Puzle Inteligencia Artificial TP1

GUILLERMO MADOERY

UNIVERSIDAD DE LA DEFENSA NACIONAL IUA

Objetivo del sistema

El objetivo del sistema es realizar un programa que conste de un algoritmo para la resolución de un puzle de 9 piezas de las cuales 8 son móviles y una es un espacio vacío para permitir el movimiento de las piezas. El algoritmo realizado sirve para que la máquina resuelva por sí misma el puzle ingresado, tomando las decisiones que considere más apropiadas para la realización de un movimiento según el criterio de dicho algoritmo y así llegar a su objetivo, que es armar correctamente el puzle con sus piezas. Lo que la maquina debe saber con anterioridad es el puzle en su estado inicial, que es el puzle con las piezas en desorden, y el puzle en su estado final, que es el puzle armado correctamente con las piezas acomodadas en su estado finalizado.

Lo más importante a considerar en este sistema es que la maquina debe resolver el algoritmo por si misma a partir del método “enseñado” a la misma, para de esta manera lograr un principio de inteligencia artificial.

Método usado

Para la realización de este programa se utilizó el siguiente método:

- **Primero el mejor:** este algoritmo usa heurística, escogiendo así, el que menor heurística tenga, esto lo hace de la siguiente manera. Cuando explora un nodo, mete sus hijos en la cola de nodos y los acomoda por heurística de mayor a menor, luego para hacer el próximo movimiento el algoritmo elige el primer nodo que tenga menor heurística, es decir el ultimo acomodado que sería el menor, y si en este nodo no encuentra la solución, explora el siguiente con menor heurística, haciendo lo mismo con sus hijos organizándolos en la cola de nodos, hasta encontrar la solución.

  En el sistema propuesto como solución, se hizo que los nodos explorados se vayan guardando en la cola, y que cuando el algoritmo busque un nodo para explorar, este explora la lista de nodos abiertos, para así, escoger el primero que encuentre con menor heurística.

Otro método analizado sin el éxito deseado fue:

- **Máxima pendiente:** este algoritmo se guía a partir de una heurística, donde a dicha heurística la utiliza de la siguiente manera. Cuando explora un nodo evalúa la heurística de los nodos hijos y el hijo que posea una heurística menor, ese será explorado. Los demás nodos son eliminados de la lista. Para lograr que el sistema implementado realice esto, se buscó que elimine hasta uno antes de explorar, para que, de esta manera, si el nodo elegido con menor heurística no posee hijos que no se termine ahí la búsqueda con un fracaso, sino que tenga la posibilidad de explorar al menos un nodo más y encontrar la solución. El método esta sin terminar ya que no encontré la forma de hacerlo funcionar correctamente.

Características del problema

En el sistema realizado no se puede decir que se valida la teoría de forma práctica, ya que, por ejemplo, la teoría dice que el algoritmo optimo es el A* con respecto a los demás, y esto no se pudo comprobar solo realizando uno de los algoritmos (primero el mejor e incompleto el de máxima pendiente) haciendo que la maquina resuelva el puzle, lo que pude comprobar es que el algoritmo de primero el mejor es bastante optimo resolviendo el puzle que se le plantee.

Usando internet en http://kali.azc.uam.mx/clc/03_docencia/posgrado/i_artificial/tareas/PUZZLE3X3HEURISTICAS.pdf pude observar que algoritmos como el de máxima pendiente necesitan más iteraciones para finalizar y resolver todo el árbol debido a su característica de explorar todo el árbol de soluciones, principalmente en el caso de primero en profundidad que explora todos los nodos abiertos.

Desarrollo en java

Como se mencionó anteriormente el algoritmo desarrollado se realizó con una pequeña modificación en su elección del nodo a desarrollar en la siguiente iteración para simplificar

la complejidad a la hora de escribir el código y cabe mencionar también que fue por cuestiones de tiempo ya que un desarrollo ideal me hubiera insumido varias semanas para dejarlo a punto con pruebas incluidas. Con respecto al sistema se hizo un menú principal estándar de diseño ampliamente utilizado sin parte gráfica, en el que se puede ingresar tanto la matriz inicial, con la que va a comenzar a tomar decisiones la máquina, y un estado final que es el estado al que la maquina va a tratar de llegar. Dentro del menú esta la opción de elegir el algoritmo con el cual la maquina desarrollara y buscara la solución del puzle ya que si se continua el desarrollo puedo ingresar algunos algoritmos más en la secuencia de usos, es por esto que si se desea hacer de nuevo la solución del puzle con otro algoritmo (a desarrollar), lo que se debe hacer es restablecer los valores predeterminados de la matriz, porque al haber sido modificados por el algoritmo planteado el resultado será diferente. Una vez restablecidas las matrices se puede elegir otro algoritmo para el desarrollo de la solución.

La solución creada por el algoritmo de “primero el mejor” es mostrada en pantalla cuando la maquina lo resuelve, mostrando la cantidad de iteraciones realizadas, la cantidad de nodos explorados y también muestra la cola de nodos cerrados que son los explorados, es decir por donde fue moviendo la pieza el algoritmo, y la cantidad de nodos abiertos que son los nodos que el algoritmo no exploro, son los hijos de un nodo explorado y que no ha sido necesario explorarlos para llegar a la solución esperada. Dentro de la impresión de la solución también es mostrado, para cada nodo su padre y su costo de movimiento, es decir cantidad de piezas en desorden.

Resultados y limitaciones

Los resultados que pude obtener varían según como la maquina comenzó a resolver el puzle.

Alguno de las matrices utilizadas fue la siguiente, matriz inicial: 1, 2, 3, 4, 5, 8, 6, 7, 0 y estado final: 1, 2, 3, 4, 5, 6, 7, 8, 0 que el algoritmo primero el mejor resolvió más eficientemente el puzle haciendo una cantidad de 174 iteraciones y 290 nodos explorados.

Con otra matriz pude observar una gran limitación para el método desarrollado con estado inicial: 1, 2, 3, 4, 0, 6, 7, 8, 5 y estado final: 1, 2, 3, 4, 5, 6, 7, 8, 0 el algoritmo se quedó dando vueltas y realizando miles de iteraciones sin llegar a la solución esperada, realizando una inspección de las muestras de movimientos llegue a la conclusión que la maquina realiza movimientos en bucle debido a que el costo varía entre 6 y 8 y se queda dando vueltas en esos valores sin poder tomar una mejor decisión.

Conclusión

Por mi parte creo que es muy interesante ver como la maquina va tomando decisiones para resolver un problema como el puzle en este caso, eso demuestra una cierta inteligencia computacional, ya que lo resuelve más rápido que el cerebro humano. Lo que sería bueno es desarrollar la propia inteligencia artificial llegando a que la computadora pueda aprender a realizar movimientos u otro tipo de operaciones en un entorno distinto, como puede ser un puzle más grande, por ejemplo. Viendo ciertas limitaciones para algunos puzles se me ocurre la posibilidad de hacer modificaciones en el diseño de software planteado para que el algoritmo o la maquina tenga guardado los movimientos realizados para los puzles ya resueltos y cuando se les presente de nuevo “recuerde” como resolverlo, así la maquina podrá llegar a soluciones de forma más efectiva ante estas situaciones repetitivas que se le puedas plantear.