

Análisis de los algoritmos utilizados para la realización de juegos por los alumnos del curso 2003/2004

Por Eduardo Salagaray Cepeda

• **Introducción**

El propósito de este pequeño trabajo es mostrar diversos ejemplos prácticos respecto al tema impartido en clase “Resolución de problemas”. En los próximos párrafos se mostraran los diferentes algoritmos utilizados por compañeros que cursaron esta asignatura el año pasado para lograr que un ordenador fuese capaz de jugar. En los casos más interesantes, se comentará la implementación elegida: lenguaje de programación, estructura de datos,....

• **El juego de las damas**

El algoritmo implementado en esta ocasión consistió en analizar todas las posibles jugadas a realizar eligiendo la mejor. Debido a la carga computacional, era imposible calcular absolutamente todas las alternativas por lo cual a la hora de realizar un nuevo movimiento solamente se procesaban todas las posibilidades en los próximos 3 turnos de cada jugador.

Una vez conocidas todas las potenciales opciones, se busca el movimiento que maximice la diferencia entre el número de piezas que quedarían en el tablero después de 3 movimientos del ordenador y 3 del usuario. La idea es generar un árbol con todos los posibles movimientos míos (del ordenador), luego para cada uno de ellos, los del usuario, luego los míos, luego del usuario (así hasta 3 veces). A continuación se evalúa en los estados finales la diferencia respecto al estado inicial usando una función de evaluación con el propósito de quedarse con el movimiento inicial del mejor camino. Para esta función de evaluación se tomó la diferencia entre el número de fichas restantes del ordenador menos el número de fichas restantes del contrario. Debido a las mayores posibilidades de movimiento y acción de la dama se le ha dado un peso a cada tipo de ficha: 1 punto por peón y 3 por dama.

$$f_{eval} = 3(damasMaquina - damasUsuario) + (peonesMaquina - peonesUsuario)$$

Finalmente, si el resultado de la búsqueda del mejor movimiento no es único se recurrió a la aleatoriedad para determinar el movimiento a realizar. El proceso consistía en asignar a cada uno de los movimientos un peso dependiendo del número de movimientos posteriores que le permitan llegar a un máximo de la función explicada anteriormente. En este momento se decidirá cual es el movimiento a ejecutar de forma aleatoria por lo que las alternativas que posean más caminos que llevan a el máximo

tendrán mayor la probabilidad de ser escogidas. Esta aleatoriedad permite que la máquina no sea tan determinista ante situaciones idénticas.

En principio, existen multitud de tipos de datos y alternativas para implementar esta solución. Una forma eficiente y manejada por nuestros compañeros del año anterior fue árboles en Java. Tomando esto como base, recorriendo todos los caminos de dichos árboles se podrá realizar todo lo mencionado anteriormente.

• Cuatro en raya

En el caso ideal, para realizar una máquina capaz de jugar a las cuatro en raya se debería calcular todos los posibles movimientos desde el momento actual hasta el final quedándose posteriormente con el mejor. Al igual que en el juego de las damas, debido a la carga computacional era imposible calcular absolutamente todas las posibles alternativas por lo cual a la hora de realizar un nuevo movimiento solamente se procesaban todas las posibilidades en los próximos 3 turnos de cada jugador.

Una vez conocidas todos los potenciales movimientos, para saber que opción es mejor utiliza el heurístico que evaluaba la situación proporcionando 10 puntos por cada trío de fichas que se encuentren en línea y 1 punto por cada pareja que se hallen en la misma disposición. Para completar la funcionalidad del heurístico, se penalizaba de la misma forma que puntúa por las situaciones semejantes que le suceden al contrario. De forma intuitiva la función a evaluar nos proporciona un factor que va de muy negativo a muy positivo, indicando la posición relativa con respecto a nuestro oponente.

$$f_{\text{eval}} = 10(\text{triosDeMáquina} - \text{triosDeUsuario}) + (\text{parejaDeMáquina} - \text{parejaDeUsuario})$$

Este es un heurístico ya que en realidad la elección del criterio fue elegido de forma intuitiva (“palabras textuales”) por lo que cabe la posibilidad que exista otra situación mejor.

La implementación de este algoritmo fue de forma muy similar al utilizado en el juego de las damas.

• BlackJack

El BlackJack es un juego practicado en los casinos ante lo cual los compañeros decidieron que el principal criterio a la hora de decidir para la máquina fuese el dinero. Basado en esto, a la hora de elegir entre si robar o no una nueva carta se calculaba el beneficio económico que obtendría en la situación actual teniendo en cuenta su puntuación, la de los demás y los jugadores que ya se han pasado de 21 puntos. Para calcular lo segundo se basaron en la probabilidad de que los oponentes tuvieran una u otra carta (excluyendo las mostradas en la mesa y las que el ordenador poseyese en la mano).

A continuación se paso a calcular un beneficio hipotético a través de multiplicar la probabilidad de obtener una carta x por el beneficio que obtendríamos con dicha carta. Sumando todos esos factores obtuvieron:

$$\text{Beneficio hipotético} = \sum (\text{probabilidad}(\text{carta } i) \cdot \text{Beneficio}(\text{carta } i))$$

Por tanto, comparando el beneficio hipotético con el beneficio actual establecían un criterio para decidir si coger una nueva carta o no.

Todo este algoritmo fue implementado en java, creando numerosas clases para cada una de los objetos que intervienen en el proceso. El hecho más significativo es la utilización de un array para todas las cartas, inicializándose al comienzo con la baraja al completo y siendo eliminadas las correspondientes según iban saliendo.

• Domino

El criterio para jugar de la maquina es encontrar una ficha para colocar en el tablero. En caso de existir varias alternativas que cumplen esta condición, se coloca aleatoriamente una.

• Siete y media

En esta ocasión, la implementación fue realmente simple. Siempre empezaba jugando el usuario. Una vez finalizado sus movimientos, pasaba a jugar el ordenador que sacaba cartas hasta que su puntuación era superior a la del jugador o hasta que se pasaba.

• El juego de la escoba

El criterio tomado para jugar es en primer lugar intentar lograr sumar 15 con mis cartas más las del tablero. Si existen varias alternativas se escogerá en el siguiente orden:

1. Se toma la opción en la que se cogen todas las cartas de la mesa.
2. Se adopta la alternativa que implique el siete de oros.
3. Se elige la alternativa que posea más sietes.
4. Se toma la alternativa que tiene más oros.
5. Se opta la alternativa que tiene más cartas.
6. Se escoge una aleatoriamente.

En caso de no poder sumar 15, se evaluará la siguiente función:

$$\text{Coste} = \sum (\text{probabilidadContrarioTenga}(\text{carta } i) \cdot \text{OpcionesParaSumar15}(\text{carta } i))$$

En este criterio influye la probabilidad de que el contrario tenga una determinada carta (se calculará excluyendo las que ya han salido y las que posee la maquina en la mano) por el numero de posibles opciones de sumar 15 con dicha carta. Este coste se repite para cada carta que pueda tirar el ordenador, eligiendo la posibilidad de menor valor.



• Bibliografía

- Apuntes de clase de los días 25 Octubre 2004 y 27 Octubre 2004 titulados “Resolución de problemas”
- [Estudio] “Resolución de problemas y juegos”, por Elena Alaña Salazar
- Trabajos de “Inteligencia de Redes de Comunicaciones” titulados:
 - **El juego de la escoba**
Rubén Grande Baquero
 - **Juego de las Damas**
Isidro García del Amo
Carlos Arias Fernández
 - **Blackjack**
Ignacio Ribas Ramos
Miguel Flecha Lozano
 - **Las cuatro en raya**
Javier Martín Manzano
Alejandro Bárcena Berzosa
 - **Dominó**
María Herrero Torres
Luis M. Corrales Ávila
 - **Diseño de un Expert-Juego con CLIPS: las siete y media**
Elena A. Álvarez Conde
- <http://www.aaai.org/AITopics/html/games.html>
- <http://aima.cs.berkeley.edu/ai.html#search>
- <http://www.gameai.com>
- <http://calisto.sip.ucm.es/curso/transparencias/19AIGameIntro.pdf>