



Inteligencia Artificial

# **Guía 5.4. Algoritmos Genéticos**

4° Licenciatura En Sistemas de Información

# 2020

Universidad: UADER FCyT Concepción del Uruguay

Profesor: Lopez De Luise Daniela, Bel Walter

Alumnos: Exequiel Gonzalez, Cepeda Leandro

**1. El código CLISP del programa HelloWorld.lisp está en el apéndice A.**  
**a) Instale Common Clisp**

```
$ sudo apt-get update
$ sudo apt-get install clisp
```

**b) Copie el código en un ASCII llamado “HelloWorld.lisp”**

```
$ cp ag.caracteres.clisp HelloWorld.lisp
```

**c) Corra el test con “clisp <path>/helloWorld.lisp population string”**

**Ej:**  
**- clisp helloWorld 10 "hello world"**

```
lean@lean:~/Documentos/Repositorios/4-sistemas-uader/Inteligencia Artificial/IA [tp
s]/IA [tp6]$ clisp HelloWorld.lisp 10 "hello world"
Gen 1. Fitness: 10
Best match: m6jcQmrDrRa.

Gen 2. Fitness: 10
Best match: :6jcf-3frRa.

Gen 3. Fitness: 10
Best match: :6jcQm0DrRa.
```

•  
•  
•

```
Gen 564. Fitness: 1
Best match: hello w6rld.

Gen 565. Fitness: 1
Best match: hello w6rld.

Gen 566. Fitness: 0
Best match: hello world.

Done! Generation: 566
```

**- clisp helloWorld 20 "hello world"**

```
lean@lean:~/Documentos/Repositorios/4-sistemas-uader/Inteligencia Artificial/IA [tp
s]/IA [tp6]$ clisp HelloWorld.lisp 20 "hello world"
Gen 1. Fitness: 10
Best match: m6jcQmrDrRa.

Gen 2. Fitness: 10
Best match: m6jcf-3frRa.

Gen 3. Fitness: 10
Best match: m6jcQmrDrba.
```

•  
•  
•

```
Gen 223. Fitness: 1
Best match: he#lo world.

Gen 224. Fitness: 1
Best match: heklo world.

Gen 225. Fitness: 0
Best match: hello world.

Done! Generation: 225
```

**- clisp helloWorld 50 "hello world"**

```
lean@lean:~/Documentos/Repositorios/4-sistemas-uader/Inteligencia Artificial/IA [tp
s]/IA [tp6]$ clisp HelloWorld.lisp 50 "hello world"
Gen 1. Fitness: 10
Best match: m6jcQmrDrRa.

Gen 2. Fitness: 10
Best match: m6jcf-3frRa.

Gen 3. Fitness: 10
Best match: 0zswF-3frRa.
```

•  
•  
•

```
Gen 182. Fitness: 1
Best match: hello;world.

Gen 183. Fitness: 1
Best match: hello;world.

Gen 184. Fitness: 0
Best match: hello world.

Done! Generation: 184
```

**y anote en una tabla los resultados siguientes:**

Argumento 1: population	Argumento 2: string	Best match	#generaciones	best-match(n-1)
10	“hello world”	hello world	566	hello w6rld
20	“hello world”	hello world	225	heklo world
50	“hello world”	hello world	184	hello;world

Explique los resultados

A medida que aumenta el tamaño de la población decrementa el numero de generaciones ya que se llega a una solución mucho mas rápido.

d) Adapte el código para que el target no incluya caracteres especiales

Las modificaciones realizadas son las siguientes:

```
;; Comentamos la linea original en donde se define los caracteres que se utilizaran para reemplazar en el parametro string
;; (defparameter *chars* ".,<=>+.-`~#ABCDEFGHIJKLMNPOQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 ' ! ? ; : \ / , " )

;; Duplicamos la linea y quitamos los caracteres especiales que se definieron anteriormente
(defparameter *chars* "ABCDEFGHIJKLMNPOQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 ")
```

Corra con un ejemplo

El resultado de una corrida es el siguiente:

```
lean@lean:~/Documentos/Repositorios/4-sistemas-uader/Inteligencia Artificial/IA [tp6]$ clisp HelloWorld.lisp 50 "hello world"
Gen 1. Fitness: 9
Best match: fePFWwanrR9.

Gen 2. Fitness: 8
Best match: fePfo VEP2m.

Gen 3. Fitness: 8
Best match: fePJo VEP2m.
```

.

.

.

```
Gen 108. Fitness: 1
Best match: hello w9rld.

Gen 109. Fitness: 1
Best match: hello world.

Corra con un ejemplo
Gen 110. Fitness: 1
Best match: hello w9rld.

Gen 111. Fitness: 0
Best match: hello world.
```

Argumento 1: population	Argumento 2: string	Best match	#generaciones	best-match(n-1)
50	“hello world”	hello world	111	hello w9rld

e) Explique cuál es la función de fitness actual

La función de fitness actual utiliza la función “hamming-distance”, con la que evalúa la distancia entre uno de los individuos de la población y la solución (\*target\* recibido como parámetro)

Hamming distance: en Teoría de la Información se denomina distancia de Hamming a la efectividad de los códigos de bloque y depende de la diferencia entre una palabra de código válida y otra. Cuanto mayor sea esta diferencia, menor es la posibilidad de que un código válido se transforme en otro código válido por una serie de errores. A esta diferencia se le llama distancia de Hamming, y se define como el número de bits que tienen que cambiarse para transformar una palabra de código válida en otra palabra de código válida. Si dos palabras de código difieren en una distancia d, se necesitan d errores para convertir una en la otra.

Por ejemplo: La distancia Hamming entre 1011101 y 1001001 es 2.

f) Cambie la función de fitness por otra que le parezca razonable. Corra y explique su cambio y resultados

El cambio realizado fue la función que calcula la distancia entre dos cadenas, para el caso seleccionamos la distancia de Levenshtein que puede definirse como el número mínimo de operaciones de edición requeridas para convertir una cadena en la otra, considerando como operaciones válidas el borrado, la inserción o la sustitución.

Algoritmo con fitness “levenshtein-distance”

```
lean@lean:~/Documentos/Repositorios/4-sistemas-uader/Inteligencia Artificial/IA [tps]/IA [tp6]$ clisp HelloWorldModified.lisp 50 "hello world"
Gen 1. Fitness: 10
Best match: nj 3y MCE b.

Gen 2. Fitness: 10
Best match: nj 3uRI9vL7.

Gen 3. Fitness: 9
Best match: Wj ly MCE 1.
```

.

.

.

```
Gen 110. Fitness: 1
Best match: hello wor d.

Gen 111. Fitness: 1
Best match: hello wor d.

Gen 112. Fitness: 0
Best match: hello world.

Done! Generation: 112
```

Algoritmo con fitness “hamming-distance”

```
lean@lean:~/Documentos/Repositorios/4-sistemas-uader/Inteligencia Artificial/IA [tps]/IA [tp6]$ clisp HelloWorld.lisp 50 "hello world"
Gen 1. Fitness: 9
Best match: fePFWwanrR9.

Gen 2. Fitness: 9
Best match: BePFP VMP2m.

Gen 3. Fitness: 8
Best match: fePLP SMP2m.
```

.

.

.

```
Gen 59. Fitness: 1
Best match: 9ello world.

Gen 60. Fitness: 1
Best match: 0ello world.

Gen 61. Fitness: 0
Best match: hello world.

Done! Generation: 61
```