



# Algoritmos Genéticos

# Integrantes

*Juan Pablo Oriana - 60621*

*Tomas Cerdeira - 60051*

*Santiago Garcia Montagner - 60352*



## Objetivos del trabajo

- Implementar **algoritmos genéticos** para resolver y **aproximar la solución de un problema**

- Dados:

$$W = \begin{pmatrix} W_0 \\ W_1 \\ W_2 \end{pmatrix} \quad w = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} \quad w_0 = (w_{01}, w_{02})$$

Se busca **minimizar el error** e informar los **argumentos y algoritmos óptimos** que más se ajusten a los reactivos iniciales dados como “ciertos”.



## Población y sus individuos

- Cada **individuo** es representado por un **arreglo de 11 elementos**:

$$X = (W_0, W_1, W_2, w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}, w_{01}, w_{02})$$

- Siendo X el genotipo de cada individuo representado con números reales, se lo utiliza para minimizar el error de la función:

$$E(W, w, w_0) = \sum_{\mu=1}^3 (\zeta^{\mu} - F(W, w, \xi^{\mu}))^2$$

## Función F

- Dados:

$$\underline{g(x)} = \frac{e^x}{1+e^x} \quad \text{Función logística}$$

$\xi^k$

Valores de entrada, con  $k = 1, 2, 3$

$\zeta$

Valores de salida

$$\Rightarrow \underline{F(W, w, w_0, \xi)} = \underline{g\left(\sum_{j=1}^2 W_j g\left(\sum_{k=1}^3 w_{jk} \xi_k - w_{j0}\right) - W_0\right)}$$

Donde el error resulta  $E(W, w, w_0) = \sum_{\mu=1}^3 (\zeta^\mu - \underline{F(W, w, \xi^\mu)})^2$



## ¿Cómo genero una función de fitness $f(x)$ ?

Necesitamos que la función de fitness sea mejor cuanto mayor sea el valor, para así poder utilizar los métodos de selección dados por la cátedra. Por lo tanto la **función de error NO es buena para el fitness**, ya que **buscamos minimizarla**. Podríamos tomar el opuesto del error.

$$f(x) = -E(x)$$

Esto solucionaría el problema de la maximización pero **agregaría valores negativos**. Los valores negativos rompen la selección roulette. Es por esto que **decidimos tomar**

$$f(x) = CantCasos - E(x)$$

**Asegurando un valor positivo** todo el tiempo. Sin embargo, **hay que tener en cuenta la pérdida de precisión**.



## Valores de entrada $\xi$

$$\xi^1 = \begin{pmatrix} 4,4793 \\ -4,0765 \\ -4,0765 \end{pmatrix}$$

$$\zeta^1 = 0$$

$$\xi^2 = \begin{pmatrix} -4,1793 \\ -4,9218 \\ 1,7664 \end{pmatrix}$$

$$\zeta^2 = 1$$

$$\xi^3 = \begin{pmatrix} -3,9429 \\ -0,7689 \\ 4,8830 \end{pmatrix}$$

$$\zeta^3 = 1$$



## Problema de la velocidad de resolución

- Con los 3 casos de prueba dados, la convergencia a una solución óptima ocurre **rápidamente**. Para poder hacer un análisis de datos con una convergencia más retardada, se nos ocurrió lo siguiente:
  - Encontrar una solución óptima para los 3 casos de prueba.

Individuo:  
Error:  $8.31 \times 10^{-19}$

```
[18.198890215902043, 19.887470841120884, 17.738368717311303,  
0.6652017218928963, 0.335103249657387, 5.854675779470559,  
0.5791663670483601, -0.7423788891309104, 7.785770197567634,  
-0.9460407296950941, -1.3617190069927492]
```

- Usar este vector de 11 posiciones con la función F para aproximar 6 valores más.
- Ahora usamos esos 9 valores para crear una **NUEVA** función de error



## Aproximación de los valores de entrada $\xi$

De esta forma se generaron los nuevos puntos.

e.j.

```
 $\xi =$  [3.62186468 1.8222188 1.56871685]  
 $\zeta =$  0.9999999963440138  
 $\xi =$  [ 1.17117577 -1.35207369 2.56329238]  
 $\zeta =$  0.9999999963441881  
 $\xi =$  [ 2.21837086 -2.96838178 -0.9127545 ]  
 $\zeta =$  9.890759665377149e-08
```





# Estrategia de implementación



# Individuos

- Incluyen, como se mencionó al comienzo de la presentación, el arreglo X

$$X = (W_0, W_1, W_2, w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}, w_{01}, w_{02})$$

- Función de aptitud
- Valor concreto de la aptitud
- Métodos para obtener de manera rápida y simple los parámetros relevantes del arreglo X
  - `get_big_w`: retorna [ W0, W1, W2 ]
  - `get_little_w(j)`: retorna [ wj1, wj2, wj3 ]
  - `get_w_0`: retorna [ w01, w02 ]



# GeneticSolver

- Cada uno de los **algoritmos** se utilizan **a través de una clase llamada GeneticSolver**.
  - La **selección de los padres**, la **cruza** y **mutación** de sus hijos, las **condiciones de corte** y **estados** actuales de cada corrida.
    - Número de generación
    - Individuo con la máxima aptitud de la generación
    - Aptitud promedio de la población actual
    - La población actual en sí
- Mediante los métodos: `evolve` → `evolve_limited` → `next_gen`, nuevas generaciones son generadas y analizadas dentro del marco del problema.



# Métodos de cruza

- Cruza Simple → *simple\_cross*
  - un punto al azar, y se intercambian los alelos
- Cruza Doble → *double\_cross*
  - dos puntos al azar, y se intercambian los alelos
- Cruza Uniforme → *rand\_cross*
  - se intercambian los alelos con una probabilidad de 0,5.



# Métodos de selección

- Elite selection → *elite\_selection*
- Rank selection → *rank\_selection*
- Roulette selection → *roulette\_selection*
- Truncated selection → *truncated\_selection*
- Tournament selection → *tournament\_selection*
- Boltzmann selection → *boltzmann\_temperature*



# Selección de padres

- Para crear una nueva generación debemos elegir los padres.
- Los padres se eligen en base a una **distribución aleatoria proporcional a su aptitud**, lo que favorece a los individuos más aptos pero también le da lugar a los peores.
  - La **población comienza con una diversidad significativa**, lo que genera una diferencia entre los mejores y peores individuos con un orden de  $10^{-1}$ .
  - A medida que **aumentan las generaciones, la diversidad disminuye**, haciendo que la **selección se torne casi uniforme**.



# Análisis de los resultados



## Metodología de testing

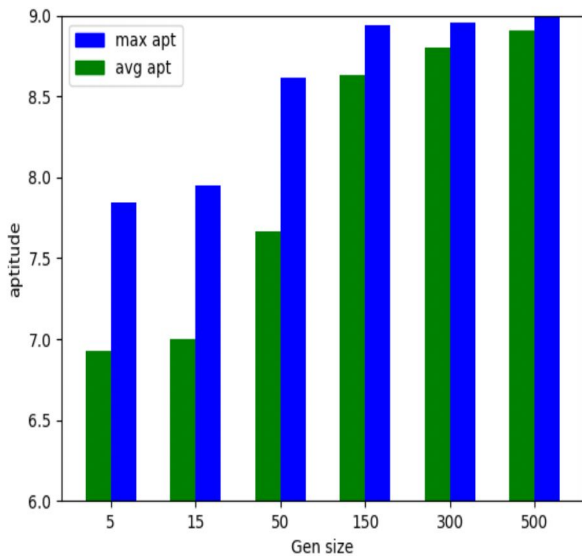
- Para cada testeo, se dejaron fijas todas las variables menos aquella que se querían testear además de la cantidad de generaciones.
- El tamaño de las generaciones aumenta de a 'chunks' y el nuevo tamaño parte de los datos que obtuvo el anterior.
  - *chunked\_generations* = ([5, 10, 35, 100, 150, 200])
  - Referencia: “Benefits of discrete generation analysis over continuous representations”
- Los valores obtenidos son un promedio de 5 corridas.
- Valores default:
  - Tamaño de generacion = 50
  - Probabilidad de mutación = 0.09 (1/11)
  - Desvío estándar para mutación = 1





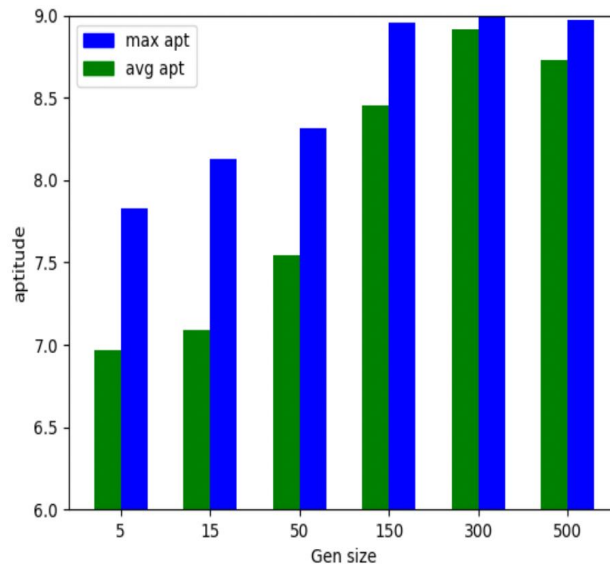
# Análisis de los *Métodos de* *cruza*

Generation change test roulette\_selection y simple\_cross ←



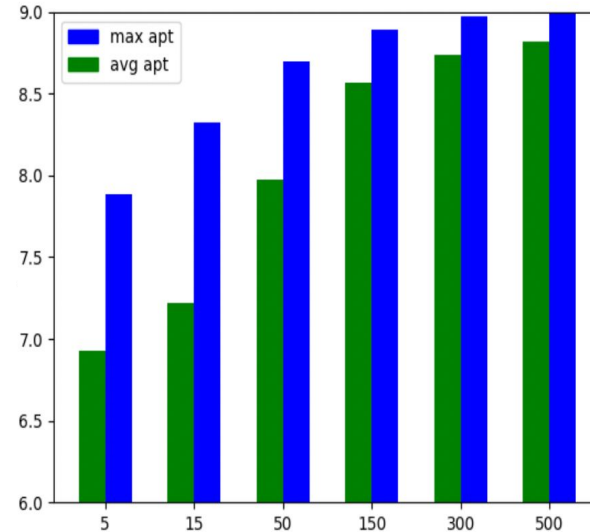
$E_{optimo} = 1.3 \cdot 10^{-13}$

Generation change test roulette\_selection y multiple\_cross ←



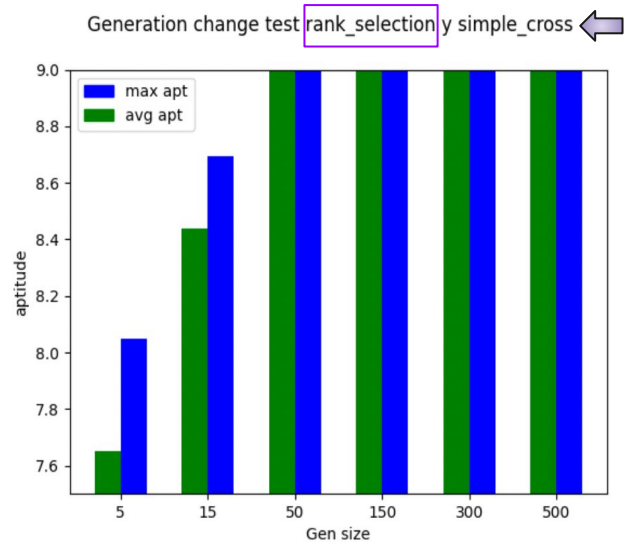
$E_{optimo} = 2.2 \cdot 10^{-6}$

Generation change test roulette\_selection y rand\_cross ←

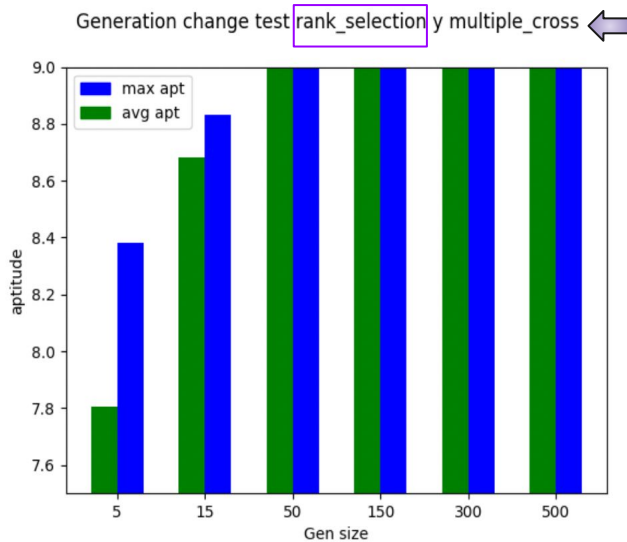


$E_{optimo} = 5.9 \cdot 10^{-16}$

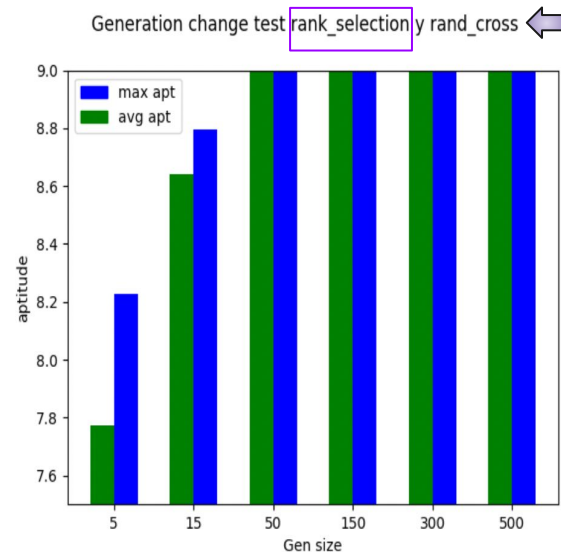
**OBS:** tamaño de la muestra = tamaño de la población = 50



$$E_{\text{optimo}} = 3.1 \cdot 10^{-18}$$



$$E_{\text{optimo}} = 1.2 \cdot 10^{-19}$$



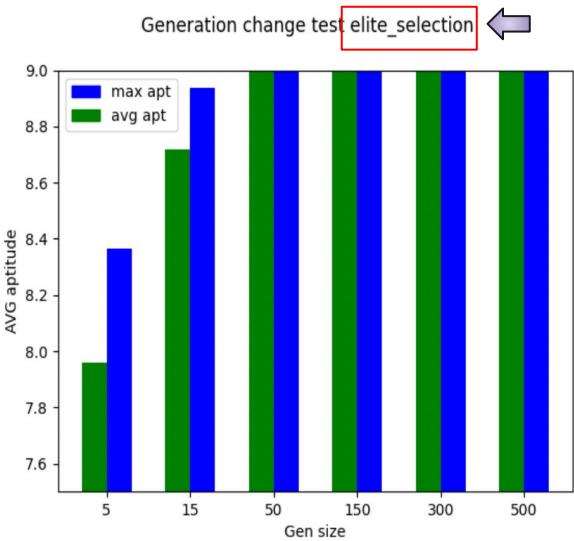
$$E_{\text{optimo}} = 7.1 \cdot 10^{-18}$$

**OBS:** tamaño de la muestra = tamaño de la población = 50

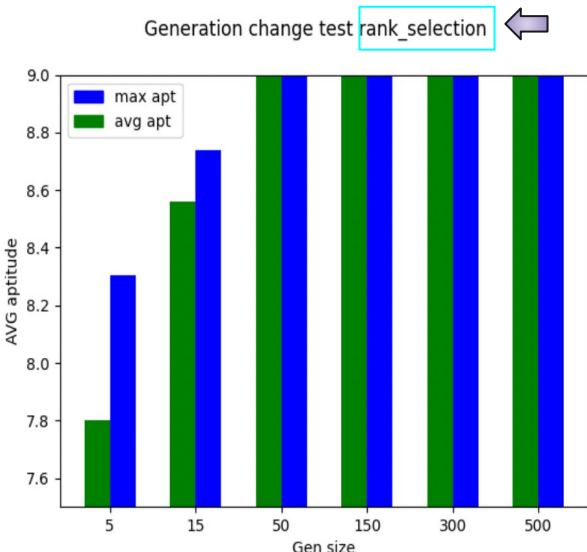


# **Análisis de los *Métodos de selección***

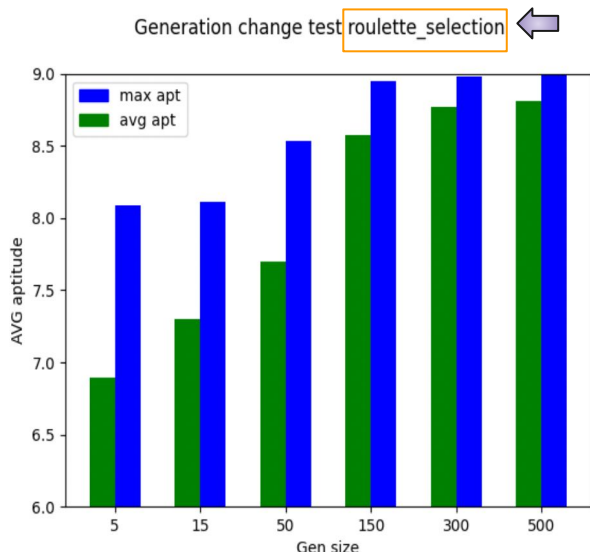
# Métodos de selección sin parámetros



$$E_{\text{optimo}} = 5.1 \cdot 10^{-19}$$



$$E_{\text{optimo}} = 3.5 \cdot 10^{-15}$$



$$E_{\text{optimo}} = 4.3 \cdot 10^{-12}$$

**OBS:** tamaño de la muestra = tamaño de la población = 50



## Métodos de selección *(sin parámetros)*

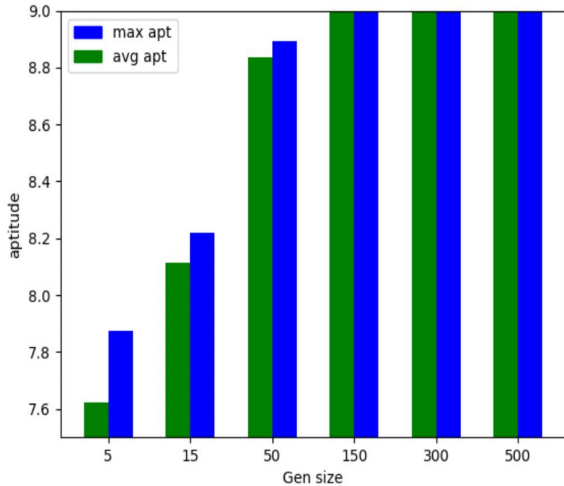
Se observa claramente como, el **promedio de las aptitudes** toma valores altos desde generaciones bajas siguiendo:

1. Elite Selection
  - selección de individuos con mayor aptitud
2. Rank Selection
  - relativa al ranking respectivo del individuo en la población
3. Roulette Selection
  - proporcional a la aptitud relativa



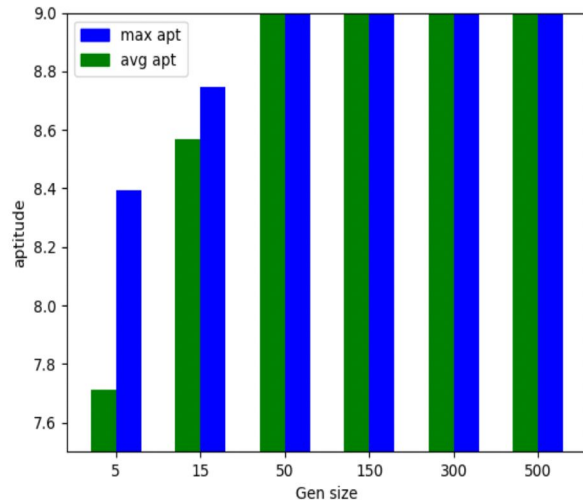
# **Análisis de la** *Probabilidad de Mutación*

Generation change test rank\_selection mutation prob:0.01 ←



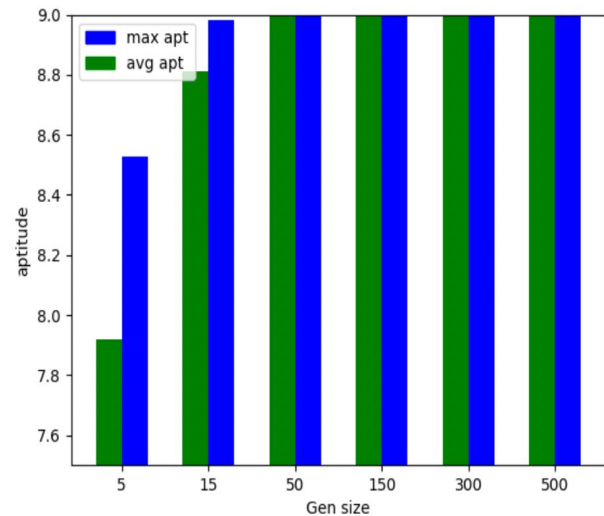
$E_{\text{optimo}} = 3.3 \cdot 10^{-18}$

Generation change test rank\_selection mutation prob:0.09 ←



$E_{\text{optimo}} = 1.2 \cdot 10^{-19}$

Generation change test rank\_selection mutation prob:0.5 ←



$E_{\text{optimo}} = 4.3 \cdot 10^{-18}$

**OBS:** tamaño de la muestra = tamaño de la población = 50





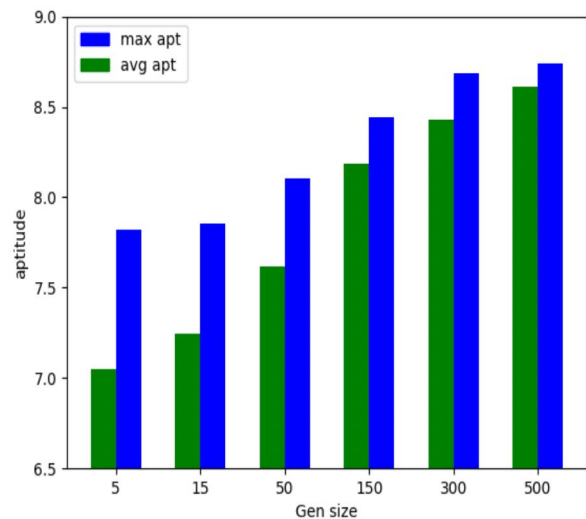
## *Probabilidad de Mutación* **con rank**

Se observa claramente como, el **promedio de las aptitudes** toma valores altos desde generaciones bajas cuanto **mayor probabilidad de mutación** hay siguiendo:

1. Probabilidad de mutación = 0.5
2. Probabilidad de mutación = 0.09
3. Probabilidad de mutación = 0.01

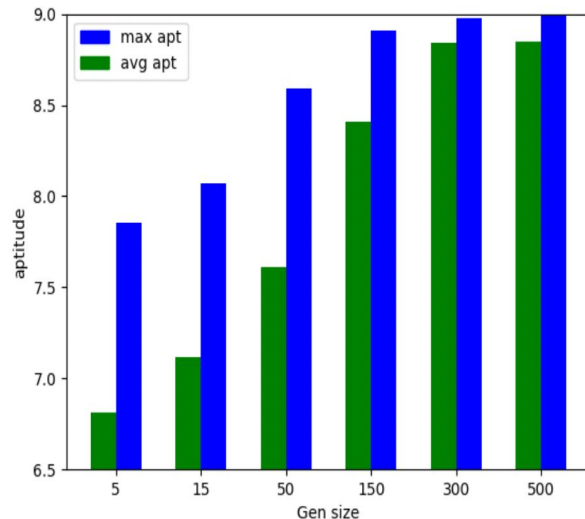
Conclusión parcial: Mayor probabilidad de mutación → Mayor diversidad → Se encuentran potenciales buenos individuos más rápido

Generation change test roulette\_selection mutation prob:0.01 ←



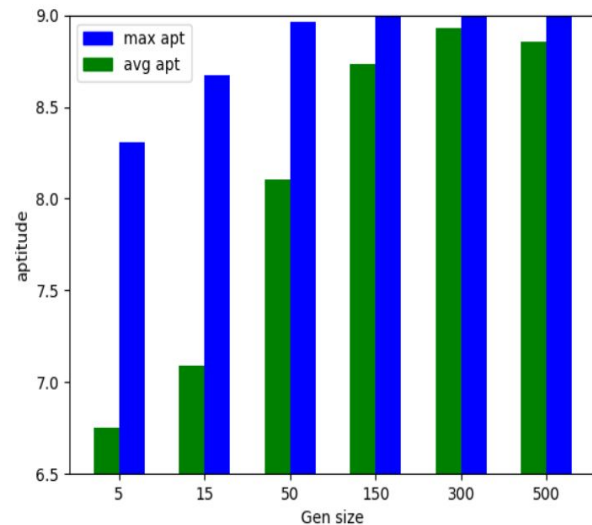
$E_{\text{optimo}} = 0.3358213$

Generation change test roulette\_selection mutation prob:0.09 ←



$E_{\text{optimo}} = 3.3 \cdot 10^{-2}$

Generation change test roulette\_selection mutation prob:0.5 ←



$E_{\text{optimo}} = 6.4 \cdot 10^{-11}$

**OBS:** tamaño de la muestra = tamaño de la población = 50



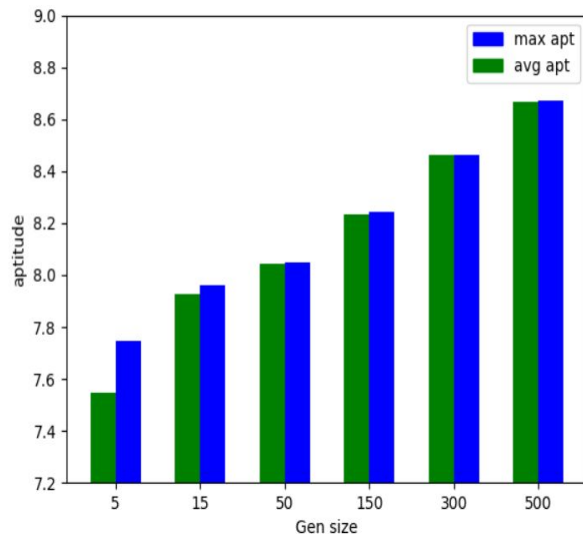
## *Probabilidad de Mutación* **con roulette**

Se observa como, en las primeras generaciones el **promedio de las aptitudes** no se ve afectado de manera positiva, pero a medida que aumentan toman valores mayores cuanto **mayor probabilidad de mutación** hay siguiendo:

1. Probabilidad de mutación = 0.5
2. Probabilidad de mutación = 0.09
3. Probabilidad de mutación = 0.01

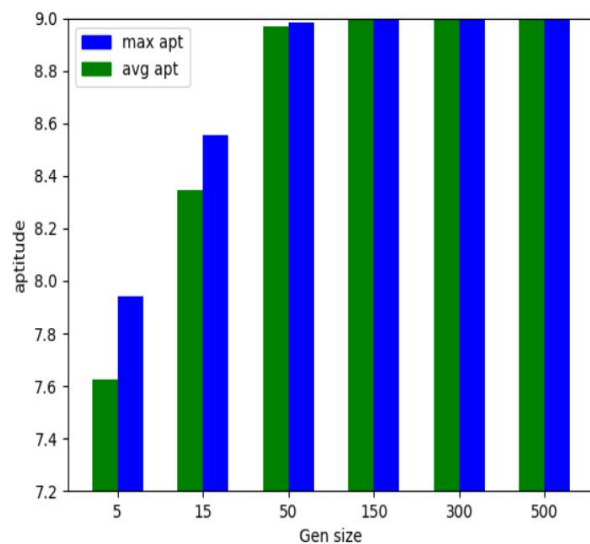
Conclusión parcial: Mayor probabilidad de mutación → Mayor diversidad → Se encuentran potenciales buenos individuos más rápido

Generation change test with elite\_selection and mutation std:0.01



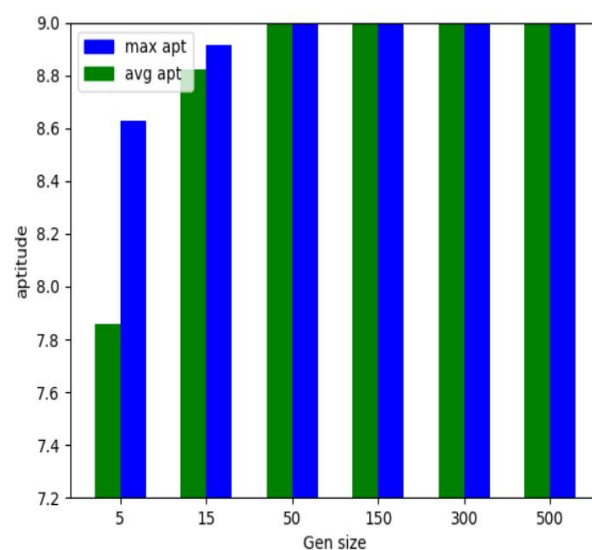
$E_{\text{optimo}} = 0.2984123$

Generation change test with elite\_selection and mutation std:0.5



$E_{\text{optimo}} = 3.4 \cdot 10^{-13}$

Generation change test with elite\_selection and mutation std:4



$E_{\text{optimo}} = 6.5 \cdot 10^{-18}$

**OBS:** tamaño de la muestra = tamaño de la población = 50



## *Desviación estándar (STD) de Mutación* **con elite**

Se observa como, el **promedio de las aptitudes** se ve afectado de manera positiva cuanto **mayor desviación estándar tiene la mutación** siguiendo:

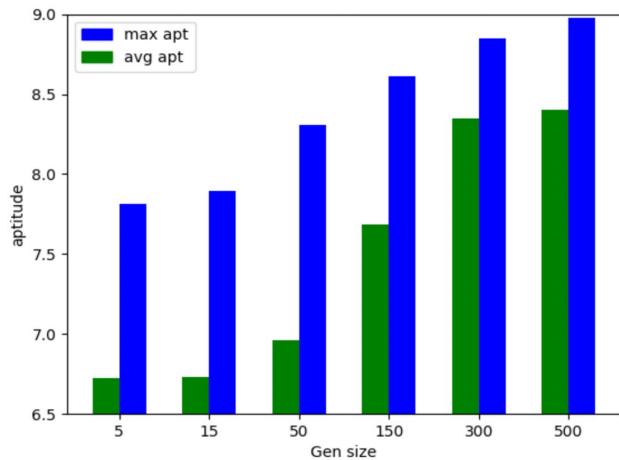
1. STD de mutación = 4
2. STD de mutación = 0.5
3. STD de mutación = 0.01

Conclusión parcial: Mayor STD → Mayor diversidad → Se encuentran potenciales buenos individuos más rápido

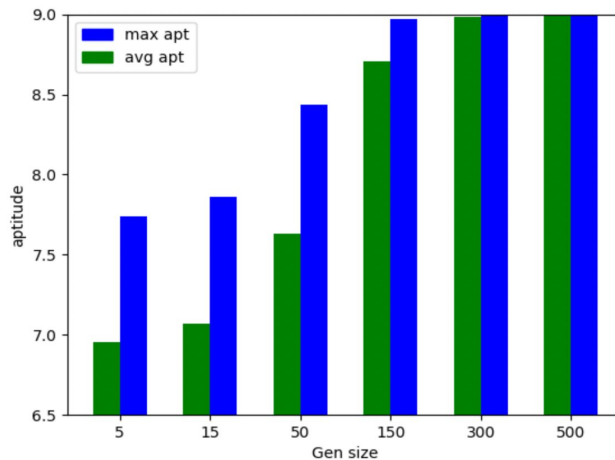


# **Análisis de la** *variación del $K$ en la selección truncada*

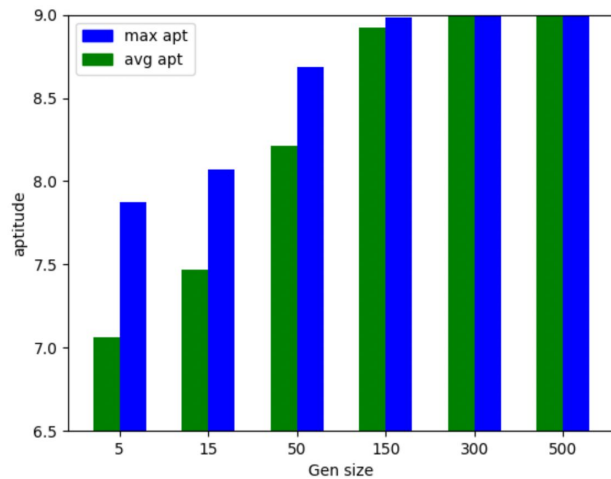
Generation change test chunked size:0 ←



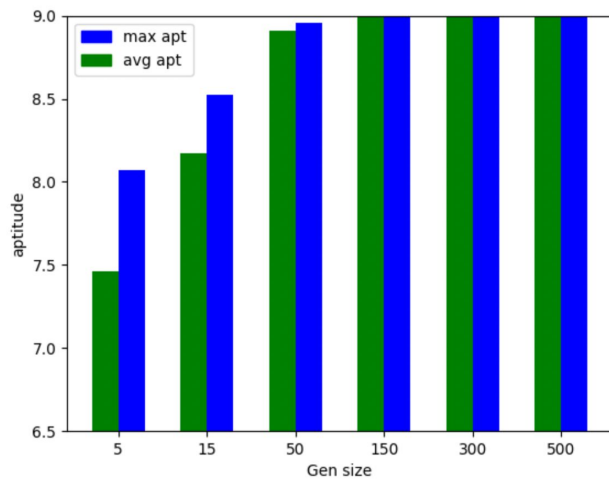
Generation change test chunked size:5 ←



Generation change test chunked size:8 ←



Generation change test chunked size:25 ←



**Crossing algo**

- *simple\_cross*

**Selection algo**

- *truncated\_selection*

- k: variable



## *Variación del K* **en selección truncada**

Analizando los resultados obtenidos con K variando de  $0 \rightarrow 25$ , se ve como:

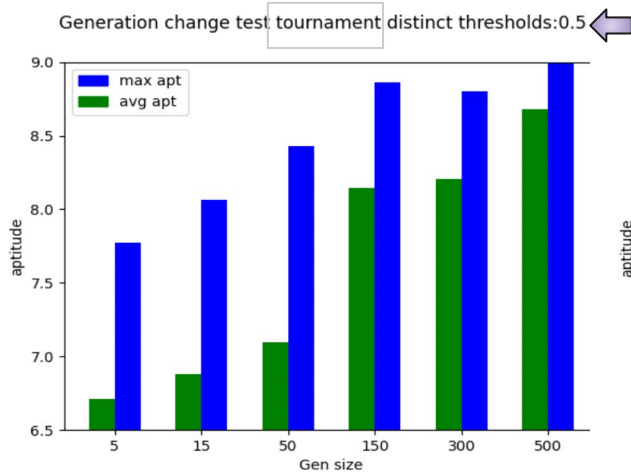
- Con K chico, la cantidad de individuos no óptimos que pueden ser elegidos en la selección aleatoria es mayor. Pero a medida que el K va aumentando, esta cantidad de individuos se reduce por lo que el método de selección se acerca a un comportamiento elitista.

Conclusión parcial: Mayor K  $\rightarrow$  Tiene un comportamiento más 'elitista'  $\rightarrow$  Hay menos variación en los individuos

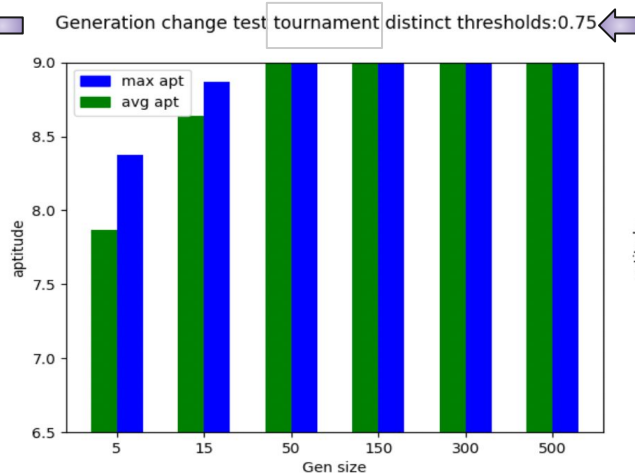




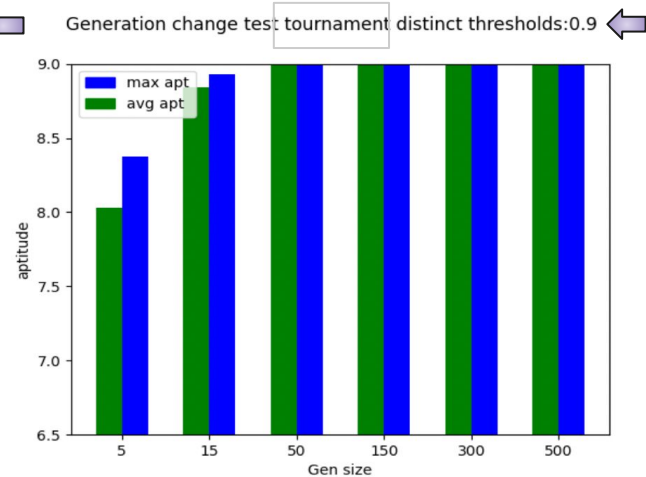
# **Análisis de la** *variación del Threshold en la selección competitiva / Tournament*



$$E_{\text{optimo}} = 5.01 \cdot 10^{-2}$$



$$E_{\text{optimo}} = 2.3 \cdot 10^{-6}$$



$$E_{\text{optimo}} = 4.12 \cdot 10^{-11}$$

**OBS:** tamaño de la muestra = tamaño de la población = 50



## *Variación del Threshold* **en selección competitiva**

Se observa como, el **máximo de las aptitudes** se ve afectado de manera positiva cuanto **mayor threshold**, porque se reduce la cantidad de rondas donde elijo una aptitud peor.

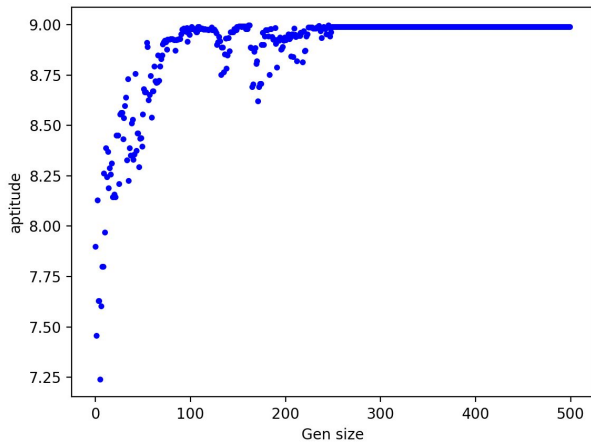
**OBS:**  $\text{threshold} \in [0.5, 1]$

Conclusión parcial: Mayor threshold  $\rightarrow$  Mayor probabilidad de elegir al de mejor aptitud  $\rightarrow$   
Converge más rápido

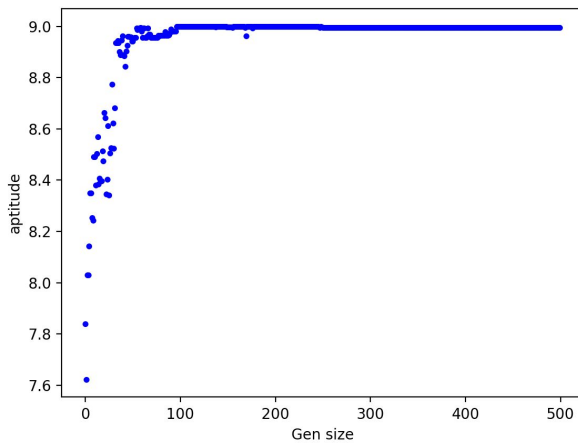


# **Análisis de la** *variación de la temperatura inicial en Boltzmann*

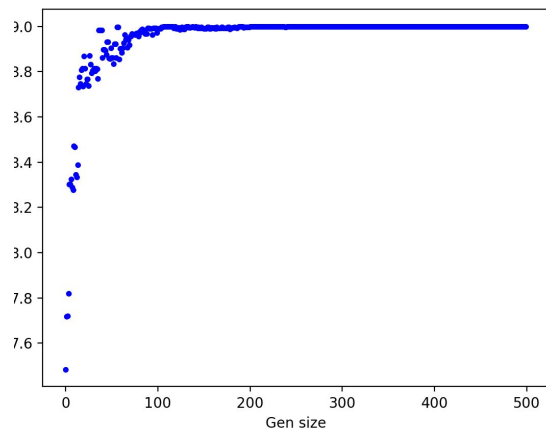
Generation change test Boltzman distinct initial\_temp:100 ←



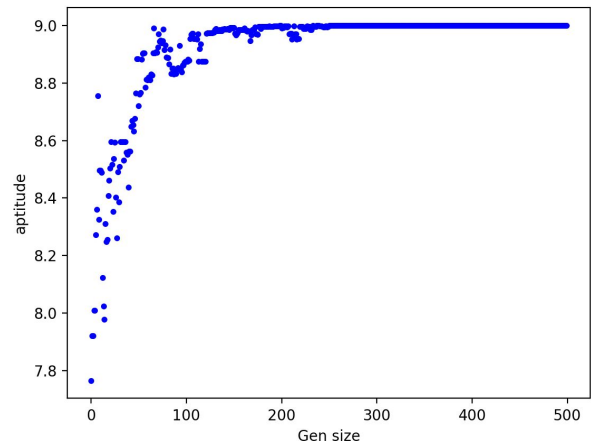
Generation change test Boltzman distinct initial\_temp:80 ←



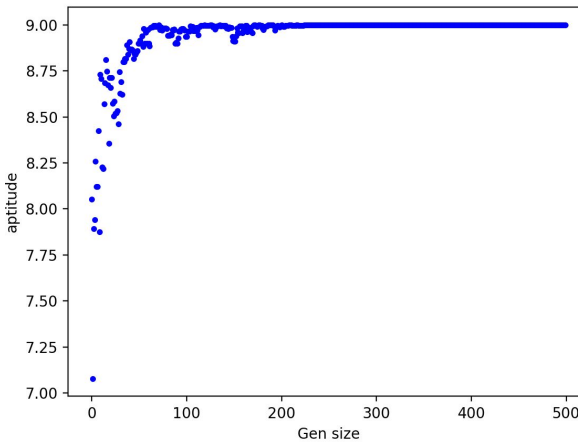
Generation change test Boltzman distinct initial\_temp:60 ←



Generation change test Boltzman distinct initial\_temp:40 ←



Generation change test Boltzman distinct initial\_temp:20 ←



$$VE(i, t) = \frac{e^{f(i, t)/T}}{\sum_{i=1}^P e^{f(i, t)/T}}$$

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



## *Variación de la temperatura inicial* **en Boltzmann**

Comenzando la selección de Boltzmann con valores altos de temperatura inicial, se observa como los resultados tienen mayor variabilidad en cuanto a su aptitud.

En otras palabras, la selección de los individuos inicialmente ocurre de manera más uniforme y a medida que transcurren más generaciones, se torna más elitista. Por eso con temperaturas iniciales bajas, se obtienen mejores aptitudes en pocas generaciones.

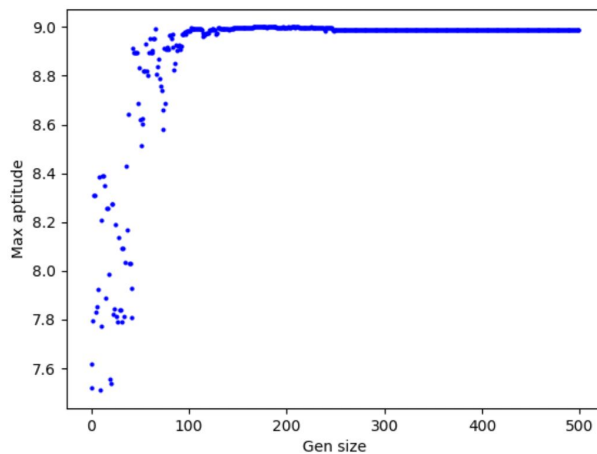
Conclusión parcial: Menor  $T_0$  → Menor dispersión en aptitudes para generaciones iniciales



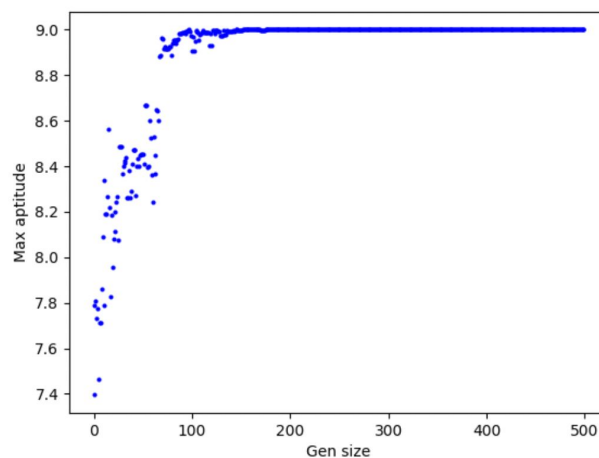
# **Análisis de la** *variación del factor de decrecimiento en Boltzmann*



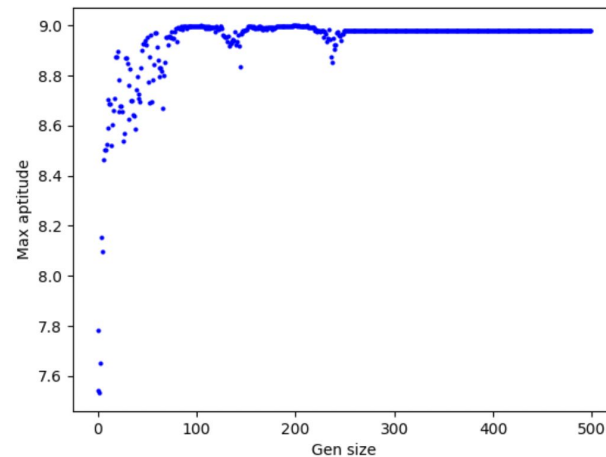
Boltz, To: 50, Tc: 10, decrease\_factor: 0.1




Boltz, To: 50, Tc: 10, decrease\_factor: 0.5



Boltz, To: 50, Tc: 10, decrease\_factor: 1



$$VE(i, t) = \frac{e^{f(i, t)/T}}{\sum_{i=1}^P e^{f(i, t)/T}}$$

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






## *Variación del factor de decrecimiento* **en Boltzmann**

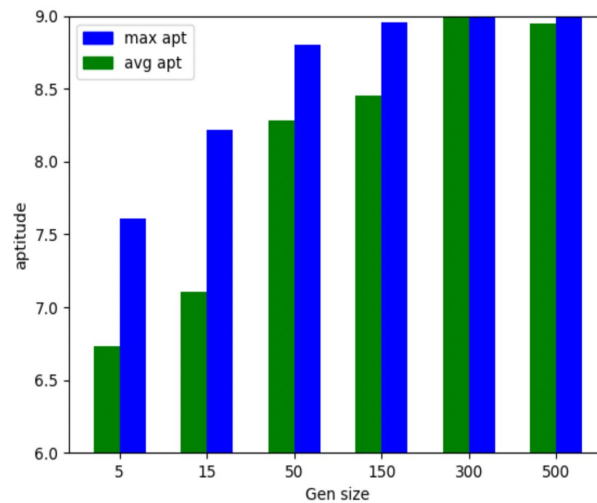
Cambiando los valores que toma el factor de decrecimiento, viendo la ecuación, observamos cómo a mayor valor de  $K$  la temperatura disminuye → se eligen los individuos con mayor aptitud

Conclusión parcial: Mayor  $K$  → Se eligen individuos con mayor aptitud → Converge más acelerada

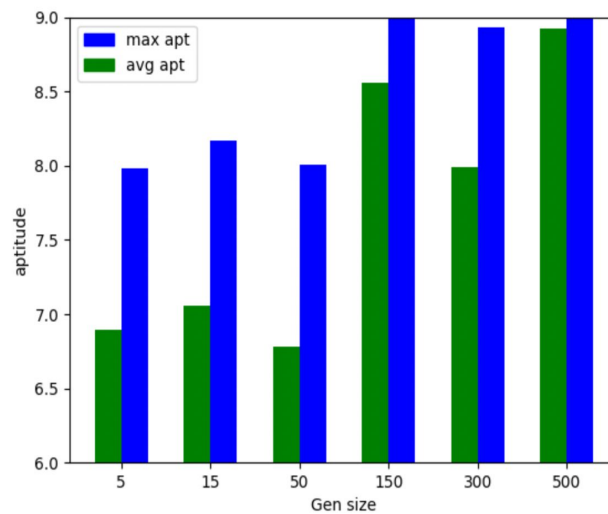


# **Análisis de la** *variación del factor de cambio en Boltzmann*

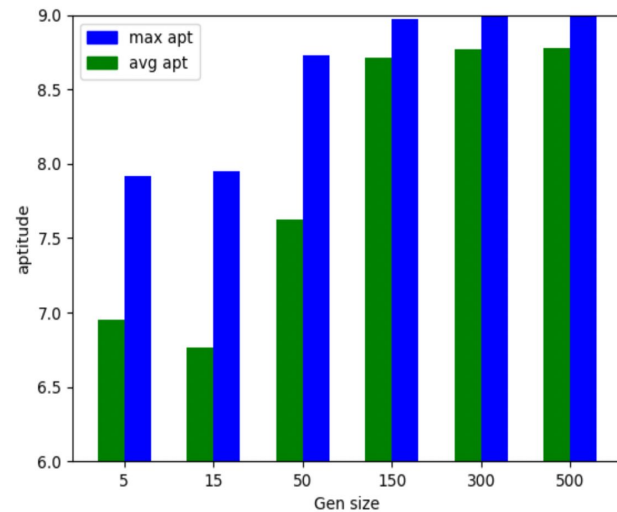
Generation change test Boltzman distinct change\_factor:10 ←



Generation change test Boltzman distinct change\_factor:20 ←



Generation change test Boltzman distinct change\_factor:50 ←



$$VE(i, t) = \frac{e^{f(i, t)/T}}{\sum_{i=1}^P e^{f(i, t)/T}}$$

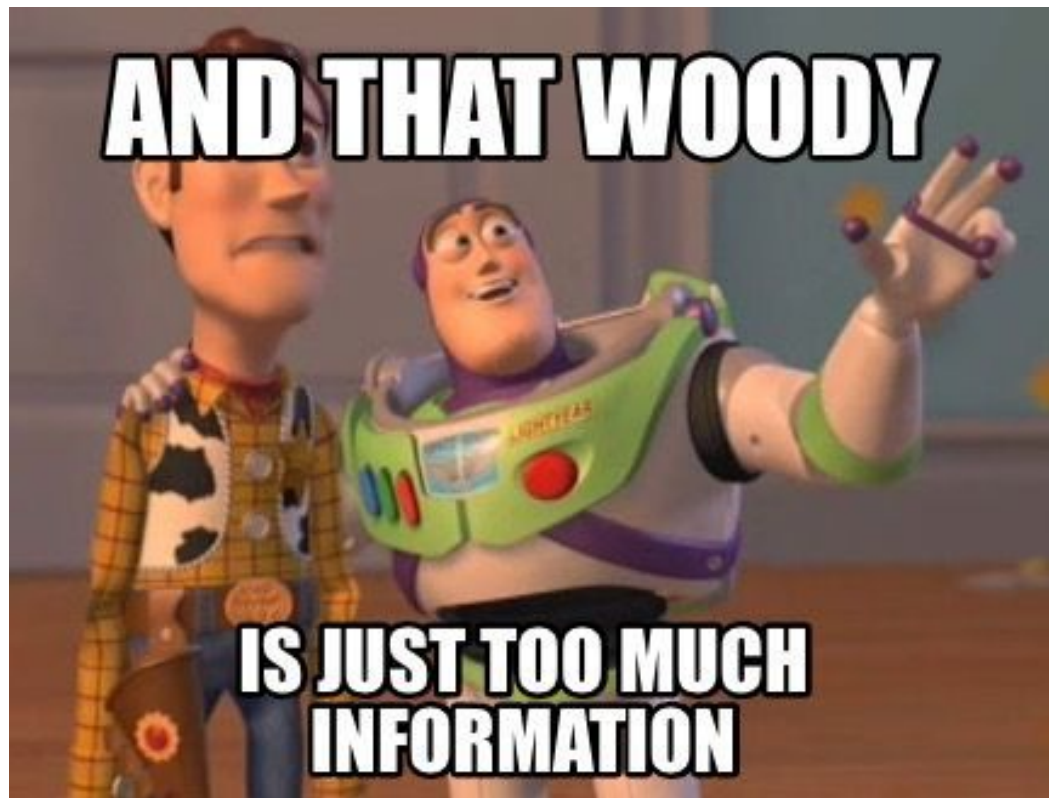
$$T(t) = T_c + (T_0 - T_c)e^{-kt}$$



## *Variación del factor de cambio* **en Boltzmann**

Al aumentar el factor de cambio, se observa como la temperatura se acerca asintóticamente a este valor, provocando que el método de selección nunca llegue a ser demasiado elitista y dejando de esta manera a los menos óptimos dentro de la selección.

Conclusión parcial: Mayor  $T_c \rightarrow$  Hay más diversidad en la población  $\rightarrow$  Converge más retardada



**AND THAT WOODY**

**IS JUST TOO MUCH  
INFORMATION**




## Conclusión

No podemos salir del análisis de nuestros resultados, que estamos solucionando un problema relativamente simple donde el peligro de caer en soluciones malas es muy bajo. Por eso es que premiamos casi siempre parámetros que nos dan una velocidad de convergencia acelerada, y vimos mejores resultados en estos casos.

Pueden existir miles de problemas de mayor complejidad que requieren de una variedad más interesante en la cruce/selección de los individuos, por lo que es fundamental tener todas estas herramientas al alcance de la mano.



# Demo



**Gracias por su  
atencion.**

 [Anexo](#)

