

MÉTODOS DE BÚSQUEDA Y ALGORITMOS GENÉTICOS

TP 1

Sistemas de Inteligencia Artificial 2C-2022

Grupo 1 - "Augusta Ada King"

Domingues Paula

Donikian Gastón

Pavan Matias

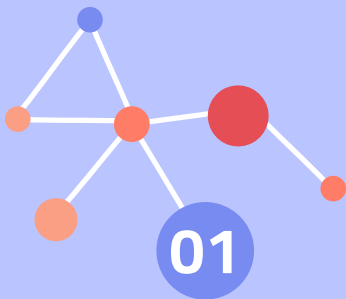
Rodriguez Manuel

60158

60067

58296

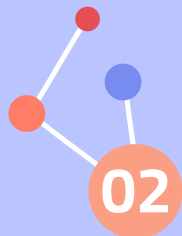
60258



MÉTODOS DE BÚSQUEDA

8-Puzzle

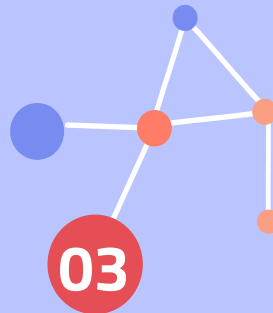
Posible implementación,
heurísticas admisibles y
métodos de búsqueda.



MÉTODOS DE BÚSQUEDA

Cubo rubik

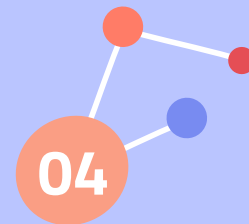
Implementación,
heurísticas admisibles,
métodos de búsqueda,
heurísticas y análisis de
resultados.



ALGORITMOS GENÉTICOS

ASCII Art

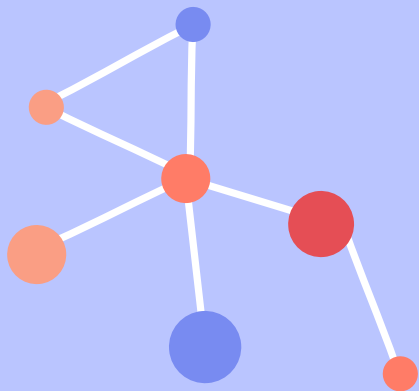
Posible implementación,
función de fitness y
métodos de optimización.



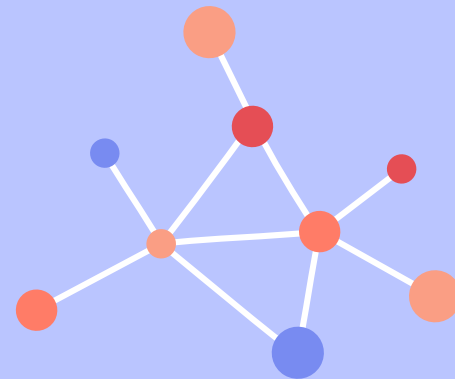
ALGORITMOS GENÉTICOS

Color Palette

Implementación,
opciones de selección,
crossovers, mutaciones y
análisis de resultados.

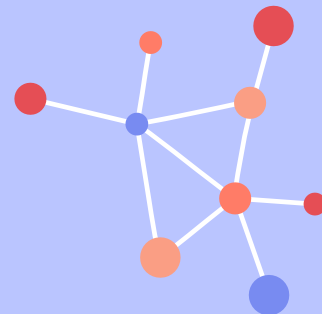


01



MÉTODOS DE BÚSQUEDA

8 Puzzle





8 Puzzle

Descripción

Ejemplo de input

5	7	3
8	2	
1	6	4

Mover el "4"



5	7	3
8	2	4
1	6	

Mover el "6"

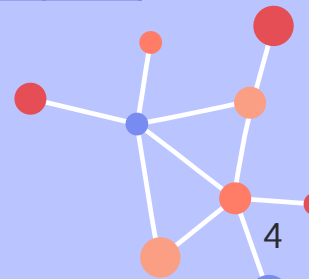


(...)



1	2	3
8		4
7	6	5

Solución





8 Puzzle

Estructura de estado

5	7	3
8	2	
1	6	4



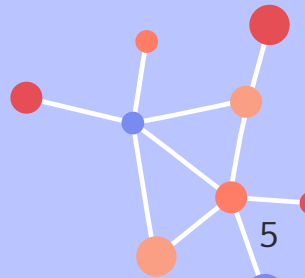
573820164

1	2	3
8		4
7	6	5



123804765
(estado solución)

Cada estado se corresponde con un id único el cual se encuentra determinado por la posición de los números en el tablero.





8 Puzzle

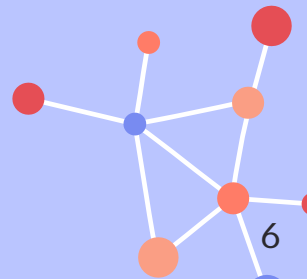
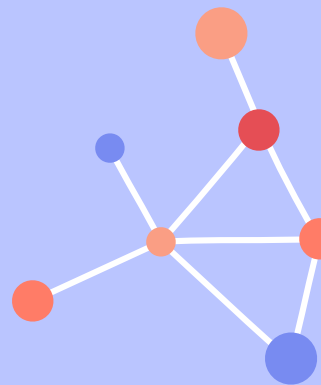
Movimientos posibles

5	7	3
8	2	
1	6	4

Se puede pensar que los movimientos posibles es el espacio en blanco, moviéndose en direcciones del vecindario Von Neumann ($r=1$).

1	2	3
8		4
7	6	5

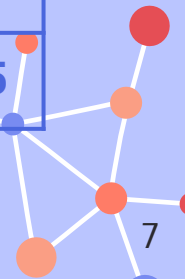
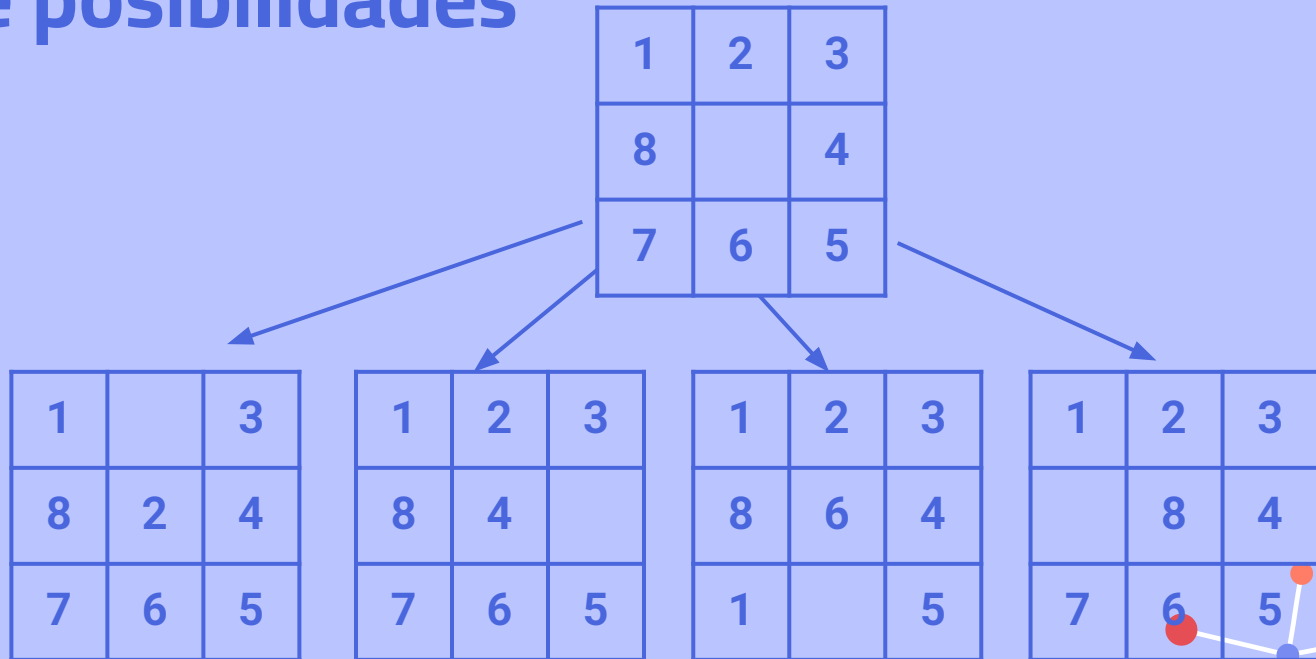
5	7	
8	2	3
1	6	4





8 Puzzle

Árbol de posibilidades





8 Puzzle

Heurísticas admisibles

Distancia Hamming

1	2	3	8	0	4	7	6	5
5	7	3	8	2	0	1	6	4

Consideramos la diferencia en las posiciones de su número identificador. Sería la cantidad de baldosas, distintas del vacío, que no se encuentran en su posición final.

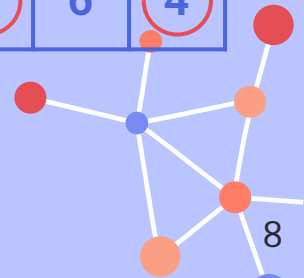
En este caso, el valor es $h(e) = 5$.

Solución

1	2	3
8	0	4
7	6	5

Input

5	7	3
8	2	0
1	6	4





8 Puzzle

Heurísticas admisibles

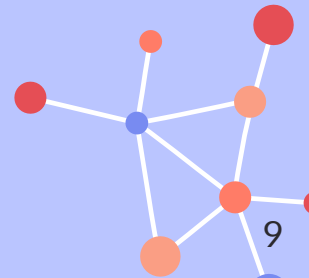
Distancia Manhattan

5	7	3
8	2	
1	6	4

Consideramos a cuántas baldosas de distancia se encuentra del estado final con norma 1.

Por ejemplo: el número 5 se encuentra a 4 baldosas de distancia de su posición final.

La heurística será la sumatoria de las distancias Manhattan de cada baldosa

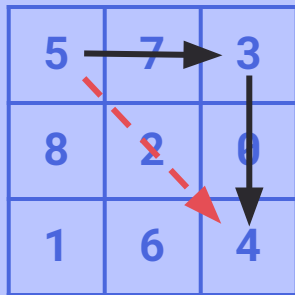




8 Puzzle

Heurísticas admisibles

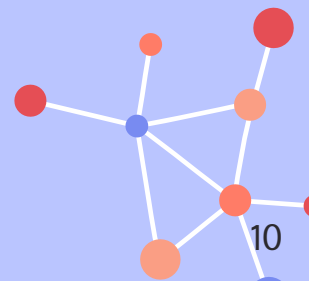
Distancia Euclideana

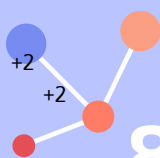


Consideramos a cuántas baldosas de distancia se encuentra del estado final con norma 2.

Por ejemplo: el número 5 se encuentra a $\sqrt{2^2 + 2^2} = \sqrt{8}$ baldosas de su posición final.

La heurística será la sumatoria de las distancias Euclidianas de cada baldosa a su posición destino.





8 Puzzle

Heurísticas admisibles

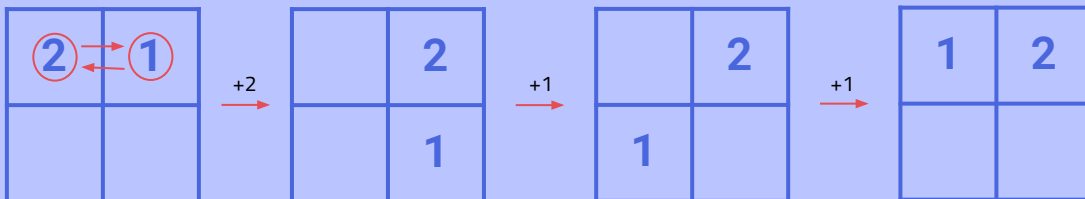
Distancia Manhattan + Reversal Penalty Search

②	①	3
8	7	
5	6	4

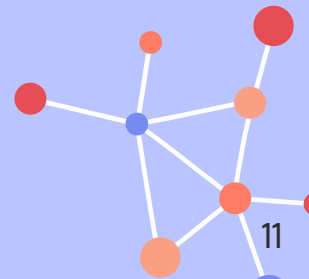
Es el cálculo de la heurística Manhattan + sumar 2 por cada par de baldosas mal-posicionadas adyacentemente.

$$h(e) = \sum distances + 2 \cdot \#reverse_pairs$$

Mejor escenario (imposible en el juego, es una subestimación)



$$\text{Total} = 4 = 2 * 2$$





8 Puzzle

Método de búsqueda no informado

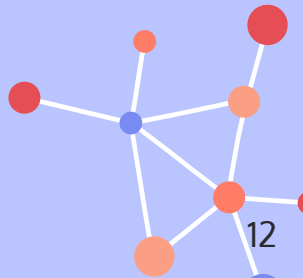
IDDFS (manteniendo el árbol)



Es **completo**.



Tomando un **step relativamente bajo** (supongamos 4), cuando el camino a la solución se encuentre cerca a un múltiplo a este step resolverá más rápido por funcionar como un **DFS limitado**.





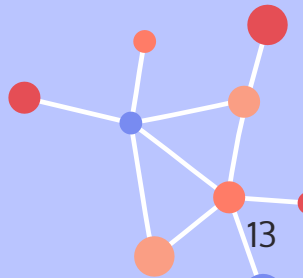
8 Puzzle

Método de búsqueda informado

A* con Distancia Manhattan + Reversal Penalty Search como heurística

Sabiendo que **A*** es realmente bueno, en la mayoría de los casos, cuando se utiliza una **buena heurística**, como es el caso de la propuesta, consideramos que es la **mejor opción** para solucionar el puzzle.

Además de dar una **respuesta rápida**, como la heurística es admisible, A* resulta ser **óptimo**.



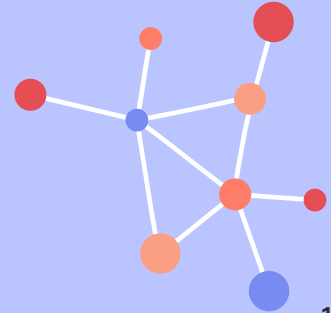


02



MÉTODOS DE BÚSQUEDA

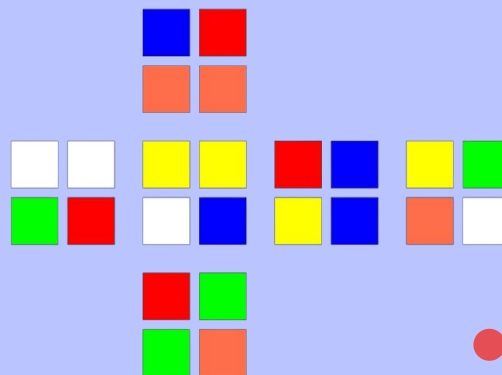
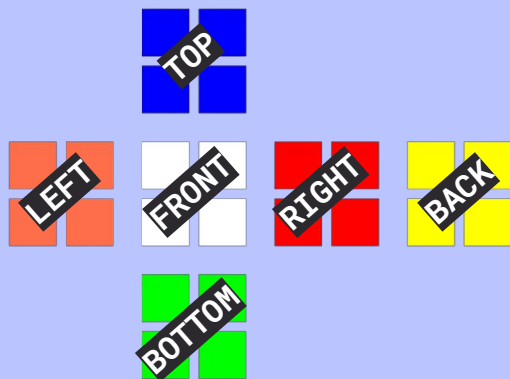
Cubo Rubik



Cubo Rubik

Estructura de estados

Un **estado** se representa por una lista de listas, que representa cada cara del cubo.



Estado ejemplo (Cubo resuelto):

[[0,0,0,0],[1,1,1,1],[2,2,2,2],[3,3,3,3],[4,4,4,4],[5,5,5,5]]

FRONT

TOP

LEFT

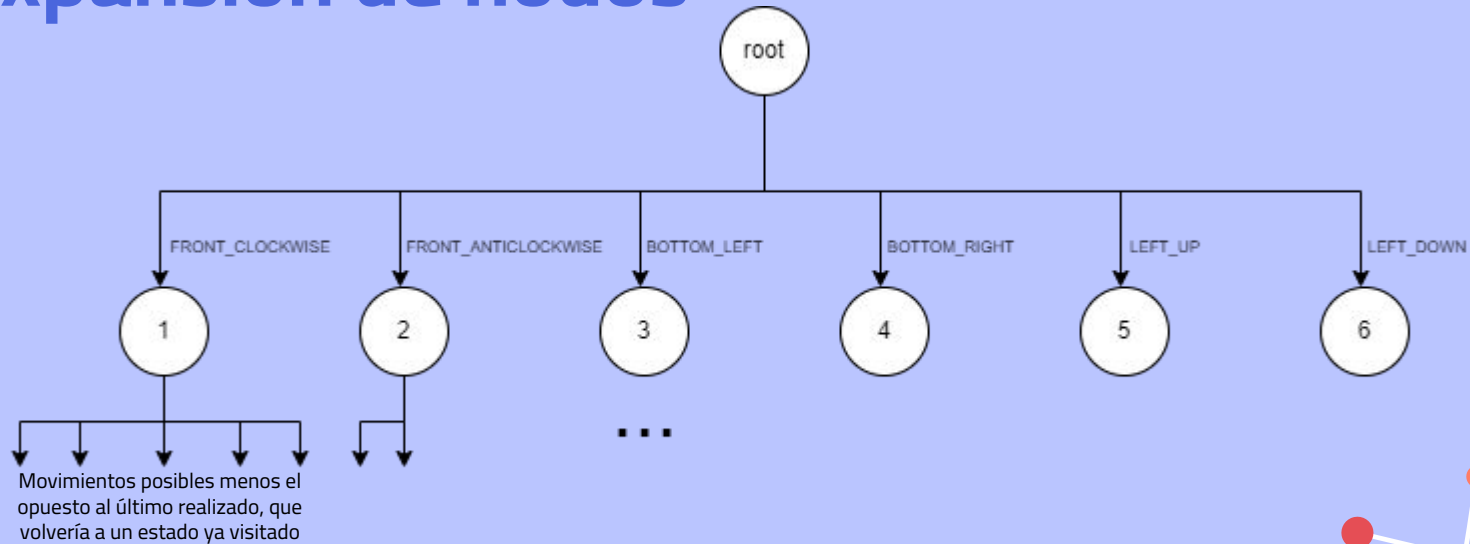
BOTTOM

RIGHT

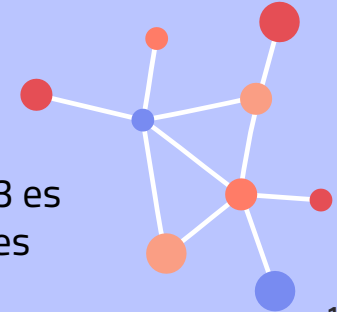
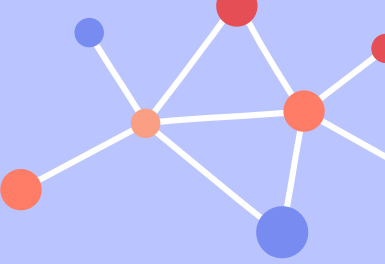
BACK

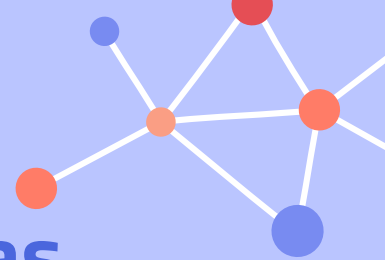
Cubo Rubik

Expansión de nodos



Este grafo enuncia los movimientos para el cubo 2x2x2, para el caso de 3x3x3 es análogo sólo que existen 12 movimientos posibles por lo que la ramificación es mucho mayor.





Cubo Rubik

Heurísticas admisibles implementadas

1. Colores

Cuenta la **cantidad de colores** erróneamente ubicados en todas las caras.

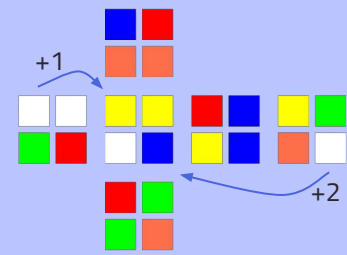


$$h(e) = \frac{3 + \dots}{4 * n}$$

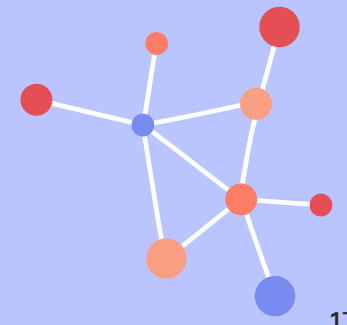
Máxima cantidad de colores que cambian de cara en un movimiento

2. Simple manhattan

Cuenta la **distancia** de los colores erróneamente ubicados hacia su cara.



$$h(e) = \frac{\sum dist}{4 * n}$$





Cubo Rubik

Benchmarks

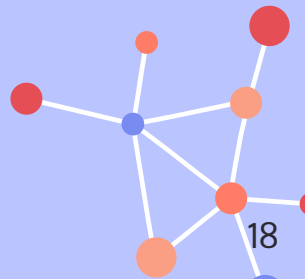
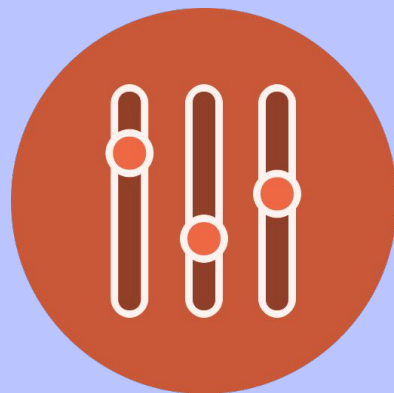
Métodos de búsqueda

Parámetros

Timeout de 90 segundos

Escenarios elegidos:

1. **Dificultad baja:** 2 movimiento a solución
2. **Dificultad media:** 6 movimientos a solución
3. **Dificultad alta:** 9 movimientos a solución



Cubo Rubik

Benchmarks

Resultados

Los siguientes algoritmos no encontraron solución en el tiempo establecido para los casos con más de 4 movimientos:



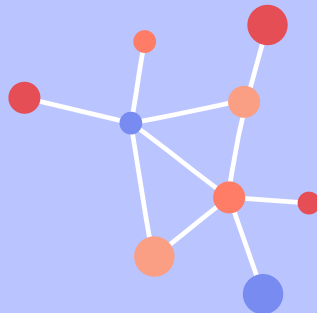
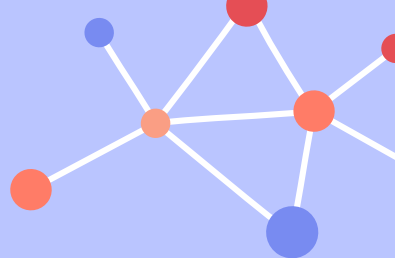
DFS



LOCAL Y GLOBAL GREEDY

Razones para este comportamiento:

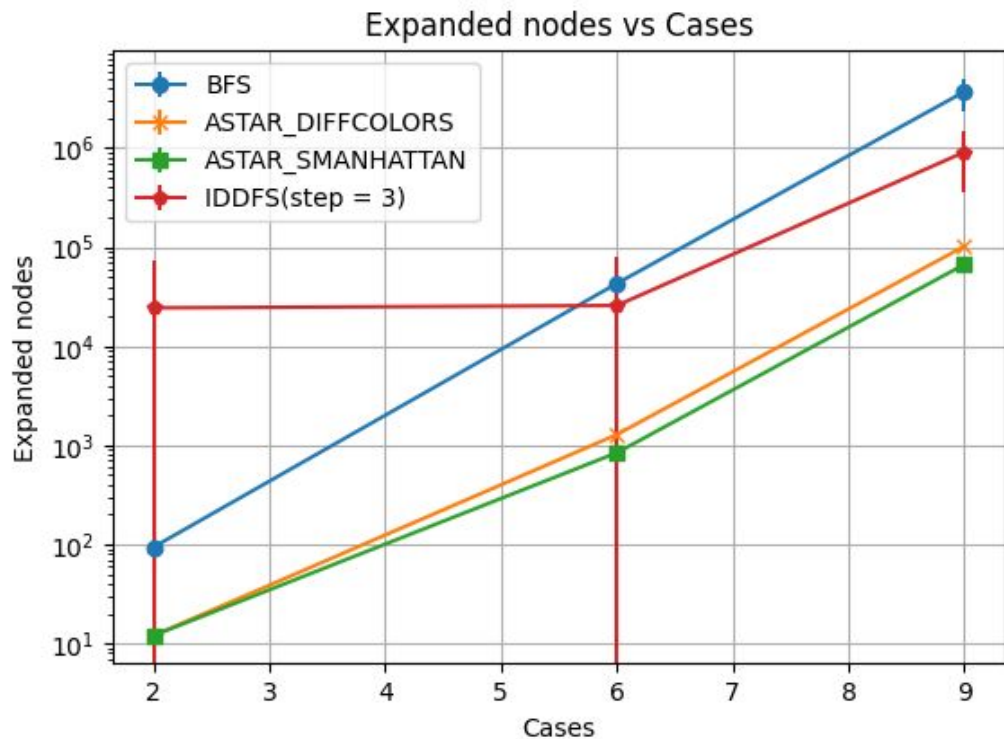
1. Gran ramificación en los nodos.
2. Gran cantidad de estados posibles.
3. Simplicidad de heurísticas.



Cubo Rubik N=2

Benchmarks

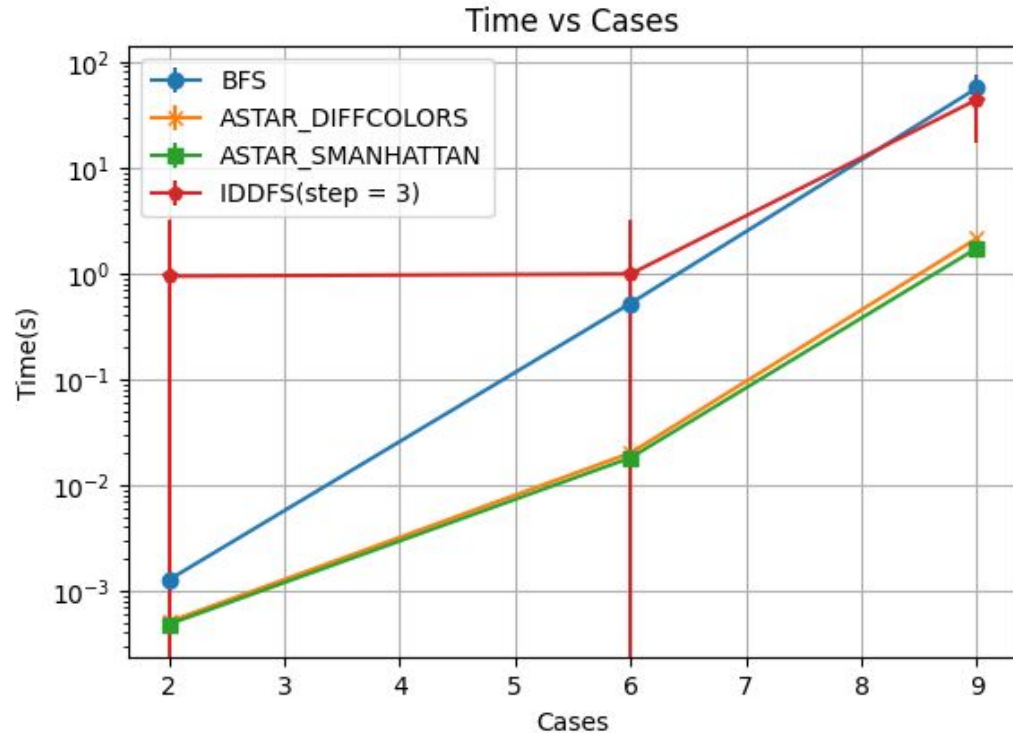
Resultados



Cubo Rubik N=2

Benchmarks

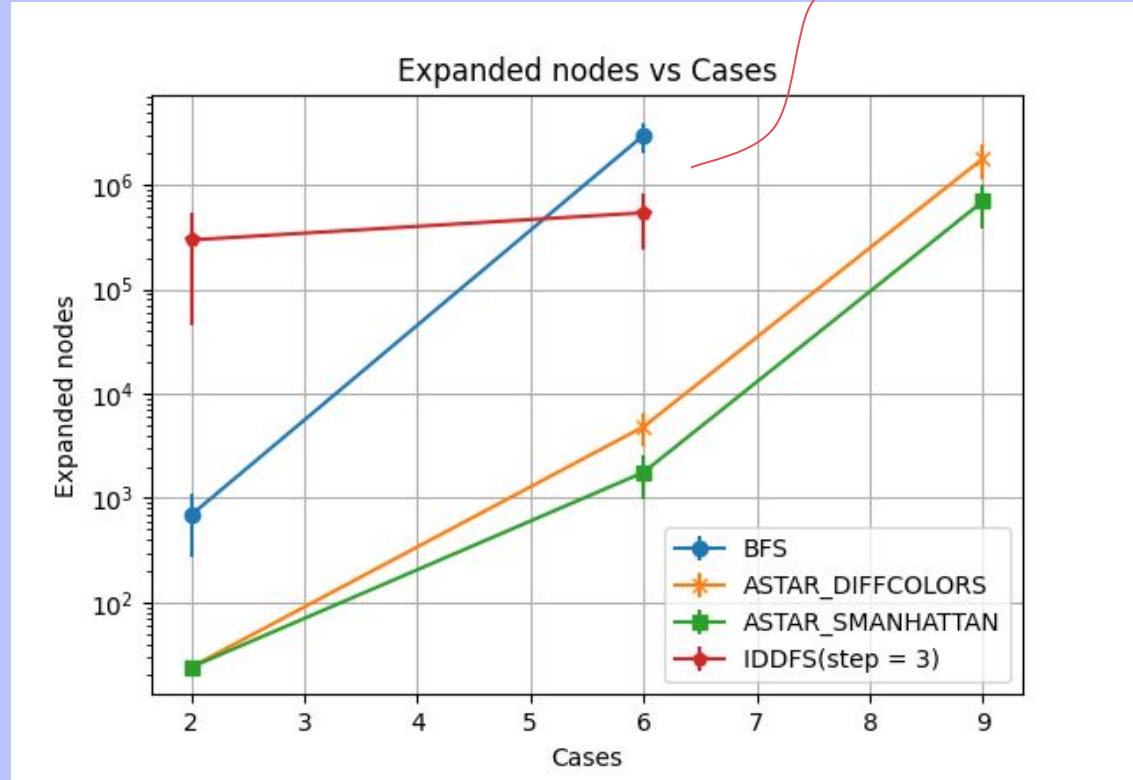
Resultados



Cubo Rubik N=3

Benchmarks

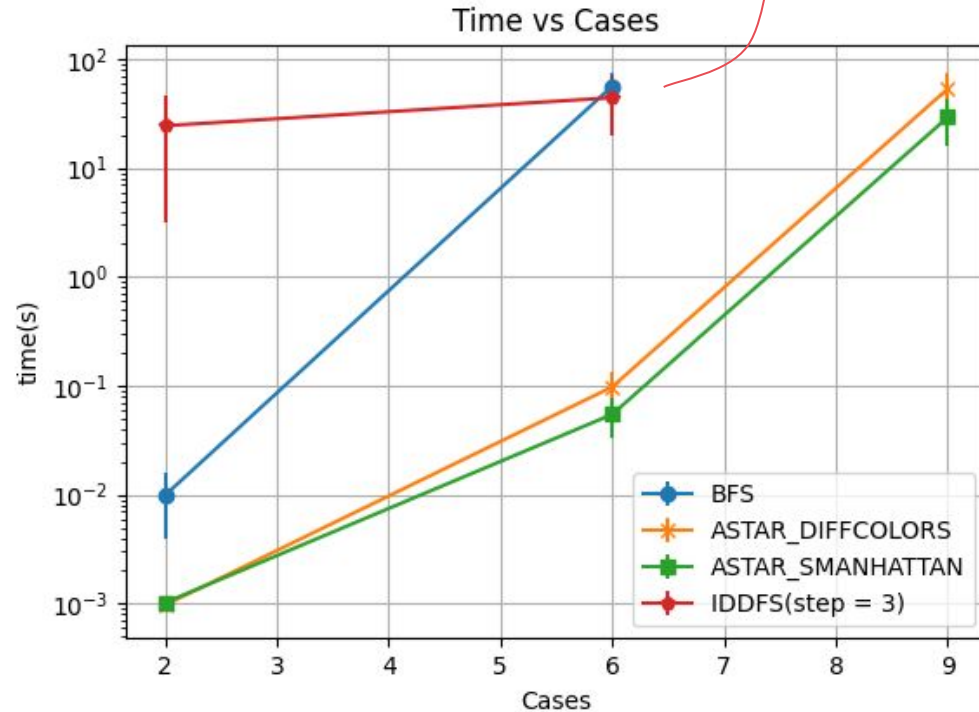
Resultados



Cubo Rubik N=3

Benchmarks

Resultados

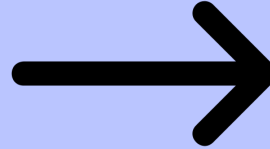


Métodos de búsqueda

Conclusiones



Información



**Menos tiempos de
búsqueda**



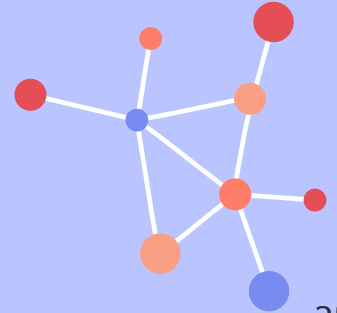
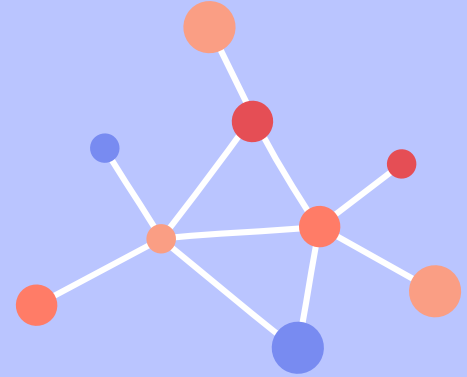
Complejidad



03

ALGORITMOS GENÉTICOS

ASCII Art

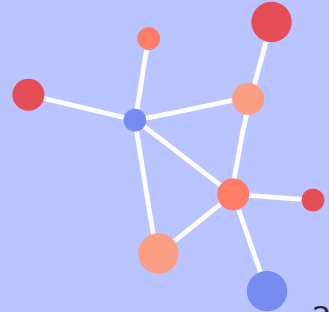
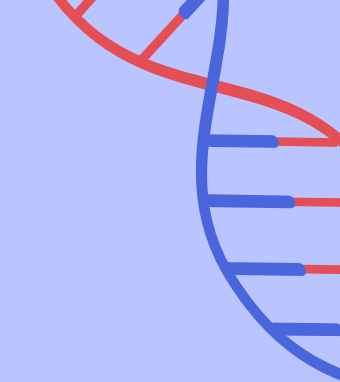


ASCII Art

Estructura de individuo

Matriz NxN donde cada elemento es un carácter ASCII y a su vez un gen

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N} \\ a_{2,1} & \dots & \dots & \dots \\ \dots & \dots & \dots & a_{N-1,N-1} \\ \dots & \dots & a_{N,N-1} & a_{N,N} \end{pmatrix}$$

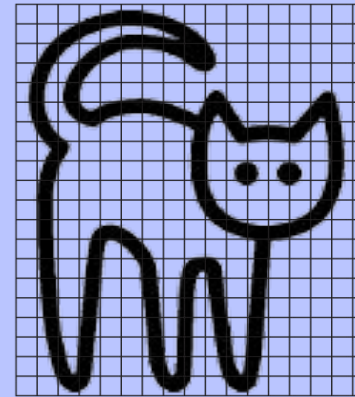


ASCII Art

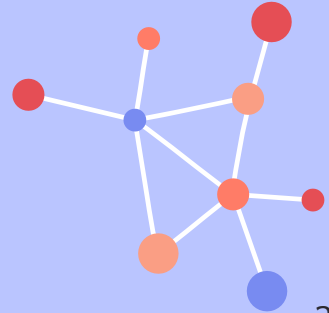
Función de fitness

Cada carácter ASCII tiene una imagen asignada y precomputada.
Se compara cada píxel de la imagen input con la imagen del ASCII correspondiente a dicha celda, contando los píxeles que matchean.

$$f(i) = \frac{\#matches(i)}{\#pixeles}$$



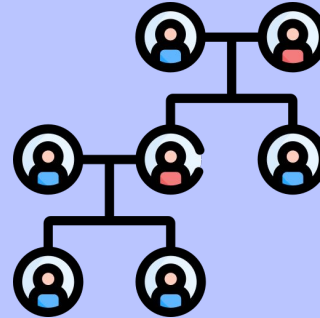
NxN



ASCII Art

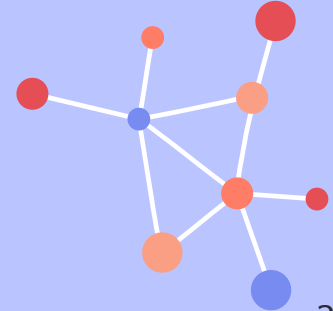
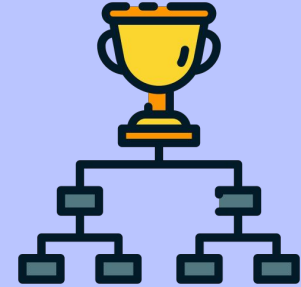
Selección

Se seleccionan mediante el muestreo estocástico de torneos probabilísticos, ya que al ser tantos genes y tantos alelos las posibilidades son muchas por lo que la velocidad es clave.



Crossover

Se elige **Cruce uniforme** para evitar correlación posicional.





ASCII Art

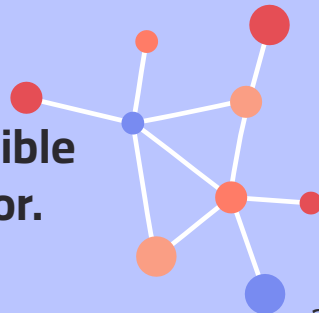
Mutación

Utilizará una mutación multigen uniforme, donde los genes mutados tomaron otro valor aleatorio. La probabilidad de que sean mutados sería alrededor del 0.05%.



Brecha generacional

Se debería conservar un 10% de los individuos en el cambio de generación. Esto se decide por que al ser tanta la diversidad es posible que la nueva generación de mucho peores resultados que la anterior.

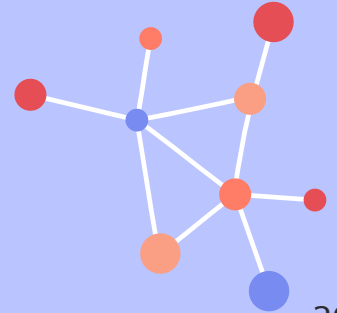
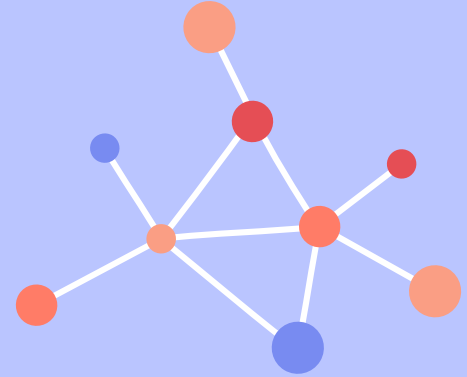




04

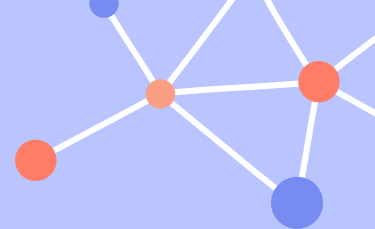
ALGORITMOS GENÉTICOS

Color Palette



Color Palette

Estructura



Población de Individuos



Compuesto por M individuos.

Array de Colores disponibles



Compuesto por N colores con
 K pigmentos (float)

Ejemplo RGB:
 $C = [[1, 0, 0], [0, 1, 0], [0, 0, 1]]$
 $N = 3, K = 3$

Color Palette

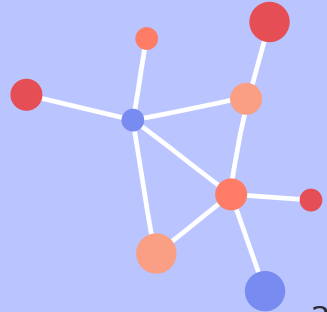
Estructura de un individuo



$$Member = [a_1, a_2, \dots, a_K]$$

Cada gen indica el coeficiente que se aplica a cada pigmento de la paleta para conseguir un color:

$$C = \min \left(1, \sum_i^K a_i \cdot C_i \right)$$



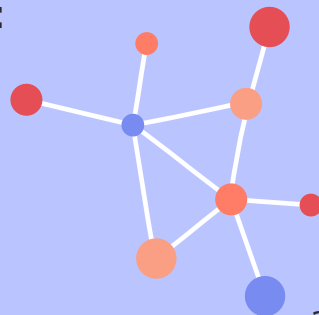
Color Palette

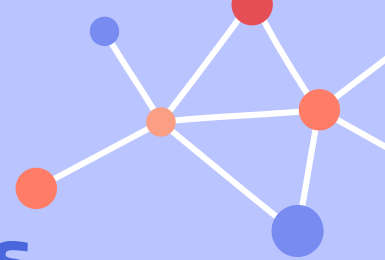
Función de fitness

$$dist(i) = \left\| target - \left[\sum_i^N a_{1,i} \cdot C_1, \dots, \sum_i^N a_{K,i} \cdot C_K \right] \right\|$$

Luego, como buscamos que la función de fitness sea **creciente**, aplicamos:

$$f(i) = e^{-1 \cdot dist}$$

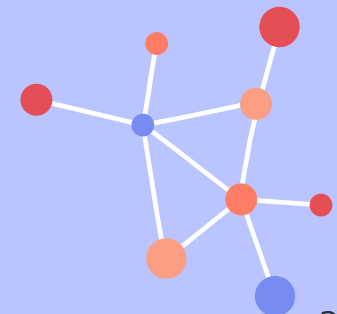




Color Palette

Métodos de Selección implementados

1. **Elite:** Se seleccionan los primeros K individuos del conjunto, ordenados por fitness.
2. **Roulette:** A partir de las aptitudes relativas, se eligen estocásticamente un número K de individuos.
3. **Selección por Torneos**
 - a. **Determinísticos:** se elige el mejor de una selección de M individuos.
 - b. **Probabilísticos:** se toman dos individuos y a partir de un threshold se elige el individuo con mayor o menor fitness.



Color Palette

Condición de Corte



Cantidad máxima de iteraciones:

Se finaliza la ejecución tras, por default, 4000 generaciones y se devuelve las proporciones con mejor fitness.



Solución aceptable:

Se finaliza la ejecución cuando se asemeja al target en 3 o más decimales.



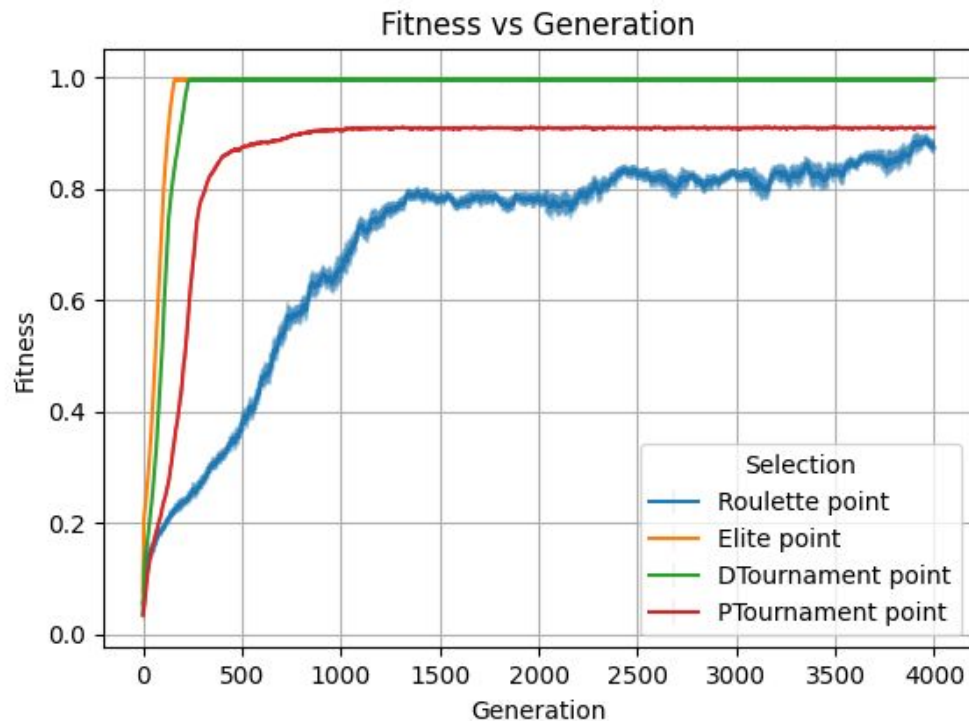
Estado Estacionario:

Se finaliza la ejecución luego de que no haya cambiado el mejor fitness encontrado tras $\frac{1}{4}$ de la cantidad máxima de generaciones posibles.

Color Palette Benchmarks

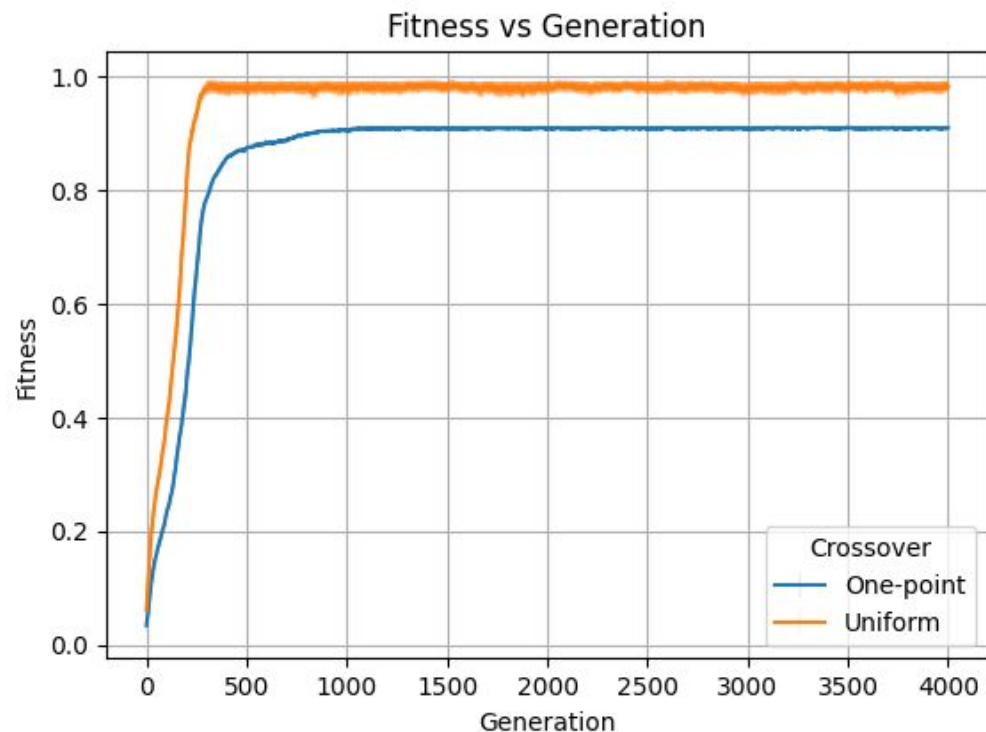
Parametros:

Crossover = Point crossover

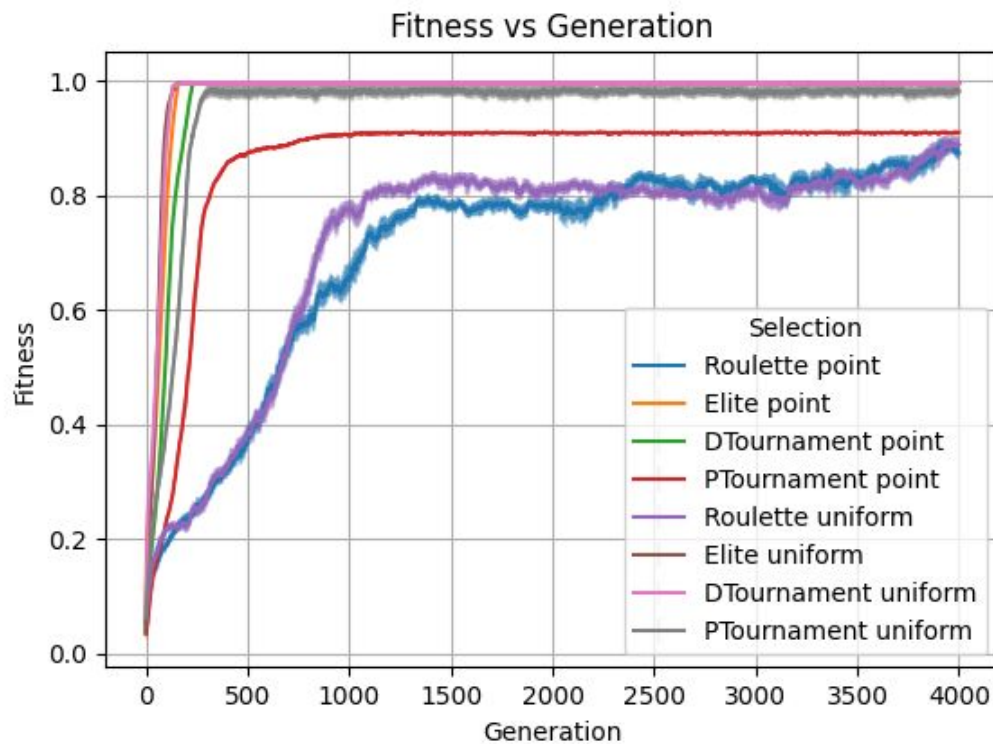
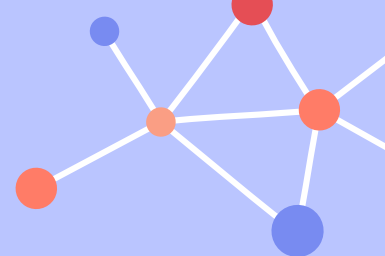


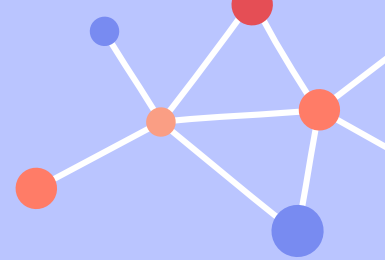
Color Palette Benchmarks

Parametros:
Selection = Probabilistic
tournament



Color Palette Benchmarks





Color Palette

Métodos más adecuados

Selección: Deterministic tournament

Mantiene un alto grado de **diversidad** y **performance**.

Crossover: Uniform

No mantiene correlación lineal entre alelos.

Mutación: Multigen Uniforme

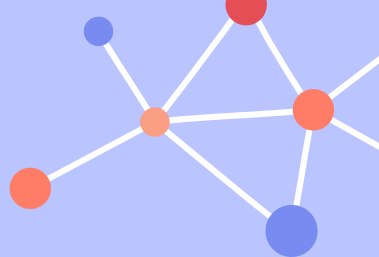
Permite la **mayor diversidad** posible, necesario para **evitar convergencia prematura**.

Se pudo observar en los benchmarks que esta es la mejor configuración.



Algoritmos genéticos

Conclusión



Se debe mantener una **diversidad alta** para **evitar** converger a un **máximo local**.



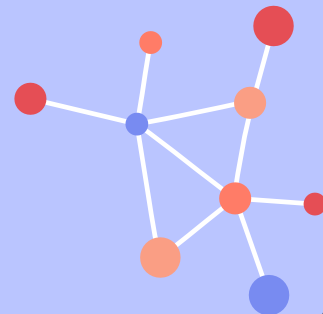
La forma de la función de **fitness** define cuáles podrían ser los métodos más adecuados a utilizar en la **optimización**.



Tener **varios factores de corte** en cuenta combinados permiten **no** procesar **iteraciones innecesarias**.



La **población inicial** tiene un **gran impacto** en la ejecución.



The background features a stylized DNA double helix with blue and red strands. In the bottom left corner, there is a small molecular structure with four colored spheres (blue, orange, red, blue) connected by white lines.

¡GRACIAS!
¿PREGUNTAS?