Introduction to genetic algorithms

Insights for IA

TEAM 2

Facultad de Ingenieria
Pontificia Universidad Javeriana

Background

Why GA?

Asses point to point, genetic algorithm is parralel, we dont need additional information about the problem.

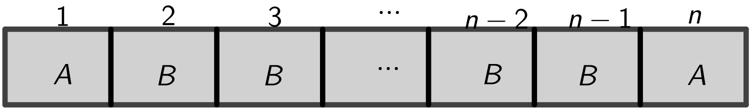
Quadratic Optimization

Gradient descendent

The process could be enlightened by extension.

Chromosome, gen, allele and locus

Figure: Chromosome representation



Codification

Binary, real...

Insights

Individuals with better features have major probability of have offspring.

Introduction to genenetic algorithm

A GA is a program, composed of strings that swap information!.

Swaping operators

The selection provided the parents that will be interchange information, with the crossover and mutation operator.

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 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.

Lingo

- 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 leght: $\delta(H)$ The distance among the first and last fixed position.

Inheritance

Features that remain of parents to offsprings.

Selection

Proportional selection

Select possible chromosomes as parents, according its fitness.

$$P_{x_i} = \frac{f_i(t)}{\sum_j^n f_j(t)} \tag{1}$$

after of get we can choose a random number among 0 and the sum of total fitness in population, this strategy is denominated wheel strategy.

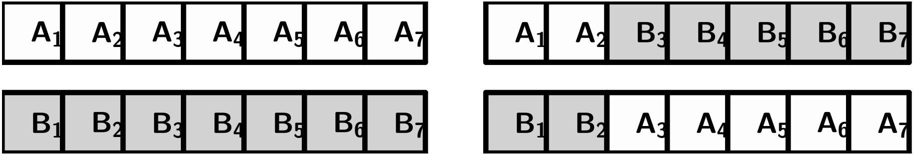
Mutation and crossover

The crossover and mutation as operators of the evolutionary process, in the search process, one is exploration and the another explotation of search space.

Crossover

fidex one point

Figure: Crossover in single point (2-index)



the offspring born choosing a random locus, in two parents, and after concatenate α_a and α_b and the locus is equal to π then the offspring will be $\alpha_a[:\pi]$ concatenate with $\alpha_b[\pi:]$.

Crossover

Inheritance-compactibility

When we consider a schema H

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 that the probability of a schem will be destroyed by $P_c * P_{dl}$

$$Ps = (1 - P_c)\frac{\delta(H)}{l(H) - 1} \tag{2}$$

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

$$P_m = (1 - P_{mu})^{o(H)} \tag{3}$$

Crossover-Mutation-Inheritance

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

$$P_{sur} = [(1 - p_c)\frac{\delta(H)}{L(H) - 1}](1 - p_{mu})^{o(H)} \tag{4}$$

Schema theorem

Inheritance in generations

First we need take in mind the settings:

- Binary codification
- $N_{t-1} = 0$ (There are not parents in the next generation)
- Roulete 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 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 :

$$\frac{f(H,t)}{F(t)} \tag{5}$$

Schema theorem

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

$$m(H,t)\frac{f(H,t)}{F(t)}(n) \tag{6}$$

will be the number of individuals that match that will be selected.

Schema theorem

In the next generation:

$$m(H, t+1) = m(H, t)\frac{f(H, t)}{F(t)}(n) \tag{7}$$

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

$$m(H, t+1) = m(H, t)\frac{f(H, t)}{\bar{F}(t)} \tag{8}$$

This means that schemas with higher fitness of avarage tend to be selected more.

Schema theorem

Lower-bound

Why is low bound? because in some cases we assume for isntance crossover with another not belonging schemas.

$$E[m(H,t+1)] \geq m(H,t)\frac{f(H,t)}{\bar{F}(t)}[(1-p_c)\frac{\delta(H)}{L(H)-1}](1-p_{mu})^{o(H)} \quad (9)$$

Implicit parallelism

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

Deap library

Mutation

Mutation

Shuffle indices

genetic TSP

Elitism process

$\mu + \lambda$ algorithm

Genetic algorithms in ML

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

- Decision trees
- K-means
- Tunning hyperparamters in Neural Networks.

Memetic algorithms

References

Following techniques to research

Simulated annealing, tabú search... ant PSO and ant-colony.