

# Trabajo Práctico de Informática 2 - Zoz

Guido Nuñez

9 de mayo de 2014

## 1. Formulación y análisis del problema:

- a) Escriba la formulación del problema: estados, estado inicial, función sucesora, verificación de la meta y costo del camino.

Estados: Un estado es una configuración del tablero con  $x$  fichas colocadas,  $1 \leq x \leq f(N) - 1$ ,  $f(N)$  = cantidad de celdas del tablero en función a un lado  $N$ , tal que se haya obtenido mediante un movimiento válido o sea la configuración del estado inicial.

Estado inicial: Cualquier configuración del tablero con  $f(N) - 1$  fichas colocadas.

Función sucesora: Mover una ficha en cualquier dirección hacia una celda desocupada pasando exactamente por encima de una ficha intermedia y quitando esta última del tablero.

Verificación de la meta: Cualquier configuración del tablero con una sola ficha colocada.

Costo del camino: Número de pasos.

- b) Encuentre una fórmula  $f(N)$  que expresa el número de posiciones que tiene el tablero en función a uno de los lados  $N$ . Por ejemplo, los tableros ilustrados en las figuras tienen un lado de 5 posiciones, luego existen  $f(5) = 15$  posiciones en el tablero. Muestre paso a paso como deduce la fórmula a su expresión algebraica más sencilla.

Un tablero con  $n$  celdas por lado puede verse como un triángulo con  $1 + 2 + 3 + \dots + (n - 1) + n$  celdas.

$$f(n) = 1 + 2 + \dots + (n - 1) + n$$

$$f(n) + f(n) = 1 + 2 + \dots + (n - 1) + n + n + (n - 1) + \dots + 2 + 1$$

$$f(n) + f(n) = (1 + n) + (2 + n - 1) + \dots + (n - 1 + 2) + (n + 1)$$

$$f(n) + f(n) = (n + 1) + (n + 1) + \dots + (n + 1) + (n + 1)$$

$$2f(n) = n(n + 1)$$

$$f(n) = \frac{n(n + 1)}{2}$$

- c) En base a la fórmula  $f(N)$  encontrada (Ejercicio 1b), halle el tamaño de los tableros para los lados  $N = \{5; 6; 7; 8; 9; 10\}$

	$f(5)$	$f(6)$	$f(7)$	$f(8)$	$f(9)$	$f(10)$
Dimensión del tablero	15	21	28	36	45	55

- d) ¿Cuál es el tamaño del espacio de estados para el juego Zoz? Una vez derivada la expresión, calcule el espacio de estados para dimensiones de tableros cada vez mayores. El tablero inicial puede ser cualquier configuración de fichas en donde exactamente una posición del tablero se encuentra libre. Puede asumir que todas las configuraciones del tablero son válidas.

Dado que para cada  $N$  existe una cantidad  $f(N)$  de celdas y cada celda puede llevar dos valores (contener o no una ficha), y exceptuando la configuración que contiene todas las celdas vacías y la configuración que contiene todas las celdas con fichas, el tamaño del espacio de estados es  $2^{f(N)} - 2$ .

	$N = 5$	$N = 6$	$N = 7$	$N = 8$	$N = 9$	$N = 10$
Espacio de estados	32766	2097150	268435454	$6,872 \times 10^{10}$	$3,518 \times 10^{13}$	$3,603 \times 10^{16}$

- e) Suponga que somos capaces de almacenar cada posición del tablero Zoz en un único átomo, y que el universo visible tiene un total de  $10^{80}$  átomos en su totalidad. ¿Cuál es el tablero de lado  $N$  más grande que se podría resolver en este universo visible?

El tablero más grande que se podría resolver en este universo visible es un tablero con  $N = 22$ , con  $1,447 \times 10^{76}$  átomos utilizados.

- f) Considere que cada posición del tablero ocupa 16 bytes en la memoria. ¿Cuánta memoria se necesita para cada caso?

	$N = 5$	$N = 6$	$N = 7$	$N = 8$	$N = 9$	$N = 10$
Total de memoria (en bytes)	524256	33554400	4294967264	$1,100 \times 10^{12}$	$5,629 \times 10^{14}$	$5,765 \times 10^{17}$

2. Una vez definido el problema, escriba el programa para resolver el juego Zoz. Halle la solución al problema empleando las siguientes estrategias de búsqueda:

- Breadth First Tree Search (BFTS)
- Breadth First Graph Search (BFGS)
- Uniform Cost Graph Search (UCGS)
- Depth First Tree Search (DFTS)
- Depth First Graph Search (DFGS)
- Recursive Depth First Search (RDFS)
- Iterative Deepening Search (IDS)
- Greedy Best First Graph Search (GBFGS)
- Recursive Greedy Best First Search (RGBFS)
- A\* (A\*S)

Las soluciones pueden verse ejecutando el programa implementado y siguiendo las instrucciones.

3. Evalúe la complejidad temporal y espacial de cada estrategia de búsqueda para el problema Zoz de dimensión  $N = 5$ . La complejidad temporal y espacial deben medirse según los criterios presentados en el libro de Russell y Norvig. Con los resultados obtenidos construya una tabla de resultados.

Estrategias de búsqueda	Complejidad Temporal	Complejidad Espacial	Costo de buscar
(BFTS)	323872	323873	647745
(BFGS)	4845	4846	9691
(UCGS)	4846	4847	9693
(DFTS)	6281	92	6373
(DFGS)	1102	92	1194
(RDFS)	6281	92	6373
(IDS)	1011045	92	1011137
(GBFGS)	942	943	1885
(RGBFS)	1008	1009	2017
(A*S)	4507	4508	9015

4. Una vez elaborada la tabla conteste a las siguientes preguntas:

- a) ¿Cómo se explican las diferencias en la complejidad temporal y espacial de las estrategias de búsqueda Breadth First Search en la versión implementada como árbol y grafo? ¿Sucede lo mismo en el caso de Depth First Search?

La estrategia de búsqueda Breadth First Three Search implementada como árbol posee una mayor complejidad espacial y temporal que la versión implementada como grafo. Esto ocurre debido a que la versión implementada como grafo mantiene una lista (explored) que almacena los nodos que ya se han expandido y esto evita expandir nodos repetidos. Esto reduce tanto la complejidad temporal como la espacial. Lo mismo ocurre con la estrategia de búsqueda Depth First Search.

- b) Describa la heurística  $h()$  que diseñó para implementar la búsqueda Greedy.

La función heurística diseñada busca evitar llegar a estados de los cuales no se puede llegar a un estado meta. En particular consideré la configuración de tablero en la que las fichas se encuentran colocadas en los bordes y no en las celdas interiores. Esta es una configuración a partir de la cual no se puede avanzar. Basándome en esto, la función heurística va sumando una unidad por cada ficha colocada en un borde y por cada celda vacía en el interior del tablero. Con esto se logra obtener una suma alta para configuraciones no deseadas y así el algoritmo de búsqueda evita elegir estos estados.

- c) Describa la heurística  $h()$  que diseñó para implementar la búsqueda A\*. Dada la heurística  $h()$  diseñada, ¿la búsqueda A\* es óptima?

La heurística utilizada es la misma que la diseñada para la búsqueda Greedy. Todas las soluciones de Zoz poseen el mismo costo de camino = 13, por lo que cualquier solución es óptima, y por lo tanto, la búsqueda  $A^*$  es óptima.

- d) ¿Las estrategias desinformadas o informadas son mejores para resolver Zoz con  $N = 5$ ?

Las estrategias informadas en conjunto son mejores para resolver Zoz con  $N = 5$ . Sin embargo, estas estrategias no contienen a la mejor estrategia para resolver el juego. Según las experimentaciones realizadas la mejor estrategia es Depth First Graph Search que se encuentra entre las estrategias no informadas. A pesar de esto, las estrategias no informadas no pueden considerarse como mejores debido a las altas complejidades de las demás estrategias constituyentes.

- e) ¿Las estrategias de búsqueda en profundidad o en anchura son mejores para resolver Zoz con  $N = 5$ ? ¿A qué se debe esto?

Para resolver Zoz con  $N = 5$  son mejores las estrategias de búsqueda en profundidad. Esto se debe a que los nodos metas se encuentran en el nivel 13 del árbol y la búsqueda en anchura debe expandir todos los nodos de cada nivel para llegar a ese nivel. En cambio, la búsqueda en profundidad no necesita expandir todos los nodos de los niveles inferiores ya que el nodo meta hallado no se encuentra entre las últimas hojas del árbol ubicadas más a la derecha.

- f) ¿Cuál es el mejor algoritmo de búsqueda para Zoz con  $N = 5$ ? ¿Se puede afirmar también que es el mejor para dimensiones mayores  $N > 5$ ?

La mejor estrategia es Depth First Graph Search debido a que obtiene el menor costo de buscar gracias a una baja complejidad temporal y espacial. Se puede considerar que es la mejor también para dimensiones mayores ya que por más que no obtenga la mejor complejidad temporal (alguna estrategia informada podría tener una mejor), es la que necesitaría menos memoria.

- g) Usted es un investigador que quiere contribuir con la ciencia encontrando mejores algoritmos de búsqueda para resolver el juego Zoz. Dados los resultados experimentales que obtuvo, ¿Qué tipo de algoritmos de búsqueda se esforzaría en investigar a continuación?

Investigaría estrategias informadas, ya sea Greedy o  $A^*$ , debido a las dependencias de éstas sobre las funciones heurísticas. Trataría de hallar heurísticas admisibles y consistentes que permitan reducir las complejidades de estas estrategias, como también ciertas mejoras o cambios que puedan aplicarse a dichas estrategias de modo a obtener el máximo provecho de ellas.

5. ¿Es posible utilizar la búsqueda bidireccional para este problema? En caso afirmativo, escriba la función predecesora.

Si, es posible, construyendo un nuevo estado ficticio cuyos predecesores inmediatos son los estados meta del juego. El estado ficticio que utilicé corresponde a una configuración de tablero con todas las celdas vacías. Los predecesores de este estado ficticio son los estados cuyas configuraciones de tablero poseen una única ficha colocada. La función predecesora puede verse en el código del trabajo, dentro de la definición del problema Zoz y puede ejecutarse ejecutando el programa implementado y siguiendo las instrucciones.

6. ¿Cualquier configuración inicial de fichas sobre el tablero sirve para jugar este juego? Implemente una función que lo prueba.

Para un tablero con  $N = 5$ , cualquier configuración inicial de fichas sirve para jugar el juego. La prueba puede observarse ejecutando el programa implementado y siguiendo las instrucciones.

7. ¿Cuál es la instancia de problema más grande ( $N$ ) que pudiste resolver? ¿Cuál es el algoritmo de búsqueda que elegiste para intentar resolver instancias más grandes del problema? ¿Con qué criterio elegiste el algoritmo de búsqueda?

La instancia más grande que pude resolver fué con  $N = 6$  para una HP Pavilion dv4 Notebook PC, con procesador AMD Turion(tm) II Dual-Core Mobile M500 2.20 GHz y memoria de 4 GB. Probé una instancia con  $N = 7$  durante 6 horas sin lograr resultado alguno. El algoritmo utilizado fué el Depth First Graph Search debido a que es el mejor algoritmo de búsqueda para Zoz según lo establecido anteriormente.

8. ¿Cuáles son las características del árbol de búsqueda completo? Utilice el algoritmo Breadth First Tree Search (BFTS) con  $N = 5$  para completar la tabla a continuación:

Dimensión completa del arbol	$N = 5$
Máximo factor de anchura (b)	8
Máxima profundidad del árbol (m)	13
Total de nodos que tiene el árbol	823873
Total de soluciones que tiene Zoz	1550

¿Coincide el espacio de estados para  $N = 5$  (calculado en el Ejercicio 1d) con el total de nodos que tiene árbol? Explique.

No coincide. Esto ocurre debido a que los nodos del árbol poseen estados repetidos. El espacio de estados no tiene en cuenta los estados repetidos. Por ende el total de nodos del árbol es mayor al espacio de estados.

9. El programa que implementaste es un:

- Agente reflejo
- Agente basado en memoria
- Agente basado en metas

- Agente basado en utilidades
- Agente que aprende
- Ninguna de las anteriores

Elija la opción más acertada para el trabajo que entregaste.

Es una clase de agente basado en metas llamado agente resolvente-problemas. Los agentes resolventes-problemas deciden qué hacer para encontrar secuencias de acciones que conduzcan a estados deseables.

Observaciones para la ejecución del programa:

Para la ejecución del programa solo es necesario ejecutar el archivo zoz.py y luego seguir las indicaciones. Si el programa pide introducir el valor de fila y columna de la celda para el estado inicial, se introduce valores según la siguiente la siguiente imagen:

