

# PREGUNTAS/RESPUESTAS INTELIGENCIA ARTIFICIAL

## PRIMER PARCIAL ORDINARIA

### 8 DE MAYO DEL 2019

#### 1.- El concepto de Agente. Agentes Racionales vs. Agentes Inteligentes. Arquitecturas de Agentes.

El concepto de agente no está aceptada plenamente por la comunidad pero una de las más utilizadas es la siguiente: “Un agente es una entidad que percibe y actúa sobre su entorno.”, es decir, tiene entradas a través de sensores, y luego después de procesar los datos internamente da un comportamiento.

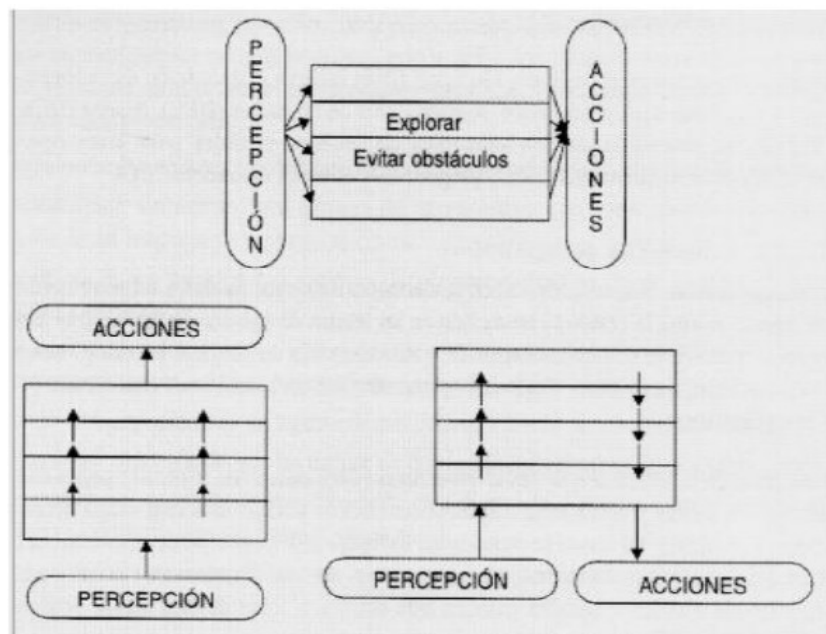
Un agente racional es un agente que responde de forma correcta a su entorno y elige la acción más óptima/favorable entre todas las que tiene en su abanico para conseguir su objetivo. El agente inteligente es lo mismo que el agente racional pero tiene una peculiaridad, es que tiene que haber conseguido pasar el test de Turing.

¿Qué es el test de Turing? Es una prueba de habilidad de una máquina para conocer si tiene comportamiento inteligente similar al de un ser humano. En pocas palabras, no saber diferenciar si estas hablando con una máquina o con una persona.

Podemos dividir la arquitectura de agentes en dos tipos: por su topología, y nivel de abstracción.

- **Topología:**

- **Arquitectura Horizontal:** En esta arquitectura todas las capas tienen acceso a los sensores y a los actuadores. En efecto, cada capa actúa por si misma como si fuese un agente, produciendo sugerencias de que acción realizar. Gracias a esto aseguramos la consistencia pero obtenemos cuello de botella.
- **Arquitectura Vertical:** Solo una capa tiene acceso a los sensores y actuadores, a través de paso de mensajes se va enviando a cada capa así hasta llegar a conseguir la acción. El problema de esta arquitectura es que no es tolerante a fallos.
- **Arquitectura Híbrida:** Une las dos arquitecturas anteriores.



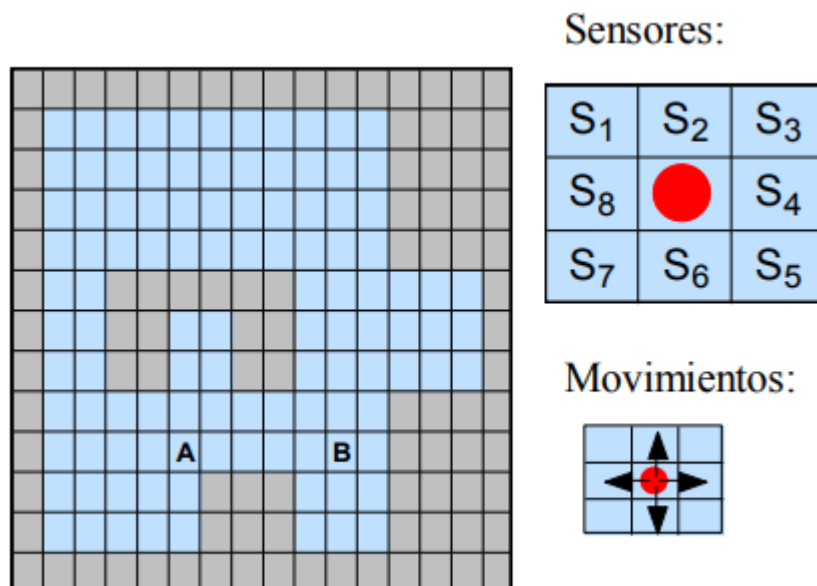
- **Abstracción:**

- **Arquitectura Deliberativa:** Contiene un mundo representado explícitamente y un modelo lógico del mismo, y en la cual las decisiones son hechas por medio de un razonamiento lógico.
- **Arquitectura Reactiva:** Se caracteriza porque opera rápidamente y efectivamente sin la necesidad de procesar una representación simbólica del entorno. La decisión que toman es basada en el presente, sin hacer uso del pasado. Su actuación es estímulo-respuesta.
- **Arquitectura Híbrida:** Une las dos arquitecturas anteriores.

## 2.- Características de los Agentes reactivos y deliberativos. Similitudes y diferencias. Arquitecturas.

**Agente Reactivo:** Este tipo de agente percibe su entorno a través de sensores, procesa la información percibida y hace o no, una representación interna de la misma. Después de este proceso escoge una acción entre todas las posibles teniendo en cuenta la información obtenida anteriormente. Transforma finalmente la acción tomada en señales para que los actuadores la hagan.

Este agente puede tener varias técnicas como: agentes en una tabla, sistema de producción, redes neuronales, etc.



Una de sus características más importantes es que se diseñan completamente y hay que anticipar todas las posibles reacciones para todas las situaciones. Realizan pocos cálculos y almacenan todo en memoria.

**Agente Deliberativo:** Este tipo de agente dispone de un modelo del mundo en donde se encuentra, disponible un modelo de los efectos de sus acciones sobre el mundo y éste es capaz de razonar sobre esos modelos para decidir que hacer para alcanzar sus objetivos.

Los agentes reactivos no incluyen ninguna representación interna del entorno, en cambio el deliberativo sí (aunque no siempre, hay a veces que un agente deliberativo no guarda la representación de forma interna). Se podría decir que un agente deliberativo es un agente que conoce todo, y el agente reactivo no, según vaya encontrándose va conociendo su entorno.

Los agentes reactivos suelen tener arquitectura horizontal, es decir, cada acción posible tiene acceso a la percepción que tiene el agente al entorno. En cambio los agentes deliberativos suelen tener arquitecturas verticales.

### 3.- Describir brevemente los métodos de búsqueda no informada.

Tenemos cuatro búsquedas sin información: anchura, costo uniforme, profundidad y bidireccional.

- **Anchura:** Desde el estado inicial se deben analizar todos los nodos hijos antes de pasar al nivel siguiente en el árbol de búsqueda. ¿Qué quiere decir esto? Si el nodo padre tiene 3 hijos, primero debemos analizar estos 3 hijos antes de bajar a alguno de ellos, si después estos 3 nodos hijos tienen a su vez 5 hijos más tendremos que verlos uno a uno.

Este algoritmo siempre encuentra solución en caso de haberla pero tarda muchísimo ya que debemos explorar todos los nodos.

- **Costo Uniforme:** En este método de búsqueda suponemos que expandir un nodo tiene un costo. La 'heurística' que toma para seleccionar el nodo a explorar es el que tenga menor costo teniendo entre los que tenemos sin explorar todavía.

Este algoritmo es óptimo y rápido comparado con el de anchura, aunque si todos los costos son iguales estamos en el mismo problema.

- **Profundidad:** Desde el estado inicial debemos ir analizando todos los nodos hijos del nodo elegido. Es decir, si el nodo padre tiene 2 hijos, primero analizaremos el de mayor nivel, y así recursivamente, en caso de no encontrar solución por ese nodo, tomaremos el otro hijo y haremos el mismo proceso.

El algoritmo encuentra solución pero no se garantiza que esta sea óptima ya que la solución encontrada será la primera explorada. Podría ser que por otro hijo se llegará antes.

Este método de búsqueda tiene dos variantes más: búsqueda con retroceso (backtraking) y búsqueda con profundización iterativa.

- **Backtraking:** Es una técnica para resolver problemas de manera recursiva para construir una solución de forma incremental. La idea es encontrar la mejor combinación posible en un momento determinado. Si se encuentra alguna alternativa incorrecta durante la búsqueda ésta retrocede hasta el paso anterior y toma la siguiente alternativa.
- **Profundización Iterativa:** En este método de profundidad lo que hacemos es ir avanzando por el árbol pero explorando nodo hijo a nodo hijo, es decir, si nuestro padre tiene dos hijos, verá primero uno y luego otro, si estos dos hijos tienen dos hijos más, veremos primero los dos hijos de uno y después los otros dos, y así todo el rato...

- **Bidireccional:** En esta búsqueda se ejecutan de forma simultánea dos búsquedas. Una empieza desde el estado inicial, y otra desde el objetivo. Se dice que se ha encontrado solución cuando las dos búsquedas se encuentran entre sí.

El problema de esta búsqueda es que necesitamos saber en todo momento donde se encuentra el objetivo. Y a veces esto no es así, sabemos que está ahí pero no dónde exactamente.

#### 4.- El concepto de heurística. Como se construyen las heurísticas. Uso de las heurísticas en IA.

Una heurística es un criterio, método o principio para decidir cual, entre una serie de alternativas acciones, promete ser la más efectiva a la hora de lograr alguna meta. Es importante que la heurística elegida funcione bien ya que una mala decisión puede causar un gran problema para el funcionamiento del algoritmo.

La construcción de funciones heurísticas normalmente son a través de modelos simplificados. Es difícil encontrar una buena nada más empezar, ya que la iremos descubriendo poco a poco.

La Inteligencia Artificial se basa mayormente en el uso de algoritmos de búsqueda los cuales usan heurísticas para que funcionen de forma correcta. Aunque es cierto que no garantizan la solución óptima siempre produce resultados que satisfacen el problema. Incluso en ciertas ocasiones la heurística puede dar soluciones óptimas.

Las heurísticas se utilizan en Inteligencia Artificial para guiar las búsquedas ya que esto es esencial en todo método en Inteligencia Artificial. Por ejemplo en algoritmos greedy se utiliza el criterio de selección para saber por donde ir para encontrar una solución, este criterio también es una heurística.

#### 5.- Los métodos de escalada. Caracterización general. Variantes.

Los métodos de escalada son técnicas de búsqueda con información, en donde buscan en un entorno local del nodo en curso. Tratan de elegir en cada paso un estado cuyo valor heurístico sea mayor que el del estado activo en ese momento.

Tenemos diferentes métodos de escalada: escalada simple, escalada por la máxima pendiente.

- **Escalada Simple:** En este método, en el momento que se encuentra un nodo más favorable que el se está expandiendo, dicho nodo es devuelto sin generar el resto de nodos hijos. Este nodo más favorable pasa a ser el nodo en curso y sigue el proceso con normalidad.
- **Escalada por la máxima pendiente:** En este caso, se generan todos los hijos de un nodo, y se calculan sus valores heurísticos. Se determina uno de los nodos de mejor valor, y se compara dicho valor con el del nodo explorado. Si es mejor, éste pasa a ser el nodo en curso y sigue el proceso.

No es un algoritmo completo, es rápido y útil si la función es monótona.

El problema de los métodos anteriores es la posibilidad de encontrarse con máximos/mínimos locales, etc.

Y además variantes del método de escalada como: estocástica, primera opción y reinicio aleatorio.

- **Estocástica:** Escoge aleatoriamente entre los sucesores con mejor valoración que el estado actual.
- **Primera opción:** Se generan aleatoriamente sucesores, escogiendo el primero con mejor valoración que el estado actual.
- **Reinicio aleatorio:** Se repite varias veces la búsqueda partiendo cada vez desde un nodo inicial distinto, generado aleatoriamente. El mejor nodo inicial es guardado y si posteriormente hay un nodo mejor que el guardado, lo reemplaza.

## 6.- Características esenciales de los métodos “primero el mejor”.

Este método, llamado “Primero el Mejor” es una de las técnicas de búsqueda con información. Avanzan a través del mejor nodo encontrado hasta el momento.

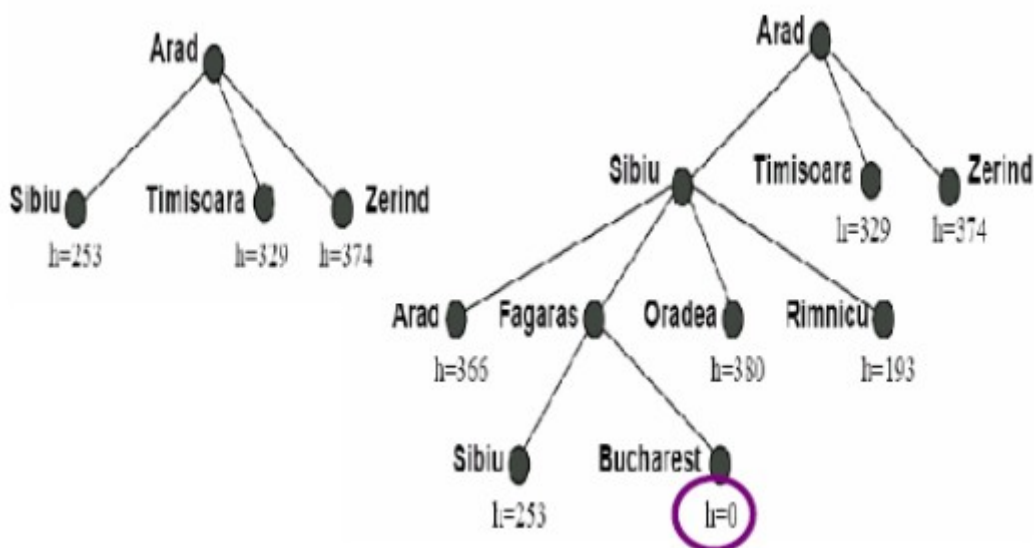
Son algoritmos basados en las variantes del algoritmo de Dijkstra, en vez de generar todos los mejores caminos nos quedamos con el primer nodo encontrado hasta ese momento.

Utilizamos dos listas, una de nodos abiertos y otra de nodos cerrados.

La lista de abiertos son los nodos que se han generado y a los que se le ha aplicado la función heurística pero que aún no han sido explorados, hay que destacar que esta lista (la de abiertos) se ordena según el valor de la heurística.

La lista de cerrados son aquellos nodos que han sido ya explorados. Es necesaria para que no exploremos un nodo que hemos explorado con anterioridad (se encontrará en esta lista).

La búsqueda primero el mejor se parece a la búsqueda primero en profundidad en el modo que prefiere seguir un camino hacia el objetivo pero volverá atrás cuando llegue a un callejón sin salida.



## 7.- Elementos esenciales del algoritmo A\*.

Se trata de un algoritmo “mejor primero” donde  $f(n)$  es la suma de dos componentes:

- Una medirá la distancia actual desde el nodo origen hasta el nodo a etiquetar.
- Otra expresará la distancia estimada desde este nodo a etiquetar hasta el nodo destino.

En sí:  $f(n) = g(n) + h(n)$ .

Donde  $g(n)$  es la distancia del mejor camino hasta el momento desde el nodo inicial I al n. Y  $h(n)$  expresa la distancia estimada desde el nodo n hasta el nodo objetivo O.

Necesitaremos para la A\* una lista de nodos abiertos y cerrados. Donde los cerrados son aquellos nodos que ya se han examinado, esto es casi obligatorio para evitar abrirlos de nuevo. Y la lista de abiertos son los nodos que se han generado pero que aún no han sido explorados.

El algoritmo A\* selecciona el nodo más prometedor que se haya generado hasta ese momento (se encuentre en la lista de abiertos). A continuación explora el nodo elegido generando así a todos sus nodos hijos. Si alguno de ellos es meta (osea, el objetivo), el proceso acaba. De no ser así, el algoritmo continúa.

En sí el A\* es una búsqueda en anchura, si en una rama por la que está explorando no aparece la solución podemos abandonar la rama y explorar la nueva. Sin embargo la vieja rama no se olvida, su último nodo se almacena en el conjunto de nodos generados sin explorar.

Uno de los problemas del A\* es que necesita mucho espacio para que sea ejecutado, y el rendimiento de los algoritmos depende de la calidad de la función heurística.

## 8.- Elementos esenciales de un algoritmo genético.

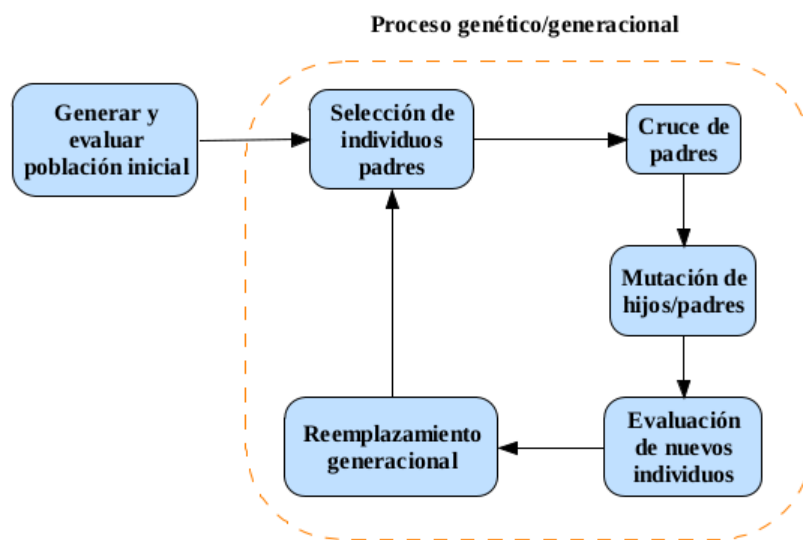
Los algoritmos genéticos son métodos de optimización basados en una simulación parcial de los mecanismos de la evolución natural. Estos algoritmos junto a los evolutivos constituyen una parte importante de la computación evolutiva, una área de la inteligencia artificial. Desde un punto de vista general este algoritmo tiene como objetivo encontrar una solución óptima para una cierta función objetivo.

- **Cromosoma:** Vector representación de una solución al problema.
- **Gen:** Característica concreto del valor de representación de una solución.
- **Población:** Conjunto de soluciones al problema.
- **Adecuación al entorno:** Valor de función objetivo.
- **Selección natural:** Operador de selección.
- **Reproducción sexual:** Operador de cruce.
- **Mutación:** Operador de mutación.
- **Cambio generacional:** Operador de reemplazamiento.

Las características esenciales de estos algoritmos son:

- Algoritmos de tipo “generar y probar”.
  - Algoritmos estocásticos basados en poblaciones.
  - Operadores de variación (cruce y mutación) crean la diversidad necesaria.
  - La selección reduce la diversidad (hacia soluciones de calidad).
- 
- Inicialización (generación aleatoria de una población inicial).
  - Variación (operadores de cruce y mutación).
  - Evaluación (aptitud de cada individuo).
  - Selección (selección probabilística).

El proceso del algoritmo genético es el siguiente:



En primer lugar se genera y se evalúa la población inicial, hay muchas formas de generar esta pero la más normal es hacerlo de forma aleatoria.

La selección de individuos padres también se hace de forma aleatoria, de igual forma las parejas. Cada individuo de la población tendrá una pareja con quién cruzarse.

En el cruce de los padres nacen hijos, vamos a poner el ejemplo que cada cruce tiene un hijo. Y el hijo en este caso va a heredar los tres primeros bits de la madre y los 2 primeros bits del padre.

¿En qué consiste la mutación? La mutación de los individuos es alterar los 0 o 1 de la tabla del individuo de forma aleatoria. Si a lo mejor en primer momento tenemos un individuo con: {0,1,1,0,0} puede ser que mute y se quede así {1,0,1,0,0}. De esta forma este individuo será distinto y tendrá otro comportamiento.

En el siguiente paso hay que evaluar los individuos tales como han quedado después del cruce/mutación. Con esto podremos ver que individuos son mejores que antes.

El último paso es volver a realizar la selección de individuos tomando como población inicial la que hemos conseguido en el último paso.