Backtracking (Vuelta atrás)

Análisis y Diseño de Algoritmos

Dr. Jaime Osorio Ubaldo

Backtracking

1. Cuando las técnicas divide y vencerás, algoritmos greedy y programación dinámica no son aplicables para resolver un problema, entonces no queda más remedio que utilizar la fuerza bruta (algoritmos exhaustivos), pero esto en algunos casos resulta inviable debido a su alto costo temporal.
2. El backtracking (búsqueda con retroceso) se aplica a problemas que requieren de una búsqueda exhaustiva dentro del conjunto de todas las soluciones potenciales. El espacio a explorar se estructura de tal forma que se puedan ir descartando bloques de soluciones que posiblemente no son satisfactorias
3. Los algoritmos de backtracking optimizan el proceso de búsqueda de tal forma que se puede hallar una solución de una forma más rápida que con los algoritmos de la fuerza bruta.

Pasos

Ya que las soluciones se construyen por etapas se procede de la siguiente manera:

1. El espacio de posibles soluciones se estructura como un árbol de exploración (cuyos nodos son denominados nodos estados), en el que en cada nivel se toma la decisión de la etapa correspondiente.
2. En cada etapa de búsqueda en el algoritmo backtracking, el recorrido del árbol de búsqueda se realiza en profundidad. Si una extensión de la solución parcial actual no es posible, se retrocede para intentar por otro camino, de la misma manera en la que se procede en un laberinto cuando se llega a un callejón sin salida.
3. Para una determinada etapa, se elige una de las opciones dentro del árbol de búsqueda y se analiza para determinar si esta opción es aceptable como parte de la solución parcial. En caso de que no sea aceptable, se procede a elegir otra opción dentro de la misma etapa hasta encontrar una opción aceptable o bien agotar todas las opciones. Si se encuentra una solución aceptable, ésta se guarda y se profundiza en el árbol de búsqueda aplicando el mismo proceso a la siguiente etapa.

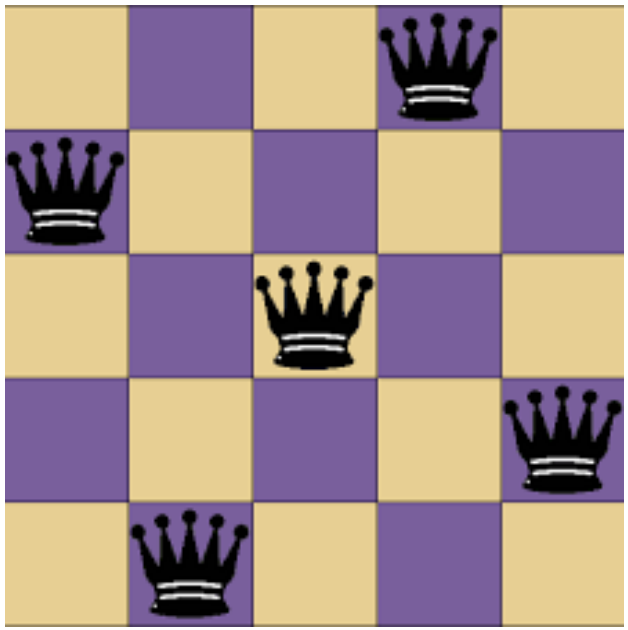
Observación. Es común que los árboles de exploración se definan de tal forma que los nodos solución solo se encuentran en las hojas

Pseudocódigo

```
Función retroceso(etapa)
  iniciarOpciones()
  Hacer
    opcion→ seleccionarOpción
    Si aceptable(opcion) entonces
      guardar(opcion)
      Si incompleta(solución) entonces
          exito → retroceso(siguiente(etapa))
          Si(exito=FALSO) entonces
              retirar(opcion)
          finSi
      Sino
        exito → VERDADERO
      finSi
    finSi
  mientras(exito= FALSO AND opcion ≠ últimaOpción)
  regresar exito
finRetroceso.
```

El problema de las N reinas

Consiste en situar N Reinas en un tablero de ajedrez de N filas y N columnas, sin que se ataquen entre si.



El problema de las N reinas

¿Cuántas alternativas tenemos que explorar?

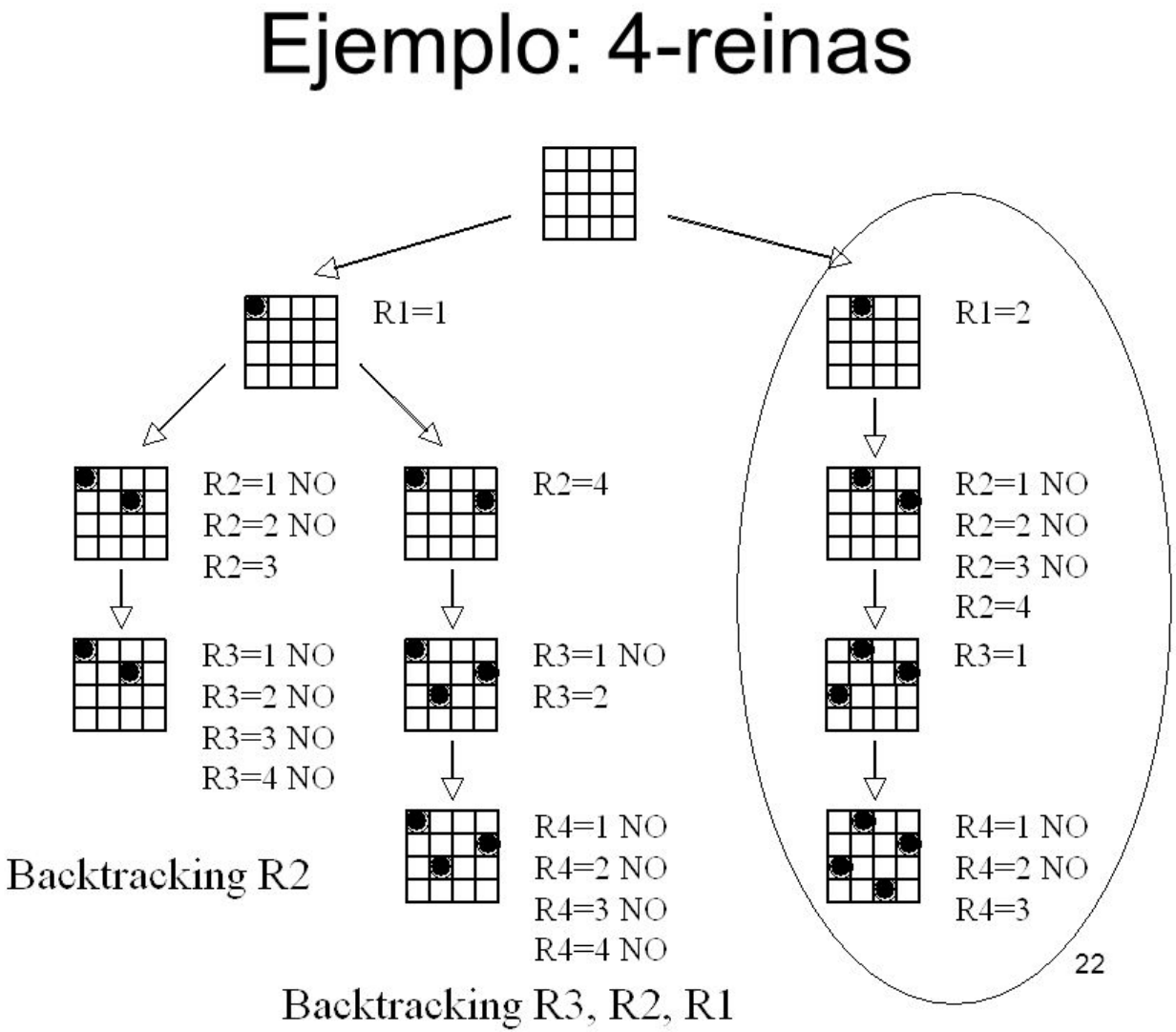
Por conteo, tenemos 64 casillas en un tablero de ajedrez de 8×8, y 8 reinas que ubicar en posiciones distintas, así que hay $\binom{64}{8}$ alternativas de solución (tableros con las 8 reinas situadas en posiciones distintas), esto es, 4 426 165 368. Al número de alternativas se le denomina tamaño del espacio de búsqueda, y siempre que tengamos un problema de búsqueda como éste de las N-Reinas es conveniente estimarlo mediante técnicas de conteo.

Algoritmo

1. Ir colocando las reinas una a una en columnas en un vector A de tal manera que no se ataquen
2. Si las primeras k reinas colocadas no se atacan avanzar a colocar la reina k+1. Si tenemos éxito k irá aumentando de 1 hasta n y habremos situado las n reinas en columnas donde no se ataquen entre sí y obtendremos una solución.
3. Si no tenemos exito, es necesario hacer una vuelta atrás para cambiar la columna en la que se colocó la última reina.

Usando Backtracking

El árbol de exploración es



Investigación

Explique el pseudocódigo del problema de las n-reinas con backtracking.