Teoría de Juegos: TP Corte Supremo

Eric Brandwein

Metodología

Se implementó un algoritmo en la función `simular` que genera tortas aleatorias de distribución uniforme de un tamaño dado, y luego calcula el mejor corte que puede hacer el primer jugador para cada torta. Para esto llama a la función `mejor_corte`, que recorre todos los cortes posibles, y en cada corte compara las dos porciones según los gustos del segundo jugador usando la función de valoración u_2. Para elegir el mejor corte, compara las valoraciones de las porciones no elegidas por el segundo jugador usando la función de valoración u_1.

Para normalizar las funciones u_i, se divide la suma de las valoraciones para cada componente de la porción por la suma de las valoraciones en toda la torta. Esto hace que u_i represente la proporción de la valoración obtenida en relación al máximo posible sobre esa torta.

La implementación no es óptima: recorre todos los cortes posibles, y por cada corte recorre todas las componentes de la torta para calcular las u_i. Esto hace que el algoritmo tenga complejidad $\mathcal{O}(n^2)$ para cada ronda de simulación, donde n es el tamaño de la torta, y $\mathcal{O}(n^2 \cdot i)$ en total, donde i es la cantidad de iteraciones. Esto podría mejorarse, ya que para calcular las valoraciones de un corte es posible reutilizar las valoraciones del corte anterior, sumándole o restándole sólo el valor de un elemento de cada lado. Esto nos llevaría a una complejidad total de $\mathcal{O}(n \cdot i)$. De todas formas, para los parámetros configurados, la complejidad alcanzada fue suficiente.

El lenguaje utilizado fue Python3. Se eligió configurar una semilla para el generador de números aleatorios para poder reproducir los resultados.

El programa se puede correr desde la terminal pasando las tablas de gustos de los jugadores como argumentos. El mismo imprime por pantalla el mejor corte para cada torta y la valoración de cada jugador, y al final imprime la suma de las valoraciones de cada jugador.

Un ejemplo de corrida del programa es el siguiente:

```
$ ./simulacion.py "[1,2,3,4]" "[1,2,3,4]"
Torta: 3413|341132
Valoraciones: 0.44 0.56

Torta: 33334|14443
Valoraciones: 0.5 0.5
...
Torta: 24244|11443
Valoraciones: 0.448 0.552
```

```
Total: [46.62777721 53.37222279]
```

Resultados

Se corrió el algoritmo con 100 iteraciones y tortas aleatorias de tamaño 10, y distintas tablas de gustos. La siguiente tabla muestra los resultados para cada configuración.

Gustos 1	Gustos 2	Suma Valoraciones 1	Suma Valoraciones 2
[1, 1, 1, 1]	[1, 1, 1, 1]	50,0	50,0
[1, 1, 1, 1]	[2, 2, 2, 2]	50,0	50,0
[1, 2, 3, 4]	[1, 2, 3, 4]	46,628	53,372
[1, 2, 3, 5]	[1, 2, 3, 5]	46,435	53,565
[1, 2, 3, 100]	[1, 2, 3, 100]	39,497	60,503
[0, 0, 0, 1]	[0, 0, 0, 1]	34,929	46,071
[0, 0, 1, 1]	[0, 0, 0, 1]	58,446	64,488
[0, 1, 1, 1]	[0, 0, 0, 1]	63,913	65,655
[1, 1, 1, 1]	[0, 0, 0, 1]	66,7	65,405
[1, 1, 1, 0]	[0, 0, 0, 1]	77,504	68,471
[1, 1, 0, 0]	[0, 0, 0, 1]	79,761	69,238
[1, 0, 0, 0]	[0, 0, 0, 1]	79,2	68,138
[1, 0, 0, 0]	[0, 0, 1, 1]	75,405	64,517
[1, 0, 0, 0]	[0, 1, 1, 1]	73,655	64,134
[1, 0, 0, 0]	[1, 1, 1, 1]	62,455	59,2
[1, 0, 0, 0]	[1, 1, 1, 0]	58,779	58,417
[1, 0, 0, 0]	[1, 1, 0, 0]	51,879	57,105

Cuando los jugadores no tienen preferencia de gustos, la suma de las valoraciones de cada jugador es siempre 50, ya que el primer jugador corta la torta en 2 partes iguales y el segundo jugador elige cualquiera de las dos. Al darle preferencia a algunos gustos, sin embargo, las diferencias empiezan a aparecer. Vemos que si configuramos la misma lista para los dos jugadores, el segundo resulta tener mayor o igual suma. Esto se debe a que no importa qué corte haga el primer jugador, el segundo elegirá la porción que sea mejor para él, que a su vez será la que sea mejor para el primer jugador, y por lo tanto le dejará la peor para el primero. Conforme agregamos diferencias entre los gustos de los jugadores, la suma de las valoraciones del primer jugador aumenta para superar a la del segundo. Igualmente, la valoración del segundo también aumenta, pero en menor medida. Esto último se debe a que pueden separarse la torta tal que cada parte tenga los gustos que más le gustan a cada jugador.

También vemos que en el caso donde los dos jugadores tienen tablas de gustos con elementos positivos no compartidos, se maximizan las valoraciones para las configuraciones probadas. Esto se debe a que el primer jugador tiene muchas

opciones de corte que tienen la misma valoración para el segundo jugador, pero que pueden cambiar para el primero, lo cual no ocurre cuando los gustos son compartidos.