



Algoritmos genéticos

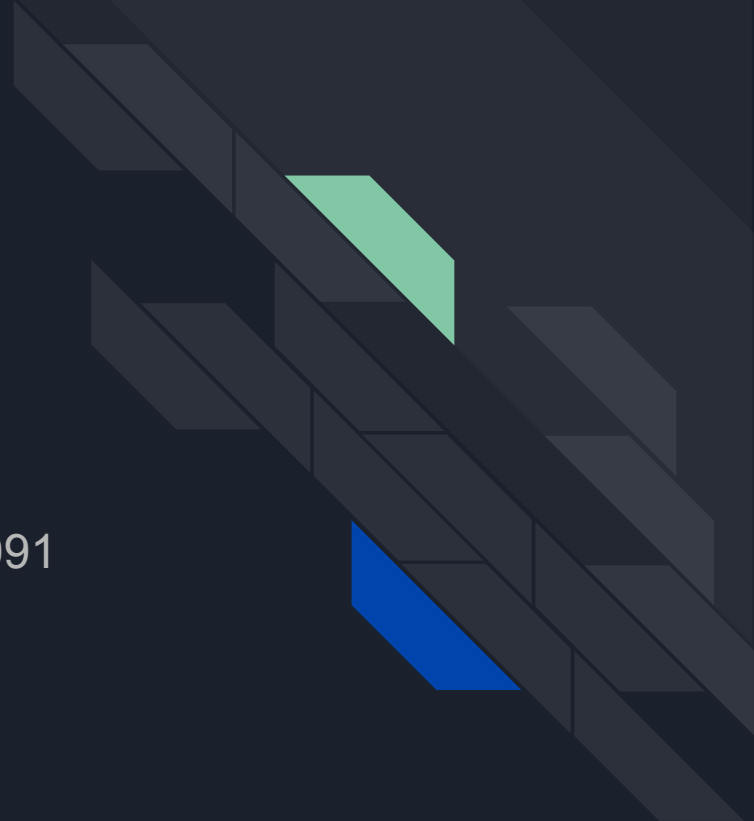
72.27 - Sistemas de Inteligencia Artificial

Grupo 7

Luque Meijide, Manuel - 57386

Karpovich, Lucía - 58131

Tarradellas del Campo, Manuel - 58091

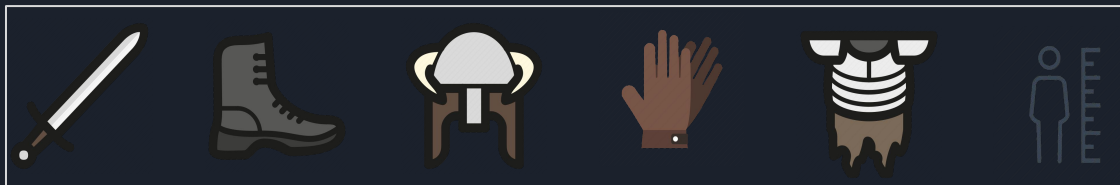




Implementación

Componentes de Algoritmo genético

Genotipo



Gen

Alelos → [chest.csv]

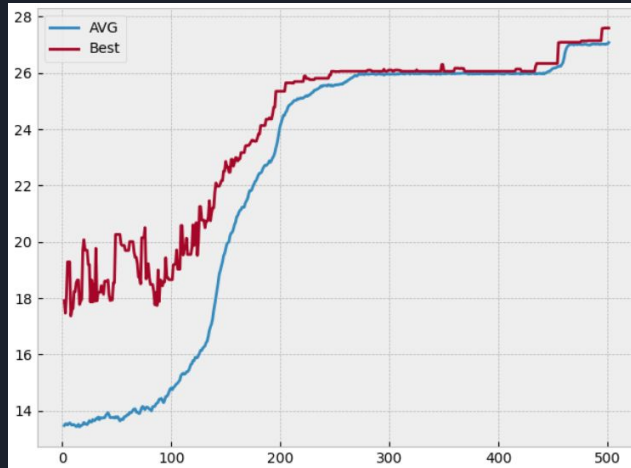


Consideraciones

Selección Entrópica - Boltzmann

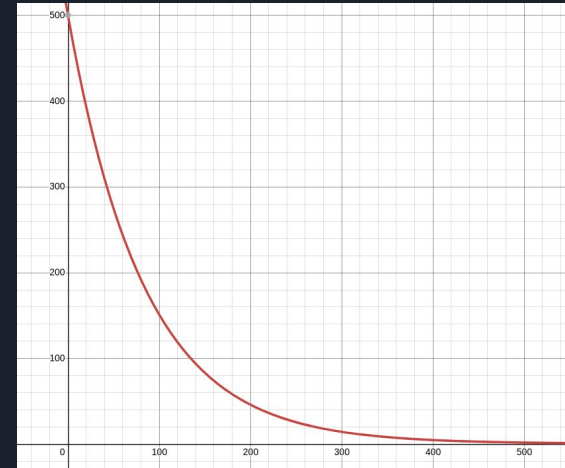
$$T(t) = T_c - (T_0 - T_c) e^{-kx}$$

Fitness



Nº de generación

$T(t)$



Nº de generación



Criterios de Corte

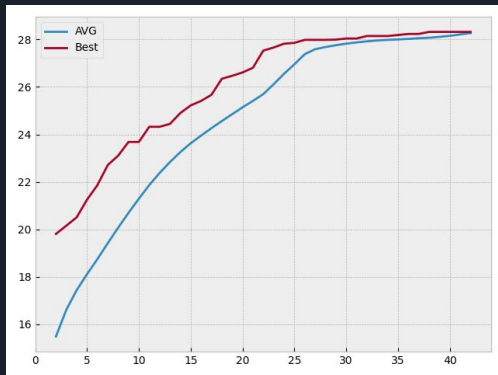
- Tiempo
- Cantidad de Generaciones
- Solución aceptable
- Estructura
- Contenido



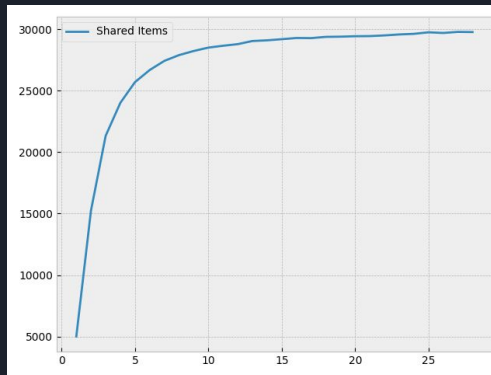
Gráficos

Gráficos en tiempo real

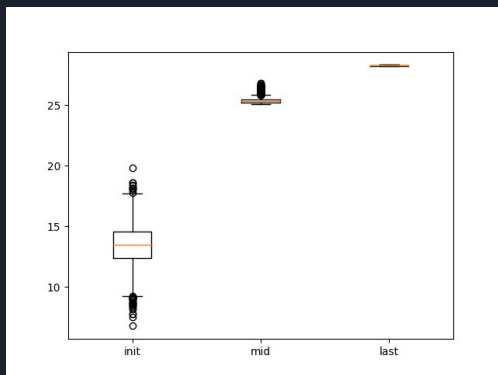
Mejor fitness vs
Avg fitness
por generación



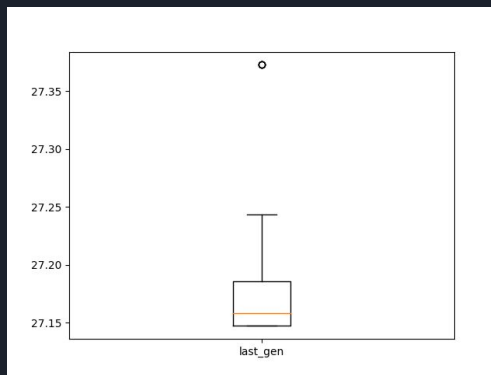
Ítems
compartidos
por generación



Boxplot de
fitness de
población
inicial, media y
final



Boxplot de
fitness
población final



Activities PyCharm Professional Edition abr 5 11:51 TP2 - config.json

File Edit View Navigate Code Refactor Run Tools Git Window Help

TP2 config.json

Project
main.py
config.json
graph.png
fitness_graph.png
fitness_boxplot.png
final_fitness_boxplot.png
items_graph.png
util.py

TP2
Project
main.py
config.json
graph.png
fitness_graph.png
fitness_boxplot.png
final_fitness_boxplot.png
items_graph.png
util.py

main.py
10
11
12
13
14
15
16
17
18
19
20
mutation

Run: main
/usr/bin/python3.8 /home/lkarpovich/SIA/TP2/main.py
Configuration:
Player class: warrior
Dataset: data
Population size: 5000
Fill: fill_all
Cross: namespace(annular_length=3, method='uniform', un
Mutation: namespace(method='limited', probability=0.5)
Select parents: namespace(amount=5000, boltzmann_k=0.1,
Select children: namespace(boltzmann_k=0.1, boltzmann_t
Cut: namespace(acceptable_fitness=19, fitness_decimals=
Retrieving data from file....
Done.
Generating random initial population...
Done.
Loading configuration...
Done.
Starting algorithm...
~~~ COMPLETE ~~~

Figure 1  
30000  
25000  
20000  
15000  
10000  
5000  
0  
0 10 20 30 40  
Shared Items  
limit=30)

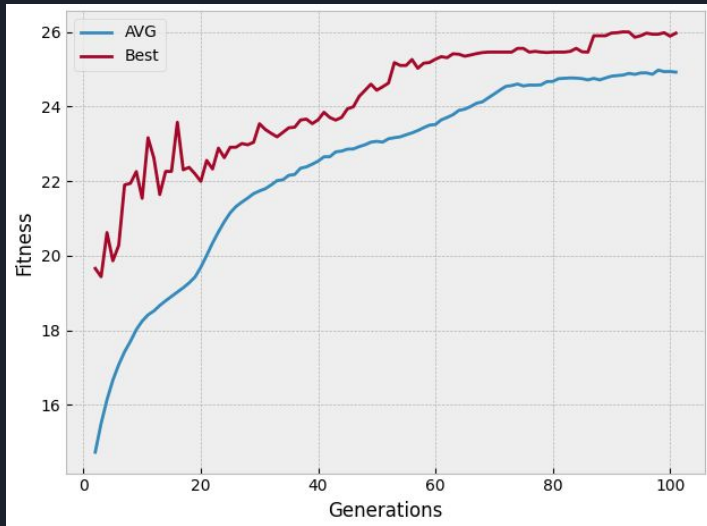
Figure 1  
25  
20  
15  
10  
5  
0  
init mid last  
X= Y=16.23

20:1 LF UTF-8 4 spaces\* JSON: config.json Python 3.8 P main



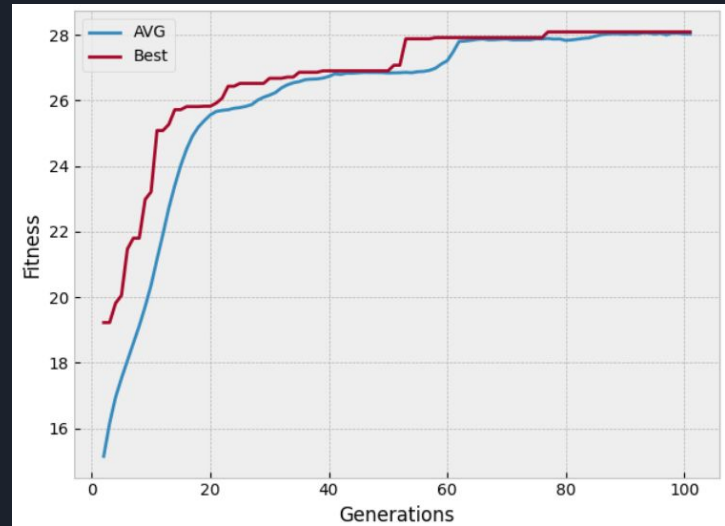
# Comparaciones

## Fill - Parents



Explorativo

## Fill - All



Explotativo

# Variación de Población



Mayor variedad de ítems



Mayor probabilidad de comenzar con ítems cercanos a los óptimos



Menos dependencias en mutaciones



En menos generaciones se llega a un resultado cercano al óptimo.



Mayor costo en memoria y en tiempo de ejecución.

# Variación de población 100 vs 1000 vs 10000

## Configuración

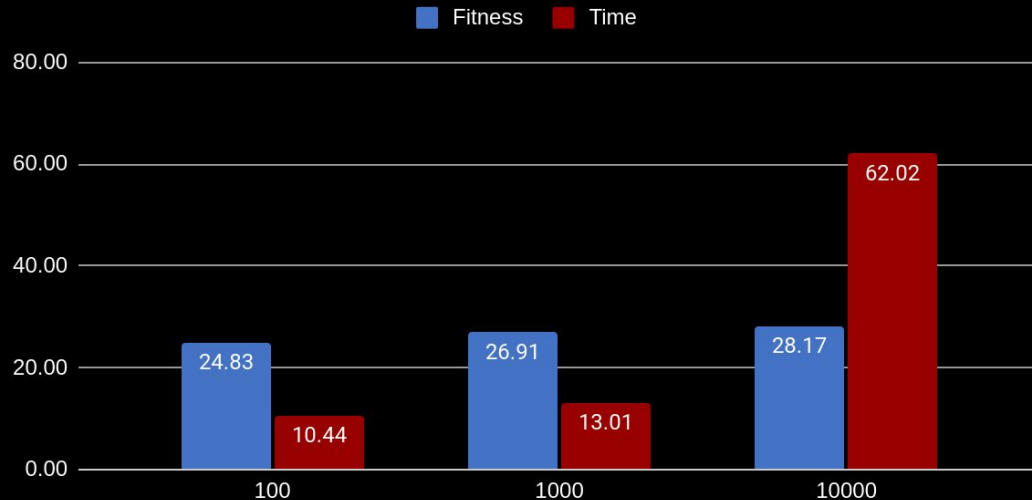
Clase: Guerrero

Criterio de corte: Contenido (límite de 10 generaciones)

Método de cruce: Anular

Mutación: Limitada

### Fitness y Time



# Mutación

## Configuración

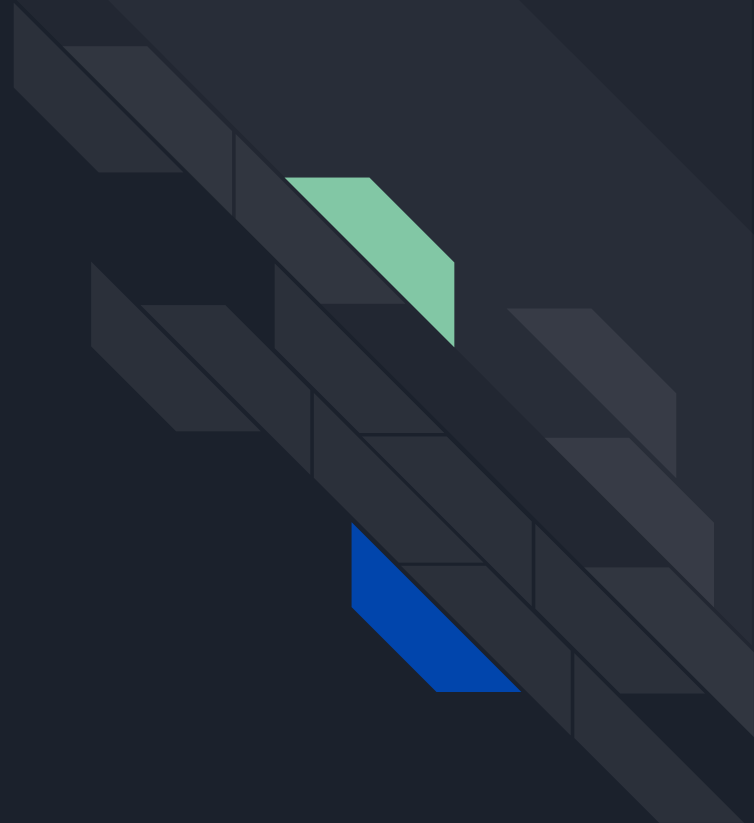
Clase: Guerrero


Criterio de corte: Generacional (límite de 50 generaciones)

Método de cruza: Elite

Cruza: Sin Cruza

Población: 10000

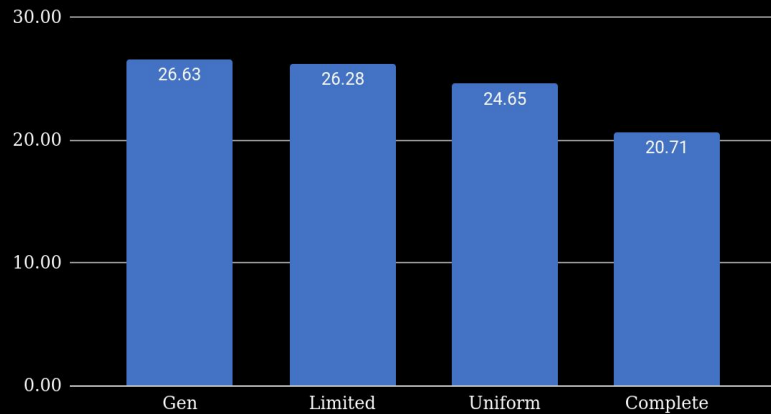




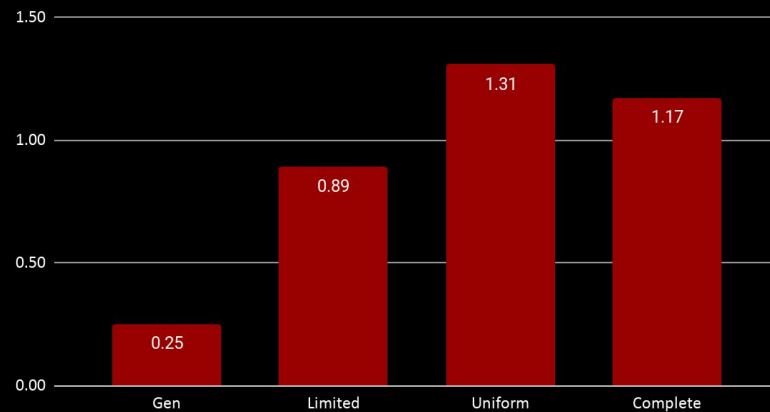
## Mutación

# Gráficos y análisis

### Best Fitness



### Time





# Selección

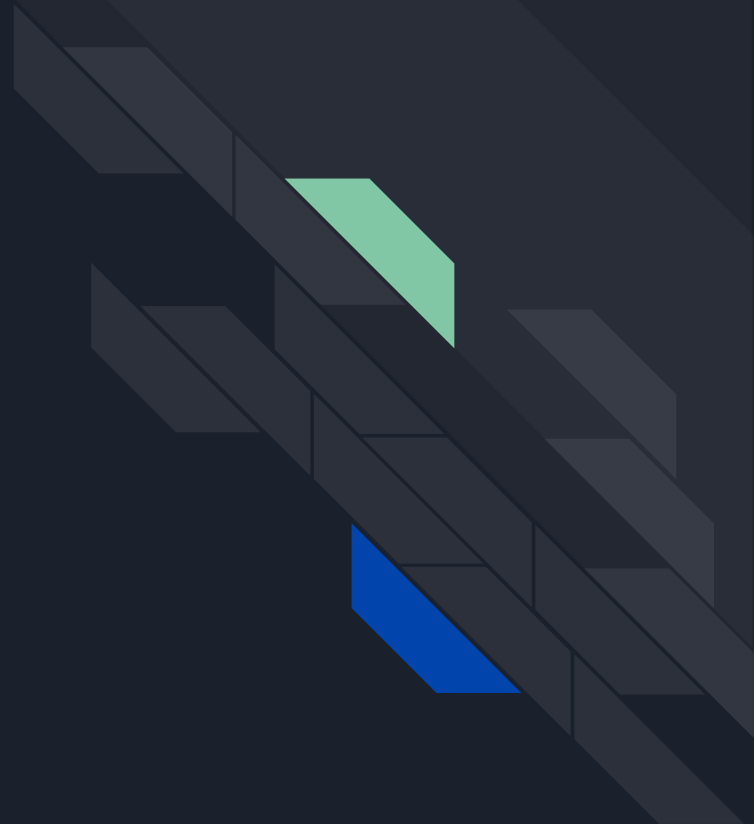
## Configuración

Clase: Guerrero

Criterio de corte: Generacional (límite de 50 generaciones)

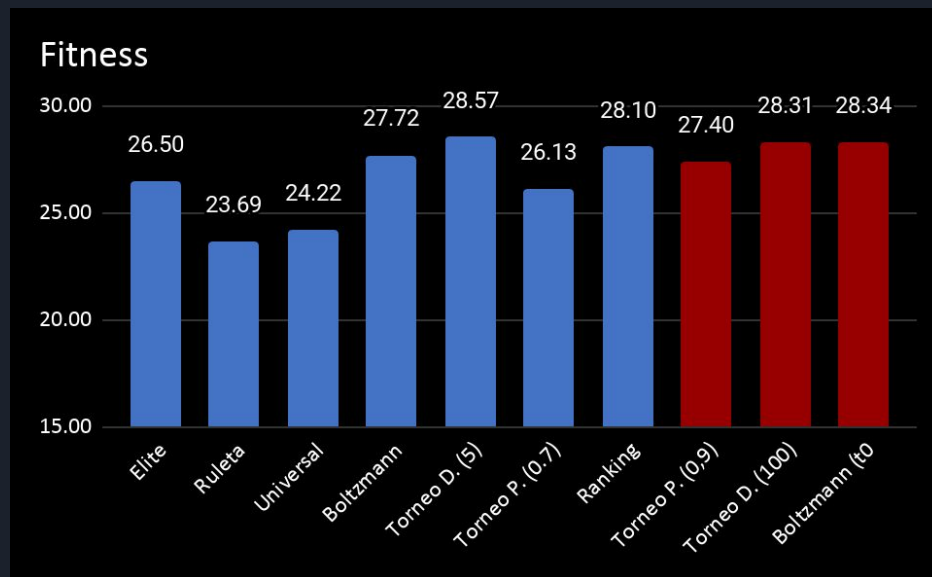
Método de cruce: Un punto


Mutación: Gen



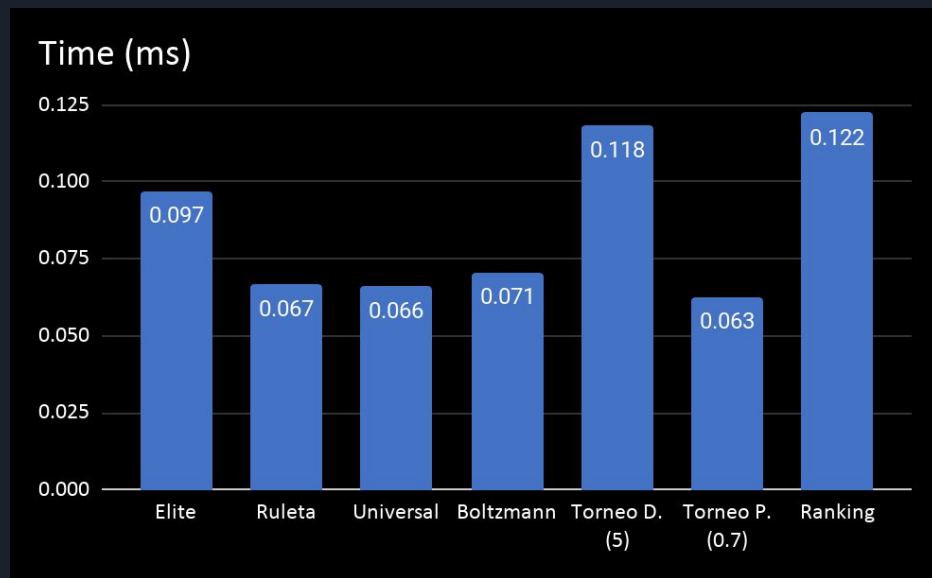
# Selección

## Gráficos y análisis



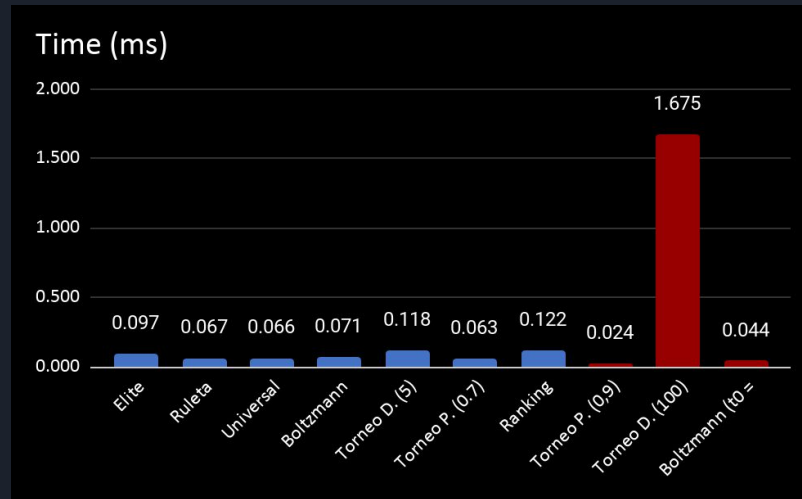


## Selección Gráficos y análisis



# Selección

## Gráficos y análisis



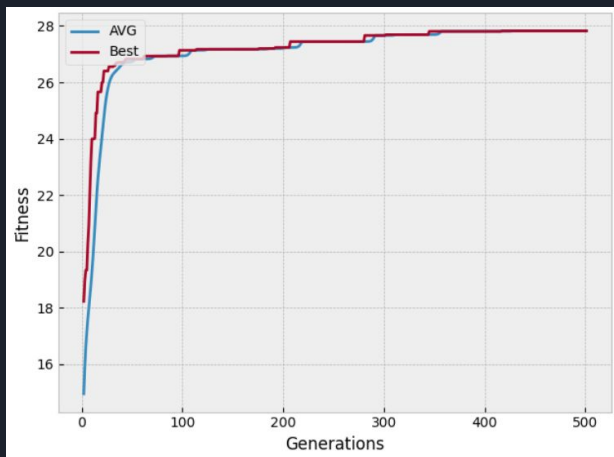
Variando los parámetros

# Selección

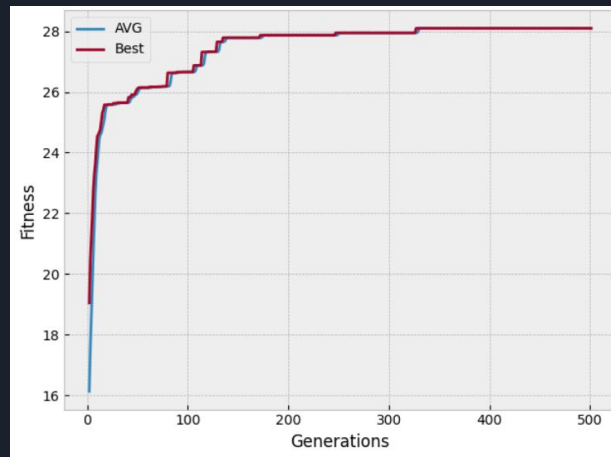
## Presión selectiva

Mayor presión selectiva

- Elite



- Torneo Determinístico (50)

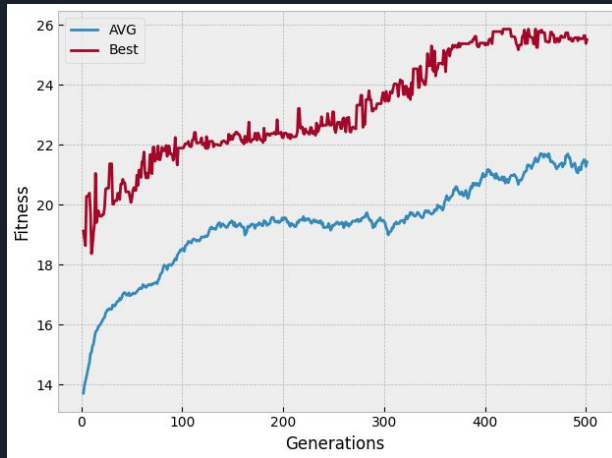


# Selección

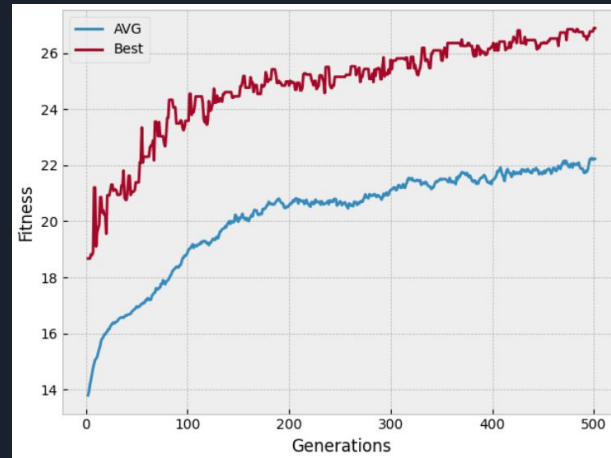
## Presión selectiva

Menor presión selectiva

- Torneo Probabilístico (0,6)



- Universal



# Cruza

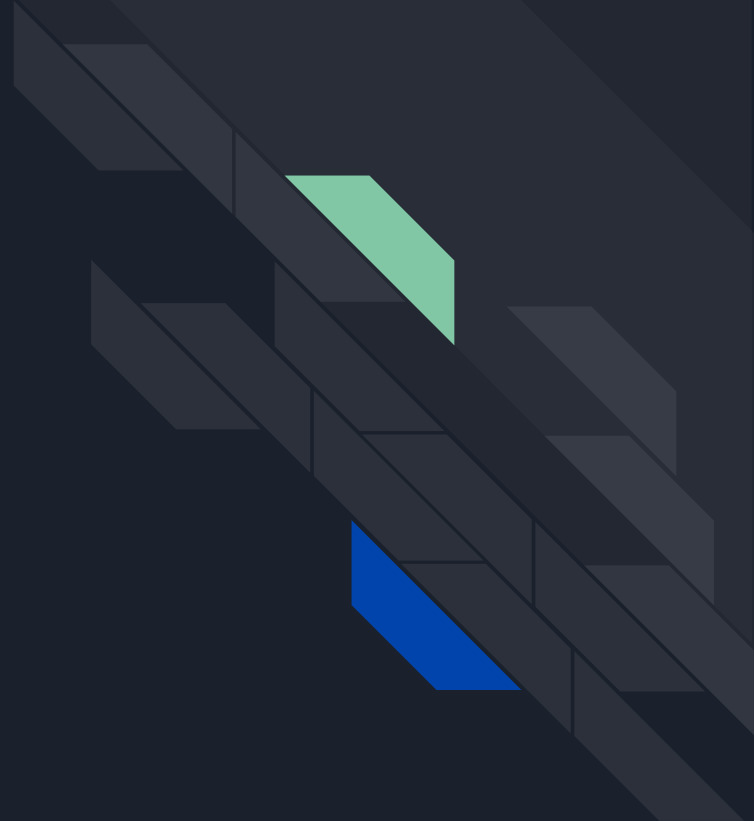
## Configuración

Clase: Guerrero

Criterio de corte: Generacional (límite de 50 generaciones)

Método de selección: Elite

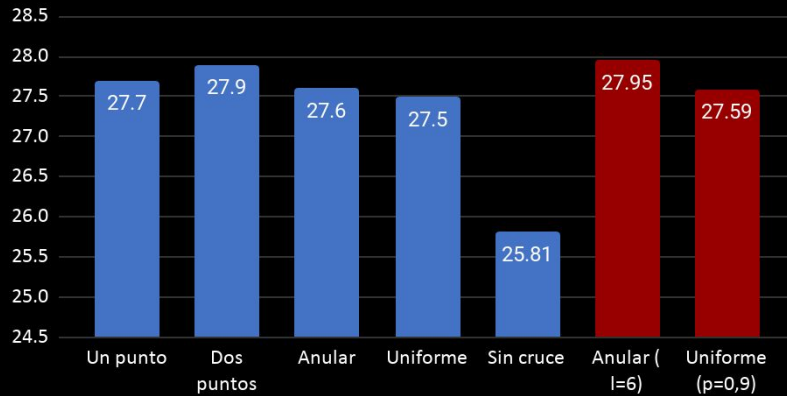
Mutación: Gen



# Cruza

## Gráficos y análisis

Fitness



### Observaciones

- Se puede ver el impacto en el fitness final.
- Ninguno se destaca particularmente sobre el otro.



## Cruza Gráficos y análisis

Time (ms)



### Observaciones

- No aportan una complejidad temporal significativa al algoritmo
- Muy similares entre sí

# Conclusiones

