

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

Arquitecturas de Agentes.

Un **agente inteligente**, es una entidad capaz de percibir su entorno, procesar tales percepciones y responder o actuar en su entorno de manera racional, es decir, de manera correcta y tendiendo a maximizar un resultado esperado. Es capaz de percibir su medioambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores (elementos que reaccionan a un estímulo realizando una acción). En este contexto la racionalidad es la característica que posee una elección de ser correcta, más específicamente, de tender a maximizar un resultado esperado. Un agente racional es por tanto aquel que actúa con la intención de obtener el mejor resultado posible.

Arquitecturas de agentes:

Por su topología:

- Horizontal

Recorre su lista de acciones posibles por capas hasta llegar a la última y entonces ejecutar la acción correspondiente.

- Vertical

Pueden ser de una pasada que entonces actuarían como las arquitecturas horizontales o de dos pasadas, éstas recorren su lista de posibles acciones por capas hasta llegar a la última y después el flujo baja hasta la capa inferior de nuevo.

- Híbrida

Por su nivel de abstracción:

- Deliberativa

Son aquellas arquitecturas que utilizan modelos de representación simbólica del conocimiento. Suelen estar basadas en la teoría clásica de planificación. Estos agentes parten de un estado inicial y son capaces de generar planes para alcanzar sus objetivos (Maes, 1989). En estos sistemas parece aceptada la idea de que es necesario dotar a los agentes de un sistema de planificación que se encargue de determinar qué pasos se deben llevar a cabo para alcanzar sus objetivos. Por tanto, un agente deliberativo (o con una arquitectura deliberativa) es aquel que contiene un modelo simbólico del mundo, explícitamente representado, en donde las decisiones se toman utilizando mecanismos de razonamiento lógico basados en la correspondencia de patrones y la manipulación simbólica, con el propósito de alcanzar los objetivos del agente.

Cuando se decide implementar una arquitectura deliberativa hay que buscar, en primer lugar, una descripción simbólica adecuada del problema, e integrarla en el agente, para que este pueda razonar y llevar a cabo las tareas encomendadas en el tiempo preestablecido. Aunque parece una cuestión trivial, debido a la complejidad de los algoritmos de manipulación simbólica, es un aspecto al que hay que prestar mucha atención, especialmente si se tiene en cuenta que los agentes se desenvuelven en dominios reales, en los que frecuentemente tienen que responder a los estímulos en tiempo real.

- Reactiva

Actúa en función al entorno que perciben a través de sus sensores, a lo denominado estímulo-respuesta. Operan rápido y eficazmente sin la necesidad de una representación interna del entorno ni de hacer uso de decisiones pasadas. Aunque también existen los agentes reactivos con memoria, que estos sí que almacenan acciones tomadas en el pasado y pueden llegar a llevar una pequeña representación del mundo en el que se mueven para poder así tomar una decisión más correcta la próxima vez que se vaya a tomar una.

- Híbrida

Como hemos visto, tanto las arquitecturas reactivas como las deliberativas presentan ciertas limitaciones.

Por ello, se han propuesto sistemas híbridos que pretenden combinar aspectos de ambos modelos.

Una primera propuesta puede ser construir un agente compuesto de dos subsistemas: uno deliberativo, que utilice un modelo simbólico y que genere planes en el sentido expuesto anteriormente, y otro reactivo centrado en reaccionar a los eventos que tengan lugar en el entorno y que no requiera un mecanismo de razonamiento complejo.

Por su propia naturaleza estas arquitecturas son propicias para una estructuración por capas, que puede ser:

- Vertical: sólo una capa tiene acceso a los sensores y actuadores.
- Horizontal: todas las capas tienen acceso a los sensores y a los actuadores

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

[Explicar agente reactivo y deliberativo]

Similitudes:

- Tienen como objetivo alcanzar una meta.
- Tienen que partir con un conjunto de reglas o acciones con las que interactuar con el entorno.

Diferencias:

- El agente deliberativo necesita tener un modelo simbólico del entorno, mientras que el reactivo no.
- El agente reactivo basa sus acciones en lo que percibe en ese momento del entorno, mientras que el deliberativo procesa su acción con más información y conocimiento.
- El agente reactivo suele realizar sus acciones más rápidas que el agente deliberativo.

3.- Métodos de búsqueda no informada.

Los algoritmos de búsqueda no informada no dependen de información propia del problema a la hora de resolverlo, sino que proporcionan métodos generales para recorrer los árboles de búsqueda asociados a la representación del problema, por lo que se pueden aplicar en cualquier circunstancia.

Hay 6 tipos:

Búsqueda en profundidad: utilizado para recorrer todos los nodos de un grafo o árbol de manera ordenada, pero no uniforme. Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando, de forma recurrente, en un camino concreto. Cuando ya no quedan más nodos que visitar en dicho camino, regresa de modo que repite el mismo proceso con cada uno de los hermanos del nodo ya procesado.

Búsqueda en anchura: utilizado para recorrer todos los nodos de un grafo o árbol. Intuitivamente, se comienza en la raíz y se exploran todos los vecinos de este nodo. A continuación, para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el árbol.

Búsqueda en profundidad limitada: es una variante de la búsqueda en profundidad, donde se establece un límite en la profundidad.

Búsqueda en profundidad iterativa: es una variante de la búsqueda en profundidad limitada, donde la cota de profundidad elegida va aumentando si no encuentra una solución. También existe esta variante para la búsqueda en anchura iterada y limitada.

Búsqueda en coste uniforme: es una variante del método “Primero el Mejor”, utilizado para recorrer sobre grafos el camino de costo mínimo entre un nodo raíz y un nodo destino. La búsqueda comienza por el nodo raíz y continúa visitando el siguiente nodo que tiene menor costo total desde la raíz. Los nodos son visitados de esta manera hasta que el nodo destino es alcanzado.

Búsqueda bidireccional: Es un método en el cual se realiza simultáneamente una búsqueda desde el estado inicial y otra búsqueda desde el estado objetivo hasta que éstas se encuentren en el estado solución.

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

Una heurística es un algoritmo que trata la resolución de problemas aplicando soluciones parciales, a menudo intuitivas. Se evalúan los resultados intermedios obtenidos para aproximarse poco a poco al resultado o solución final.

Una heurística se caracteriza por carecer o bien de buenos tiempos de ejecución, o bien de no proveer buenas soluciones; por ejemplo, normalmente encuentran buenas soluciones, aunque no hay pruebas de que la solución no pueda ser arbitrariamente errónea en algunos casos; o se ejecuta razonablemente rápido, aunque no existe tampoco prueba de que siempre será así. Las heurísticas generalmente son usadas cuando no existe una solución óptima bajo las restricciones dadas del problema.

Las heurísticas agrupan criterios, métodos, o principios para decidir cuál puede ser la solución más efectiva hacia un objetivo, por tanto, se caracterizan en por tener criterios simples y poder diferenciar buenas de malas decisiones.

Una heurística es **admisible** si nunca sobrepasa el costo de alcanzar el objetivo, es decir, en el punto actual del problema, la estimación del costo de alcanzar el objetivo nunca es mayor que el menor de los costos reales posibles.

Las heurísticas admisibles se construyen a partir de una versión simplificada del problema, o por información desde un patrón de base de datos que almacene la solución exacta de un problema.

Si se tiene de una heurística admisible, la que mejor funcione va a ser la de mayor valor sin sobrepasar el valor real.

De ahí que si tenemos varias heurísticas admisibles y no sabemos cuál elegir, podemos crear una nueva que sea el máximo de todas las otras.

Las heurísticas se utilizan por ejemplo en simplificación de problemas, una manera de conseguir una solución al problema principal es proponer un problema más sencillo cuya solución es igual al problema principal. Dicha heurística puede no ser capaz de encontrar todas las soluciones del problema principal, pero puede encontrar una más rápido porque el problema simplificado es mucho más fácil de resolver.

También en problemas de búsqueda para eliminar los peores caminos cuanto antes, problemas de juegos, optimización, etc.

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

Los métodos de escalada son un tipo de búsqueda con información basados en la búsqueda en profundidad. Son métodos de búsqueda local porque parten de un nodo inicial y profundizan en la búsqueda seleccionando el mejor hijo, realizando esta misma acción para el hijo y sus descendientes sólo si resulta ser una mejor opción que el padre. Por esta razón puede no encontrar una solución ya que se puede quedar en extremos locales, siempre que la función no sea monótona.

Este método intenta optimizar una función objetivo ($f(x)$), donde x es un conjunto de valores discretos y/o continuos. En cada iteración, el método de escalada ajustará un único elemento en x y determinará si el cambio mejora el valor de la función. Cualquier cambio que mejore la función es aceptado, y el proceso continúa hasta que no pueda encontrarse un cambio que la mejore. Se dice entonces que es "óptima localmente".

Existen las siguientes variantes:

Escalada simple: En este método, en el momento en el que se encuentra un estado que es más favorable que el que se está expandiendo actualmente, dicho estado es devuelto sin generar el resto de estados hijos.

Escalada por la máxima pendiente: En este caso, se generan todos los hijos de un estado, se calculan sus valores heurísticos y se determina uno de los estados de mejor valor heurístico; se compara dicho valor con el de el estado expandido y si es mejor, se devuelve ese estado como expansión

En ambos casos, si ningún estado hijo es mejor que el estado que está siendo expandido, no se devuelve nada, lo que conllevará que la búsqueda termine sin haber encontrado un objetivo.

Escalada estocástica: Selecciona un vecino aleatorio en lugar de examinar todos los vecinos, y decide (basado en la cantidad de progreso en ese vecino) si moverse a él o examinar otro.

Descenso coordinado: realiza una búsqueda en línea a lo largo de una dirección de coordenadas a partir del punto actual en cada iteración. Pudiendo ser elegida de forma aleatoria.

Reinicio aleatorio: comienza cada vez con una condición inicial aleatoria y se va iterando sobre el conjunto de elementos guardándose la mejor condición (más óptima). Gasta más CPU explorando el espacio, que optimizando desde una condición inicial.

6.- Características esenciales de los métodos "primero el mejor".

La búsqueda voraz 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. Sufre los mismos defectos que la búsqueda primero en profundidad, no es óptima, y es incompleta (porque puede ir hacia abajo en un camino infinito y nunca volver para intentar otras posibilidades). En el peor de los casos recorrerá una rama infinita tamaño del árbol, antes de encontrar la solución que se encuentre en una rama contigua, aunque con una buena función, sin embargo, se puede intentar reducir la complejidad de este problema. La cantidad de la reducción depende del problema particular y de la calidad de la heurística.

El proceso en estos algoritmos es el siguiente. Se utiliza, normalmente un árbol en el cual se va iterando, cogiendo el hijo con mejor heurística hasta que no quedan más hijos, entonces se vuelve al

padre del nodo actual. Se escoge el hijo con mejor heurística, sin contar por el que ya se ha pasado y se repite el proceso hasta que se haya recorrido todo el árbol.

7.- Elementos esenciales del algoritmo A*.

El problema de algunos algoritmos de búsqueda en grafos informados, como puede ser el algoritmo voraz, es que se guían en exclusiva por la función heurística, la cual puede no indicar el camino de coste más bajo, o por el coste real de desplazarse de un nodo a otro (como los algoritmos de escalada), pudiéndose dar el caso de que sea necesario realizar un movimiento de coste mayor para alcanzar la solución. Es por ello bastante intuitivo el hecho de que un buen algoritmo de búsqueda informada debería tener en cuenta ambos factores, el valor heurístico de los nodos y el coste real del recorrido.

Así, el algoritmo A* utiliza una función de evaluación $f = g + h$, donde h representa el valor heurístico del nodo a evaluar desde el actual, n , hasta el final. Y g , el coste real del camino recorrido para llegar a dicho nodo, n , desde el nodo inicial. A* mantiene dos estructuras de datos auxiliares, que podemos denominar *abiertos*, implementado como una cola de prioridad (ordenada por el valor de cada nodo), y *cerrados*, donde se guarda la información de los nodos que ya han sido visitados. En cada paso del algoritmo, se expande el nodo que esté primero en abiertos, y en caso de que no sea un nodo objetivo, calcula la f de todos sus hijos, los inserta en abiertos, y pasa el nodo evaluado a cerrados.

Como todo algoritmo de búsqueda en amplitud, A* es un algoritmo completo: en caso de existir una solución, siempre dará con ella.

8.- Elementos esenciales de un algoritmo genético.

Los algoritmos genéticos son métodos sistemáticos para resolver problemas de búsqueda y optimización que aplican estrategias biológicas: selección basada en el más adaptado, reproducción sexual y mutación.

Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados. Los algoritmos genéticos (AG) funcionan entre el conjunto de soluciones de un problema llamado fenotipo, y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena, generalmente binaria, llamada cromosoma. Los símbolos que forman la cadena son llamados genes. Cuando la representación de los cromosomas se hace con cadenas de dígitos binarios se le conoce como genotipo. Los cromosomas evolucionan a través de iteraciones, llamadas generaciones. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud. Las siguientes generaciones (nuevos cromosomas), son generadas aplicando los operadores genéticos repetidamente, siendo estos los operadores de selección, cruzamiento, mutación y reemplazo.

Funcionamiento:

Se genera aleatoriamente la población inicial, que está constituida por un conjunto de cromosomas los cuales representan las posibles soluciones del problema.

A cada uno de los cromosomas de esta población se aplicará la función de aptitud para saber cómo de "buena" es la solución que se está codificando.

El algoritmo se deberá detener cuando se alcance la solución óptima, pero esta

generalmente se desconoce, por lo que se deben utilizar otros criterios de detención.

Normalmente se usan dos criterios: un número máximo de iteraciones (generaciones) o detenerlo cuando no haya cambios en la población. Mientras no se cumpla la condición de término se hace lo siguiente:

Selección: Después de saber la aptitud de cada cromosoma se procede a elegir los cromosomas que serán cruzados en la siguiente generación. Los cromosomas con mejor aptitud tienen mayor probabilidad de ser seleccionados.

Recombinación o cruzamiento: La recombinación es el principal operador genético, representa la reproducción sexual, opera sobre dos cromosomas a la vez para generar dos descendientes donde se combinan las características de ambos cromosomas padres.

Mutación: Modifica al azar parte del cromosoma de los individuos, y permite alcanzar zonas del espacio de búsqueda que no estaban cubiertas por los individuos de la población actual.

Reemplazo: Una vez aplicados los operadores genéticos, se seleccionan los mejores individuos para conformar la población de la generación siguiente.

Un algoritmo genético no garantiza la obtención del óptimo, pero, si está bien construido, proporcionará una solución razonablemente buena.