

Trabajo Práctico Especial 1

Sokoban

Grupo 8

Colloca, Tomás

López, Noelia Belén

Suárez Bodean, Joaquín

Vera, Juan Sebastián

Índice

Índice	2
Introducción	4
Modelo	5
Engine	5
State	5
Problem	5
Rule	5
Heurística - Regla	6
Simple	6
Player-Box, Near Box-Goal Distances (PB Near BG)	6
Costo	6
Conclusiones	7
Ningún método no informado es claramente mejor que otro	7
Greedy resulta más performante que A*, aunque no da resultado óptimo	7
Anexo	8
Tabla de resultados	8
Other 2 Board	8
Tiempo estimado	8
Distancia	8
Complicated 2 Board	8
Tiempo estimado	8
Distancia	9
Complicated Board	9
Tiempo estimado	9
Difficult Board	9
Tiempo estimado	9
Distancia	10
Other 2 Board	10
Tiempo estimado	10
Distancia	10
Gráficos	11

Introducción

Mediante la implementación de algoritmos de búsqueda, informados y no informados, se buscó comparar las distintas soluciones obtenidas, a distintos tableros posibles del clásico juego Sokoban, analizando las distintas alternativas para llegar a los mismos.

Ante el juego de ingenio, creado por Hiroyuki Imabayashi, en el que un jugador debe ir empujando cajas hasta ubicarlas en sus objetivos, se evaluaron diferentes reglas de heurística con las que después se resolvieron los posibles tableros.

A fin de elegir, luego, la mejor sucesión de pasos hacia el resultado (el mejor camino), fueron comparados los diferentes costos de seguir las reglas de heurística.

Modelo

Engine

A partir de los modelos de *gps* provistos por la cátedra, se implementaron modelos que pudieran ser aplicados al problema específico del Sokoban.

Se mantuvo el paradigma en el que el *Engine* fuera quién crease una *PriorityQueue*, utilizando el comparador especificado (*DFS*, *BFS*, *ASTAR*, *GREEDY*, *IDDFS*); y en el que los *States* son quienes tienen la información del nodo.

State

A fin de mantener la simplicidad y de no perder la totalidad de la memoria en la búsqueda de un camino óptimo, se resolvió que cada estado sólo tendría la disposición de las cajas y el personaje en el tablero en cuestión. A su vez, se le permite aplicar una regla.

Problem

El problema es el que tiene la heurística y la aplica sobre un estado. También es el encargado de determinar si un estado es un estado de solución; en nuestro caso, comparando la posición de las cajas con la de los objetivos.

Rule

Cada regla tiene la capacidad de retornar su nombre de manera que pueda ser entendida por un ser humano, a la vez de evaluarse ante un estado y de retornar el costo, de ser aplicada, hasta ese momento.

Heurística - Regla

Con fines de claridad y eficiencia, todas las heurísticas son representadas en clases separadas que, a su vez, implementan una interfaz general que las hermana en el cálculo del valor de la misma.

Simple

La primera regla heurística presentada es, inevitablemente, la más simple, ver cuántas cajas se encuentran ya en un goal, y obtener la diferencia con la cantidad de cajas que, originalmente, se encuentran en juego.

Si bien es una heurística simple que ayuda a resolver el problema planteado, no es la mejor porque no incluye, dentro de su análisis, la distancia de cada caja a sus respectivos objetivos, o la cantidad de movimientos que tuvo que hacer el personaje para poder ubicar las cajas en posición.

A su vez, llegar a la conclusión de que, en algunos tableros, el camino óptimo resulta de sacar una caja de su goal, termina por concluir que está no es una heurística óptima.

Esta heurística es admisible, ya que si una caja no está en su objetivo, el costo de acomodar la misma es mayor o igual a uno.

Player-Box, Near Box-Goal Distances (PB Near BG)

A partir de la heurística anterior y con la ayuda de varios caminos tomados hacia una heurística mejor, se llegó a *PB Near BG*, que evalúa la distancia del personaje a cada una de las cajas y, a su vez, la distancia de cada caja a su objetivo más cercano; o 0 si ya están ubicadas en un objetivo.

De esta manera se puede tener en cuenta, también, la distancia que separa al jugador de una caja, la cantidad de movimientos que tiene que realizar para llegar a ella, y la cantidad de movimientos que le cuesta posicionar la caja en su respectivo goal.

De igual modo, la posibilidad de que, en algún momento, una caja tenga que ser removida de su objetivo, para poder dar paso a una solución del problema.

Esta heurística no es admisible, puesto que la suma de las distancias del jugador a las distintas cajas puede ser mayor al costo del jugador de alcanzar las cajas una tras otra.

Costo

Vale remarcar que el costo es calculado a partir de cada movimiento del jugador, por lo que, en cada estado, la diferencia de costo es 1.

Conclusiones

Ningún método no informado es claramente mejor que otro

Cada tablero resulta en una performance distinta ante cada método de búsqueda, por lo que se encontraron tableros en los que uno fue mejor que otro, y tableros en donde la situación se dio a la inversa.

Greedy resulta más performante que A^* , aunque no da resultado óptimo

Es interesante como, dentro de los métodos informados, *greedy* resulta ser, en la gran mayoría de los casos, mucho más performante en cuanto a la velocidad en la búsqueda, pero dicha velocidad puede no significar un resultado óptimo; cosa que sí garantiza A^* .

Anexo

Tabla de resultados

Other 2 Board

Tiempo estimado

Board	Heuristic	Strategy				
		ASTAR	BFS	DFS	GREEDY	IDDFS
OtherBoard.txt	PBBG	263	401	372	72	6,217
	PBNearBG	263	396	369	74	6,110
	Simple	590	604	455	228	6,408

Distancia

Board	Heuristic	Strategy			
		ASTAR	BFS	DFS	GREEDY
other2.txt	PBBG	97	89	153	99
	PBNearBG	89	89	153	99
	Simple	89	89	153	107

Complicated 2 Board

Tiempo estimado

Board	Heuristic	Strategy			
		ASTAR	BFS	DFS	GREEDY
Complicated2Board	PB near BG	19,553	107,751	89,351	1,318
	PBBG	1,786	107,503	79,752	1,748
	Simple	108,636	95,514	57,550	1,925

Distancia

Board	Heuristic	Strategy			
		ASTAR	BFS	DFS	GREEDY
Complicted2Board	PB near BG	41	41	707	73
	PBBG	49	41	707	57
	Simple	41	41	707	183

Complicated Board

Tiempo estimado

Board	Heuristic	Strategy				
		ASTAR	BFS	DFS	GREEDY	IDDFS
complicatedBoard.txt	PB near BG	48,194,501	15,449,248	14,518,616	93,593,487	148,903,807
	PBBG	47,559,856	12,117,001	9,429,498	50,927,371	265,967,835
	Simple	20,105,441	73,013,610	30,098,699	57,153,761	333,976,167

Distancia

Board	Heuristic	Strategy				
		ASTAR	BFS	DFS	GREEDY	IDDFS
complicatedBoard.txt	PB near BG	38	38	38	38	38
	PBBG	38	38	38	38	38
	Simple	38	38	38	38	38

Difficult Board

Tiempo estimado

Board	Heuristic	Strategy				
		ASTAR	BFS	DFS	GREEDY	IDDFS
difficultBoard.txt	PB near BG	26	21	9	100	195
	PBBG	56	14	11	63	280
	Simple	32	111	47	44	302

Distancia

Board	Heuristic	Strategy				
		ASTAR	BFS	DFS	GREEDY	IDDFS
difficultBoard.txt	PB near BG	39	39	43	39	39
	PBBG	39	39	43	39	39
	Simple	39	39	43	39	39

Other 2 Board

Tiempo estimado

Board	Heuristic	Strategy			
		ASTAR	BFS	DFS	GREEDY
other2.txt	PBBG	14,932	14,788	11,029	5,693
	PBNearBG	56,241	14,073	10,281	4,392
	Simple	24,693	15,297	9,183	3,361

Distancia

Board	Heuristic	Strategy			
		ASTAR	BFS	DFS	GREEDY
other2.txt	PBBG	97	89	153	99
	PBNearBG	89	89	153	99
	Simple	89	89	153	107

Gráficos



Figura 1. Barras comparativas en distancia y tiempo sobre el tablero 'complicatedBoard' en todas las estrategias posibles

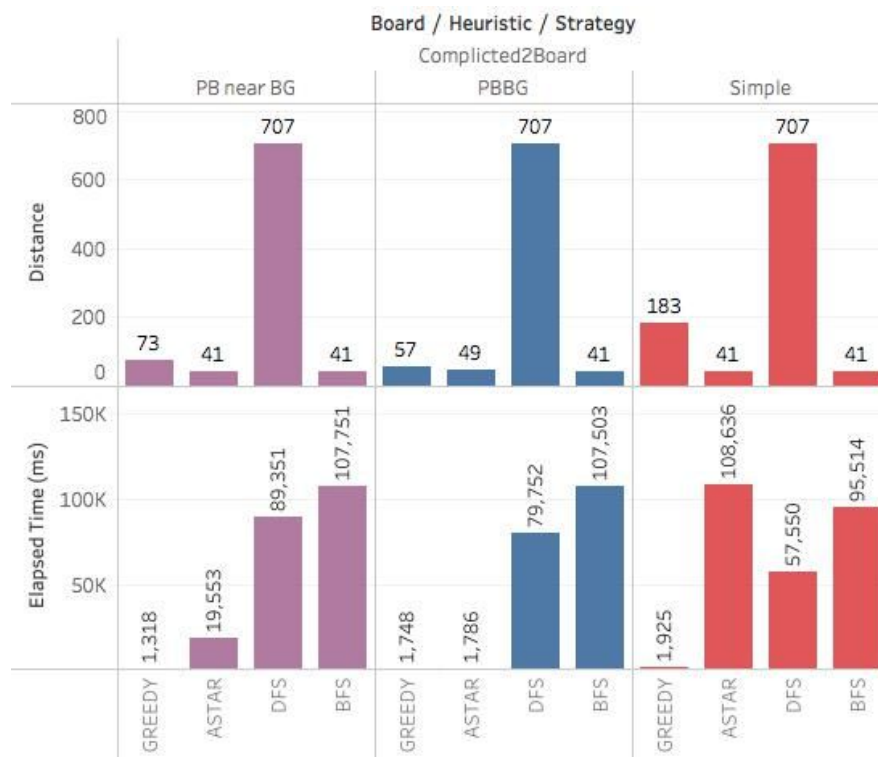


Figura 2. Barras comparativas en distancia y tiempo sobre el tablero 'complicated2Board' en todas las estrategias posible



Figura 3. Barras comparativas en distancia y tiempo sobre el tablero 'otherBoard' en todas las estrategias posibles

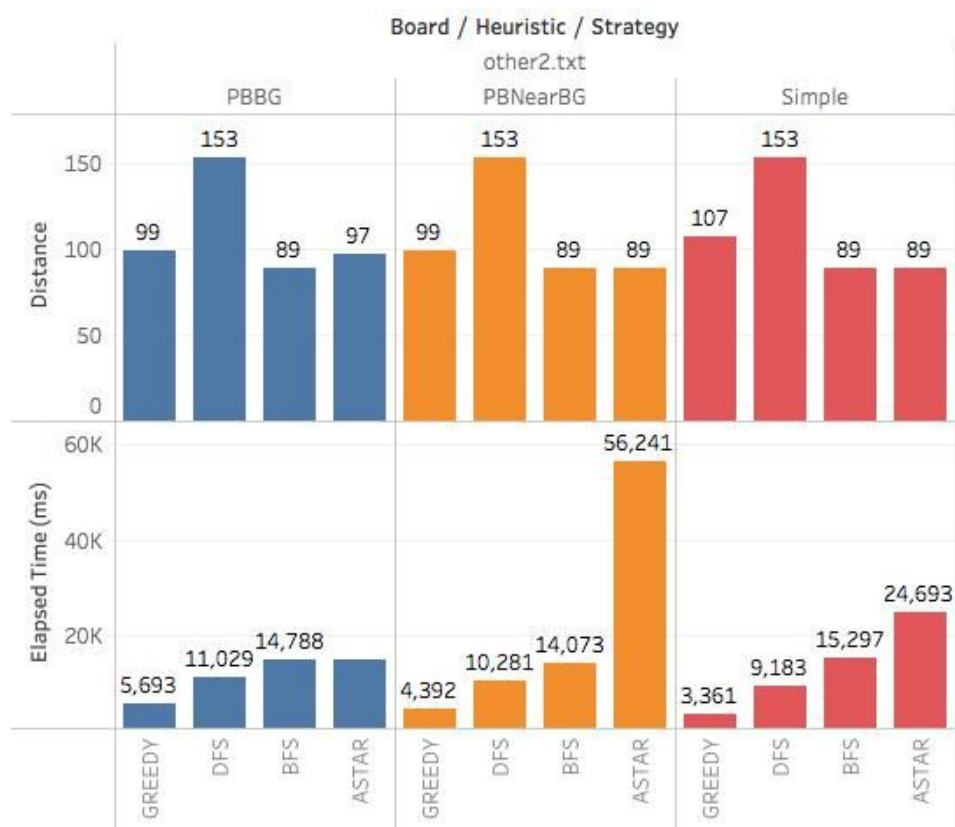


Figura 4. Barras comparativas en distancia y tiempo sobre el tablero 'other2Board' en todas las estrategias posibles