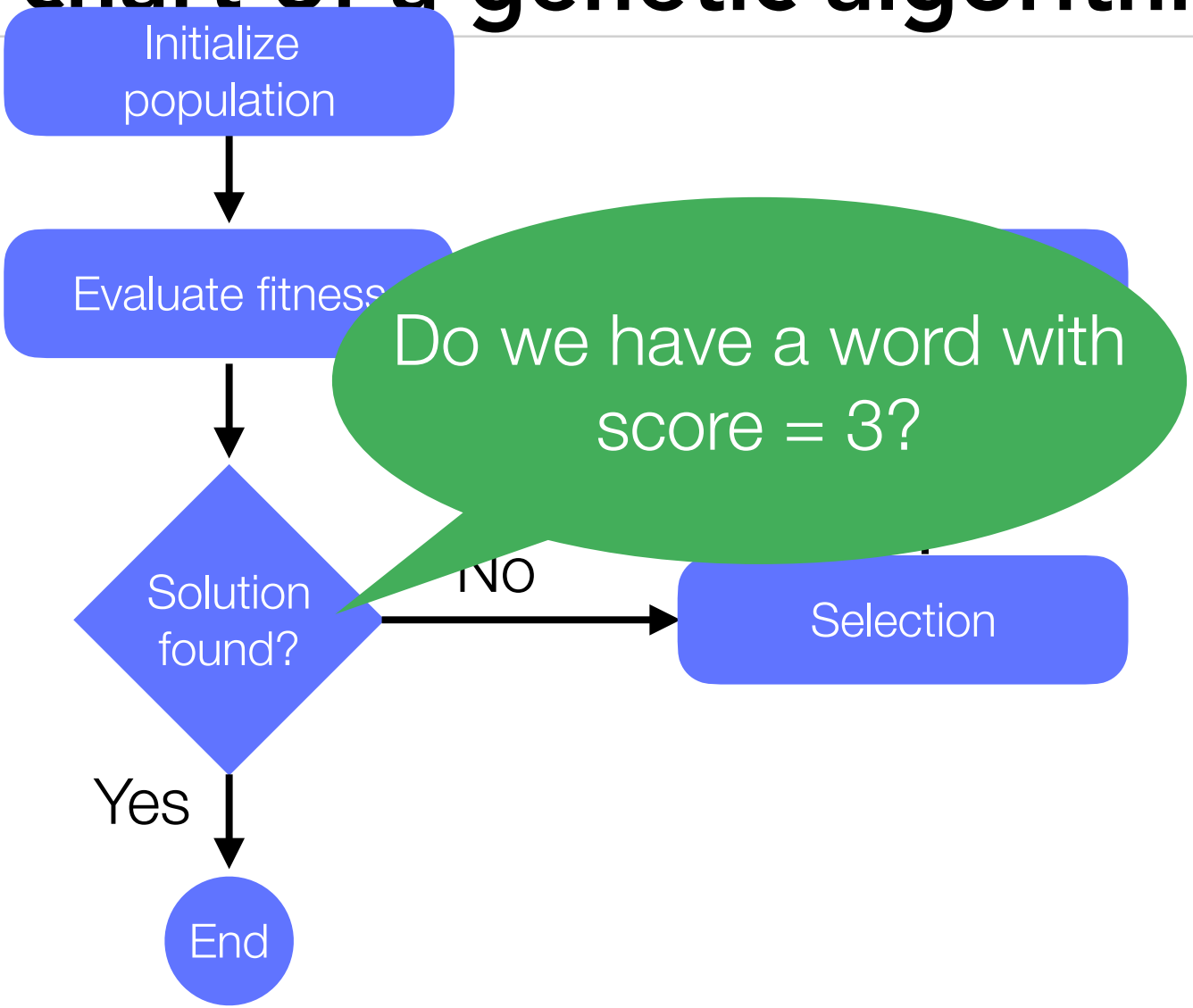
Flow chart of m



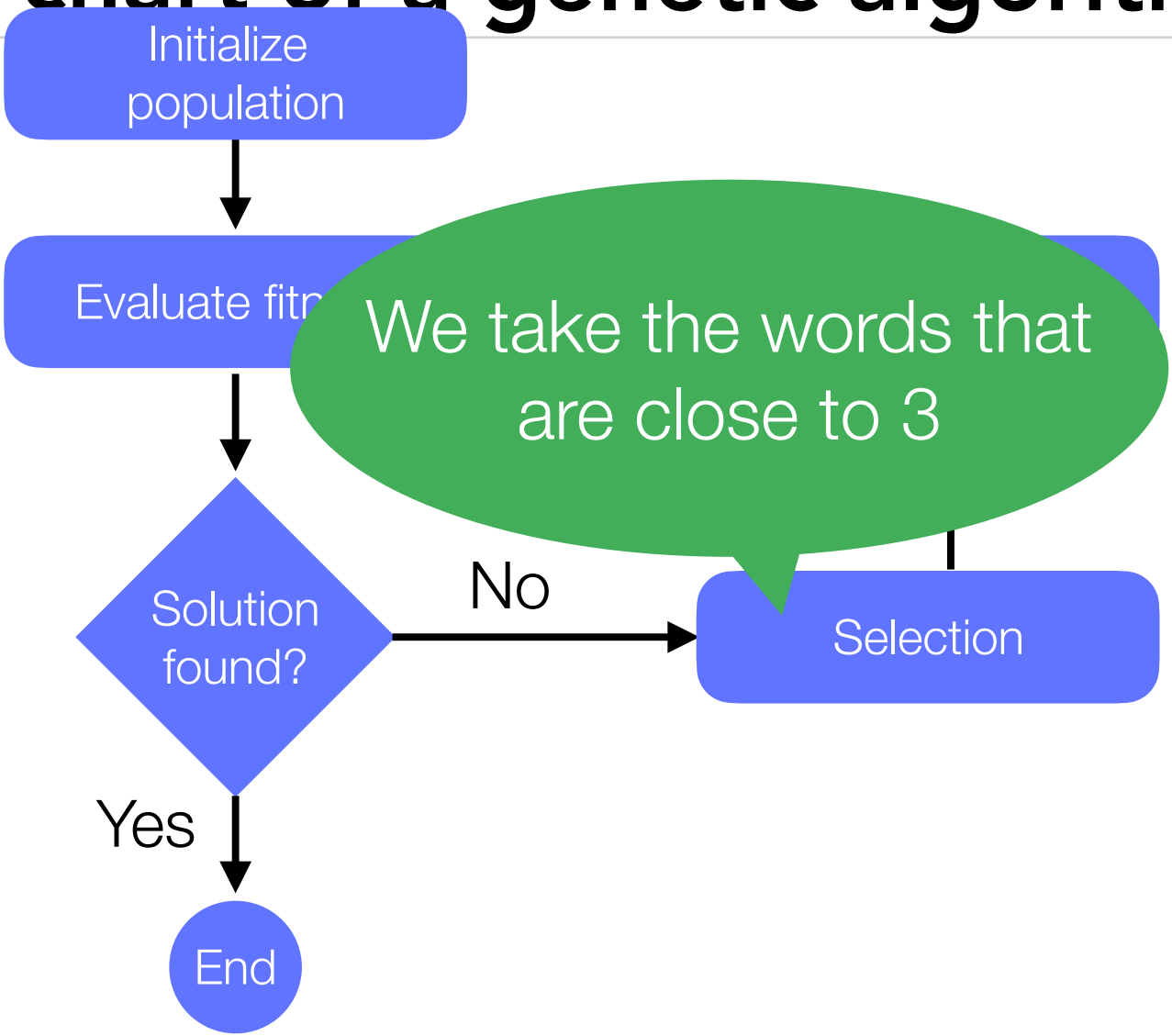
Flow chart of a genetic algorithm



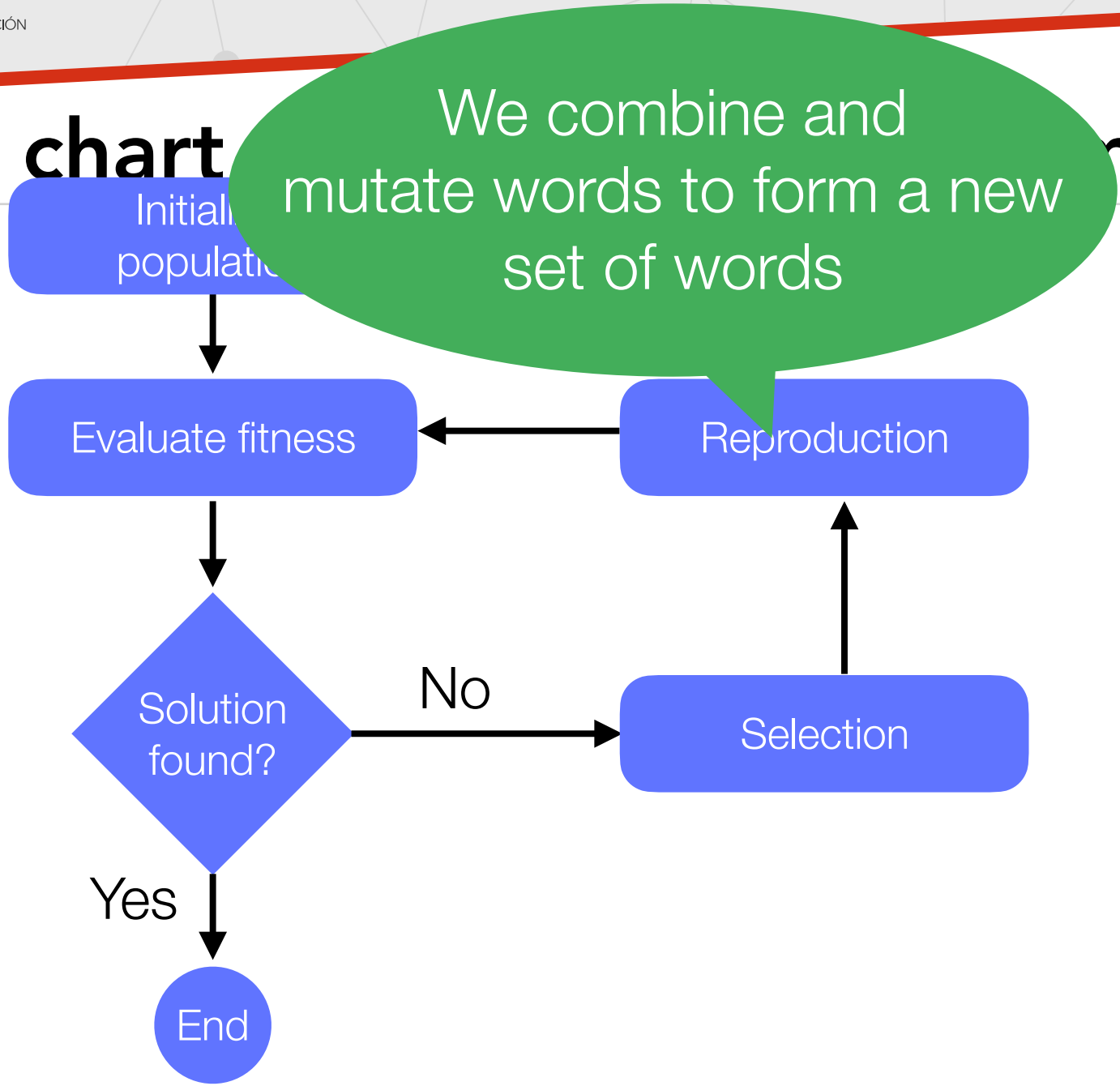
Flow chart of a genetic algorithm



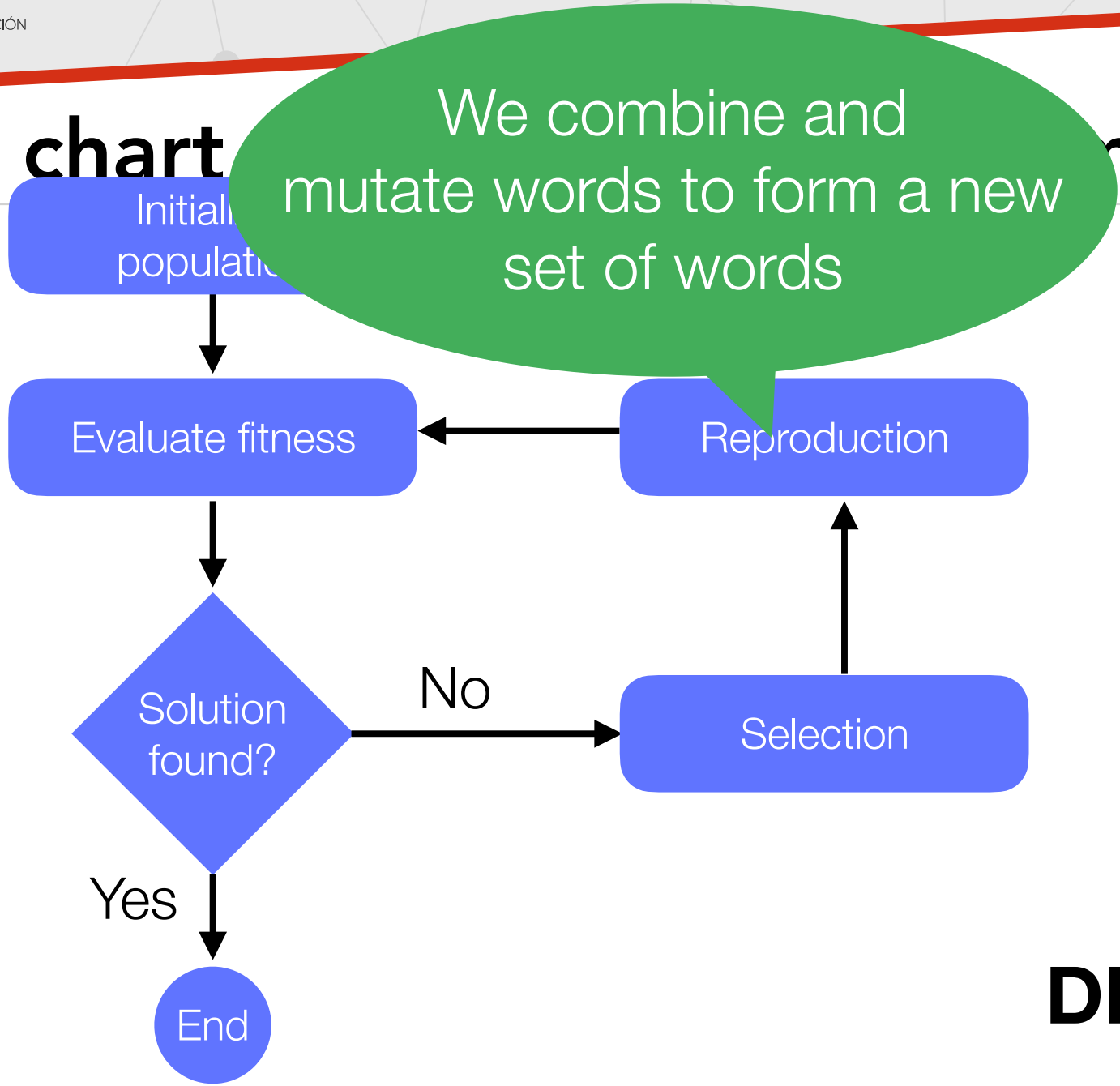
Flow chart of a genetic algorithm



Flow chart


We combine and mutate words to form a new set of words
Initial
Evaluate fitness
Reproduction
Solution found?
No
Selection
Yes
End

Flow chart



DEMO

Genetic Algorithm in a Nutshell

Evolutionary computation technique that automatically solves problems without specifying the form or structure of the solution in advance

Generally speaking, *genetic algorithms are simulations of evolution*, using biological genetic operations

Finding x such as $f(x) = y$ is maximal, x is a tuple of any arbitrary dimension and domain value, $y \in \mathbb{R}$

Terminology: Individual

An *individual* represents an element in a population

Each individual has a chromosome, composed of *genes*

Individual

Gene = index in a chromosome

Allele = value of a gene

Terminology: Population

A *population* is a set of individuals

Generally, the *population size is fixed* over time. Individuals are replaced at each generation, but the number of individuals remains constants.

All individuals of the population *have the same size*

Terminology: Fitness function

The fitness function evaluates how *fit* an individual is

$f($  $) = y \qquad y \in \mathbb{R}$

The whole idea of genetic algorithm is to search for the individual that maximizes the fitness function

Example: optimizing a server

Consider a server running in Java

The Java virtual machine, which has over 200 options

What are the options that consume the least amount of memory to answer HTTP requests

$$f(x) = y$$

x is a set of options, e.g., `[ -Xgc:parallel, -ms32m, -mx200m]`

y is the amount of memory (in bytes)

$f(x)$ is computed by launching the server using the options and sending 1000 requests

Example: optimizing a server

C

T 0 options

What a least amount of memory to answer

We look for the optimal x that minimize f(x)

$$f(x) = y$$

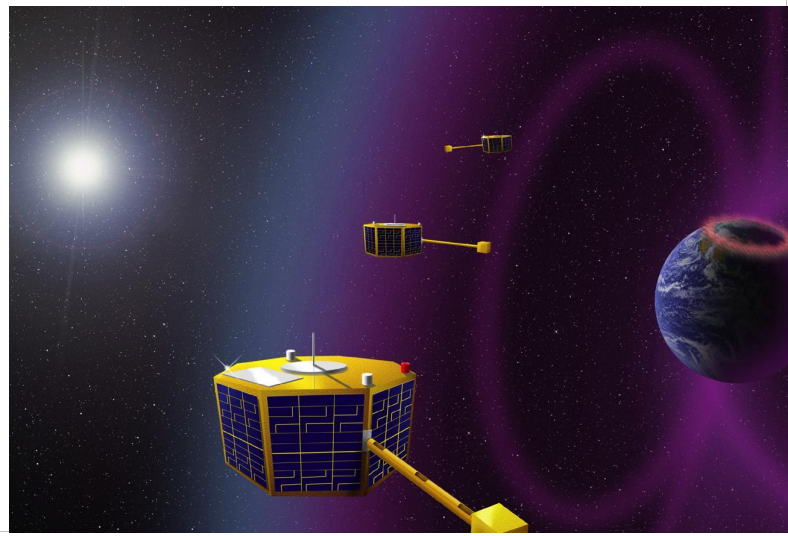
x is a set of options, e.g., `[ -Xgc:parallel, -ms32m, -mx200m]`

y is the amount of memory (in bytes)

$f(x)$ is computed by launching the server using the options and sending 1000 requests

Example: space antenna

Evolved antenna, produced by NASA in 2006.
Used in 3 satellites that take measurement Earth magnetosphere.
Satellites used this antenna to communicate with the ground



Example: space antenna

List of commands:
forward(length, radius)
rotate-x(angle)
rotate-y(angle)
rotate-z(angle)

$f(x) = y$

voltage and impedance for radio waves

Automated Antenna Design with Evolutionary Algorithms

Gregory S. Hornby* and Al Globus
University of California Santa Cruz, Mailtop 269-3, NASA Ames Research Center, Moffett Field, CA
Derek S. Linden
JEM Engineering, 8683 Cherry Lane, Laurel, Maryland 20707
Jason D. Lohn
NASA Ames Research Center, Mail Stop 269-1, Moffett Field, CA 94035

https://ti.arc.nasa.gov/m/pub-archive/1244h/1244%20(Hornby).pdf

Example: software testing

$$f(x) = y$$

Actions on the user interface (e.g., pressing a button, entering a value in the text field, clicking on a menu)

Number of tested functionalities

Randomly testing an application has many applications:

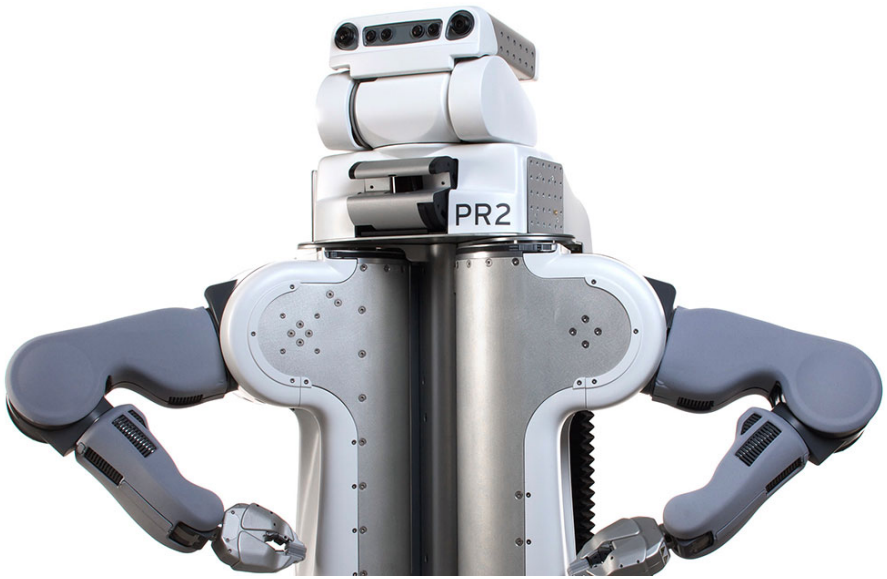
- Finding functional bugs (i.e., which actions crashes my application)
- Finding dead code (i.e., which part of my application is not used)

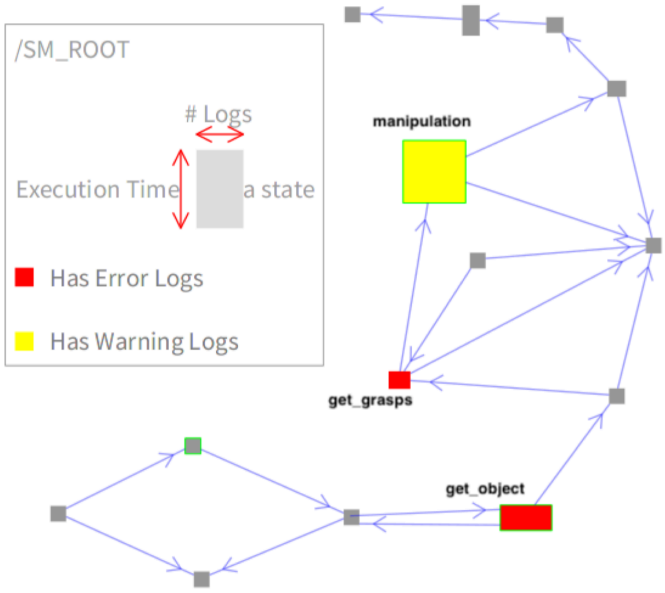
Example: robot testing

$$f(x) = y$$

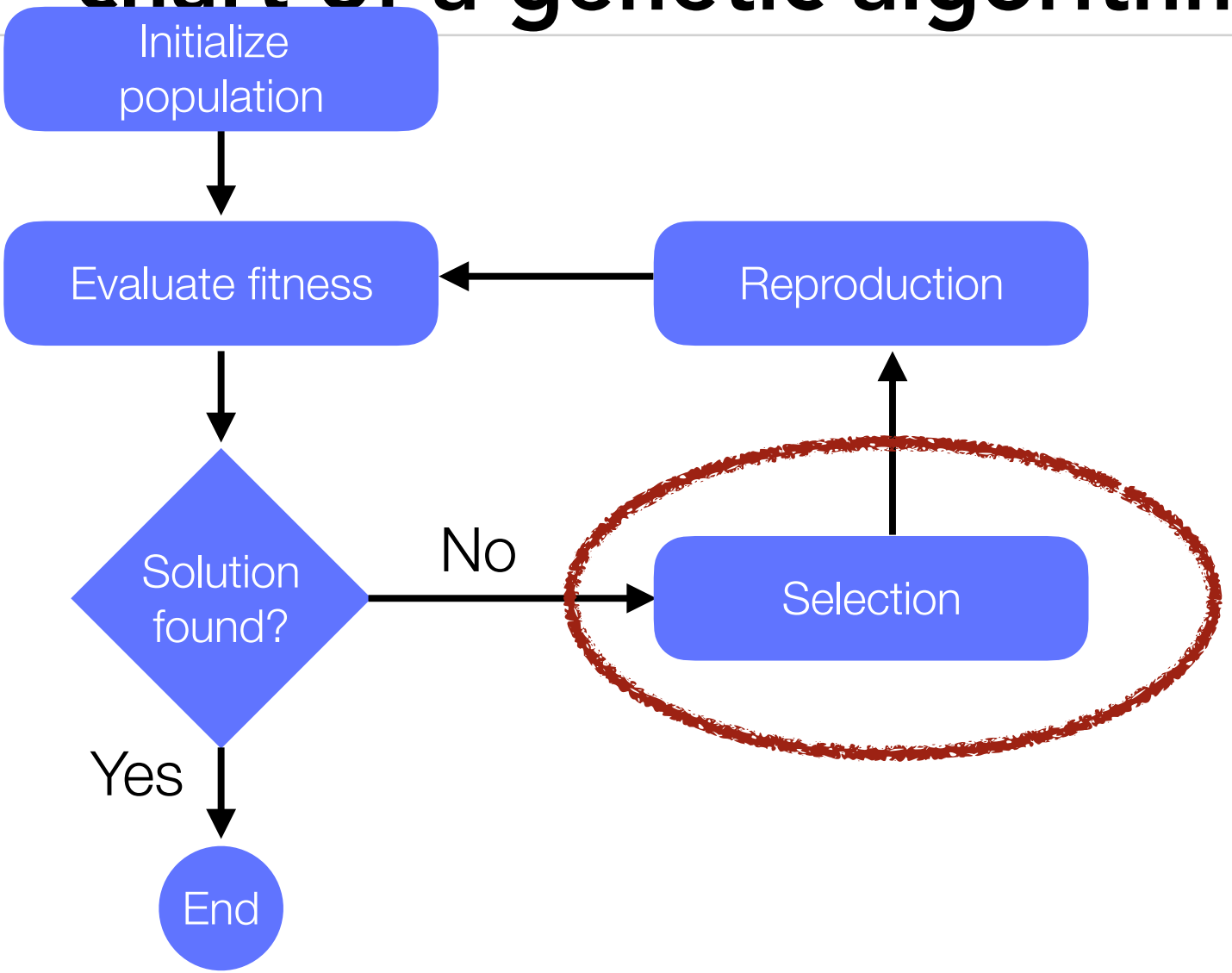
List of random inputs of a robot state

Number of raised warning/errors





Flow - chart of a genetic algorithm



Selection phase

Several selections algorithms exist

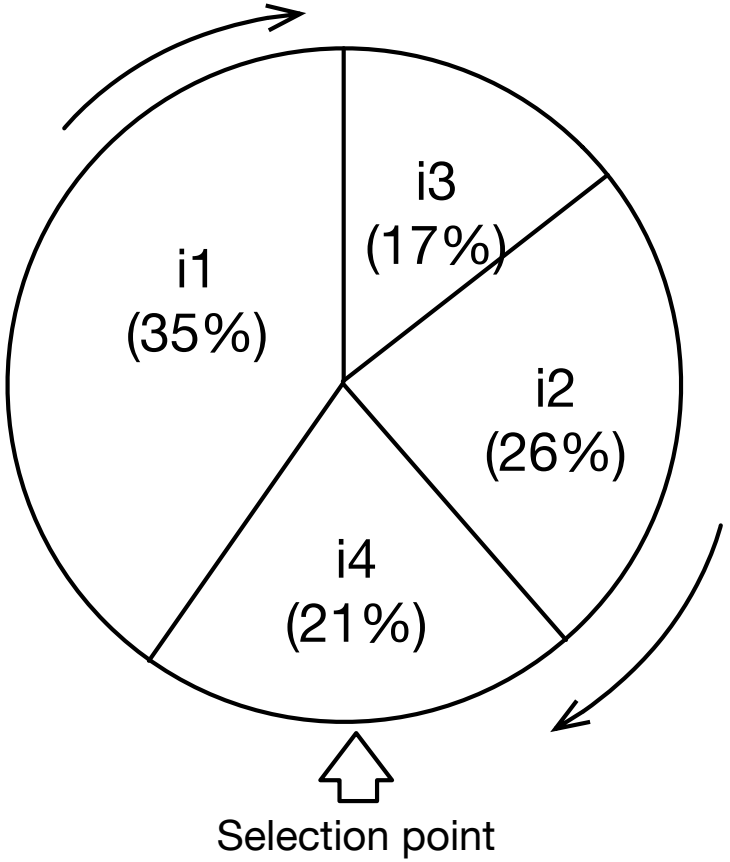
The two most popular are


Roulette associates a probability of selection to each individual


Tournament between a set of randomly picked individuals

Roulette selection

f(i1)=200
f(i2)=150
f(i3)=100
f(i4)=120



Roulette selection

If f_i is the fitness of an individual i, then its probability of being selected is $p_i = \frac{f_i}{\sum_j f_j}$

This can be simulated by the following (naive) algorithm:
1 - Calculate the sum of all fitnesses in population (sum S).
2 - Generate a random number r in the interval [0; S].
3 - Go through the population and sum fitnesses. When the sum s is greater than r, stop and return the individual where you are.

Tournament

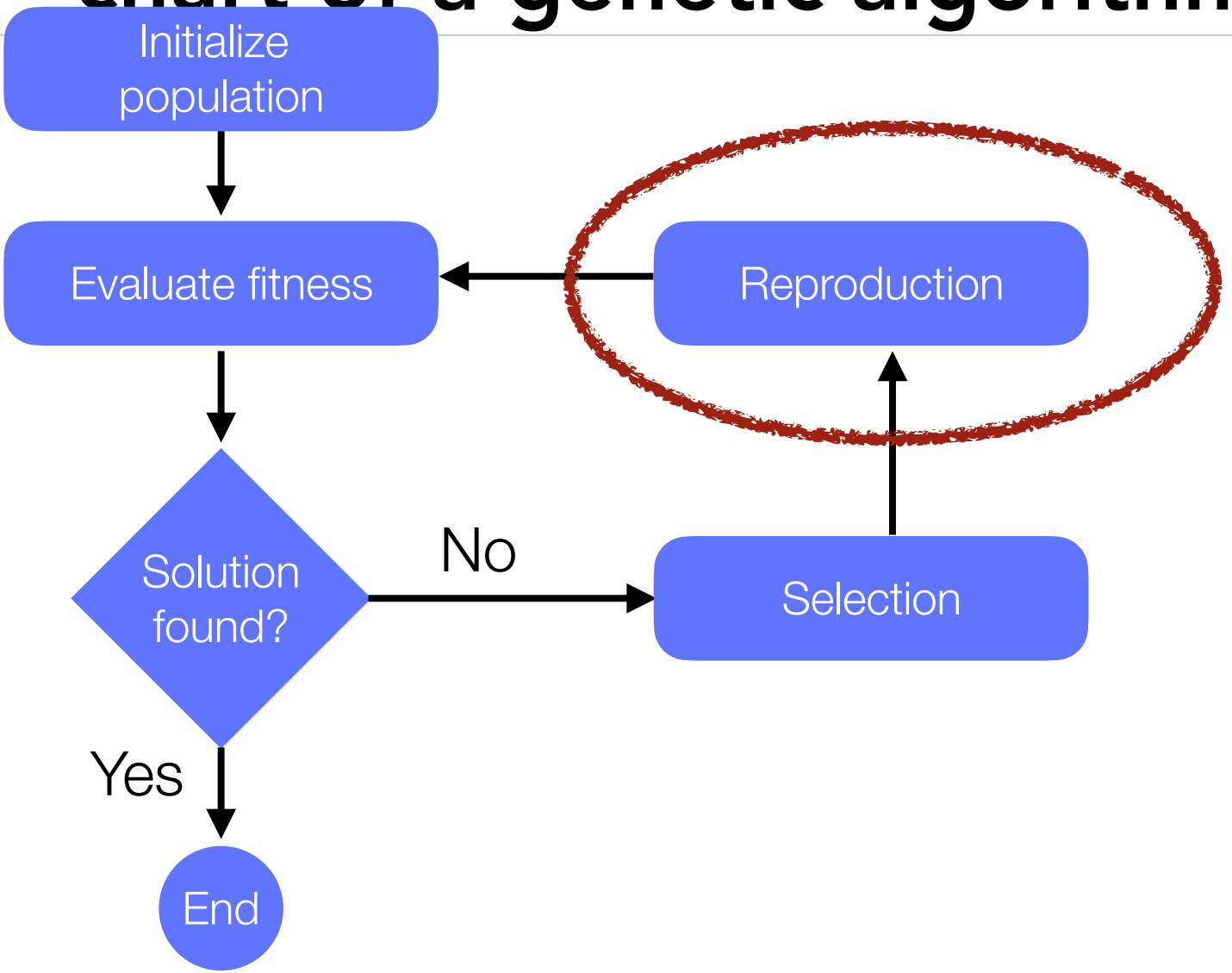
The tournament selection algorithm:

1 - Choose few individuals (e.g., 5) at random from the population (a tournament)

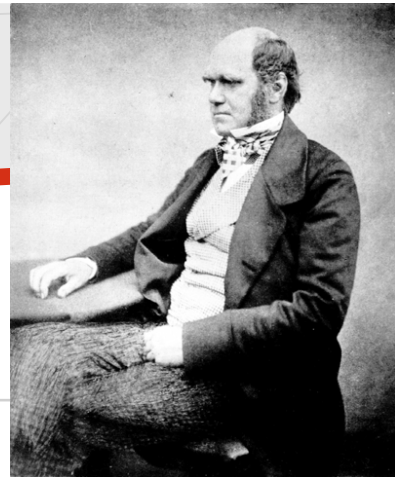
2 - The individual with the best fitness (the winner) is selected for crossover

The main advantage is to be highly parallelizable across multiple CPU Cores

Flow - chart of a genetic algorithm



Darwinian Natural Selection



In order to have a natural selection, we need to have:

Heredity: a child receives properties of its parents. In particular, if the parents are robust and can live long enough, then the child should too

Variation: some variation may be introduced in children. Children should not be identical copy of their parents

Selection: some members of a population must have the opportunity to be parents and have offsprings in order to pass their genetic information. Typically referred to as “survival of the fittest”

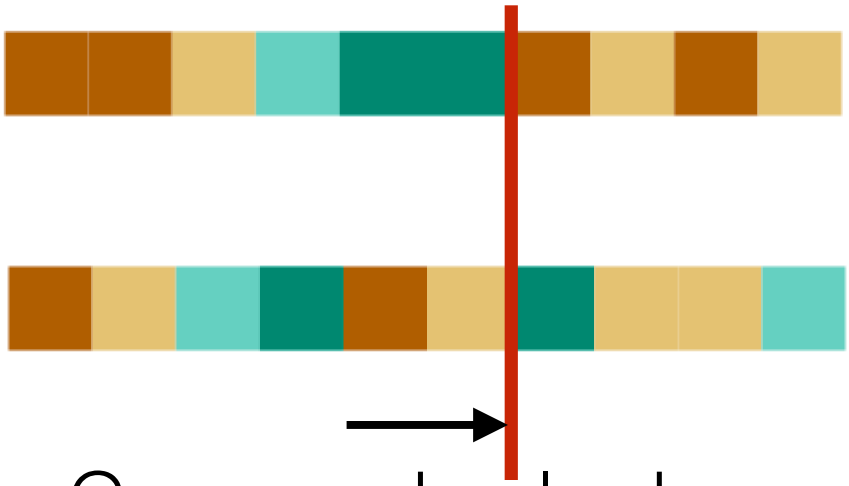
https://en.wikipedia.org/wiki/Charles_Darwin#/media/File:Charles_Darwin_seated_crop.jpg

Genetic operators: Crossover

Operations that takes two individuals and produces a new one, result of a combination

Genetic operators: Crossover

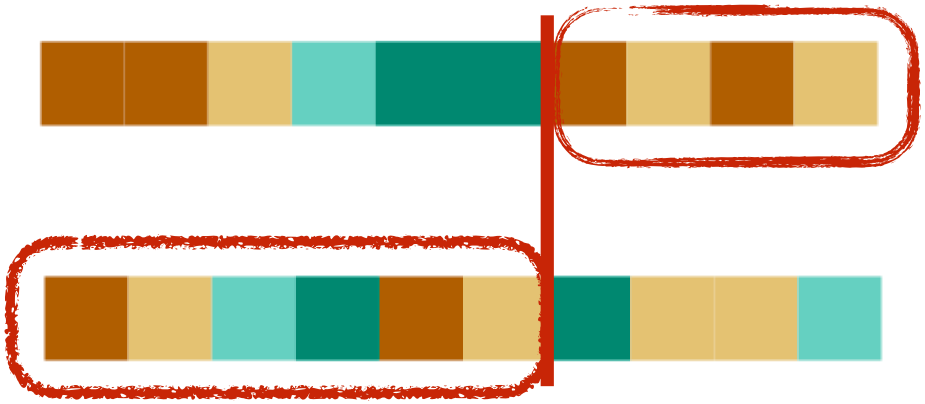
Operations that takes two individuals and produces a new one, result of a combination



Gene randomly chosen

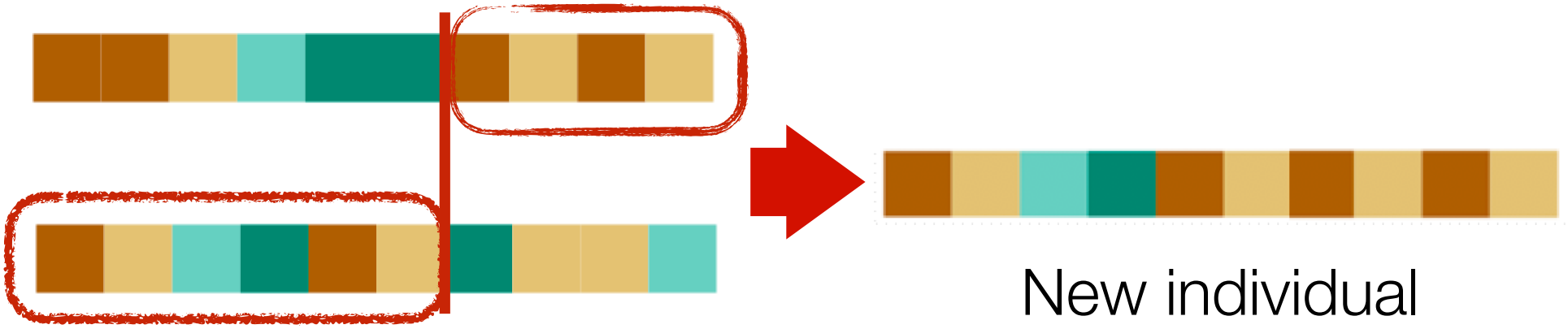
Genetic operators: Crossover

Operations that takes two individuals and produces a new one, result of a combination



Genetic operators: Crossover

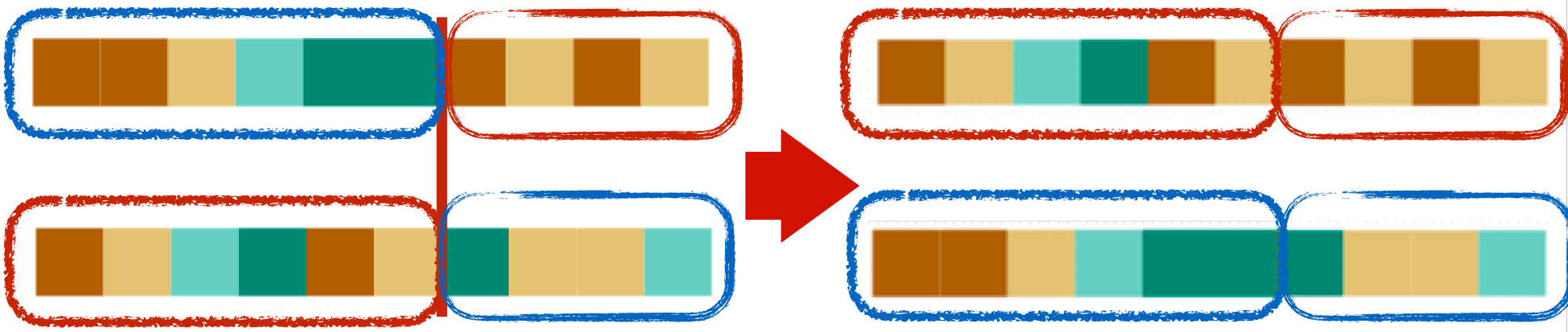
Operations that takes two individuals and produces a new one, result of a combination



New individual

Genetic operators: Variant of Crossover

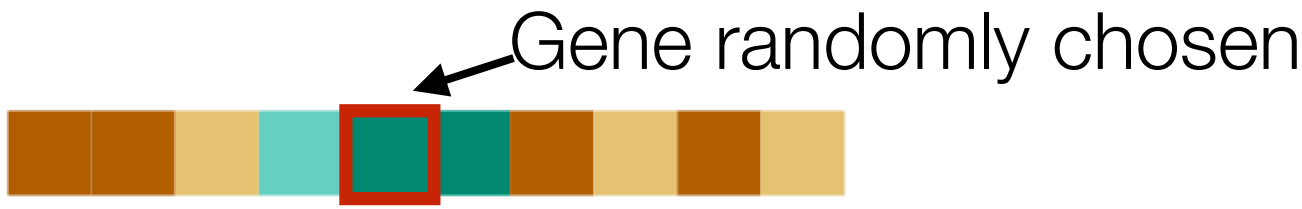
Operations that takes two individuals and produces a new one, result of a combination



Two new individuals

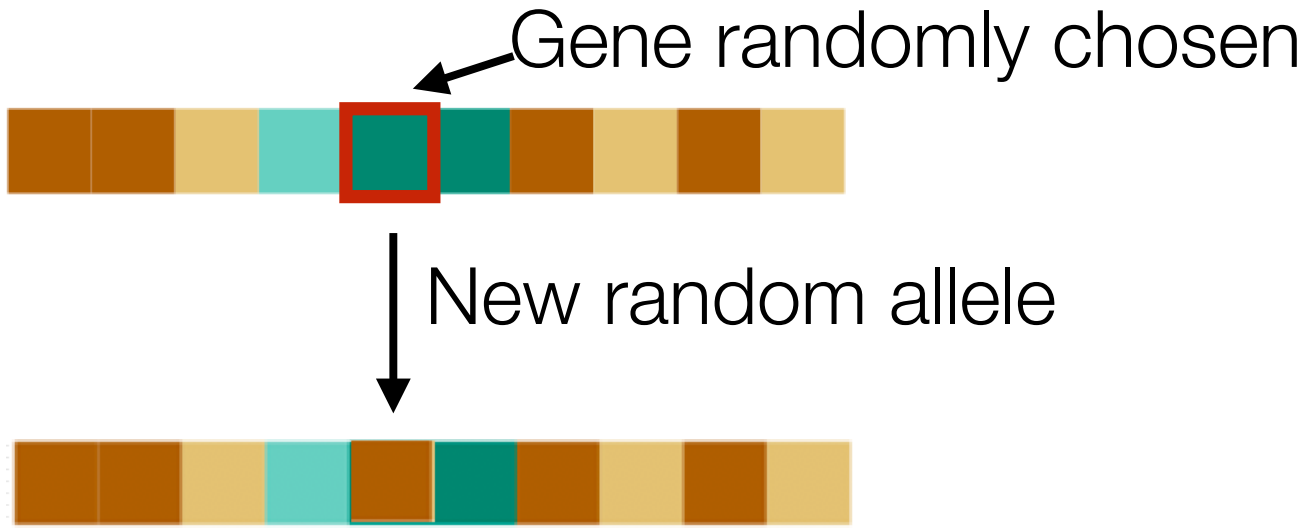
Genetic operators: Mutation

Operations that takes one individual and produces a new one, result of a mutation

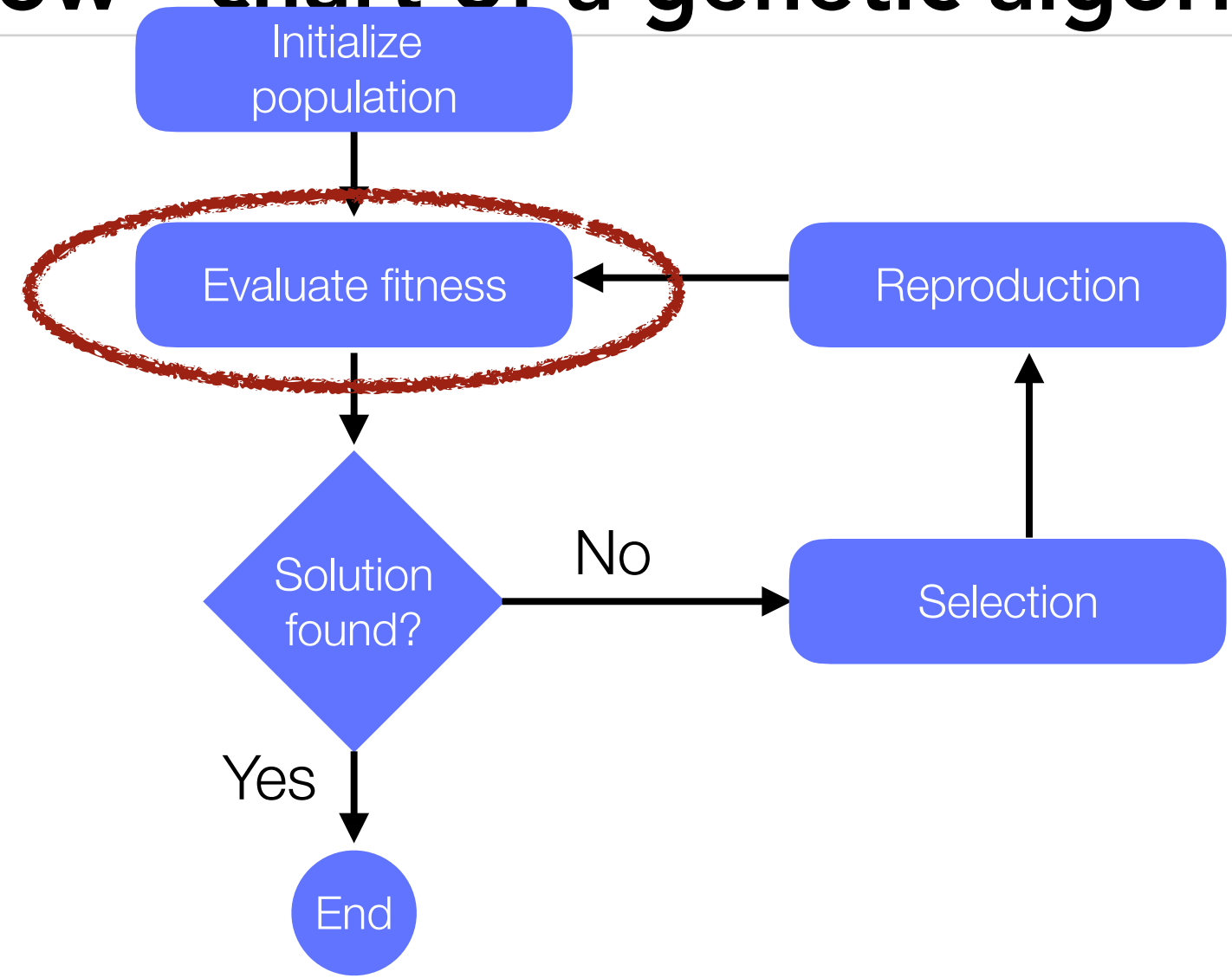


Genetic operators: Mutation

Operations that takes one individual and produces a new one, result of a mutation



Flow - chart of a genetic algorithm



Configuring the algorithm

Mutation rate: % to change a gene when creating a child.
E.g., a number between 0.00 and 0.20

Population size: number of individual to consider each time.
E.g., a number between 10 and 1000

Number of genes: how many genes contains each individual

Fitness function: Function that tells how good / far an individual is from the (ideal) solution

Some benefits of Genetic Algorithm

GA provides a compelling way to *not be trapped in local optima*

GA allows optimization of systems in which *variables may be discrete or categorical*, and not only continuous

e.g., direction of a robot or characterization of an antenna segment

GA can be combined with *other AI techniques*

Some benefits of Genetic Algorithm

GA provides a compelling way to *not be trapped in local optima*

GA allows optimization of systems in which *variables may be discrete or categorical*, and not only continuous

e.g., direction of a robot or characterization of an antenna segment

GA can be combined with *other AI techniques*

DEMO!

Artificial Intelligence Landscape

GA is often used to solve AI problems in terms of optimization

GA just need a way to say how good a solution is

Deep learning requires many examples

GA is easy to learn and implements

Exercise

1. Implement a genetic algorithm engine
    1. A new generation is built with: newIndividual = mutate(crossover(selectIndividual(), selectIndividual())
    2. Use the first version of the crossover (2 parents -> 1 child)
    3. Use the tournament selection algorithm
2. Use it to solve the word finding example
    1. Use it to find a simple word
    2. Use it to find a long sentence
3. Produce a hotmap to explore the performance space
    1. Use of Matrix of configuration (population size, mutation rate)
    2. Mesure the number of generations to find the solution (or the fitness after X generations)