
Algoritmos en exploración de grafos

Backtracking

Daniel Bolaños Martínez

José María Borrás Serrano

Santiago de Diego de Diego

Fernando de la Hoz Moreno

1. Análisis del problema

El problema que tenemos ante nosotros es el problema de resolver un Sudoku. Un Sudoku consiste en un pasatiempo matemático que trata de rellenar con números del 1 al 9 una tabla de 9×9 elementos, dividida en subcuadrículas de 3×3 , con casillas ya rellenas previamente. Las normas que hay que cumplir son:

- No puede haber dos casillas en la misma fila con el mismo número
- No puede haber dos casillas en la misma columna con el mismo número
- No puede haber dos casillas, en la misma subcuadrícula de 3×3 con el mismo número.

Para resolver el problema tenemos que ir escribiendo números en las cuadrículas que se encuentran vacías sin romper las reglas anteriormente mencionadas. Se encontrará una solución cuando todas las casillas estén rellenas.

2. Diseño de la solución empleando la metodología Backtracking

Para la resolución de dicho problema mediante la técnica de Backtracking. De forma general, los pasos para diseñar un algoritmo de Backtracking son:

- Buscar una representación del tipo $T = (x_1, x_2, \dots, x_n)$ para las soluciones del problema
- Diseñar las restricciones implícitas, es decir, los valores que puede tener cada x_i para contruir la solución
- Identificar las restricciones explícitas, es decir, las restricciones externas al proceso de encontrar la solución
- Diseñar la estructura de árbol o grafo implícito que define los estados y transiciones entre estados de búsqueda de soluciones.
- Diseñar una función objetivo, la cual actuará como criterio de parada para encontrar las soluciones requeridas.
- Diseñar una función de poda $B_k(x_1, x_2, \dots, x_n)$ para eliminar la exploración de ramas que deriven en soluciones inadecuadas.
- Adaptar la estructura general del procedimiento Backtracking al problema y resolverlo.

-
3. Pseudocódigo que soluciona el problema
 4. Explicación del funcionamiento del algoritmo sobre el ejemplo
 5. Enunciado de un problema o caso real donde se pueda aplicar el algoritmo
 6. Cálculo del orden de eficiencia teórica del algoritmo
 7. Instrucciones sobre como compilar y ejecutar el código de la práctica