



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2613 - INTELIGENCIA ARTIFICIAL

# Informe Tarea 2

18 de octubre de 2019

Santiago Andrés Laguna Altamirano - 17636671

---

## Descripción heurística utilizada

La heurística utilizada corresponde a la suma de las distancias manhattan de cada cuadrante, dividido por el máximo entre el largo y el ancho. Cabe destacar, que para tomar la distancia manhattan se consideró la naturaleza del problema, es decir, que el movimiento se puede realizar como si fuera un toroide.

Por el teorema de la relajación, esta heurística es admisible. Podemos notar que es una relajación al problema, pues están permitidos todos los movimientos del problema original. Otra forma de entender esta relajación es considerando el problema original, pero eliminando restricciones.

En el problema original podemos mover máximo  $n$  cuadrantes al mismo tiempo en una dirección (donde  $n$  es el máximo entre la cantidad de filas y la cantidad de columnas), sin embargo, tienen que estar juntas en una fila o columna. En el problema relajado, podemos mover  $n$  cuadrantes por turno en una dirección, pero no es necesario que estén juntos en fila o columna y también podemos superponer números en los cuadrantes. Al incluir estas restricciones al problema relajado, obtenemos nuevamente el problema original. Luego, como la heurística determina la solución óptima a este problema relajado, entonces es admisible.

## Comparación Anytime $wA^*$ v/s Restarting $wA^*$

A continuación se muestran dos tablas comparativas, la primera respecto a los tiempos en que encuentran el óptimo (timebound de 60 segundos) y la segunda, respecto a la calidad de la solución encontrada respecto a un tiempo fijo 20 segundos). En todos los casos, la heurística utilizada es la misma (explicada en el inciso anterior).

weight	tamaño problema	Anytime wA*	Restarting wA*
5	2 x 2	0.15625 s	0.15625 s
5	3 x 3	0.15625 s	0.15625 s
5	3 x 4	33.5625 s	23.921875 s
5	5 x 5	tiempo excedido	tiempo excedido
10	2 x 2	0.140625 s	0.046875 s
10	3 x 3	0.15625 s	0.15625 s
10	3 x 4	57.828125 s	26.78125 s
10	5 x 5	tiempo excedido	tiempo excedido
20	2 x 2	0.140625 s	0.046875 s
20	3 x 3	0.15625 s	0.03125 s
20	3 x 4	tiempo excedido	50.703125 s
20	5 x 5	tiempo excedido	tiempo excedido

  

weight	tamaño problema	Anytime wA*	Restarting wA*
5	2 x 2	óptimo	óptimo
5	3 x 3	óptimo	óptimo
5	3 x 4	7 pasos	7 pasos
5	5 x 5	tiempo excedido	tiempo excedido
10	2 x 2	óptimo	óptimo
10	3 x 3	óptimo	óptimo
10	3 x 4	8 pasos	7 pasos
10	5 x 5	20 pasos	22 pasos
20	2 x 2	óptimo	óptimo
20	3 x 3	óptimo	óptimo
20	3 x 4	tiempo excedido	tiempo excedido
20	5 x 5	20 pasos	20 pasos

Como se puede apreciar en la mayoría de los casos, a medida que aumentamos la dificultad del problema (que es justamente cuando estos algoritmos son necesarios), el tiempo para encontrar el óptimo que requiere Restarting wA\* es menor que el requerido por Anytime wA\*. Luego para problemas complejos Restarting wA\* será preferible frente a Anytime wA\*. Por otro lado, podemos ver que si se desea encontrar el óptimo, no es recomendable aumentar w. Sin embargo, aumentar w es útil cuando el tiempo estimado para encontrar el óptimo es muy alto y buscamos la mejor solución posible en el tiempo del que se dispone.