

Primera práctica de Inteligencia Artificial

Curso 2016-2017

Departament d'Informàtica i Enginyeria Industrial
Universitat de Lleida
carlos@diei.udl.cat
jponfarreny@diei.udl.cat

1. Enunciado

El objetivo de esta práctica es evaluar el conocimiento del alumno sobre los problemas y algoritmos de búsqueda descritos hasta el momento en la asignatura. Los algoritmos se aplicarán en el contexto de los proyectos docentes de inteligencia artificial sobre el popular juego, Pac-Man ¹.

1.1. Búsqueda Informada en proyectos Pac-Man

En la página central de los proyectos Pac-Man podemos leer: “Los proyectos Pac-Man se desarrollaron para un curso introductorio de inteligencia artificial (CS 188) de UC Berkeley. Se aplican una serie de técnicas de IA para jugar al Pac-Man. Sin embargo, estos proyectos no se centran en la construcción de la IA de los videojuegos. En su lugar, se enseñan conceptos fundamentales, tales como búsqueda informada en el espacio de estados de un problema, inferencia probabilística y aprendizaje por refuerzo. Estos conceptos subyacen en áreas de aplicación del mundo real, tales como el procesamiento de lenguaje natural, la visión artificial y robótica.”

Nosotros nos centraremos en esta práctica en el proyecto de búsqueda ².

1.2. Algoritmos: 5 puntos

Los algoritmos que se deberán implementar son los siguientes:

- Búsqueda informada (búsqueda en grafo) - **5 puntos**
 - UCS - **1 punto**
 - Best-H first - **1 punto**
 - A* - **3 puntos**

Observaciones:

- Considerar que el coste de transición de un estado a otro es 1.
- Utilizar como heurísticas distancia de Manhattan y distancia Euclídea.
- Siempre que sea posible debes dar la versión optimal del algoritmo.

En este apartado se valora la corrección de los algoritmos implementados.

Opcional: Implementar un algoritmo de búsqueda bidireccional.

Opcional: Proponer una heurística adicional.

¹http://ai.berkeley.edu/project_overview.html

²<http://ai.berkeley.edu/search.html>

1.3. Implementación: 1 punto

En este apartado se valora la calidad y eficiencia de los algoritmos implementados. Es necesario tener en cuenta los siguientes aspectos de la implementación:

- El lenguaje de programación es Python.
- La utilización de un diseño orientado a objetos.
- La simplicidad y legibilidad del código.
- Se recomienda seguir la guía de estilo ³.

Nota 1: Tal como se ha comentado en clase, la corrección inicial será realizada con una herramienta automática. Por lo tanto para garantizar su correcto funcionamiento tendréis que ceñiros a la siguiente nomenclatura de las funciones:

- | | |
|-----------------------|-----------------|
| ■ Bidireccional → bds | ■ Best-H → bfsh |
| ■ UCS → ucs | ■ A* → astar |

En el caso de las heurísticas los nombres son los siguientes:

- | | |
|---------------------|----------------------------------|
| ■ Manhattan → mandH | ■ Propuesta por vosotros → custH |
| ■ Euclídea → eucdH | |

A vuestras funciones podéis ponerles el nombre que más os guste y luego al final del archivo *search.py* vereis una sección de *Abbreviations* donde podréis asignar otro nombre a las funciones que tengáis definidas, por ejemplo:

```
# ...
# Abbreviations
ucs = uniformCostSearch
bds = bidirectionalSearch
bfsh = greedyBestFirstSearch
astar = aStarSearch
# ...
```

Nota 2: La salida por pantalla del programa que espera la herramienta es la que tiene el proyecto Pac-Man por defecto, si para hacer *debug* necesitáis imprimir algo por pantalla, tomad precauciones para evitar confundir a la herramienta. Dos posibles soluciones:

- Quitad las líneas que imprimen por pantalla antes de entregar la práctica.
- Utilizad la salida estándar de errores `'print >> sys.stderr, "texto"'`.

1.4. Evaluación Experimental - 1 punto

A partir de las estadísticas proporcionadas tras la ejecución de los algoritmos sobre los diferentes mapas proporcionados, debéis valorar el rendimiento de los algoritmos.

Mostrar una tabla con los mapas y el rendimiento de los algoritmos. Utiliza como criterio el tiempo de ejecución y el número de nodos generados. La opción '-q' de la herramienta pacman anula el modo gráfico.

Opcional: Utiliza el generador de mapas aleatorios realizado en la sesión 1.

1.5. Problemas Bien definidos - 2 puntos

Blocks world problem: disponemos de un robot con un brazo articulado, de k bloques distinguibles y de una superficie con m posiciones. Los bloques pueden estar situados encima de otro bloque o de la superficie en una posición determinada. Debido a restricciones de la estructura de la superficie, cada posición de la superficie no puede sostener más de c bloques. El brazo articulado nos permite mover un bloque, que no tenga nada encima, a una posición permitida (sobre una posición de la superficie o sobre otro bloque). Dada una configuración

³<https://www.python.org/dev/peps/pep-0008/>

inicial de los bloques, y una configuración final, el objetivo del problema es determinar el número mínimo de movimientos necesarios para colocar todos los bloques según la configuración final.

Define adecuadamente el problema anterior, indicando, el estado inicial, el test de objetivo, las acciones, el modelo de transición y el coste de un paso.

Describe una heurística admisible y qué algoritmo utilizarías para resolver el problema.

Opcional: Incorpora este problema en la herramienta edulog-search.

1.6. Documentación - 1 punto

Documento en pdf (máximo ≈ 4 páginas) que incluya:

- una descripción de las decisiones que se han tomado en la implementación de los algoritmos y los resultados de la evaluación experimental.
- el problema bien definido del mundo de bloques

La página inicial ha de contener el nombre de los integrantes del grupo. La práctica puede realizarse en grupos de dos personas o individualmente.

Se valorará la redacción y presentación del documento.

1.7. Material a entregar

El material evaluable de esta práctica es:

- Todos los archivos de código fuente, modificados o añadidos. Estos ficheros se descargarán sobre el contenido del directorio *search*.
- Documento de la práctica.

Todo el material requerido se entregará en el paquete de nombre `ia-prac1.[tgz|tar.gz|zip]`

```
ia-prac1.[tgz|tar.gz|zip]
├── pacman
│   └── *.py
├── problem
│   └── *.py
└── informe.pdf
```

1.8. Utilidades para la realización de la práctica

Para limitar el tiempo y la memoria de la ejecución de la herramienta Pacman puedes utilizar el script *limited.sh*.

Ejemplos:

```
# Limitar tiempo de CPU a 30 segundos
$ ./limited.sh -t 30 python pacman.py -p SearchAgent -l mediumMaze -a fn=treeBFS
# Limitar la memoria a 150 MBytes
$ ./limited.sh -m 150 python pacman.py -p SearchAgent -l bigMaze -a fn=graphBFS
# Limitar tiempo de CPU a 120 segundos y memoria a 1 GB
$ ./limited.sh -t 120 -m 1024 python pacman.py -p SearchAgent -l bigMaze -a fn=astar,...
```