

Técnicas de Búsqueda Ciega (2ª semana)

21 de febrero de 2019

Índice

1. Las 8 reinas	3
1.1. Enunciado	3
1.2. Resolución	3
1.2.1. Representación de la información	3
1.2.2. Reglas aplicadas	3
2. Busquedas	5
2.1. Enunciado	5
2.2. Resolución	5
2.2.1. Aplicación de distintas búsquedas	5
2.2.2. Búsqueda bidireccional	8
2.2.3. Otros métodos de resolución	8
3. Coste de las búsquedas	9
3.1. Enunciado	9
3.2. Resolución	9
3.2.1. Nodos generados en búsqueda por profundidad	10
3.2.2. Nodos generados en búsqueda por anchura	10
3.2.3. Nodos generados en búsqueda por descenso iterativo	10
4. Sol y Sombra	11
4.1. Enunciado	11
4.2. Resolución	11
4.2.1. Búsqueda en anchura bidireccional	11
4.2.2. Búsqueda en profundidad	12

1. Las 8 reinas

1.1. Enunciado

La formulación medianamente formal (el lenguaje natural no es formal) de los estados, los estados iniciales, los estados objetivo y las reglas de transición de alguno de los apartados, a tu elección, del ejercicio 2 de la relación de ejercicios de búsqueda.

1.2. Resolución

Estudiaremos el puzle de las 8 reinas.

1.2.1. Representación de la información

Representaremos las posiciones de las reinas en un vector de dimensión 8, cuyo valores serán enteros entre 1 y 8, estos representan la posición de la reina en un eje y su índice su posición en el otro.

Por ejemplo, el vector (1,3,5...) significa que la primera reina está en el cuadro (1,1), la segunda en el (2,3), la tercera en el (3,5)...

1.2.2. Reglas aplicadas

Las reglas del juego son las siguientes:

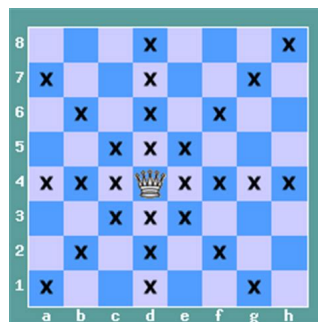


Figura 1: Posibles movimientos de una reina en el tablero.

- Se coloca una reina cada turno
- La nueva reina no puede estar en una posición en la cual no sea atacada por ninguna otra reina, esto se puede dividir en dos partes:
 - La reina no puede estar al lado de otra reina en las coordenadas x e y, es decir, no puede existir dos números iguales seguidos.
 - La reina no puede estar en la diagonal de otra reina, esto lo podemos comprobar con la siguientes formulas

$$i - j = k - l \quad (1)$$

$$j - l = i - k \quad (2)$$

Donde una reina está en las coordenadas (i,j) y la segunda en las coordenadas (j,k)

Si estas equaciones se cumplen, la reina está en la diagonal de otra reina, siendo la primera para la diagonal de pendiente negativa y la segunda la de pendiente positiva.

2. Busquedas

2.1. Enunciado

El ejercicio 3.8 del libro "Artificial Intelligence. A Modern Approach", Segunda Edición, de Stuart Russel y Peter Norvig (descrito a continuación). Para este ejercicio, te sugiero que dibujes los estados generados, con enlaces entre los estados padre y los hijos, y que junto a cada estado pongas entre paréntesis la iteración del proceso de búsqueda en que dicho estado se generó.

El ejercicio es el siguiente:

Consider a state space where the start state is number 1 and the successor function for state n returns two states, numbers $2n$ and $2n + 1$.

Se considera el espacio de estados del 1 al 15.

1. ¿En qué orden se visitan los estados si el estado objetivo fuese el 11 y la técnica de búsqueda fuese: a) anchura; b) profundidad; c) profundidad con límite = 3; d) profundidad iterativa?
2. ¿Tiene sentido aplicar la búsqueda bidireccional? ¿Por qué?
3. ¿Se puede resolver este problema sin realizar la búsqueda sobre el árbol?

2.2. Resolución

2.2.1. Aplicación de distintas búsquedas

a) Búsqueda en anchura:

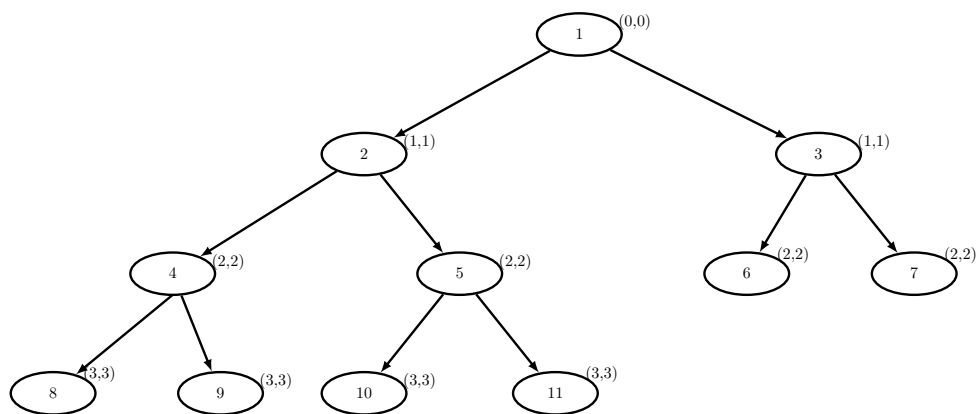


Figura 2: Diagrama búsqueda en anchura.

b) Búsqueda en profundidad:

Se ha escogido expandir primero por $2n$ y después por $2n+1$

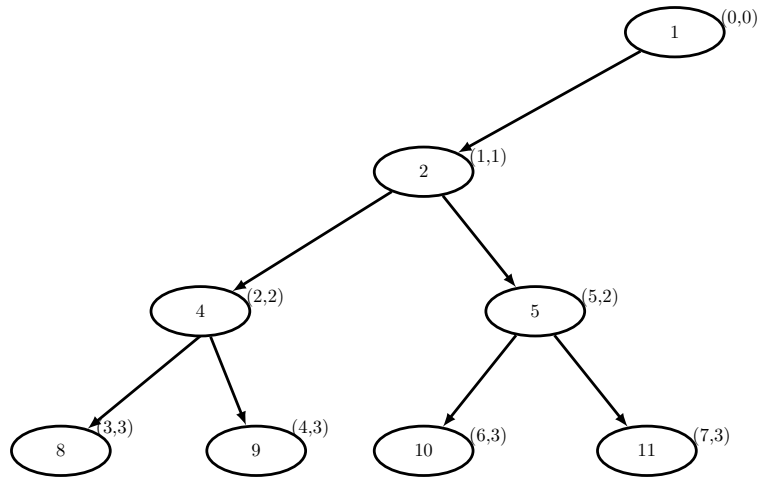


Figura 3: Diagrama búsqueda en profundidad.

c) Búsqueda en profundidad limitada a profundidad 3:

Se ha escogido expandir primero por $2n$ y después por $2n+1$

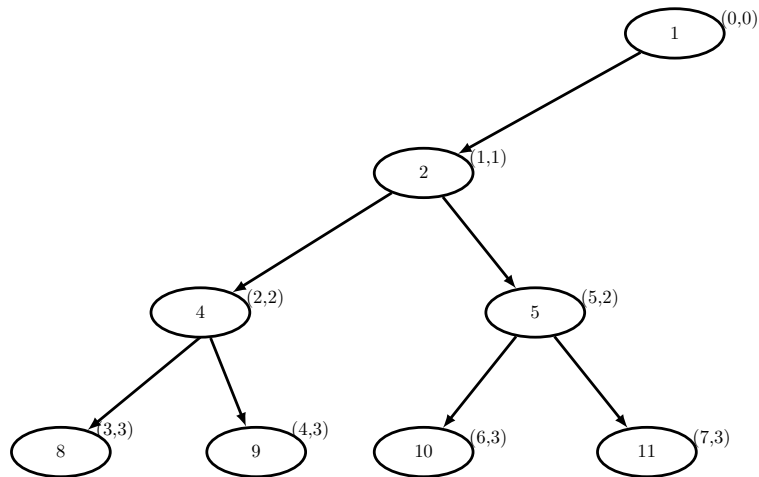


Figura 4: Diagrama búsqueda en profundidad limitada a 3.

En este caso la solución es la misma que en el caso anterior puesto que el nivel máximo que se puede alcanzar en el árbol específico a este problema es 3, ya que al multiplicar ocho por dos el resultado es mayor que quince, límite superior del dominio del problema.

- c) Búsqueda en profundidad iterativa:
Se ha escogido expandir primero por $2n$ y después por $2n+1$

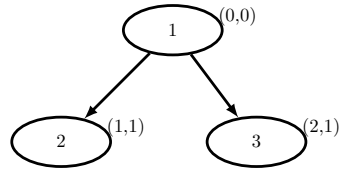


Figura 5: Diagrama búsqueda en profundidad iterativa limite 1.

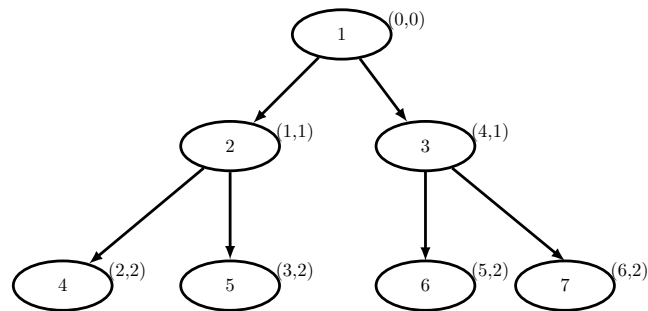


Figura 6: Diagrama búsqueda en profundidad iterativa limite 2.

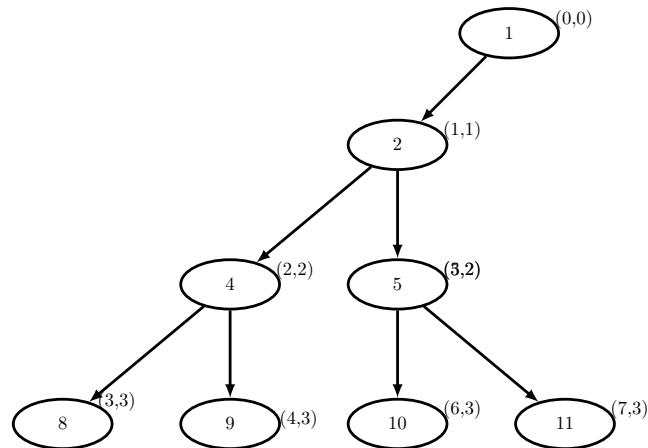


Figura 7: Diagrama búsqueda en profundidad iterativa limite 3.

2.2.2. Búsqueda bidireccional

No, ya que una de las operaciones no daría un elemento válido del dominio del problema, por ejemplo, si al 11 lo dividimos entre 2, el resultado es 5,5, el cual está fuera del dominio del problema.

2.2.3. Otros métodos de resolución

Si, siempre se puede probar a aplicar las acciones aleatoriamente, pero no es muy recomendado en este tipo de problemas.

3. Coste de las búsquedas

3.1. Enunciado

Consideremos un árbol finito de profundidad d con un factor de ramificación b (un sólo nodo raíz, con profundidad 0, y b sucesores por cada nodo, etc.). Supongamos que el nodo objetivo menos profundo se encuentra a una profundidad $g \leq d$.

1. ¿Cuál es el número mínimo de nodos generados por la búsqueda primero en profundidad con una profundidad límite d ? ¿Y el máximo?.
2. ¿Cuál es el número mínimo de nodos generados por la búsqueda primero en anchura? ¿Y el máximo?.
3. ¿Cuál es el número mínimo de nodos generados por el descenso iterativo? ¿Y el máximo? (Considérese que comenzamos con una profundidad límite inicial de 1 y la vamos incrementando una unidad cada iteración).

3.2. Resolución

Planteamos entonces el siguiente árbol genérico para la resolución del problema.

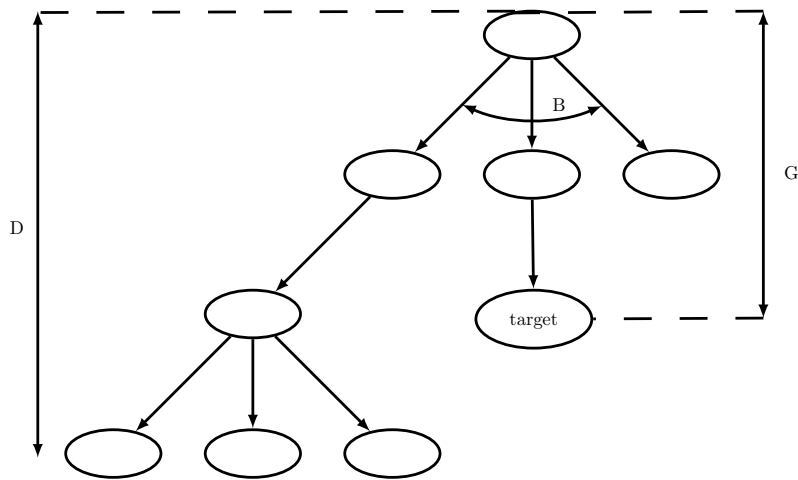


Figura 8: Árbol genérico.

3.2.1. Nodos generados en búsqueda por profundidad

Nodos mínimos generados El caso más favorable sería que nos encontremos el objetivo en la profundidad G de la cola de más a la izquierda. En este caso los nodos generados serían iguales a G

Nodos máximos generados El caso contrario sería el más desfavorable, puesto que tendríamos que explorar casi todo el árbol, el número de nodos generados podría ascender hasta el número dictado por la siguiente fórmula.

$$B^D - B^{D-G} \quad (3)$$

3.2.2. Nodos generados en búsqueda por anchura

Nodos mínimos generados El caso más favorable sería en este caso peor que en profundidad y viene definido por la siguiente fórmula.

$$B^{G-1} + 1 \quad (4)$$

siendo el primero del nivel G el nodo objetivo.

Nodos máximos generados El caso más desfavorable sería cuando se encuentre a la derecha de el nivel G siendo el número de nodos explorados el siguiente.

$$B^G \quad (5)$$

3.2.3. Nodos generados en búsqueda por descenso iterativo

El descenso iterativo va aumentando el límite de profundidad en consecutivas búsquedas por profundidad, con el objetivo de limitar cada vez menos restrictivamente cuanto puede bajar el algoritmo por el árbol, tratando de mejorar la velocidad en la búsqueda de árboles que pueden tener un gran tamaño.

Nodos mínimos generados En este caso, el número mínimo de iteraciones serían, si empezamos en 1 e incrementamos en la unidad, de $G-1$ iteraciones de la búsqueda en profundidad, a su vez, estas tendrán un número de nodos explorados cada vez mayor. El número de nodos explorados totales asciende, como mínimo al valor dado por la siguiente fórmula:

$$\left(\sum_{i=1}^{G-1} B^i \right) + 1 \quad (6)$$

Nodos máximos generados Por otra parte, en el caso más desfavorable el número de nodos explorados sería:

$$\sum_{i=1}^G B^i \quad (7)$$

4. Sol y Sombra

4.1. Enunciado

El ejercicio 3 de la relación de ejercicios de búsqueda, juego Sol y Sombra, pero con un tablero de 5 casillas (dos fichas blancas y dos fichas negras), mediante búsqueda bidireccional con dos anchuras y otro método a tu elección. En este ejercicio, también te sugiero dibujar los estados con enlaces entre los estados padre y los hijos, y que junto a cada estado pongas entre paréntesis la iteración del proceso de búsqueda en que dicho estado se generó.

4.2. Resolución

4.2.1. Búsqueda en anchura bidireccional

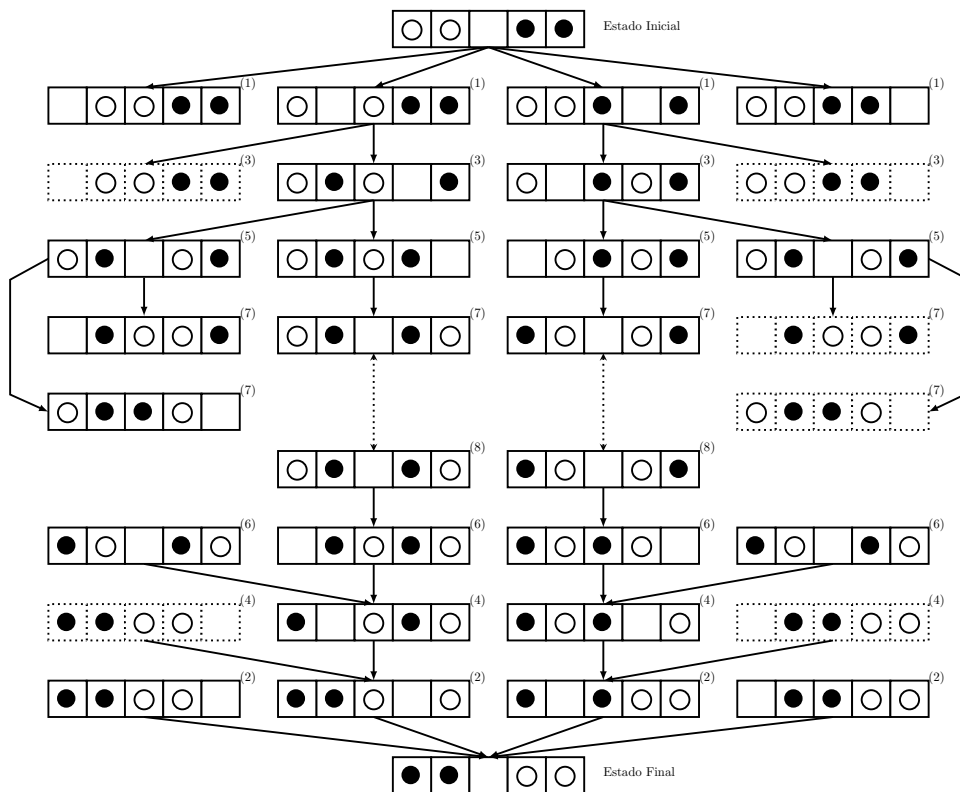


Figura 9: Sol y Sombra por búsqueda por anchura bidireccional.

4.2.2. Búsqueda en profundidad

El orden de prioridad de las acciones son las siguientes:

1. Mover el sol de la derecha a la derecha
2. Mover la luna de la izquierda a la izquierda
3. Mover el sol de la izquierda a la derecha
4. Mover la luna de la derecha a la izquierda

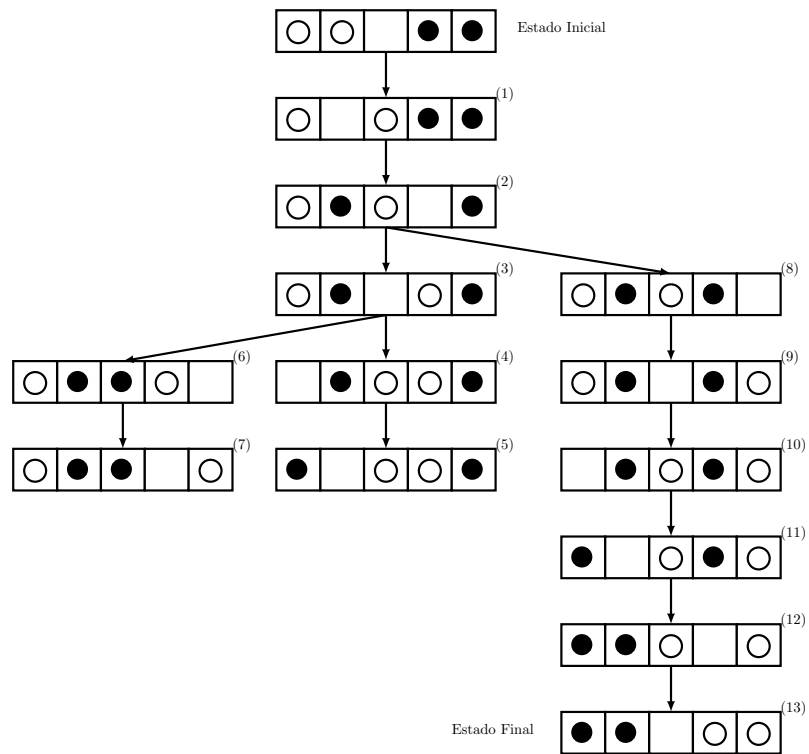


Figura 10: Sol y Sombra por búsqueda por profundidad.