

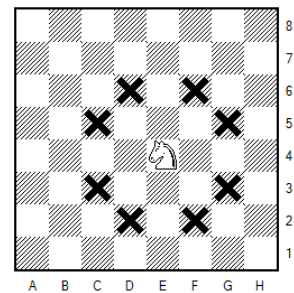
Inteligencia Artificial
4º Ingeniería Informática. Curso 2012/13
22 de octubre de 2012
Práctica 1: Implementación de métodos de búsqueda

*“La mitad de las variaciones que son calculadas
en una partida de torneo, resultan ser
completamente superfluas. Desafortunadamente,
nadie sabe por adelantado cual mitad.”*

JAN TINMAN

1. El problema

El ajedrez es un juego de inteligencia entre dos jugadores que representa una batalla entre dos ejércitos. Se desarrolla sobre un tablero cuadrulado de 64 casillas sobre el que cada jugador dispone de 16 piezas (un rey, una reina, dos alfiles, dos caballos, dos torres y ocho peones), que van moviendo alternativamente. El movimiento del caballo ♞ es en forma de L, es decir, dos casillas hacia adelante, dos hacia atrás, dos hacia la izquierda o dos hacia la derecha, y una casilla perpendicular a la anterior. La figura muestra gráficamente los movimientos permitidos a los que nos referimos.



En esta práctica se trata de obtener la ruta **más rápida** que debe seguir un caballo desde una casilla origen a una casilla destino de un tablero 10x10. Para ello formularemos el problema de búsqueda correspondiente, implementaremos distintos algoritmos de búsqueda ciega e informada, evaluaremos y extraeremos las conclusiones pertinentes. Para el cálculo de las rutas consideraremos que el coste puede verse modificado si la casilla destino de cada movimiento tiene o no tiene una penalización n añadida.

2. Los objetivos

Los objetivos de esta práctica son los siguientes:

1. Aplicar estrategias de búsqueda *ciega* y búsqueda *informada* con el fin de planificar una secuencia de acciones que permitan encontrar una solución cumpliendo con las restricciones del problema.
2. Realizar una comparación entre los distintos métodos aplicados.
3. Determinar el método de búsqueda más adecuado para este tipo de problemas.
4. Desarrollar heurísticas apropiadas para el problema, valorar su aportación y evaluar su calidad.

En cuanto a las estrategias de búsqueda ciega, se deberá realizar un análisis del problema que ayude en la elección de uno de los dos métodos clásicos (recorrido en anchura o profundidad) **antes** de implementarlos. Esta elección deberá **justificarse** en términos del objetivo del problema y los costes computacionales y espaciales de los algoritmos. Como estrategia de búsqueda informada se aplicará el algoritmo A*.

3. La implementación

La práctica se podrá realizar en cualquier lenguaje y será ejecutable bajo UNIX/Linux en las máquinas asignadas a las prácticas de la asignatura sin necesidad de ninguna plataforma o entorno de desarrollo (p.e. Netbeans, Eclipse, etc...), es decir, en la línea de comandos, como un applet en un navegador, etc.



Se desarrollará una interfaz (**muy sencilla**) que permita:

1. Indicar el estado inicial del problema, I , es decir, la posición del caballo en el tablero, las casillas con penalización (si hay) y su valor¹. En el caso de que existan casillas con penalización, seleccionar el método de asignación:
 - a) El usuario indica qué casillas tienen penalización y cuál es su valor.
 - b) El usuario indica cuántas casillas tienen penalización y cuál es su valor (único), y es el programa el que, de forma aleatoria, determina las posiciones de las casillas.
2. Seleccionar el estado meta deseado, O , es decir, la posición final que debe alcanzar el caballo.
3. Seleccionar el método de búsqueda y, si es A*, la función heurística a aplicar.

Como norma general, el programa deberá visualizar la solución mostrando el camino desde el estado inicial I al objetivo O junto con todos los pasos seguidos. Es decir, la salida por pantalla deberá permitir seguir la traza de ejecución del algoritmo seleccionado.

Además, en cada iteración de cada uno de los algoritmos se mostrará:

- Algoritmo A*:
 - Lista de nodos abiertos (estados candidatos para continuar la búsqueda)
 - Mejor nodo seleccionado de entre los candidatos
 - Camino actual hasta el mejor nodo seleccionado
 - Nuevos descendientes generados
- Algoritmo en Anchura:
 - Nivel actual de exploración (lista de nodos abiertos)
- Algoritmo en Profundidad
 - Camino actual

¹El estado inicial no puede tener penalización

- Nuevos descendientes generados
- Mensaje en caso de *backtracking*

Para mostrar los nodos del árbol, seguir OBLIGATORIAMENTE la siguiente estructura:

- Algoritmos de Búsqueda Ciega: (ID)
- Algoritmo A*: ($ID, g(n), h(n)$)

donde ID es el nuevo nodo generado, $h(n)$ es el valor de la heurística y $g(n)$ es el coste del camino óptimo actual (encontrado hasta ese momento).

En cualquier caso, al final de la ejecución se mostrará el camino encontrado y el coste del mismo. Si no fuese posible encontrar una solución al problema se mostrará un mensaje indicando que el éste no es resoluble.

4. La memoria de prácticas

La memoria se realizará de dos volúmenes y se ajustará a las normas publicadas en la web (existen modelos de documentos en L^AT_EX o Microsoft Word/OpenOffice).

4.1. Primera parte

La primera parte de la memoria de prácticas tendrá una extensión total máxima de 10 páginas (excluyendo portada, resumen e índice) y contendrá **OBLIGATORIAMENTE** los siguientes apartados:

1. **Portada.** Indicará el título (*Resolución de problemas de búsqueda. Memoria de Prácticas de Inteligencia Artificial. Primera Entrega*), los autores, el “login” de cada uno, el directorio de depósito y la fecha.
2. **Resumen.** Escribir con no más de 150 palabras los objetivos fundamentales de esta parte de la práctica.²
3. **Índice.** Incluir la tabla de contenidos.
4. **Métodos.** (Página 1)

a) Definir los conceptos de:

- *Espacio de estados* del problema, en términos de estado inicial, estado meta y conjunto de reglas que describen las acciones (operadores y sus restricciones) disponibles.
- Prueba de meta
- Función de coste (algoritmo A*)

y aplicar las definiciones anteriores al problema en cuestión para describir **formalmente** estos conceptos.

²El resumen no debería ser una copia del enunciado de la práctica.

- b) Análisis (sencillo) que justifique la elección de uno de los dos métodos de búsqueda ciega (anchura o profundidad) para este problema, teniendo en cuenta los requisitos (teóricos) de memoria y tiempo de ejecución y sus características en cuanto a óptimos y completos.
- c) Definir el concepto de función heurística. Proponer al menos dos heurísticas para el problema³. Para cada una de ellas, describir:
 - 1) La expresión en términos matemáticos y una explicación de su idoneidad.
 - 2) La demostración—si no es posible formalmente, si al menos teóricamente—de que en ningún caso sobreestima el coste real.
 - 3) Una explicación acerca de si la heurística da lugar a la aparición de mínimos locales en el espacio de estados. Un espacio de estados contiene un mínimo local si la heurística proporciona un valor menor (mejor) para un estado que para otro que realmente se encuentra más cerca de la meta.
- d) Suponer una situación particular en la que además de casillas con penalización, existan muros verticales del grosor de una casilla que el caballo no pudiera “saltar”. Contestar a las siguientes preguntas:
 - 1) ¿Seguirían siendo válidas todas las heurísticas? Si no fuera así, ¿cómo podríamos adaptarlas?
 - 2) ¿Qué algoritmos de los propuestos y bajo qué condiciones alcanzarían la solución al problema? ¿Sería óptima igualmente?
- e) Suponer otra situación particular en la que la pieza que debe moverse sea el rey ♔: avanza sólo una posición pero en cualquier dirección—horizontal, vertical o diagonal. Contestar a las siguientes preguntas:
 - 1) ¿Seguirían siendo válidas todas las heurísticas?
 - 2) Si no fuera así, ¿cómo podríamos adaptarlas?
- f) Sobre la situación anterior, suponer ahora que el coste de algunos operadores, como los de desplazamiento horizontal, es superior al de otros.
 - 1) ¿Seguirían siendo válidas todas las heurísticas?
 - 2) Si no fuera así, ¿cómo podríamos adaptarlas?

5. Bibliografía

4.2. Segunda parte

La segunda parte de la memoria de prácticas contendrá **OBLIGATORIAMENTE** los siguientes apartados:

1. **Portada.** Indicará el título (*Resolución de problemas de búsqueda. Memoria de Prácticas de Inteligencia Artificial. Segunda Entrega*), los autores, el “login” de cada uno, el directorio de depósito y la fecha.
2. **Resumen.** Escribir con no más de 150 palabras los objetivos fundamentales de esta parte de la práctica.

³Puede usarse el método de generación de heurísticas por relajación de restricciones (véanse transparencias).

3. **Índice.** Incluir la tabla de contenidos.

4. **Resultados.**

a) Ejemplos de ejecución.

Se mostrará la traza completa de la ejecución de cada uno de los algoritmos de búsqueda (al menos en un ejemplo) siguiendo estrictamente el formato expresado en el apartado 3.

b) Caracterización de la calidad de las heurísticas empleadas.

Sea $h^*(s)$ la función que devuelve el coste real de un camino de coste mínimo desde el estado s hasta el estado meta O . La función de evaluación heurística $h(s)$ es una estimación de $h^*(s)$. Una heurística admisible es aquella que nunca sobrestima $h^*(s) : h(s) \leq h^*(s)$. La medida de precisión de heurísticas que utilizaremos será el *error absoluto esperado*: $E(h^*(s) - h(s))$. Este error se puede calcular computando la media de los errores absolutos a través de un conjunto n de estados aleatoriamente seleccionados (en nuestro caso, $n \geq 15$). Una vez obtenidos los valores implicados, se construirá una tabla con la siguiente cabecera:

<i>Ejemplo</i>	$h(s)$				$h^*(s)$
estado s_1	$h_1(s_1)$	$h_2(s_1)$...	$h_n(s_1)$...
...
estado s_n	$h_1(s_n)$	$h_2(s_n)$...	$h_n(s_n)$...
<i>Promedio</i>	$E(h_1(s))$	$E(h_2(s))$...	$E(h_n(s))$	$E(h^*(s))$

Tabla 1: Tabla comparativa de rendimiento de heurísticas

donde el estado s_i , con $1 \leq i \leq n$, es un ejemplo en particular, y $h_1(s_i)$, $h_2(s_i)$, ..., $h_n(s_i)$ son los valores correspondientes a las heurísticas propuestas (al menos 2) en s_i , y E es el operador esperanza.

c) Comparación del rendimiento de los distintos métodos de búsqueda implementados.

Se construirá una tabla con la siguiente cabecera:

	Coste de la búsqueda			Factor Ramificación Efectivo		
d	Ciega	A*(h1)	A*(h2)	Ciega	A*(h1)	A*(h2)
-	-	-	-	-	-	-

Tabla 2: Tabla comparativa de rendimiento de métodos de búsqueda

Para rellenarla es necesario generar n problemas aleatorios (en nuestro caso, $n \geq 15$), resolverlos con los métodos de búsqueda implementados, y agruparlos en función de la longitud de la solución obtenida para obtener valores PRO-MEDIOS en cada una de las filas (con una precisión de 2 decimales).

El significado de los parámetros de la tabla es el siguiente:

- d : es el valor de profundidad del árbol-grafo de exploración en el cual se ha hallado la solución.
- Coste de la búsqueda: número total de nodos seleccionados para expandir durante el proceso de búsqueda (nodos en la lista de cerrados).

- Factor de ramificación efectivo: Si el número total de nodos generados es N y la profundidad de la solución es d , entonces b^* es el factor de ramificación que tendría un árbol uniforme de profundidad d con N nodos⁴. Esta medida representa el número medio de caminos intentados para cada estado. Si el factor de ramificación efectivo fuera 1, nuestro camino de búsqueda obtenido por el algoritmo sería un camino solución. Es decir, cuanto más se aproxima a 1 nuestra heurística, la búsqueda estará más dirigida al objetivo, con muy pocas ramificaciones en el árbol. Normalmente, para una heurística h , b^* es mas o menos constante en varias instancias del problema.

El factor de ramificación eficaz b se calcula solucionando la incógnita b en la ecuación siguiente (donde N es el número de nodos visitados y d la profundidad de la solución), asumiendo un árbol uniforme:

$$N = 1 + b + b^2 + \dots + b^d = \frac{(b^{d+1} - 1)}{(b - 1)} \quad (1)$$

En el [Campus Virtual](#) se encuentra el código que resuelve los cálculos.

5. Discusión

- a) Conclusiones sobre los resultados obtenidos en la comparativa entre heurísticas del apartado [4b](#).
- b) Comentar el origen de las ventajas e inconvenientes que aportan para este problema los métodos de búsqueda propuestos (no se espera en este apartado una copia de las comparativas generales entre estos métodos que se pueden encontrar en la bibliografía). Se incluirá una discusión sobre los resultados obtenidos en el apartado [4c](#).
- c) Inventa una función heurística para el problema que en ocasiones sobreestime y muestra, utilizando el algoritmo A*, cómo produce una solución subóptima en un ejemplo en particular (es obligatorio incluir el proceso completo, no sólo la solución).

6. Bibliografía

7. Apéndices

Código de las principales unidades de la práctica (algoritmos, generación de sucesores, comprobación de repetidos, etc.) y cualquier otra cosa de interés.

5. Los plazos de entrega

La práctica tendrá dos plazos de entrega improrrogables:

- Primera parte de la práctica hasta el **21 de diciembre de 2012**
- Segunda parte de la práctica hasta el **20 de febrero de 2013**.



⁴Un árbol uniforme es aquél en el cual todos los nodos expandidos (no hojas) tienen el mismo factor de ramificación y la profundidad de todos los caminos de búsqueda es constante.

La entrega de las dos partes de la memoria y el **código** (junto con un fichero *makefile* o, en su defecto, instrucciones claras de compilación y ejecución exclusivamente “en línea”) se efectuará **electrónicamente** depositando los ficheros en los servidores de docencia, para su recogida automática, bajo el directorio /PRACTICAS/EI/IA/P1/login_alumno.

6. La evaluación



Una práctica estará suspensa si no está completa su MEMORIA o su IMPLEMENTACIÓN. No se valorarán elementos ajenos al tema de la práctica tales como gráficos, música, o cualquier otro aspecto de la interfaz o funcionalidad que no se hubiera requerido en el enunciado.

Se podrá citar a la defensa de la práctica a los autores de los trabajos que tengan que aclarar algún aspecto (apartados incompletos, falta de instrucciones claras de ejecución, ...).

7. NOTAS IMPORTANTES

- Se recuerda que las prácticas DEBERÁN realizarse por parejas.
- Cualquier duda o consulta generalizada se comentará en un hilo específico del foro *Preguntas frecuentes* en el [Campus Virtual](#)
- No existen horas de docencia presencial en el laboratorio. El profesorado atenderá las dudas en el despacho, a través del [Campus Virtual](#) o por correo electrónico.
- Contacto: Mariano Cabrero Canosa.
 - Tutorías en <http://www.fic.udc.es>
 - email: mariano.cabrero@udc.es

Referencias

- [1] V. Moret, A. Alonso, M. Cabrero, B. Guijarro, E. Mosqueira. *Fundamentos de Inteligencia Artificial (2ª Ed)*. Servicio de Publicaciones, UDC, 2000
- [2] N. Nillson. *Inteligencia artificial: una nueva síntesis*. McGraw-Hill, 2001
- [3] Prieditis, R. Davis. Quantitatively relating abstractness to the accuracy of admissible heuristics, *Artificial Intelligence*, vol.74, pp. 165-175, 1995
- [4] E. Rich, K. Knight. *Inteligencia Artificial (2ª ed)*. McGraw-Hill, 1994
- [5] S. Russell, P. Norvig. *Inteligencia Artificial: Un enfoque moderno*. Prentice-Hall, 2004.