Universidad de Granada

GRADO EN INGENIERÍA INFORMÁTICA

Tema 4
Inteligencia Artificial

Búsqueda con adversario y juegos

Fernández Fernández, Sergio

GRUPO B

Profesor: González, Antonio

Granada, junio de 2018

JUEGOS BIPERSONALES CON INFORMACIÓN PERFECTA

Estas situaciones se estudian y resuelven utilizando la **Teoría de Juegos**, inventada por **John von Neumann** y **Oskar Morgenstern** en 1944. Un **juego** es cualquier situación de decisión, caracterizada por poseer una independencia estratégica, gobernada por un conjunto de reglas y con un resultado bien definido. En un juego, cada jugador intenta conseguir el mayor beneficio para sus intereses. La solución de un juego permite indicar a cada jugador qué resultado puede esperar y cómo alcanzarlo. Por ejemplo:

- El dilema de los prisioneros: tiene una estructura no cooperativa.
- El juego de los palillos: el jugador puede quitar 1, 2 o 3 palillos.

Un juego de **información perfecta** es aquel en el que los jugadores tienen a su disposición toda la información de la situación del juego.

ÁRBOLES DE EXPLORACIÓN DE JUEGOS

Agente deliberativo que tiene delante a otro jugador. Un árbol del juego es una representación explícita de todas las formas de jugar a un juego. El árbol consiste en las opciones que tiene cada jugador. Cada camino es una partida a dicho juego. Se utiliza un árbol y no un grafo ya que éste no permite volver a un estado anterior, cosa que en los juegos no es posible.Correspondencia entre árboles de juegos y árboles Y/O: en los árboles Y/O se indica la consecución de tareas para conseguir un objetivo, lo que es parecido al orden de los movimientos para llegar a una situación en un juego. Cuando no se puede controlar algo es un árbol Y/O.

NOTACIÓN MIN-MAX

En la resolución exacta de un juego, se basa en el uso de algoritmos de búsqueda sobre árboles Y/O.

- **MAX**: primer jugador.
- **MIN:** segundo jugador.
- Hay nodos **MAX** y nodos **MIN**.
- Los nodos terminales se etiquetan con V(victoria/gana), D (derrota/pierde), E(empata) desde el punto de vista de **MAX**.

ALGORITMO STATUS

Si J es un nodo MAX no terminal, entonces STATUS(J) =

- V si alguno de los sucesores de J tiene STATUS V.
- D si todos los sucesores de J tienen STATUS D.
- E en otro caso.

Si J es un nodo MIN no terminal, entonces STATUS(J) =

- V si todos los sucesores de J STATUS V.
- D si alguno de los sucesores de J tiene STATUS D.
- E en otro caso.

NUEVO MODELO DE SOLUCIÓN

Nos fijamos como juegan los seres humanos. Los juegos complejos no se pueden resolver ya que es imposible la exploración total hasta la terminación. El nuevo objetivo es encontrar una buena jugada inmediata. La heurística en el proceso es muy importante. Búsqueda parcial: ve lo que hay en la frontera. En la frontera usamos heurísticas. Propagan los valores de la heurística hacia arriba mediante métodos de propagación hacia atrás.

EL MODELO BÁSICO

- Arquitectura de percepción/planificación/actuación.
- Búsqueda con horizonte
- Uso de heurísticas.

LA REGLA MINIMAX

El valor de V(J) de un nodo J de la frontera de búsquedas es igual al de su evaluación estática; en otro caso:

- Si J es un nodo MAX, entonces su valor V(J) es igual al máximo de los valores de sus nodos sucesores.
- Si J es un nodo MIN, entonces su valor V(J) es igual al mínimo de los valores de sus nodos sucesores.

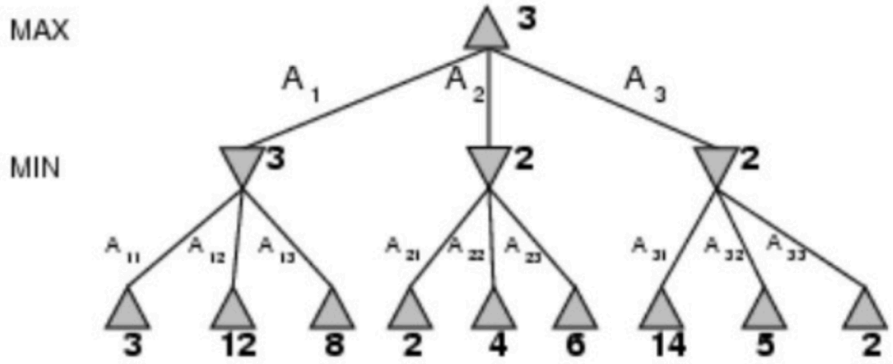
ALGORITMO MINIMAX

Método de decisión para minimizar la pérdida máxima esperada en juegos con adversario y con información perfecta. Se elige el mejor movimiento para uno mismo, suponiendo que el contrincante elegirá el peor para ti. Se utiliza búsqueda retroactiva para minimizar el espacio en memoria.

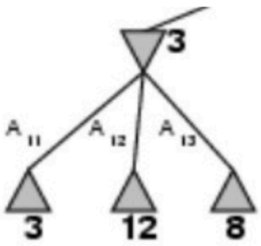
Para determinar el valor minimax, V(J) de un nodo J, hacer lo siguiente:

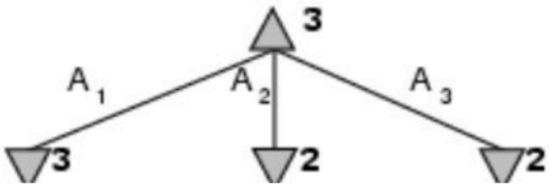
- Si J es un nodo terminal, devolver V(J) = f(J); en otro caso:
- Para k=1, 2,..., b hacer:
  - Generar J_k , el k-ésimo sucesor de J.
  - CalcularV(J_k)
  - Si k=1, hacer AV(J) ← V(J_1); en otro caso, para k>=2,
  - HacerAV(J)←max{AV(J),V(J_k)}si J es un nodo **MAX**
  - Hacer AV(J) ← min {AV(J), V(J_k)} si J es un nodo **MIN.**
- Devolver V(J) = AV(J).

Ejemplo.



Baja a los nodos hoja de cada nodo. En este ejemplo, busca el menor valor que hay en sus hijos. El 3.





En este caso, busca el máximo de los hijos, que es 3.

PODA ALFA-BETA

Técnica de búsqueda que reduce el número de nodos evaluados en un árbol de juegos para el algoritmo minimax. Con una ordenación 'primero el mejor' podríamos explorar con el mismo esfuerzo un árbol de doble profundidad.

Para calcular el valor V(J,alfa,beta), hacer lo siguiente:

1. Si J es un nodo terminal, devolver V(J)=f(J). En otro caso, sean $J_1,\ldots,J_k,\ldots,J_b$ los sucesores de J. Hacer k ←1 y, si J es un nodo MAX ir al paso 2; si J es un nodo MIN ir al paso 5.
2. Hacer alfa ← max(alfa, V(J_k,alfa,beta)).
3. Si alfa >= beta devolver beta; si no, continuar
4. Si k=b, devolver alfa; si no, hacer k ← k+1 y volver al paso 2.
5. Hacer beta ← min(beta, V(J_k,alfa,beta)).
6. Si beta <= alfa devolver alfa; si no, continuar
7. Si k=b, devolver beta; si no, hacer k ← k+1 y volver al paso 5.

Preguntas tema 4

1. **La mayor parte de los programas para juegos no almacenan los resultados de las búsquedas de un movimiento para el siguiente movimiento, ya que normalmente empiezan de cero cada vez que es el turno del ordenador, ¿por qué?**

Porque la situación sobre la cual desarrollar el árbol de posibilidades suele ser completamente distinto, ya que en el turno del adversario la situación del juego puede haber cambiado completamente y ya no resultar válido todo lo calculado anteriormente.

2. **¿Por qué la estructura básica para representar un juego es un árbol y no un grafo?**

Un árbol de juego es una representación explícita de todas las formas de jugar a un juego. En un grafo se permitiría volver a un estado anterior del juego, lo cual en la mayoría de los juegos no es posible y no dejaría evaluar rápidamente y eficientemente las posibles situaciones del juego.

3. **¿Por qué los procedimientos de búsqueda en juegos parten de la posición inicial en vez de empezar por la posición objetivo y realizar la búsqueda hacia atrás?**

Porque no se tiene el control sobre los movimientos del adversario.

4. **Relación entre árbol de juego y árbol Y/O**

En los árboles Y/O se indica el orden de consecución de tareas a realizar para alcanzar un objetivo. Esto es parecido al orden de movimientos que se deben de dar en un juego para llegar a una situación determinada.

5. **¿Por qué en los juegos se llama a la función heurística función de evaluación estática?**

Porque una heurística trata de evaluar una situación de manera rápida sin ser del todo óptima la solución en todos los casos. En cambio, una función de evaluación estática examina exhaustivamente una situación y aporta una valoración a ésta asociada a la probabilidad de perder, ganar o empatar.

6. **¿Por qué en el algoritmo minimax es conveniente utilizar estrategias de búsqueda retroactivas para la búsqueda parcial?**

Por ahorrar espacio en memoria.