

# HERRAMIENTAS DE APOYO EN LA DOCENCIA DE LA INTELIGENCIA ARTIFICIAL

Luis Antonio Alvarez Llorente, Pilar Bachiller Burgos  
Departamento de Informática. Escuela Politécnica  
Universidad de Extremadura  
Avda. de la Universidad s/n  
10071 Cáceres  
{luisalva, pilarb}@unex.es

## Resumen

*La asignatura de Inteligencia Artificial se imparte en la Universidad de Extremadura desde el curso académico 95-96. Se trata de una materia troncal de segundo ciclo, con un alumnado muy heterogéneo y con un perfil amplio. El temario se ha diseñado para abarcar unos conocimientos generales que sirvan de base a todos los alumnos. Se ha dividido en tres grandes apartados: Representación del conocimiento, Resolución de problemas y Adquisición de conocimiento. Asimismo, se proponen tres prácticas complementarias, cuyo principal objetivo es dar a conocer los dos lenguajes típicos de la IA: Prolog y Lisp. Como apoyo en la docencia, se presentan tres herramientas que tratan de facilitar la comprensión de otros tantos temas de la materia. La primera es un simulador del algoritmo A\*, que permite estudiar el comportamiento del mismo resolviendo el conocido problema del 8-puzzle. El segundo desarrolla el sistema Strips, generando planes para un mundo de bloques ideal. Finalmente, el tercero es un simulador de una red neuronal del tipo backpropagation, que puede ser entrenada para diferentes patrones. Todas ellas han sido desarrolladas por alumnos de la Escuela en diferentes proyectos fin de carrera.*

## 1 Objetivos

La asignatura de *Inteligencia Artificial* se imparte en la Universidad de Extremadura desde el curso académico 95/96 en los estudios de Ingeniería Informática. Se trata de una materia troncal de cuarto curso, recogida en los nuevos planes de estudio que se implantaron en la Escuela Politécnica de Cáceres hace ahora seis años.

Al ciclo superior se puede acceder directamente dentro de la Ingeniería Informática o como Ingeniero Técnico de Sistemas o Gestión, o bien como Diplomado en el caso de alumnos provenientes del plan antiguo. Esta forma de acceso implica que los conocimientos con los que llegan los alumnos sean muy heterogéneos. De los 93 alumnos matriculados en el presente curso, aproximadamente un 30% son Diplomados que desean completar sus estudios; pero entre ellos, hay quienes cursaron la carrera en esta misma Escuela y quienes estudiaron en otras Universidades; los hay que terminaron el curso anterior y los hay con más de 5 años de experiencia laboral...

Otro aspecto importante a destacar es el perfil dado a este ciclo superior. Como hemos mencionado anteriormente, la asignatura de *Inteligencia Artificial* es troncal. Sin embargo, es la única de obligado estudio relacionada con esta disciplina. Es cierto que para los más interesados en la materia se ofertan algunas materias optativas, tales como *Redes Neuronales*, *Robótica* o *Programación Declarativa*, pero en cualquier caso, no ofrecen un panorama demasiado amplio para lo extensa que puede llegar a ser la *Inteligencia Artificial* como disciplina.

Estos dos factores condicionan bastante los objetivos de la asignatura. Por un lado, no podemos olvidar que la mayoría de los alumnos desconocen aspectos básicos de la IA; y por otro lado, debe servir de base para todos los que deseen ampliar sus conocimientos cursando materias optativas.

Por tanto, el objetivo fundamental será dotar al alumno de unos conocimientos generales que le permitan tener una visión global de lo que es la Inteligencia Artificial. De esta forma, tendrán la base necesaria para continuar cursando otras materias optativas afines. O si deciden seguir otros perfiles, al menos conocerán las cuestiones básicas de la IA.

## 2 Temario de la asignatura

Visto el contexto en el que se desenvuelve la asignatura de *Inteligencia Artificial*, vamos a centrarnos en la misma. La carga lectiva de la asignatura es de 9 créditos anuales, repartidos en 6 créditos teóricos y 3 créditos prácticos, lo que se traduce en que los alumnos reciben dos horas de teoría y una hora de prácticas semanalmente. El temario es esencialmente similar al impartido en asignaturas análogas de otras universidades españolas [1][2]. Para intentar cumplir los objetivos marcados se ha confeccionado un temario amplio. Básicamente lo podemos dividir en tres partes fundamentales:

- Representación del conocimiento.
- Resolución de problemas.
- Adquisición de conocimiento.

Dentro de la primera parte se estudia la *lógica matemática* como método básico de representación del conocimiento [3]. La estrategia que se sigue es la de dar una idea de qué es la lógica matemática, para qué sirve y cuál es su uso dentro de la IA. Para los alumnos más interesados en el tema se oferta en quinto curso una asignatura optativa, en la que pueden ampliar sus conocimientos en esta disciplina.

Como complemento a la lógica, se tratan las *redes semánticas* y los *frames (marcos)* [3][6] como forma de representación estructurada del conocimiento. En ambos casos, los ejemplos son básicos para entender y manejar estas dos técnicas de representación.

La segunda parte, dedicada a la resolución de problemas, es quizás la más amplia en cuanto a dedicación y profundidad. Básicamente está dividida en tres bloques :

- En un primer apartado se revisan diferentes algoritmos que emplean la búsqueda guiada mediante heurísticas como forma habitual de

trabajo, frente a otros métodos clásicos de búsqueda no guiada. Así, quedan recogidos dentro de este bloque los *algoritmos de escalada, enfriamiento simulado, A\*, AO\**, etc [3]. De entre todos ellos cabe destacar por el especial interés que tiene para los alumnos el tema de la resolución de juegos, al que se dedica un capítulo.

- En un segundo bloque se estudian algunos programas clásicos dentro de la Inteligencia Artificial, como el LT (*Logic Theorist*) o el GPS (*General Problem Solver*) [7]. Se trata de programas que, como alternativa a la búsqueda heurística, emplean procedimientos que emulan ciertas técnicas del ser humano: dividir el problema a resolver en subproblemas de menor entidad, e ir solucionando cada uno de ellos comenzando por los de mayor dificultad. Se ilustra el análisis de medios y fines como método general.
- Finalmente nos centramos en otros sistemas más evolucionados que emplean la planificación como alternativa a la búsqueda, y para ello se toma como base el sistema STRIPS [3][7]. Se trata de un planificador muy general para el que se definen operadores básicos con los que pueda trabajar en un universo simplificado. En concreto se trabaja con un brazo de robot que apila y desapila bloques en un mundo ideal.

En la última parte, dedicada a la adquisición de conocimiento, se estudian diferentes formas de aprendizaje: *memorístico, por inducción, por analogía, por descubrimiento*, etc [3][4]. En todos los casos se toman como base ejemplos de la forma de aprendizaje en cuestión, de los que se extraen los conceptos básicos, la metodología general del procedimiento y diferentes conclusiones.

También se plantean los fundamentos de los *algoritmos genéticos* [4] y las *redes neuronales* [3], cuestiones en las que posteriormente se puede profundizar y ampliar conocimientos en distintas materias optativas.

Por último, como forma de adquisición y representación de información un tanto especial, se aborda el tema del *procesamiento de lenguaje natural*. Se trata de una materia muy amplia, a la que bien se podría dedicar una asignatura optativa completa. En un recorrido básico por la IA no podemos pasar por alto este tema, pero dado que el

objetivo es dotar al alumnos de conocimientos generales, se tratan los contenidos sin profundizar en las distintas técnicas que se emplean en el uso del lenguaje natural como interfaz.

### 3 Las prácticas

Hasta el momento no hemos hablado de los 3 créditos de prácticas de la asignatura. El objetivo que nos marcamos en las prácticas es doble: por un lado pretendemos que los alumnos puedan ver la Inteligencia Artificial como algo tangible, que tiene una utilidad más allá de lo puramente teórico, aprovechando para ello los conocimientos adquiridos en las clases de teoría. Por otro lado es interesante que conozcan los dos lenguajes más utilizados en Inteligencia Artificial: Lisp y Prolog. En este curso académico se han realizado tres prácticas.

- La primera práctica se realiza sobre el lenguaje Prolog [8]. Las sesiones de laboratorio se dedican a explicar los fundamentos de este lenguaje, que para la mayoría de los alumnos supone romper con todos los esquemas de lenguajes de programación conocidos, tales como Pascal o C++, que emplean un modelo de programación muy distinto al de Prolog. En las primeras sesiones intentamos que el alumnado se familiarice con el lenguaje, comenzando con la redacción de pequeños programas y acabando con la resolución de problemas de cierta entidad. Estas experiencias se llevan a cabo en el laboratorio con la ayuda del profesor. Para finalizar con este bloque, se plantea una práctica de mayor entidad, que los alumnos deben completar por sí mismos.

En concreto, La práctica que entregaron este año consistía en resolver en Prolog el conocido problema de *los misioneros y los caníbales*[5]. Se trata de mostrar la secuencia de pasos necesarios para que 3 misioneros y 3 caníbales atraviesen un río, teniendo en cuenta que sólo disponen de una barca (en la que pueden montar dos personas) y que en ninguna de las dos orillas puede haber mayoría de caníbales, ya que se comerían a los misioneros...

En cuanto al software empleado, para esta primera práctica usamos el intérprete de libre distribución SWI Prolog versión 1.9.5 [13], que permite escribir programas Prolog en un fichero y pasárselo al intérprete para su ejecución.

- Durante el tiempo en que se desarrolla esta primer práctica, en las clases de teoría se ha avanzado en el programa para explicar uno de los temas que los alumnos siguen con mayor interés: la resolución de juegos. En este tema se explican los dos algoritmos fundamentales de la teoría de juegos: la *regla Min-Max* y el *algoritmo de poda Alfa-Beta* [3][7]. De igual modo, aunque sin tanto detalle, se enumeran diferentes mejoras que se pueden introducir en estos algoritmos según el tipo de juego a resolver.

Conocidos los fundamentos teóricos de la resolución de juegos, se plantea la segunda práctica: implementar un juego *inteligente*, que permita que un jugador humano desafíe a la computadora. Este año el juego propuesto es el de *las 4 en raya*. Además, en este curso académico hemos puesto en marcha una experiencia que ha resultado de gran interés: combinar esta práctica con la tercera práctica de la asignatura de *Redes de Ordenadores*<sup>1</sup>. En esta última se propone desarrollar un sistema de comunicaciones entre varios ordenadores equipados con tarjetas *ethernet*, de tal forma que en un momento determinado dos equipos puedan establecer comunicación y transmitirse información. Esta información se corresponderá con los datos necesarios para que esos dos equipos puedan jugar una partida a *las 4 en raya*.

El aspecto más importante de esta práctica no es tanto la presentación final del programa, sino las heurísticas empleadas, que permiten a la máquina jugar mejor o peor. Así, no se ha valorado tanto la interfaz de usuario como las funciones heurísticas diseñadas y los algoritmos empleados.

La práctica conjunta se implementó en Borland C++ versión 4.0 [11][12], principalmente por ser este el lenguaje empleado en la asignatura de *Redes de Ordenadores*.

- Finalmente, la última práctica trata de introducir a los alumnos en el manejo del lenguaje Lisp [9][10], uno de los más usados en el desarrollo de programas dentro de la Inteligencia Artificial.

---

<sup>1</sup> Redes de Ordenadores es una asignatura obligatoria de cuarto curso de Ingeniería Informática.

Esta vez el trabajo se ha desarrollado a medio camino entre el laboratorio y el aula, puesto que no se ha planteado una práctica final, sino una serie de cuestiones y ejercicios prácticos sobre lo estudiado. Las primeras sesiones se dedican al estudio del lenguaje, su sintaxis y sus características, para terminar realizando programas en Lisp que implementan funciones sencillas. Se plantean 5 problemas de mayor complejidad para ser resueltos por los alumnos y entregados a modo de trabajo.

Para esta última práctica usamos Allegro Common Lisp versión 3.0.1 [14]. Este intérprete permite escribir programas Lisp directamente o en ficheros que después son procesados.

#### 4 Herramientas de apoyo en la docencia

Con el fin de facilitar la comprensión de cada uno de los conceptos expuestos a lo largo de la asignatura, se dispone de un conjunto de herramientas de apoyo para distintas partes del temario, desarrolladas como proyectos fin de carrera por alumnos de la propia escuela.

##### 4.1 Simulador de algoritmo A\*

Para la parte de resolución de problemas, se han utilizado dos herramientas. La primera es una aplicación que implementa el algoritmo de búsqueda A\* [3][5] para resolver el conocido problema del 8-puzzle[5][7]. Este problema consiste en la reordenación de 8 piezas numeradas, situadas en un tablero de 3x3, hasta alcanzar una configuración final denominada *estado objetivo*. Uno de los cuadros del tablero queda siempre vacío, lo cual posibilita el movimiento de las fichas adyacentes a él. Mediante estos movimientos es posible obtener una configuración objetivo partiendo de una situación inicial de las fichas del tablero. El alumno selecciona una situación inicial y una situación objetivo del tablero (figura 1) y puede ir visualizando los distintos pasos que va dando el algoritmo para resolver el problema.

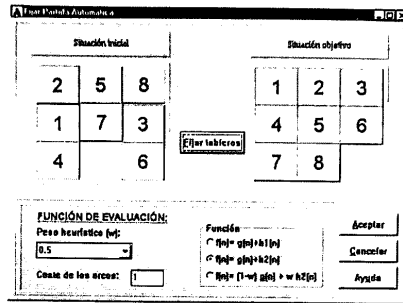


Figura 1: Estados inicial y objetivo del tablero.

La herramienta permite que el alumno elija una de entre varias funciones de evaluación heurística que el algoritmo utiliza para pasar de un estado a otro. Dichas funciones realizan una estimación, en cada uno de los nodos del árbol generado, del coste de pasar del nodo actual al nodo objetivo; es decir, miden lo *prometedor* que es cada movimiento para alcanzar la situación objetivo. Para calcular ese valor, cada función se descompone en la suma de dos subfunciones que llamamos  $g(n)$  y  $h(n)$ , siendo  $n$  el nodo sobre el que se realiza la evaluación. La función  $g(n)$  mide el coste necesario para llegar al nodo  $n$  desde el nodo inicial;  $h(n)$  es una función heurística que se encarga de estimar el coste de alcanzar la situación objetivo a partir del nodo  $n$ . Distintas combinaciones de  $g$  y  $h$ , permiten definir las siguientes funciones:

- $f(n)=g(n)+h1(n)$ : La función heurística  $h1$  tiene en cuenta el número de piezas mal situadas en el tablero evaluado con relación al tablero objetivo. El valor de la función se obtiene sumando un punto por cada pieza del tablero que esté mal colocada, sin tener en cuenta la posición del hueco.
- $f(n)=g(n)+h2(n)$ : En la función heurística  $h2$  se tienen en cuenta dos aspectos:
  - *Piezas que están mal colocadas con respecto al tablero objetivo.* Se suman dos puntos por cada pieza mal colocada. Además si la posición central no está libre, se suma otro punto. A este valor lo vamos a llamar  $s$ .
  - *Distancia de cada pieza a su posición en el tablero objetivo (distancia de Manhattan)* [5]. La suma de estas distancias la denotamos como  $p$ .

Utilizando estos criterios,  $h2$  se calcula de la siguiente forma:  $h2=p+3*s$ .

- $f(n)=(1-w)g(n)+ wh_2(n)$ : En función del valor que se asigne al peso ( $w$ ), se dará más importancia a la función heurística  $h_2$ , es decir, premiar el hecho de que el estado actual esté próximo al estado objetivo, o a la distancia  $g(n)$ , es decir, intentar alcanzar la solución en el menor número de pasos posibles. Desde el punto de vista del árbol generado, variará la componente en anchura que interviene en el proceso de búsqueda. Cuanto mayor valor tenga  $w$ , más importancia tienen los movimientos próximos a la solución y, por lo tanto, la anchura del árbol generado disminuye.

La elección de una u otra función de evaluación variará visiblemente el comportamiento del algoritmo, pudiendo apreciarse dicho comportamiento en el árbol de exploración generado.

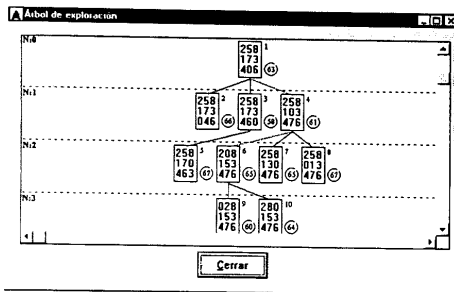


Figura 2: Árbol de exploración generado.

Una vez que el algoritmo ha finalizado su ejecución, el árbol de exploración generado es mostrado por pantalla de manera detallada, pudiendo apreciarse el valor de la función de evaluación para cada uno de los nodos (figura2). De esta forma, el alumno puede comprobar los pasos seguidos por el algoritmo para alcanzar la solución, lo que permite establecer una comparación entre las distintas funciones de evaluación.

#### 4.2 Planificador STRIPS

La segunda herramienta desarrollada implementa el sistema de planificación STRIPS, que trabaja sobre un mundo ideal de bloques compuesto por un brazo de robot, una superficie a la que llamaremos *mesa* y un conjunto de bloques con los que el robot puede realizar varias operaciones: coger bloques, apilarlos, desapilarlos, etc.

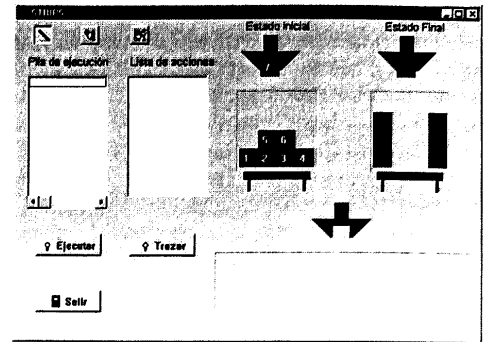


Figura 3: Definición de los estados inicial y final.

El algoritmo de planificación debe partir de una situación inicial y alcanzar una situación objetivo aplicando distintos operadores. Para ello, el alumno debe especificar el estado inicial y final del mundo de bloques (figura 3). Usando el ratón, puede marcar las posiciones de los distintos bloques que van a componer cada estado. Además, el estado final puede tener variables (bloques no identificados), lo cual no está permitido en el estado inicial.

STRIPS dispone de una pila de objetivos en la que se almacena la tarea que le queda para completar el plan. El primer paso del algoritmo es, por lo tanto, almacenar en dicha pila el estado final del mundo de bloques, para lo cual se utilizan los siguientes predicados:

- **SOBRE(A,B)**: El bloque  $B$  está situado encima del bloque  $A$ .
- **SOBREMESA(A)**: El bloque  $A$  está sobre la *mesa*.
- **LIBRE(A)**: El bloque  $A$  no tiene nada encima.
- **COGIDO(A)**: El brazo del robot tiene cogido el bloque  $A$ .
- **MANOVACIA**: El brazo del robot no tiene cogido ningún bloque.

Mediante la conjunción de estos literales es posible expresar cualquier estado de nuestro mundo de bloques.

El proceso de resolución consiste en explorar iterativamente la cima de la pila. Si el literal examinado forma parte de la descripción del mundo actual, se saca de la pila. Si no, se busca un

operador que permita modificar el estado actual de forma que se verifique el predicado en cuestión. Los operadores de que dispone STRIPS son:

- DESAPILAR(X,Y): Coger el bloque Y que está situado sobre el bloque X.
- APILAR(X,Y): Apilar el bloque Y sobre el X.
- COGER(X): Coger el bloque X que debe estar situado sobre la mesa.
- DEJAR(X): Colocar el bloque X sobre la mesa.

Cada uno de los operadores anteriormente definidos está formado por una lista de precondiciones, que deben cumplirse para poder ejecutarlo, y una lista de adición y otra de substracción que modifican el estado actual.

El algoritmo finalizará su ejecución cuando la pila quede vacía. En ese caso, se habrán verificado todos los predicados mediante los cuales se expresaba el estado final y, por lo tanto, el estado actual del mundo de bloques coincidirá con la situación objetivo que se pretendía alcanzar.

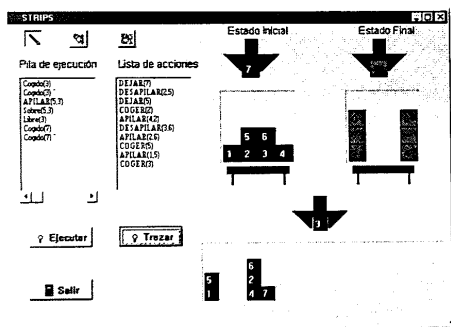


Figura 4: Ejecución del algoritmo de planificación.

La herramienta empleada para simular el sistema STRIPS permite visualizar en cada paso de ejecución el estado de la pila de objetivos. Además, muestra también una lista de acciones que va almacenando en orden los operadores aplicados (figura 4). Esta lista contendrá, por lo tanto, el plan generado que permite pasar del estado inicial al final. La aplicación presenta en todo momento el estado actual del mundo de bloques, pudiendo apreciar los cambios producidos en él como resultado de la ejecución de cada uno de los operadores que forman el plan (figura 4).

### 4.3 Red Neuronal Backpropagation

Como ejemplo de un sistema de adquisición de conocimiento, utilizamos una herramienta que implementa una red neuronal backpropagation[3]. La red se emplea principalmente como clasificador de patrones, pudiendo ser entrenada con cualquier conjunto de patrones que contengan hasta un máximo de 30 características.

Red Neuronal Tipo BackPropagation										
Valores a la salida de la Red para cada Patrón										
Pat.1	Pat.2	Pat.3	Pat.4	Pat.5	Pat.6	Pat.7	Pat.8	Pat.9	Pat.10	
0.104	0.104	0.100	0.104	0.101	0.105	0.103	0.108	0.099	0.098	
0.099	0.097	0.100	0.104	0.099	0.101	0.103	0.101	0.097	0.097	
0.094	0.092	0.099	0.098	0.102	0.105	0.105	0.096	0.103	0.104	
0.102	0.100	0.102	0.108	0.103	0.105	0.112	0.105	0.100	0.102	
0.097	0.094	0.100	0.095	0.100	0.106	0.106	0.099	0.104	0.103	
0.099	0.095	0.103	0.098	0.102	0.107	0.109	0.102	0.105	0.107	
0.099	0.097	0.104	0.103	0.101	0.103	0.111	0.104	0.107	0.108	
0.106	0.101	0.100	0.101	0.100	0.104	0.103	0.105	0.103	0.103	
0.101	0.099	0.107	0.101	0.107	0.108	0.110	0.098	0.121	0.126	
0.100	0.099	0.107	0.102	0.107	0.109	0.111	0.099	0.121	0.121	
Valores a la salida de la Red para la consulta										
Pat.1	Pat.2	Pat.3	Pat.4	Pat.5	Pat.6	Pat.7	Pat.8	Pat.9	Pat.10	
Datos de interés										
Paso n°	2821	PI=0.050	error=90.068	Encocas=15	Ocultas=30	Salidas=10				
<input type="button" value="F5 Cambiar PF"/> <input type="button" value="F1 Computar"/> <input type="button" value="F2 Salvar"/> <input type="button" value="F3 Cargar"/> <input type="button" value="ESC Pausa"/> <input type="button" value="F8 Salir"/>										

Figura 5: Inicio del proceso de aprendizaje.

Los distintos parámetros de la red se establecen mediante un fichero de configuración. En él se indican las dimensiones de la red (número de neuronas de entrada, ocultas y de salida), el valor inicial del factor de aprendizaje, el nombre del fichero que contiene los patrones de entrenamiento y el fichero en el que se almacenan los valores de las salidas deseadas para cada uno de esos patrones. Estos ficheros se pueden modificar con cualquier editor de texto.

En cada iteración del proceso de aprendizaje, pueden visualizarse los valores de las salidas de la red para los distintos patrones de entrenamiento (figura 5). Esto permite observar la evolución del error cometido con respecto a las salidas deseadas de cada uno de dichos patrones. Además, es posible variar el valor del factor de aprendizaje en cualquier instante del entrenamiento, el cual influye directamente en la velocidad del proceso.

La herramienta permite consultar cualquier patrón cuyo número de características coincida con el de los patrones utilizados en el entrenamiento. En la figura 6 se muestra el proceso de aprendizaje para una red entrenada con patrones del siguiente tipo:



En este caso, las características de los patrones se obtienen de cada figura pixel a pixel, de tal forma que se considera un valor 0 si el pixel es blanco y un 1 si es gris.

Red Neuronal Tipo BackPropagacion										
Valores a la salida de la Red para cada Patrón										
Pat.1	Pat.2	Pat.3	Pat.4	Pat.5	Pat.6	Pat.7	Pat.8	Pat.9	Pat.10	
0.991	0.007	0.000	0.000	0.000	0.000	0.005	0.000	0.005	0.000	
0.006	0.992	0.001	0.004	0.004	0.000	0.000	0.000	0.001	0.004	0.000
0.000	0.000	0.992	0.004	0.006	0.002	0.004	0.000	0.000	0.000	0.001
0.003	0.004	0.003	0.994	0.000	0.001	0.003	0.000	0.000	0.000	0.003
0.001	0.002	0.006	0.000	0.991	0.003	0.001	0.002	0.005	0.000	
0.004	0.000	0.004	0.000	0.003	0.994	0.000	0.000	0.000	0.000	0.003
0.000	0.000	0.006	0.003	0.001	0.000	0.993	0.004	0.001	0.000	
0.007	0.000	0.000	0.000	0.000	0.000	0.005	0.992	0.001	0.000	
0.000	0.002	0.000	0.000	0.005	0.000	0.001	0.003	0.992	0.007	
0.000	0.000	0.000	0.004	0.000	0.004	0.000	0.000	0.007	0.992	
Valores a la salida de la Red para la consulta										
Pat.1	Pat.2	Pat.3	Pat.4	Pat.5	Pat.6	Pat.7	Pat.8	Pat.9	Pat.10	
0.000	0.023	0.946	0.018	0.000	0.000	0.002	0.000	0.000	0.000	
0.004	2.064	95.524	1.864	0.034	0.034	0.164	0.004	0.004	0.014	
Datos de interés										
Patrón nº	125993	PI=0.500	error=0.076	Entrada=15	Ocultas=30	Salidas=10				
/r/,FS Cambiar P2 P1 Consultar P2 Salvar P3 Cerrar ESP Pausa ESC Salir										

Figura 6: Aprendizaje y consulta de patrones.

Cuando se reconoce un patrón, la herramienta muestra, además de los valores de las salidas para ese patrón, el tanto por ciento de semejanza con los patrones empleados en el aprendizaje. Ese tanto por ciento nos permite identificar la clase a la que, con mayor probabilidad, pertenece el patrón consultado. La figura 6 refleja el resultado de reconocer el patrón que se muestra a continuación para la red anteriormente mencionada:



Como puede apreciarse en la figura 6, el resultado del reconocimiento indica que el patrón consultado pertenece a la clase 3, en la cual están incluidos los caracteres semejantes a la C.

Siempre que se quiera, es posible guardar en un fichero los valores de los pesos obtenidos hasta el momento en el proceso de aprendizaje. De esta forma, la red podrá ser utilizada posteriormente para continuar un entrenamiento que fue previamente interrumpido o simplemente para realizar consultas.

## 5 Conclusiones

En este trabajo hemos presentado una visión general de la asignatura de Inteligencia Artificial, que se imparte en el cuarto curso de los estudios de Ingeniería Informática en la Universidad de Extremadura.

Se ha realizado un análisis del temario y de las prácticas que se llevan a cabo en esta asignatura. Nos hemos centrado en las herramientas de apoyo a la docencia que se han desarrollado en la propia Escuela como parte de los proyectos fin de carrera que realizan los alumnos. Actualmente se están realizando algunos proyectos más, entre los que cabe destacar un simulador de agentes inteligentes, que permitirá programar las heurísticas de juego de varios agentes para jugar un partido de fútbol.

## Referencias

- [1] *Actas de las III Jornadas de Enseñanza Universitaria de Informática, Jenui'97*. Facultad de Informática de la Universidad Politécnica de Madrid, 1997.
- [2] Asunción Gómez, César Montes. *Enseñanza de Inteligencia Artificial e Ingeniería del Conocimiento*. Revista Iberoamericana de Inteligencia Artificial de la AEPIA. Nº3, otoño 97, páginas 2-9.
- [3] Elaine Rich, Kevin Knight. *Inteligencia Artificial. Segunda Edición*. McGraw-Hill, 1994.
- [4] Patrik Henry Winston. *Inteligencia Artificial. Tercera Edición*. Addison-Wesley, 1994.
- [5] Stuart Russell, Peter Norvig. *Inteligencia Artificial: un enfoque moderno*. Prentice-Hall Iberoamericana, 1996.
- [6] J. Mira, A.E. Delgado, J.G. Boticario, F.J. Díez. *Aspectos básicos de la Inteligencia Artificial*. Sanz y Torres, 1995.
- [7] Nils J. Nilsson. *Principios de Inteligencia Artificial*. Díaz de Santos, 1987.
- [8] J. Orenge, J.P. Sánchez. *Prolog: Introducción a la Programación de Sistemas Expertos*. RaMa, 1987.
- [9] Harald Wertz. *LISP: Introducción a la Programación*. Masson, SA, 1986.
- [10] P.H. Winston. *LISP (Third edition)*. Addison-Wesley, 1991.
- [11] Luis A. Álvarez, Juan M. Murillo, Jorge Quirós, Fernando Sánchez. *Introducción al lenguaje C/C++*. Editado por los autores, 1998.

- [12] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. *Design Patterns. Elements of Reusable Object Oriented Software*. Addison-Wesley, 1996.
- [13] Distintas versiones del intérprete SWI-Prolog, desarrollado por Jan Wielemaker en la Universidad de Amsterdam (Holanda) pueden encontrarse en la dirección:  
<http://www.swi.psy.uva.nl/usr/jan/SWI-Prolog.html>
- [14] La versión 3.0.1 de Allegro Common Lisp está disponible en la dirección:  
<http://www.franz.com/dload/offer.main.html>



# Una propuesta docente para la asignatura Heurísticas en el marco de la Inteligencia Artificial

J. Marcos Moreno-Vega, José A. Moreno Pérez

Dpto. de Estadística, Investigación Operativa y Computación  
Centro Superior de Informática  
Universidad de La Laguna  
e-mail: jmmoreno@ull.es

## Resumen

*El plan de estudios de la Ingeniería Técnica en Informática de Gestión de la Universidad de la Laguna contempla la asignatura optativa Heurísticas. A pesar de que actualmente esta asignatura está adscrita al área de Estadística e Investigación Operativa, es intención del Departamento de Estadística, Investigación Operativa y Computación asignarla al área de Ciencias de la Computación e Inteligencia Artificial en la revisión de los planes de estudio que se está llevando a cabo. Además, se pretenden modificar los descriptores para dar cabida a otras técnicas heurísticas no contempladas o que puedan surgir en el futuro. Todo ello adaptando, tanto los objetivos, contenidos y metodología, como la coordinación entre asignaturas, desde el área de Ciencias de la Computación e Inteligencia Artificial. Presentamos aquí una propuesta docente en la que se describen todos los elementos anteriores.*

## 1 Introducción

La palabra *Heurística* proviene del griego *Heuriskein* que puede traducirse por encontrar, descubrir, hallar. Así, la definición que de heurística da la Enciclopedia Salvat [2] es

*arte de inventar o descubrir hechos valiéndose de hipótesis o principios que, aun no siendo verdaderos, estimulan la investigación.*

No es de extrañar, por tanto, que las diferentes nociones que se tienen de este término involucren esta idea. Por ejemplo, en Inteligencia

Artificial se suele interpretar que *heurístico* es el calificativo apropiado para las técnicas o procedimientos que, empleando conocimiento acerca de un problema, tratan de buscarle solución usando una cantidad de recursos razonable. Por otra parte, en los problemas de optimización de Investigación Operativa, el término heurístico se aplica, en contraposición a exacto, a un procedimiento de solución que no puede garantizar la optimalidad o factibilidad de la solución que aporta. Una de las definiciones más usuales en Investigación Operativa establece que *una heurística es una técnica que encuentra soluciones buenas (casi óptimas) con un coste computacional razonable, pero que no es capaz, en muchos casos, de garantizar ni la factibilidad, ni la optimalidad, ni establecer lo cerca de la optimalidad que está una solución factible particular* [3].

A pesar de las diferencias que pueden existir, no cabe duda de la importancia que tiene lo heurístico en ambos campos. Así se refleja, por ejemplo, en muchas definiciones alternativas de la Inteligencia Artificial como la que aparece en el texto de Rich y Knight [4] según la cual la Inteligencia Artificial es *el estudio de técnicas de resolución de problemas exponenciales complicados en tiempo polinomial mediante el uso de conocimiento sobre el campo de aplicación del problema*. En el mismo sentido, cabe destacar el interés creciente por el estudio y aplicación de procedimientos heurísticos en Investigación Operativa, pasando de ser considerados pobres herramientas a instrumentos fundamentales y, en muchos casos, imprescindibles para la resolución práctica de un problema.