Métodos de búsqueda no informados

Gonzalo V. Castiglione, Alan E. Karpovsky, Martín Sturla
Estudiantes Instituto Tecnológico de Buenos Aires (ITBA)

15 de Marzo de 2012

Entrega Preliminar 1 - Informe

***Resumen*—El presente informe busca analizar y comparar distintas estrategias de búsqueda no informadas sobre un problema en particular haciendo uso de un motor de inferencias.**

***Palabras clave*—DFS, BFS, General Problem Solver, depth-first, breadth-first, search strategy**

I. Introducción

El juego *Edificios* también conocido como *Skyscraper puzzle* es una variante del conocido *Sudoku* y consiste de una grilla cuadrada con números en su borde que representan las pistas sobre cuántos edificios se visualizan en esa dirección. El tablero, visto desde arriba, representa un espacio cubierto de edificios. Cada casillero debe ser completado con un dígito que va entre 1 y N, siendo N el tamaño de la grilla; haciendo que cada fila y cada columna contengan sólo una vez a cada dígito (como sucede en el Sudoku).

En este puzzle, cada dígito puesto en la grilla podría ser visualizado como un edificio de esa altura. Por ejemplo, si ingresamos un 5, estaríamos colocando un edificio de altura 5. Cada uno de los numeros que están por fuera de la grilla revelan la cantidad de edificios que pueden ser vistos al mirar la línea o columna en esa dirección. Cada edificio bloquea a todos los edificios de menor altura de ser vistos, mientras que los edificios de mayor altura son vistos a través de él.

II. Estados del problema

A. Estado inicial

El estado inicial del problema es un tablero que contiene sólo las pistas en sus bordes (no contiene ningún edificio en la grilla). Para que el problema tenga solución éste tablero debe ser válido; es decir que no cualquier combinación de pistas sobre la visibilidad de edificios en esa dirección conducirán a un problema que resoluble.

B. Estado final

El estado final del problema es un tablero con todos los casilleros completos (lleno de edificios) que cumpla con las raglas del juego citadas anteriormente.

15 de Marzo, 2012.

III. Modelado del problema

Aaaaaa

IV. Reglas

Dado un tablero de tamaño $n \times n$ podemos describir las reglas del problema en forma general como sigue:

Poner un edificio de tamaño 1 *a* n *en la casilla* (i, j).

Por lo que, que dado un tablero de $n \times n$, el total de reglas está dado por $n^{n^n} = n^3$: por cada fila, por cada columna, se tienen n edificios distintos para colocar.

Ejemplos de reglas:

- Poner un edificio de altura 1 en la posición $(0,0)$
- Poner un edificio de altura 1 en la posición $(0,1)$
- Poner un edificio de altura 2 en la posición $(0,0)$
- Poner un edificio de altura 4 en la posición $(3,1)$

A. Metareglas

Si una restriccion (o pista) dice n (siendo n la dimendion del tablero) podemos definir una metaregla como sigue:

Colocar edificios de altura 1 *hasta* n *en forma ordenada y ascendente en esa fila o columna*

Esto reduce considerablemente toodos los algoritmos dado que quedarian menos casillas por explorar.

V. Costos

Debido a la naturaleza de nuestro probema, la aplciacion de todas las reglas tiene el mismo costo y éste es unitario. Es decir que la transición de un estado X a un estado Y, en nuestro caso, siempre nos cuesta 1.

VI. Algoritmos de búsqueda implementados

Hemos implementado cuatro algoritmos de búsqueda no informada distintos.

DFS Depth-first search
BFS Breadth-first search
IDFS Iterative DFS o profundización iterativa
HIDFS Hybrid Iterative DFS

VII. Heurísticas

Completar con heurísticas

VIII. Other

Método	Tiempo medio Octave (*sec*)	Tiempo medio Propio (*sec*)
Eliminación Gaussiana	0,011430	0,013100
Cholesky	0,009559	0,012204
QR	0,028912	0,031182

TABLE I
Tabla de comparación de performance de los métodos propios versus los métodos nativos de GNU Octave

Estos resultados son consistentes con la siguiente tabla[1], la cual presenta el número de operaciones necesrias para resolver el problema de cuadrados mínimos cuando $A \in \Re^{mxn}$:

Algoritmo	Número de operaciones
Eliminación Gaussiana	$n^2(m + \frac{2n}{3})$
Cholesky	$n^2(m + \frac{n}{3})$
QR	$2n^2(m - \frac{n}{3})$

TABLE II
Complejidad del problema de cuadrados mínimos

Si las columnas de la matriz A fueran casi linealmente dependientes, hubiera sido preferible la utilización del algoritmo **QR** frente a las ecuaciones normales. Como en este problema en particular esta condición no se cumple, el algoritmo más performante ha sido el de Cholesky.

[1] Fierens, P. (2011), *Cuadrados mínimos: repaso*, Buenos Aires: Instituto Tecnológico de Buenos Aires.

Referencias

[1] Fierens, P. (2011), *Cuadrados mínimos: repaso*, Buenos Aires: Instituto Tecnológico de Buenos Aires.
[2] Abdi, H., *Least-squares: Encyclopedia for research methods for the social sciences*, Thousand Oaks (CA): Sage. pp, 2003.
[3] Farebrother, R.W. (1988), *Linear Least Squares Computations, STATISTICS: Textbooks and Monographs*, New York: Marcel Dekker.
[4] Lipson, M.; Lipschutz, S. (2001), *Schaum's outline of theory and problems of linear algebra*, New York: McGraw-Hill, pp. 69–80.

Anexo A: Gráficos ilustrativos

Figura 1. Curva que ajusta las velocidades medidas en la estación RPT, obtenida mediante eliminación gaussiana.

Figura 2. Error de la curva de ajuste obtenida por eliminación gaussiana, con respecto a las velocidades de la estación RPT.

Anexo B: Código

A. Código para la obtención de las matrices A y b