Introduction to genetic algorithms

Insights for IA

Iván Andrés Trujillo Abella

Facultad de Ingeniería
Universidad de los andes

Why GA?

Numerical methods are based on a set of neccesary and sufficent conditions, and assess each point, genetic algorithm is a multipoint guided random search technique, we dont need additional information about the problem.

Quadratic function

$$f(x) = ax^2 + b^x + c \tag{1}$$

In a analytical way it is enough $f'(x^*) = 0$.

$$x^* = \frac{-b}{2a} \tag{2}$$

where x^* is the exact solution. Another way to find the solution is to use gradient descendent, that could climb the curve to reach the point.

Optimal ilustration

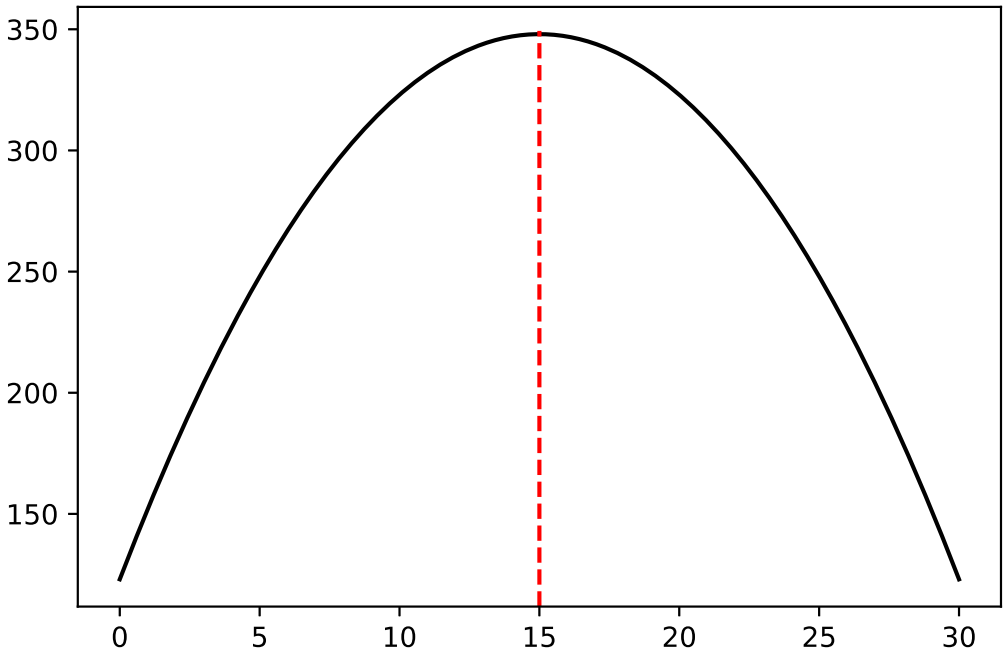


Figure: Quadratic function, $x^* = 15$

Two possible candidates

Swap information

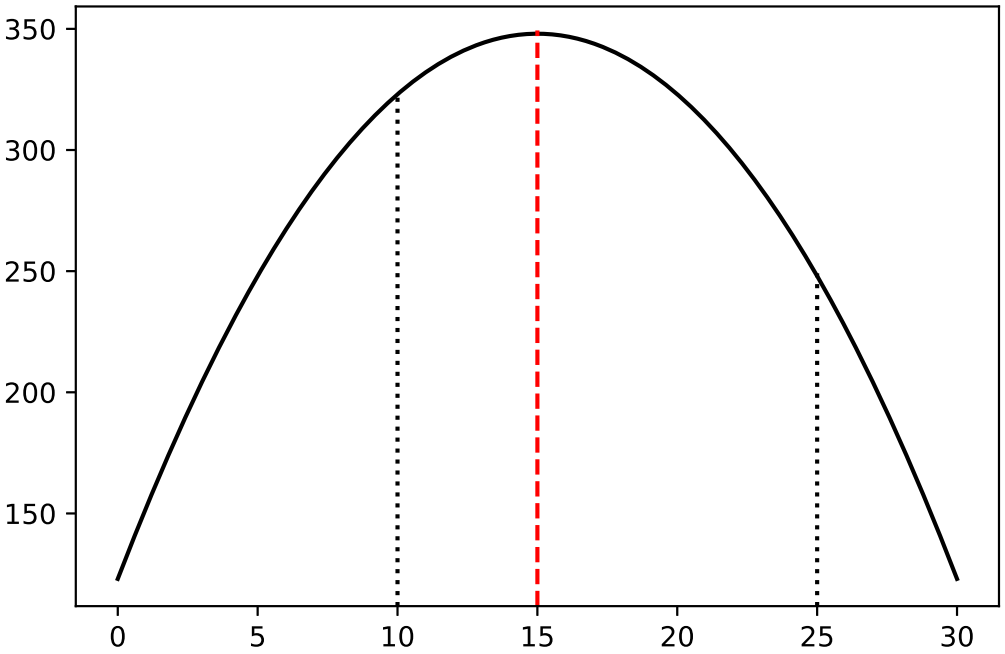


Figure: Quadratic function, $x^* = 15$

Swap information

We could represent the candidates in an iterable object and swap the digits. For instance:

```
x,y  = [1,0],[2,5]
x,y = swap(x,y) # Interchange the numbers
x,y = number(x,y) # pass the output arrays to real numbers
```

Now, we can transform them to assess the both points in the function:

$$f'(15) = 0, f'(20) \neq 0 \tag{3}$$

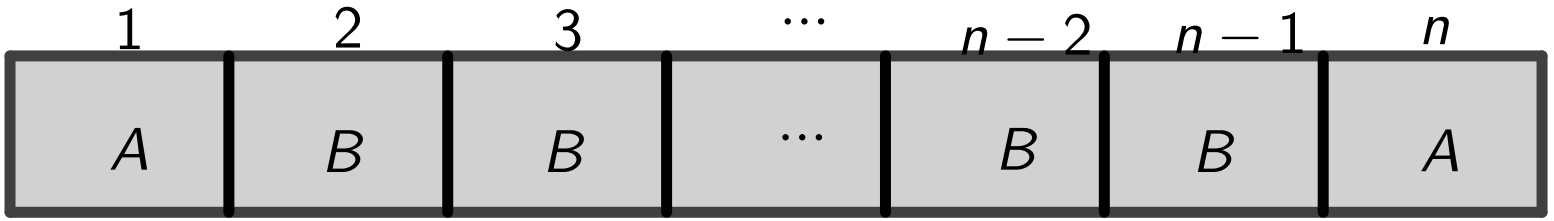
The solution was contained in the candidates!

Introduction to genenetic algorithm

A GA is a program, composed of strings that swap information!. A **Schema** theorem give us a cornerstone to understand how they works!

Chromosome, gen, allele and locus

Figure: Chromosome representation



The chromosome is genotype and the candidate solution is phenotype($x = 15$). the adjustment to the environment of x_i could be measure with the fitness function $f(x_i)$.

Insights

Individuals with better features have major probability of have offspring.

Swapping operators

The selection will provide those parents that will interchange information, with the crossover and mutation of the genetic material.

Genetic algorithm

Implementation

Algorithm 1: Canonical genetic algorithm

initialization of population;
while *not fill stoping criteria* **do**
    Select Parents ;
    Crossover pairs of parents ;
    Mutate Offspring ;
    Update population ;
end

Schema

Constructed over the alphabet $\{0, 1, *\}$ it a abstract representation of a chromosome. Where $*$ is a metasymbol(taking either 1 or 0).

Schema

If we have the following schema $***$ we could represent 2^3 possible instances, for example 111 or 100 and so on. 10101 could be represented by:

- 10101
- *0101
- 1*101
- **101
- 1**01
- ***01
- *****

in a total of $\sum_{i=0}^{k} \binom{n}{k} = 2^n$ where k is the number of wildcars.

Schema properties

- Order: $o(H)$ the number of fixed position; or the length of the schema minus the number of $*'s$.
- Length: $l(H)$ is the number of bits that composed H.
- Defining legth: $\delta(H)$ The distance among the first and last fixed position.

Inheritance

Features that remain of parents to offsprings.

Selection

Proportional selection

Selection determines what parents will be used to create offspring. Select possible chromosomes as parents, according its fitness.

$$P(x_i) = \frac{f(x_i)}{\sum_j^n f(x_j)} \tag{4}$$

The times that an individual participate in crossover and mutation it is proportional to the fitness of the individuals.

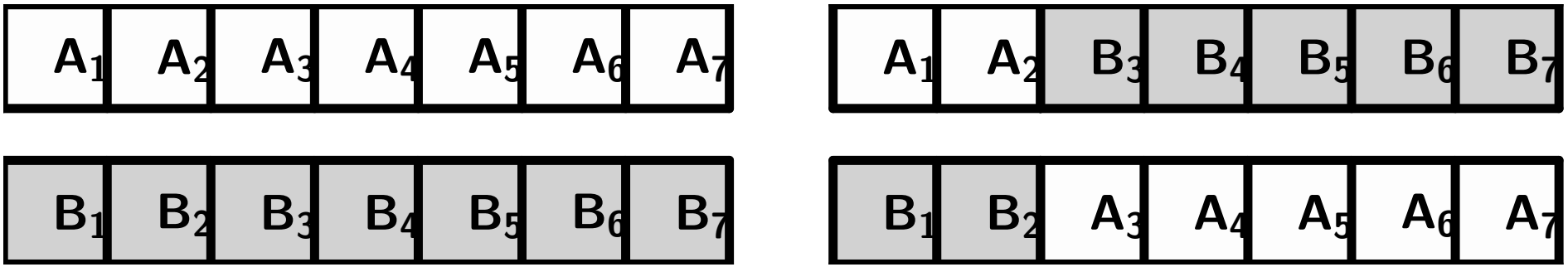
Mutation and crossover

The crossover and mutation are operators of the evolutionary process, in the search process, one is exploration and the another explotation of search space. Mutation play a key in reaching the global optima.

Crossover

fidex one point

Figure: Crossover in single point (2-index)

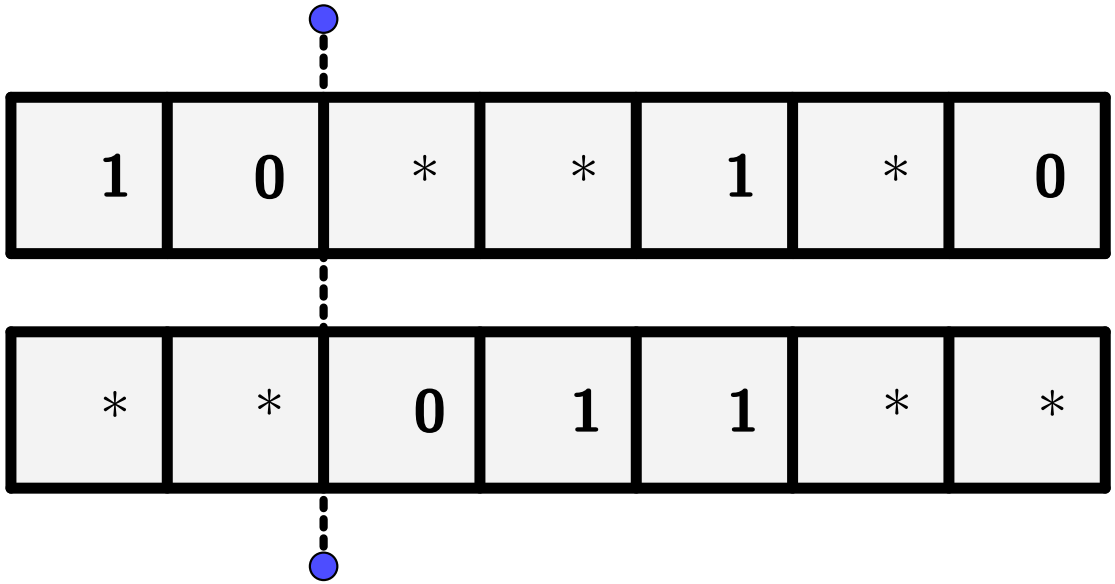


The offspring is generated choosing a random locus in two parents, after concatenate α_a and α_b if the locus selected is equal to π then $\alpha_a[:\pi]$ concatenated with $\alpha_b[\pi:]$.

Crossover

Vulneravility-compactibility

Figure: Crossover in single point (2-index)



When crossover occur within defining length, the schema have a lower probability of survive, quantifying $\frac{\delta(H)}{l(H)-1}$.

Crossover-inheritance

We can define the probability of that the crossover occur among two parents P_c, and the probability of a schema will be destroyed as p_d (related with the vulnerability) We can define the probability of destruction of a schema by $P_c * P_{dl}$

$$Ps = 1 - P_c \frac{\delta(H)}{l(H) - 1} \tag{5}$$

Therefore the probability that a schema survive will be $1 - Pd$.

Mutation-inheritance

All bits in a string have P_{mu} of undergoes a mutation, therefore the probabiltiy of not mutate is $1 - P_{mu}$ then for the number of fixed bits we have $o(H)$ that the probablity of complete schema survival to mutation will be

$$P_m = (1 - P_{mu})^{o(H)} \tag{6}$$

Crossover-Mutation-Inheritance

The probability of a schema survive to crossover and mutation will be:

$$P_{sur} = [1 - p_c \frac{\delta(H)}{L(H) - 1}](1 - p_{mu})^{o(H)} \tag{7}$$

Schema theorem

Inheritance in generations

First we need take in mind the settings (assumptions):

- Binary codification
- $N_{t-1} = 0$ (There are not parents in the next generation)
- Roulette wheel (proportional selection)
- Crossover in a single point
- Uniform mutation

Schema theorem

Population(t)	Fitness f(x)
$x_1 = 1010$	$f(x_1)$
$x_2 = 0101$	$f(x_2)$
$\vdots$	$\vdots$
$x_n = 1110$	$f(x_n)$
	$\sum_i^n f(x_i) = F(t)$

We can ranking that $\frac{f(x_i)}{F(t)} > ... > \frac{f(x_j)}{F(t)}$ and select pairs to crossover.

Schema theorem

$m(H, t)$ number of strings that match (instances) with H schema in t generation.

Chromosome	**Schema (H = 1*01)**	$f(H, t)$
x1 = 1011	0	-
x2 = 1001	1	$f(x_2)$
x3 = 0011	0	-
x4 = 1101	1	$f(x_4)$
	$m(H, t) = 2$	$\frac{f(x_2)+f(x_4)}{2}$

Schema theorem

Goal

Determine the **probability** that H survive in the evolutionary process.
The average probability of individuals that $Match(H, x_i) = 1$ be selected in the t-generation to crossover is :

$$\frac{f(H,t)}{F(t)} \tag{8}$$

Schema theorem

If we have n individuals in the population, the probability of my first parent will be a *match* (instance) will be $m(H,t)\frac{f(H,t)}{F(t)}$ and therefore for n parents we have:

$$m(H,t)\frac{f(H,t)}{F(t)}(n) \tag{9}$$

Namely, the last equation output the number of instances of H that will be selected.

Schema theorem

In the next generation:

$$m(H, t+1) = m(H, t)\frac{f(H, t)}{F(t)}(n) \tag{10}$$

The avarage fitness of the population will be $\bar{F}(t) = \frac{F(t)}{n}$ retwriting we have:

$$m(H, t+1) = m(H, t)\frac{f(H, t)}{\bar{F}(t)} \tag{11}$$

This means that schemata with higher fitness of average tend to be selected more.

Schema theorem

Lower-bound

Why is a lower bound?

$$E[m(H,t+1)] \geq m(H,t)\frac{f(H,t)}{\bar{F}(t)}[1-p_c\frac{\delta(H)}{L(H)-1}](1-p_{mu})^{o(H)} \quad (12)$$

Building Block Hypothesis

BBH

Now define the rate of growth as $\gamma = \frac{f(H,t)}{\bar{F}(t)}[1 - p_c \frac{\delta(H)}{L(H)-1}](1 - p_{mu})^{o(H)}$.

$$E(m(H, t+1)) \geq m(H, t)\gamma \tag{13}$$

if $\gamma \geq 1$ then H is a **Building block**.

Exponential growht of schemata

$$\frac{dm(H,t)}{dt} = \gamma m(H,t)$$
$$\frac{1}{m(H,t)} dm(H,t) = \gamma dt \tag{14}$$
$$\int_0^t \frac{1}{m(H,t)} dm(H,t) = \int_0^t \gamma dt$$

Remember that $\frac{d \ln(u)}{du} = \frac{1}{u}$, also that that $\gamma t - \gamma 0 = \gamma t$.

$$\ln m(H,t) - \ln m(H,0) = \gamma t$$

(15)

Exponential growht of schemata

According to the last expression we have;

$$\ln(\frac{m(H,t)}{m(H,0)}) = \gamma t \tag{16}$$

Now we can see that:

$$m(H,t) = e^{\gamma t} m(H,0) \tag{17}$$

The number of **instances** of H increase exponentially. According to **BBH** GA's search the optimal solution with building blocks and its crossover and mutation.

Deception

BBH could be loose the optimal soultion!, rejecting solutions that seems not good, the increase of homogenity of solution, could lead to lose important genetic material.

Considerations

In the following slides we try to summarize the details of implementations.

Selection

Roulette wheel bias

The main problem with proportional selection is the sensitivity to outliers values. Could be exist a bias due a higher fitness value in the population, that lead to premature convergence. Also, with increase of generations, could be appear a loss of discrimination among alternatives with $f()$.

Wheel selection

offspring

The probability of (x_i) will be selected in the generation is $\frac{f(x_i)}{\sum f(x_j)}$ in n trials then its number of descedents is;

$$n\frac{f(x_i)}{\sum f(x_j)} = \frac{f(x_i)}{\bar{f}}. \tag{18}$$

Note that equation not imply integers numbers, we can uses a wheel to solve the problem.

$$2\pi r \tag{19}$$

Now we can reeplace r with the proportion of wheel to each chromosome.

$$r = \frac{f(x_i)}{\bar{f}}\frac{1}{n} \tag{20}$$

Roulette Wheel

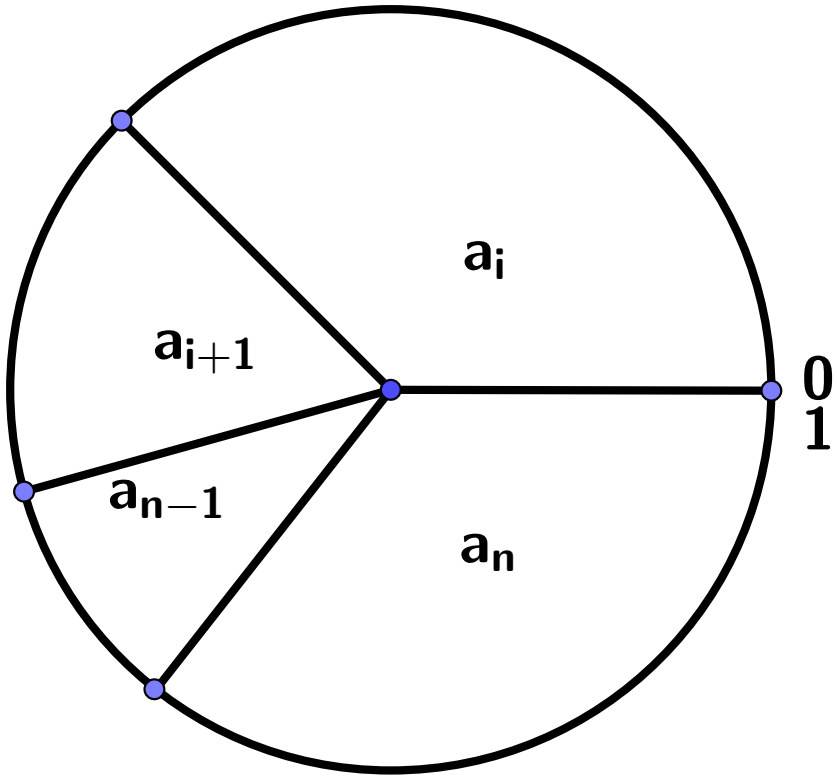
Implementation

Take in mind that $\sum \frac{f(x_i)}{\bar{f}} = n$. Now that each individual have a proportion of the wheel$(2\pi \frac{f(x_i)}{\bar{f}n})$ we only need create a vector of random numbers among $[0, 2\pi]$ and pick the number inside the interval or we could dividive each result by 2π to standardize and generate random numbers among $[0, 1]$.

Wheel

Individual	Wheel proportion pw(x)	R∼[0,1]
x_i	a_i	$[0, a_i]$
x_{i+1}	a_{i+1}	$(a_i, a_{i+1}]$
x_{n-1}	a_{n-1}	$(a_{i+1}, a_{n-1}]$
x_n	a_n	$(a_{i+1}, 1]$
$\sum a_j = 1$		

Note that R is a uniform number over the interval $[0, 1]$.



Tournament selection

In a population of n individuals; $x_1, x_2, x_3, ..., x_n$. We sample pairs of indivudals.

Sampled	**Fitness**	**Winner**
(x_3, x_2)	$f(x_3) > f(x_2)$	x_3
(x_1, x_1)	$f(x_1) = f(x_1)$	x_1
(x_5, x_6)	$f(x_5) < f(x_6)$	x_6
(x_n, x_8)	$f(x_n) < f(x_8)$	x_8
		(x_3, x_1, x_6, x_8)

Take in mind, the individuals that *Not being sampled*$(x_j : j \neq 3, 2, 1, 5, 6, 8, n)$, and *Not being selected*(x_2, x_5, x_n).

Loss of diversity

Motoki (2002)

Selection schemes must tackle loss of diversity,**not being sampled** and **not being selected** introduce the bias in tournament selection.

$$Loss(t, n) = \frac{1}{n}\sum_{k=1}^{N}(1 - \frac{k^t - (k-1)^t}{n^t})^n \tag{21}$$

where n is population size, and k is the tournament size, Where k increase the loss of diversity also increase.

Selection

Tournament selection

Given that is a random process, the selection of the better individuals not is guaranteed.
The size of the tournament(k) in each tournament will be select the individual with major fitness, to crossover. When k tournament size increase the probabilty of the inidividuals with less fitness is low.

Crossover

The crossover not neccesarily is applied to all individuals we select a number $p_c \in [0.26, 0.9]$ after we draw a random number r among $[0, 1]$ if $r < p_c$ then crossover is applied to a pair of parents.
The following are popular Crossover operators:

- Fixed point
- Multiple points (Two or more)
- PMX
- ...

Mutation

Mutation could be computationaly expensive; because there is the probability of mutate a string and by string assess bit by bit if exist a mutation (a lot of if conditionals), a practical approach is:

$$E(M) = I(x)p_c n \tag{22}$$

and after select a random number in the $[0, I(x)n]$.
In real codification we need another operator of mutation for instance increase of a amount $N \sim (0,\sigma)$.

Implicit parallelism

Process 3^n schemate, with only n as input.

Stopping criteria

- Overcome the number of generations
- time of execution
- reach a minimal value of fitness
- homegenity in solutions

Stopping critaria is a challange!

Genetic algorithms in ML

Here there are important remkar are used to avoid greedy algorithms for instance;

- Feature selection
- Image processing
- Neural networks

Popular industrial implementations in financial sector:

- Decision trees
- K-means
- Tunning hyperparamters and topoly design in Neural Networks.

References

- Blickle, T., Thiele, L. (1995, July). A Mathematical Analysis of Tournament Selection. In ICGA (Vol. 95, pp. 9-15).
- Motoki, T. (2002). Calculating the expected loss of diversity of selection schemes. Evolutionary Computation, 10(4), 397-422.
- Bhandari, D., Murthy, C. A., Pal, S. K. (2012). Variance as a stopping criterion for genetic algorithms with elitist model. Fundamenta Informaticae, 120(2), 145-164.
- Sastry, K., Goldberg, D., Kendall, G. (2005). Genetic algorithms. In Search methodologies (pp. 97-125). Springer, Boston, MA.