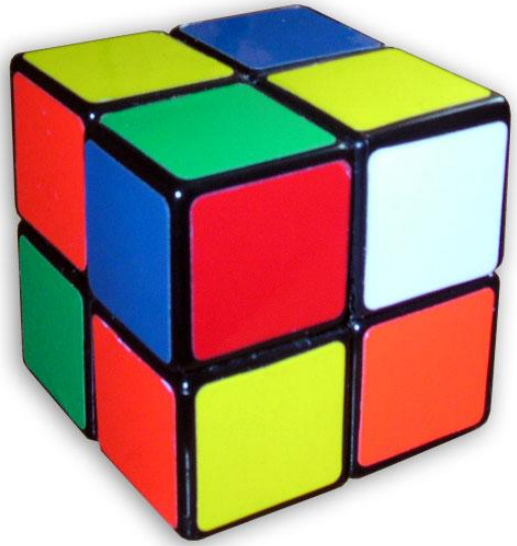
Métodos de búsqueda para cubo de Rubik 2x2x2

Gomez, Lucas

Volcovinsky, Bruno

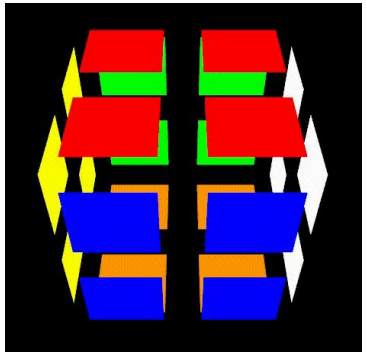
Sartorio, Alan

Problema a resolver

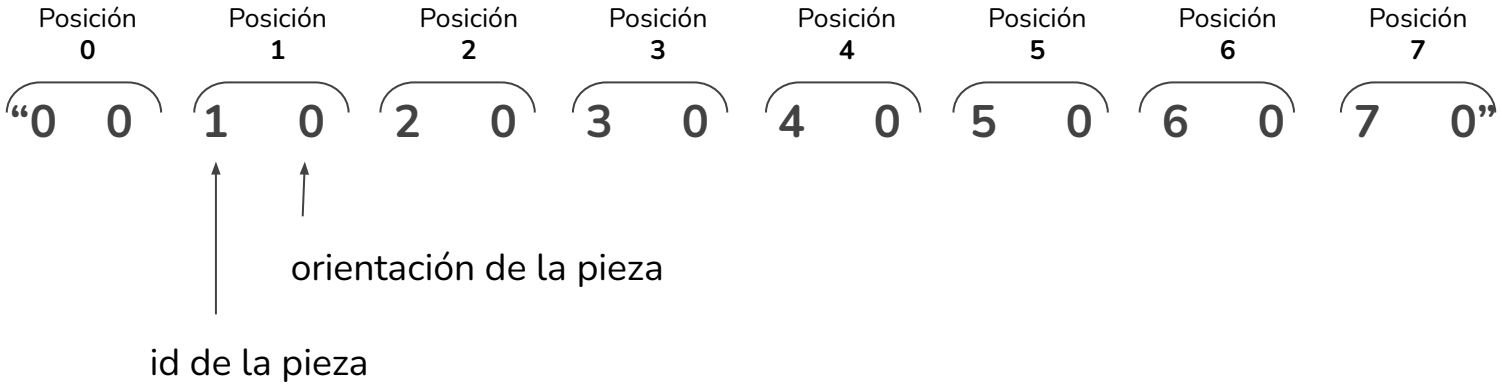


Búsqueda de soluciones al Cubo de Rubik de 2x2x2 mediante algoritmos de búsqueda.

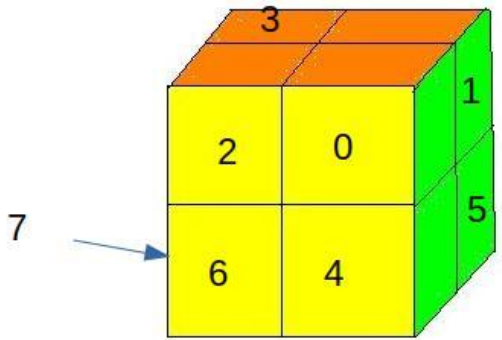
Modelo



Cada uno de los estados del cubo es representado mediante una cadena de 16 números los cuales indican ‘id de la pieza - orientación de la misma’



Modelo



Identificación de las piezas del cubo

Visualizando el orden de los colores de cada una de las piezas se puede definir la orientación de las mismas (por ejemplo, asumiendo que el color naranja/rojo en una posición vertical es la orientación ‘0’)

Búsqueda en amplitud

Estados iniciales:

- "1122304250610271"
- "7202225062124230"
- "2071411031025160"
- "0051316040711120"
- "1242325202622272"

Búsqueda en amplitud
Resultados

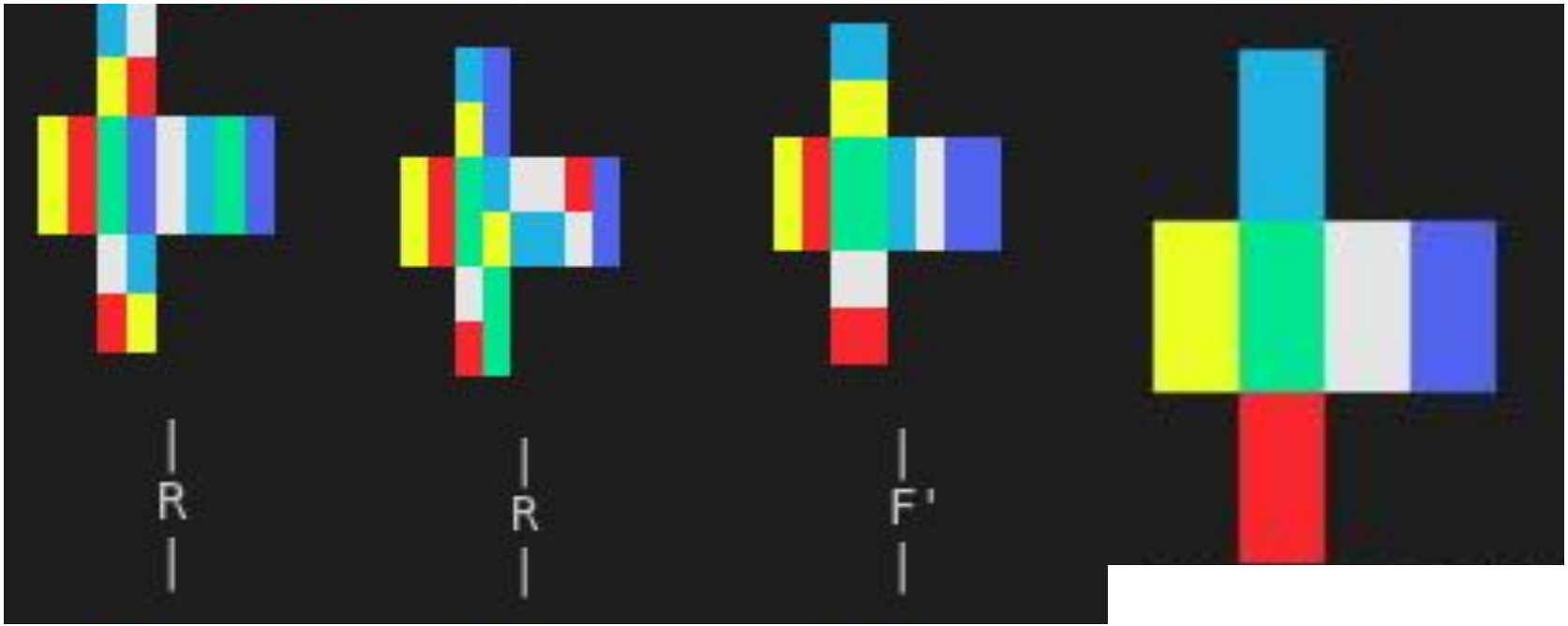
Tiempo medio de procesamiento:

(77.48530 + 24.30874 + 0.89605 +0.01272 + 0.00214) / 5 = 20.54099

Movimientos necesarios: 8, 7, 5, 3, 1

Búsqueda en amplitud

Ejemplos de estados intermedios



Búsqueda en profundidad

Estados iniciales:

- “1122304250610271”
- “1240007130216152”
- “7202225062124230”
- “2140605230700111”
- “3002617212514021”

Búsqueda en profundidad
Resultados

Tiempo medio de procesamiento:,

Movimientos necesarios:

Ejemplo de estados intermedios:

ACA PONER IMAGEN DE LOS STATES QUE NOS TIRA LA CONSOLA

Búsqueda en profundidad variable

Estados iniciales:

- “1122304250610271”
- “1240007130216152”
- “7202225062124230”
- “2140605230700111”
- “3002617212514021”

Búsqueda en profundidad variable
Resultados

Tiempo medio de procesamiento: 11.59766650, >60, >60, 13.28966569, 57.3282680

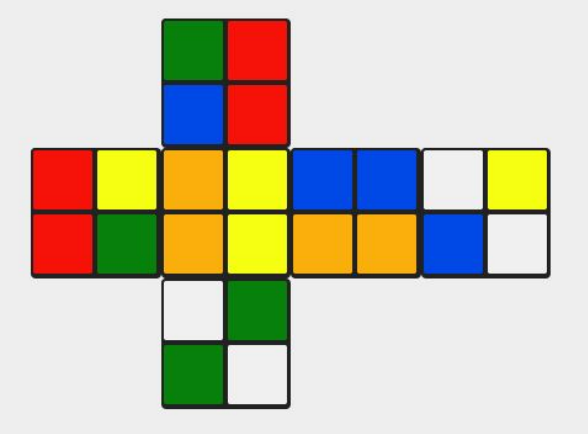
Movimientos necesarios: 10, DNF, DNF, 10, 12

Ejemplo de estados intermedios:

Heurísticas contempladas

Sticker groups

(No admisible)

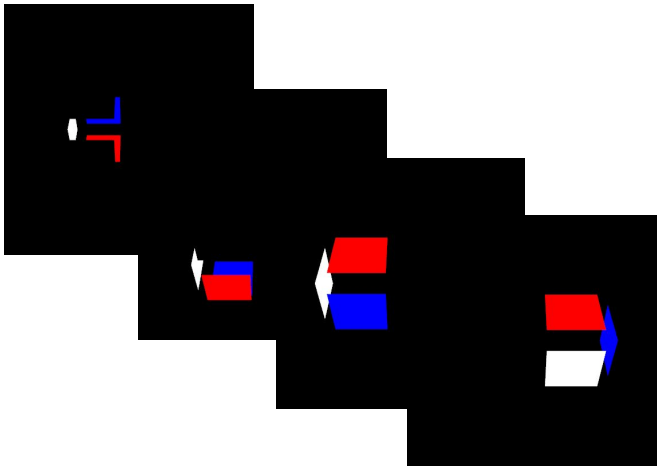


Move count combination

(Admisible)

Manhattan distance

(Admisible)



Heurística local

Estados iniciales:

- “1122304250610271”
- “7202225062124230”
- “2071411031025160”
- “0051316040711120”
- “1242325202622272”

Heurística local
sticker groups

Tiempo medio de procesamiento:

(2.59019 + 0.22756 + 1.58700 + 0.00828 + 0.00112) / 5 = 0.88283

Movimientos necesarios: 2808, 289, 1965, 9, 1

Nodos expandidos: 12990, 1307, 9052, 43, 6

Heurística local

Move count combination

Tiempo medio de procesamiento:

(0.12650 + 0.40818 + 0.68504 + 0.00122 + 0.00047) / 5= 0.24428

Movimientos necesarios: 356, 1041, 1651, 3, 1

Nodos expandidos: 1694, 4897, 7750, 16, 6

Heurística local

Manhattan distance

Tiempo medio de procesamiento:

(142.38221 + 0.03902 + 0.01433) / 3 = 47.47852

Movimientos necesarios: 11453, 3, 1

Nodos expandidos: 54498, 16, 6

Nota: Algunos de los estados iniciales planteados requieren demasiado tiempo de procesamiento

Heurística global

Estados iniciales:

- “1122304250610271”
- “7202225062124230”
- “2071411031025160”
- “0051316040711120”
- “1242325202622272”

Heurística global
Sticker groups

Tiempo medio de procesamiento: >60, >60, >60, 0.0016655921936, 0.000242233276

Movimientos necesarios: -, -, -, 3, 1

Nodos expandidos: -, -, -, 43, 6

Heurística global

Move count combination

Tiempo medio de procesamiento: 0.37185, 0.67219, 0.00070, 0.00030, 0.00012

Movimientos necesarios: 20, 23, 5, 3, 1

Nodos expandidos: 15410, 24485, 36, 16, 6

Heurística global

Manhattan distance

Tiempo medio de procesamiento: 40.70985, >60, >60, 0.00908, 0.00313

Movimientos necesarios: 12, -, -, 3, 1

Nodos expandidos: 64324, -, -, 16, 6

Método A*

Estados iniciales:

- “1122304250610271”
- “7202225062124230”
- “2071411031025160”
- “0051316040711120”
- “1242325202622272”

Método A*

Sticker groups

Tiempo medio de procesamiento:

(22.69897 + 33.37624 +59.33469 + 0.00882 +0.00112) / 5 = 23.083968

Movimientos necesarios: 20, 11, 25, 3, 1

Nodos expandidos: 55488, 76548, 127297, 43, 6

Método A*

Move count combination

Tiempo medio de procesamiento:

(0.10355 + 0.01706 + 0.00412 + 0.00220 + 0.00056) / 5 = 0.025498

Movimientos necesarios: 8, 7, 5, 3, 1

Nodos expandidos: 946, 140, 36, 20, 6

Método A*

Manhattan distance

Tiempo medio de procesamiento:

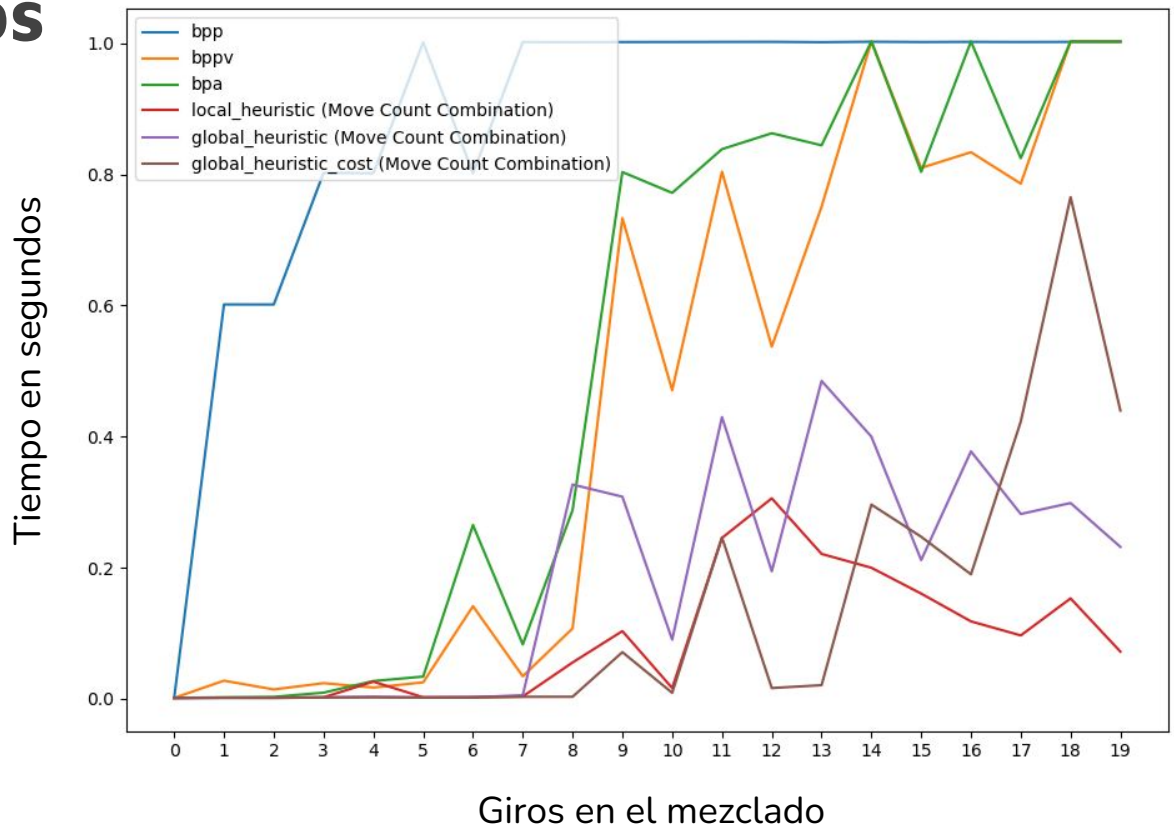
(103.04751 + 22.96441 + 1.20487 + 0.07226 +0.01372) / 5 = 25.460554

Movimientos necesarios: 8, 7, 5, 3, 1

Nodos expandidos: 30012, 7521, 422, 25, 6

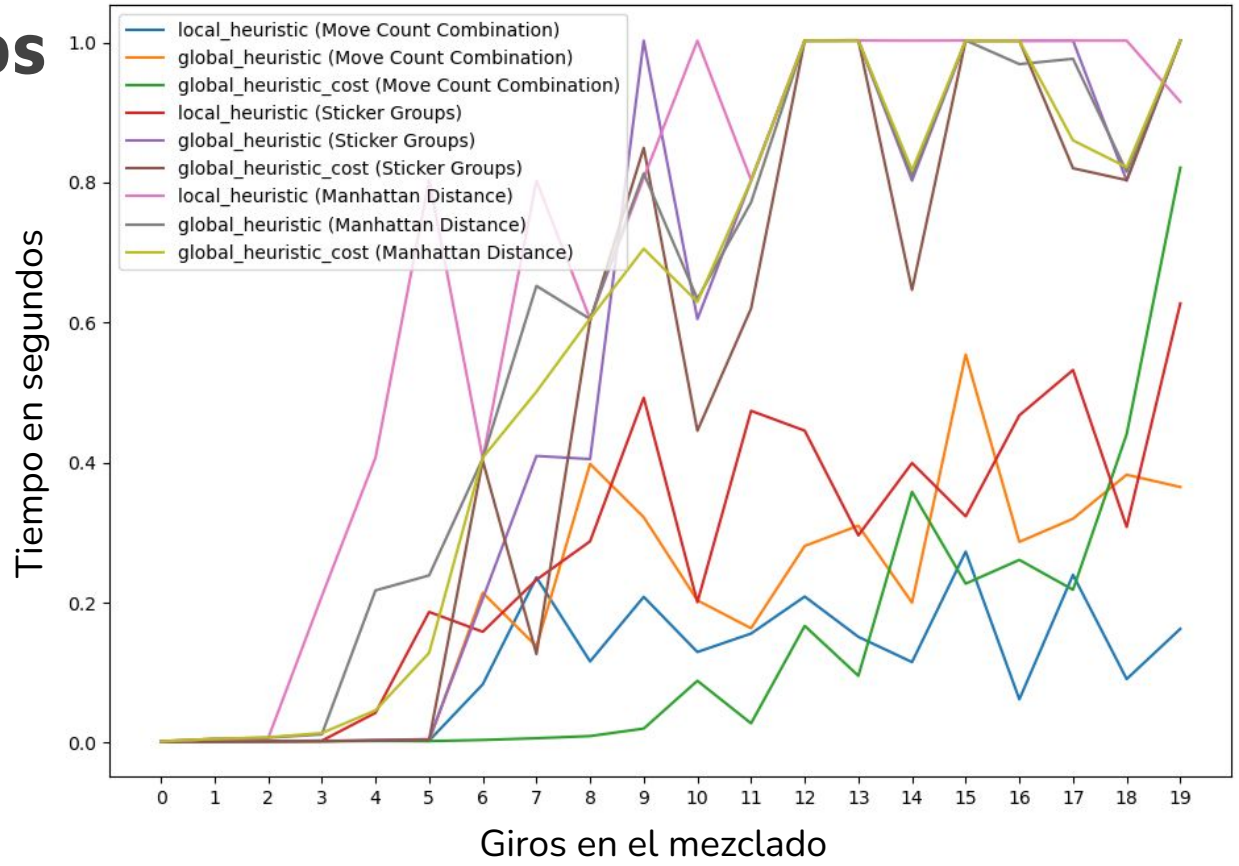
Tiempos

Comparativa métodos de búsqueda



Tiempos

Comparativa funciones heurísticas



Conclusiones

- Al definir una heurística acertada, la complejidad espacial y temporal se reduce considerablemente.
- El uso del algoritmo BPA obtiene la solución óptima (al igual que el método A*) pero con tiempos de procesamientos mucho mayores
- Se noto que el método BPP es el que peor rendimiento nos ofrece, sobretodo al aumentar la cantidad de pasos para llegar a la solución
- El método BPP tiene una gran mejora de rendimiento si se establece un límite de profundidad (BPPV)