

Examen final

Inteligencia Artificial
JULIO WAISSMAN VILANOVA

12 de mayo de 2016

1. Búsquedas

El *rompecabezas deslizante* es una versión diferente del 15 puzzle, en la cual cada línea y cada columna se deslizan, como si se encontrara en una esfera (por supuesto que este tipo de rompecabezas no se puede hacer en madera, pero en la computadora es facilísimo). Un esquema del entorno es el siguiente:

	↑	↑	↑	↑	
←	1	2	3	4	→
←	5	6	7	8	→
←	9	10	11	12	→
←	13	14	15	16	→
	↓	↓	↓	↓	

Las acciones que el agente puede realizar sobre el ambiente son: a) Girar por la derecha el renglón i ($i \in \{1, 2, 3, 4\}$); b) Girar por la izquierda el renglón i ($i \in \{1, 2, 3, 4\}$); c) Girar por arriba la columna j ($j \in \{1, 2, 3, 4\}$); y d) Girar por abajo la columna j ($j \in \{1, 2, 3, 4\}$). Se asume que el ambiente es completamente observable. La idea es que, después de aplicar un cierto número de movimientos aleatorios y no observados, el agente pueda realizar las acciones necesarias para regresar el sistema al estado mostrado en el esquema anterior.

1. Programa una clase del tipo de problema de búsquedas para el puzzle deslizante.
2. Desarrolla una función que aplica n movimientos aleatorios al puzzle deslizante.
3. Resuelve el problema después de 5 y 7 movimientos aleatorios, utilizando búsquedas no informadas (al menos un algoritmo). ¿Funcionará mejor una búsqueda en grafos o en árboles?
4. Desarrolla una heurística admisible y demuestra que efectivamente la heurística es admisible.
5. Resuelve el problema utilizando una búsqueda A^* . Muestra el número de nodos explorados y compáralos con los nodos explorados en la búsqueda no informada.

2. Búsquedas locales y satisfacción de restricciones

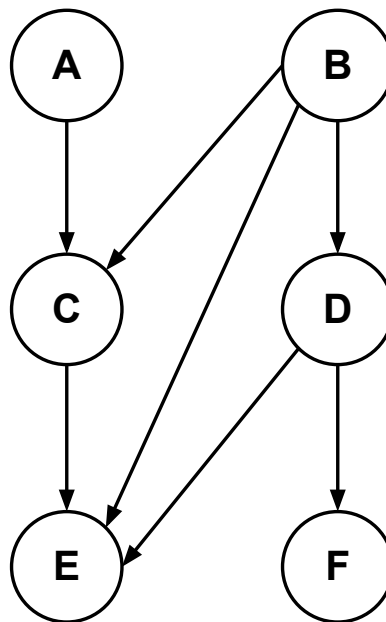
Considera el problema de colocar k caballos de ajedrez en un tablero de $n \times n$ tal que no se ataquen entre sí, donde k es un número dado, $k \leq n^2$.

1. Representa el problema como un problema de búsqueda local.

2. Programa el problema.
3. Aplica los métodos de descenso de colinas, temple simulado y un algoritmo genético. Discute los resultados.
4. Representa el problema como un problema de satisfacción de restricciones.
5. Resuelve el problema utilizando AC-3 (o al menos consistencia de tipo 1).

3. Redes bayesianas

Para la red bayesiana presentada en la figura, se asume que cada uno de los nodos puede tener 3 valores diferentes. Determina el número mínimo de parámetros que es necesario calcular.



4. Aprendizaje supervisado

En el «*Machine Learning Database Repository*» se encuentra la base de datos «*Car Evaluation*», que se puede consultar en la dirección:

<http://archive.ics.uci.edu/ml/datasets/Car+Evaluation>

La base de datos consta de 1728 objetos con 6 atributos y una clase.

Utiliza el 80% de los datos seleccionados al azar como datos de aprendizaje y 20% como datos de prueba. Utiliza los métodos siguientes:

- Vecino más próximo (1-NN),
- Naïve Bayes utilizando Laplaciano,
- Árbol de decisión.

Reporta los resultados.