

TP2 Algoritmos genéticos

Sistemas de Inteligencia
Artificial ITBA 2023 Q1

Grupo 4

Integrantes



Gaspar Budó
Berra



Marcos Dedeu



Marcus Galea
Jacobsen

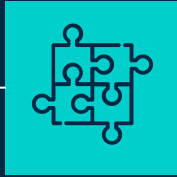


Santiago Hadad



Bruno Squillari

CONTENIDO



01

Ejercicio 1

Representación de
imágenes en ASCII



02

Ejercicio 2

Mezcla de colores

Representación de imágenes en ASCII

Suponemos que todas las imágenes CUADRADAS son blanco y negro.

Genotipo:

- Usamos los caracteres - `_ / \ |` y el espacio en blanco.
- A la imagen original la dividimos en una matriz de $3N \times 3N$ con valores 1 o 0.
- En cada posición de la imagen original que haya color negro, la marcamos con un 1. Si no, la marcamos con un 0.
- Para dibujar el carácter, utilizamos una matriz 3×3 , según los ejemplos de la siguiente diapositiva.

Representación de imágenes en ASCII

Alelos:

0 0 0	0 0 1	1 0 0
1 1 1	0 1 0	0 1 0
0 0 0	1 0 0	0 0 1
Representa -	Representa /	Representa \
0 1 0	0 0 0	0 0 0
0 1 0	0 0 0	0 0 0
0 1 0	1 1 1	0 0 0
Representa 	Representa —	Representa Espacio

Cromosoma ejemplo:

Cromosoma								
0 0 0	0 0 1	1 0 0						
1 1 1	0 1 0	0 1 0						
0 0 0	1 0 0	0 0 1						
0 1 0	0 0 0	0 0 0						
0 1 0	0 0 0	0 0 0						
0 1 0	1 1 1	0 0 0						
0 1 0	0 0 0	0 0 0						
0 1 0	0 0 0	0 0 0						
0 1 0	1 1 1	0 0 0						

Configuración del Algoritmo Genético

Población Inicial: Suponemos una Matriz compuesta aleatoriamente por estas matrices de 3x3.

Fitness: La cantidad de bits que están bien ubicados menos los que están mal ubicados.

Método de Selección: Torneos determinísticos

Cruza: Cruza uniforme.

Mutación: Multigen limitada

Condición de corte: Por tiempo o por la siguiente ecuación, donde epsilon es la similitud a la imagen que queremos alcanzar.

$$-\varepsilon < fitness - MaxFitness < \varepsilon$$

Ejercicio 2

Mezcla de colores

- Colores = vectores rgb.
- Similitud entre colores:
 - Mínima distancia entre los vectores rgb.



Implementación

Genotipo

- Se utilizan las proporciones:

$$Gen = (p_1, p_2, p_3, \dots)$$

$$p_i \in [0, 1] \wedge \sum_i p_i = 1$$

- Combinación lineal:

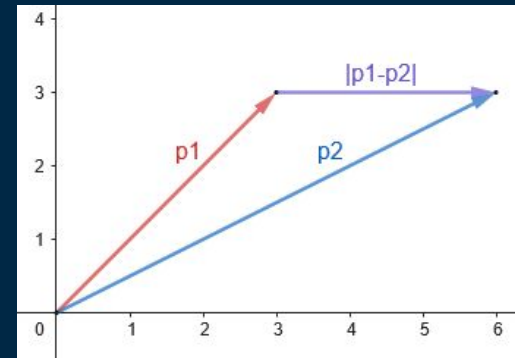
$$(r, g, b) = p_1 \cdot (r_1, g_1, b_1) + p_2 \cdot (r_2, g_2, b_2) + p_3 \cdot (r_3, g_3, b_3) \dots$$

Fitness

Definimos fitness o función de adaptabilidad como:

$$f(\vec{p}_1, \vec{p}_2) = \sqrt{3 \cdot 255^2} - |\vec{p}_1 - \vec{p}_2|$$

Rango de fitness posibles = [0, 461.67]



Algoritmos de selección:

- Elite
- Por ruleta
- Torneos probabilísticos
- Torneos determinísticos

Algoritmos de Mutación:

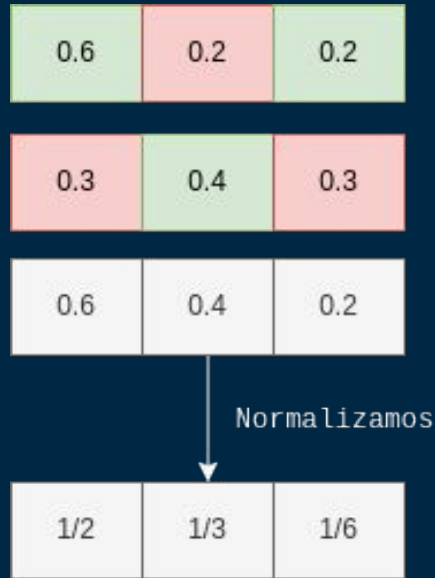
- Multigen limitada
- Multigen uniforme
- Completa

Algoritmo de cruza:

- Cruza uniforme

Aclaraciones para la cruce y la mutación

Cruza:



Mutación:



Condición de corte

★ **Por cantidad de iteraciones:** 2000 iteraciones máximo.

★ **Por estructura:** 25 iteraciones sin cambios → Solución estancada

- Definimos una cota con $\epsilon = 10$:

$$-\epsilon < MaxFitness - PrevMaxFitness < \epsilon$$

Análisis y resultados

Selección + Mutación: Parámetros

Fijos:

- Tamaño de la población: $N = 20$
- Cantidad de hijos en cada iteración: $K = 15$
- Probabilidad de mutación: $P_M = 0.5$
- Cantidad de participantes en los torneos: $N_T = 15$
- Umbral para el torneo probabilístico: $\eta = 0.5$

Selección + Mutación: Parámetros

Fijos:

➤ Paleta de colores:



➤ Color objetivo:

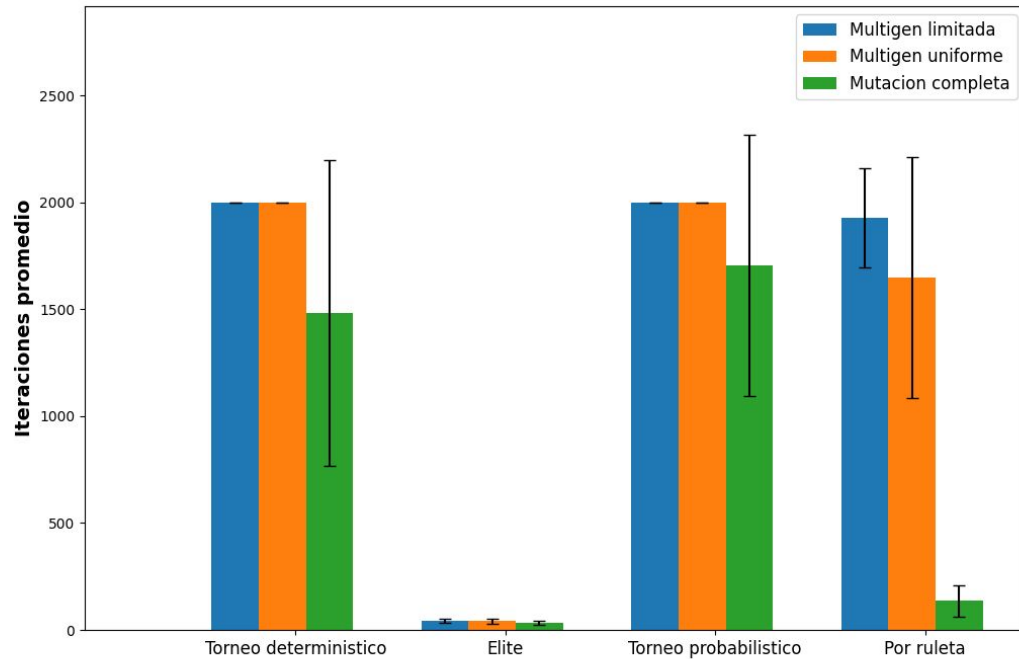


Selección + Mutación: Parámetros

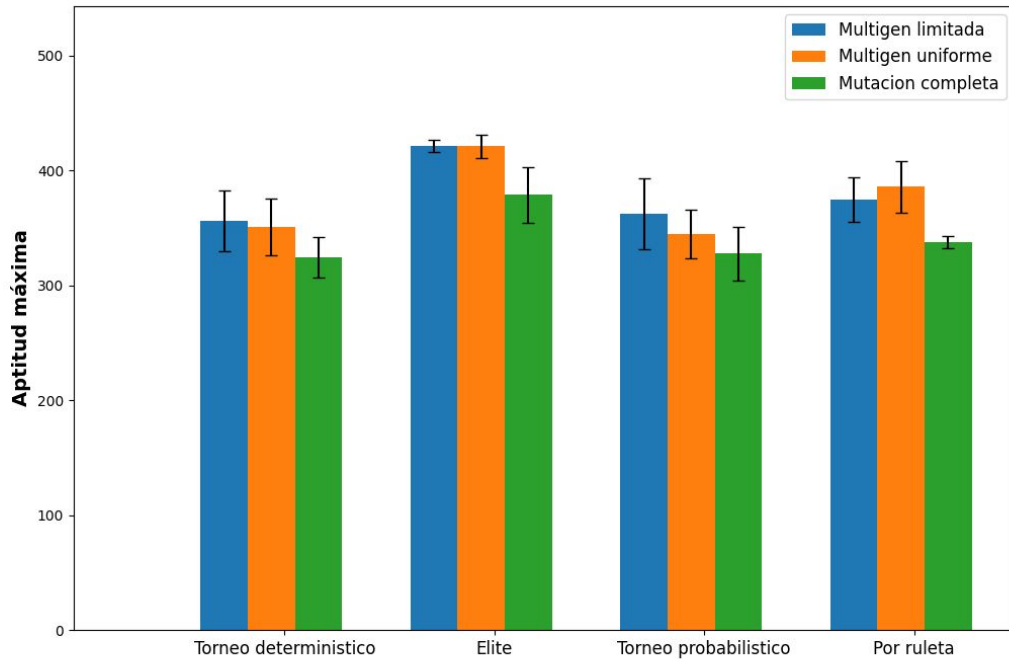
Variables:

- Métodos de selección
- Métodos de mutación

Resultados: Cantidad de Iteraciones



Resultados: Aptitud máxima



Resultados: Ejemplo visual

Objetivo

Elite & Uniform mutation

Probabilistic Tournament & Complete

Selección + Mutación: Conclusiones

- ★ Mejores resultados con Elite.
- ★ Sin máximos locales, si no uno global.
- ★ La mutación multigénica limitada obtiene mejores resultados.

Población + Hijos generados: Parámetros

Parámetros fijos:

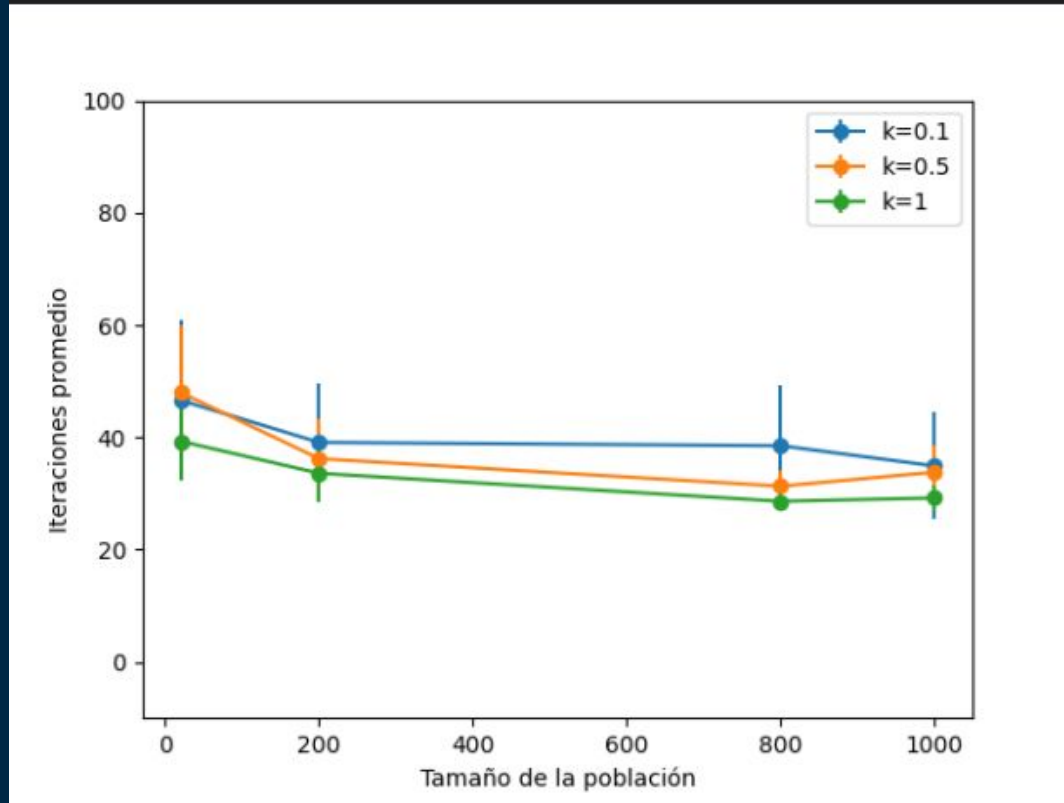
- Probabilidad de mutación: $P_M = 0.5$
- Selección Elite
- Multigen multigénica limitada
- Misma paleta inicial y color objetivo.

Población + Hijos generados: Parámetros

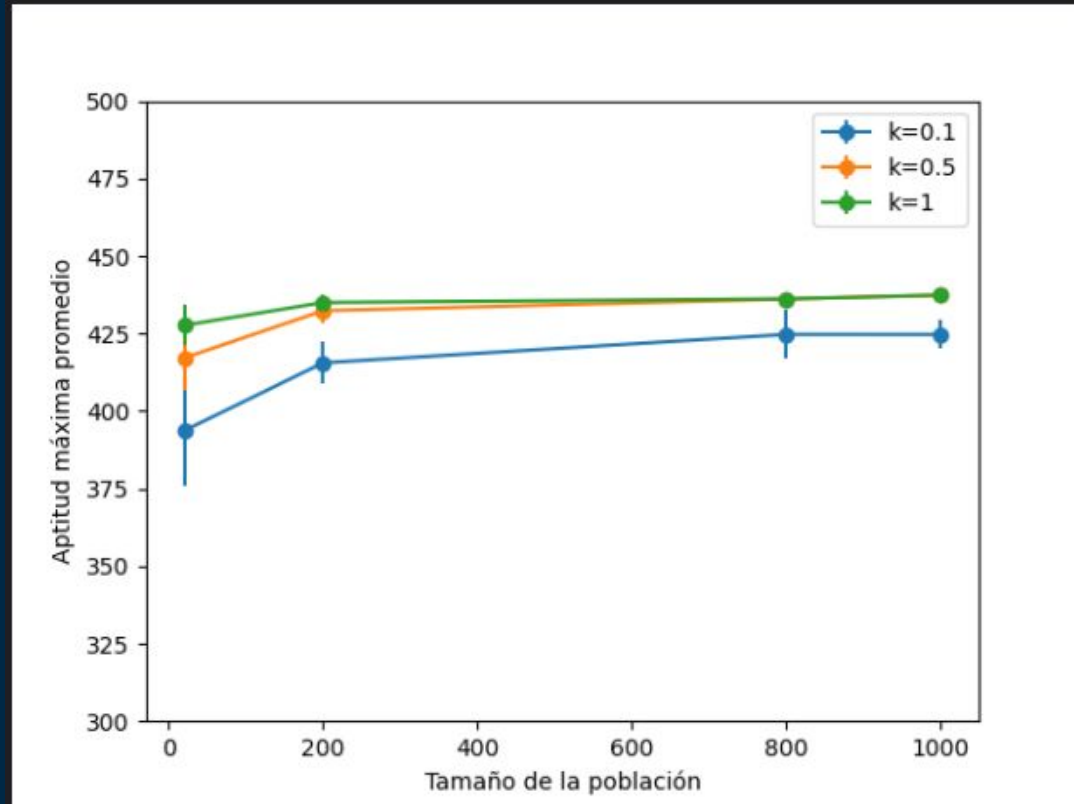
Parámetros variables:

- Tamaño de la población:
 - $N \in [20, 200, 800, 1000]$
- Cantidad de hijos generados:
 - $K \in [N * p : p \in [0.1, 0.5, 1]]$

Población + Hijos: resultados



Población + Hijos: resultados



Poblacion + Hijos: Conclusiones

- A mayor cantidad de hijos generados, menor cantidad de iteraciones.
- Incrementar el tamaño de la población incrementa la aptitud máxima.
- A partir de cierto tamaño tiende a ser constante.

Probabilidad de mutación: Parámetros

Parametros fijos:

- Selección Elite
- Multigen uniforme
- Misma paleta inicial y color objetivo.
- Tamaño de la población: $N = 200$
- Cantidad de hijos en cada iteración: $K = 200$

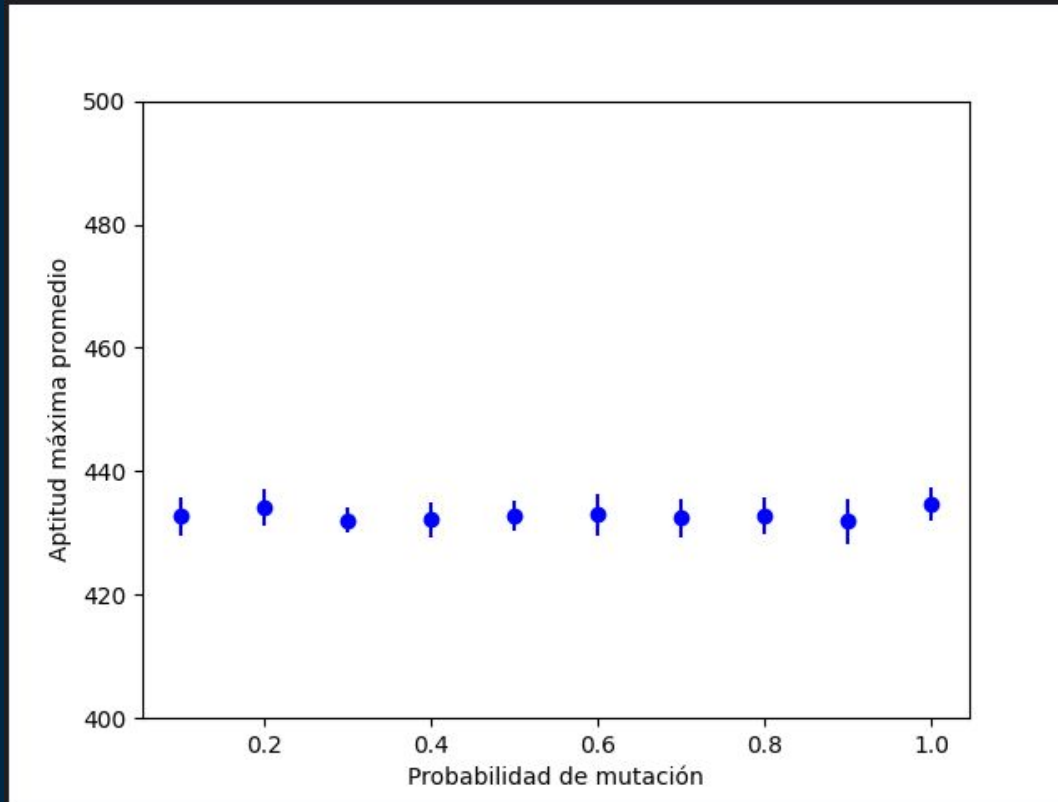
Probabilidad de mutación: Parámetros

Parametros Variables:

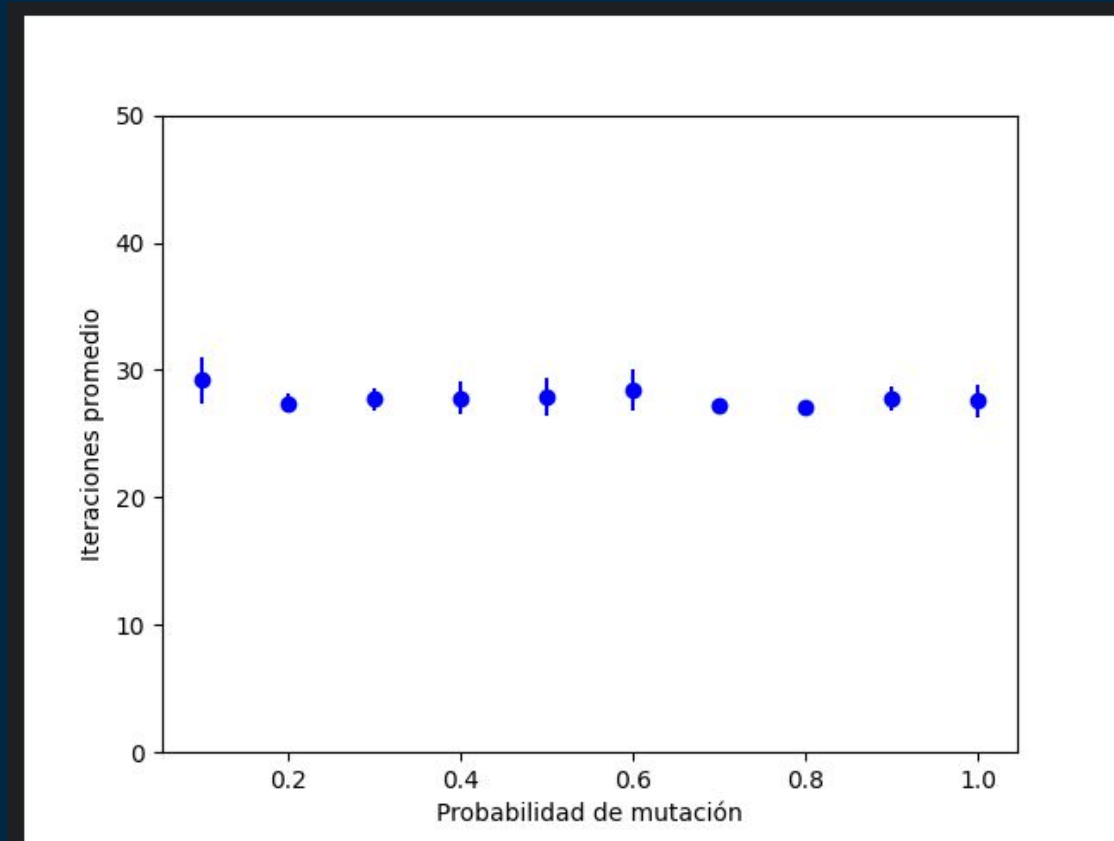
- Probabilidad de mutación

$$P_M \in \{0.1, 0.2, \dots 1\}$$

Probabilidad de mutación: Resultados



Probabilidad de mutación: Resultados



Probabilidad de mutación: Conclusiones

- ★ Variar la probabilidad de mutación usando elite, no tiene un gran impacto.