El problema de eliminación de béisbol

Mario Alberto Gutiérrez Carrales
1549273

20 de mayo de 2019

1. Introducción

El problema de máximo flujo es un problema muy relevante dentro del campo de la optimización en redes ya que es usado como herramienta para resolver otros problemas que son más elaborados, tales como el problema de emparejamiento máximo, el problema de corte mínimo, y problemas de flujo en procesos en general.

El objetivo que se busca cumplir es modelar mediante una red una instancia del problema de eliminación de béisbol como una para el problema de flujo máximo y posteriormente resolverla utilizando una implementación para el problema de máximo flujo que la librería NetworkX ofrece.

2. Ambiente computacional

Las implementaciones computacionales se realizan en `python` [1] utilizando librerías como la ya mencionada **NetworkX** [2] que sirve para la manipulación de grafos, así como algoritmos de solución para problemas de optimización en redes y **MatplotLib** [3] que es empleada para la visualización de gráficos.

Los experimentos realizados en esta práctica se realizaron en una PC con un sistema operativo de 64 bits, procesador AMD A9-9410 Radeon R5, 2 núcleos + 3G con 2.90 GHz de velocidad y 12 GB de memoria RAM.

3. El problema de eliminación de béisbol

El problema de eliminación de béisbol (que también aplica para hockey y basquetbol) se popularizo en 1960 por Alan Hoffman [4, 5, 6]

El problema de eliminación de béisbol se define de la siguiente manera:

Dada la clasificación en una división deportiva en algún momento de la temporada, determine si un equipo ha sido eliminado matemáticamente y no tiene oportunidad de ganar su división. (No se permiten empates en los encuentros)

Supóngase que se tiene la siguiente instancia del problema de eliminación de béisbol

Equipo	PG	PR	1	2	3	4
1	33	8	-	1	6	1
2	29	4	1	-	0	3
3	28	7	6	0	-	1
4	27	5	1	3	1	-

Es decir, 4 equipos se están disputando el campeonato y en la tabla se tienen los partidos que han ganado hasta el momento (PG), los partidos restantes que le restan por jugar a cada equipo (PR) y la matriz de juegos pendientes entre los equipos.

Se desea saber si el equipo 2 está eliminado o no.

Primero se puede observar que el número máximo de partidos ganados que puede conseguir el equipo 2 es $PG_n(2) = 29 + 4 = 33$ (si gana todos sus juegos).

Si gana todos sus juegos, no será eliminado solamente si:

- 1 no gana más de $C(1) = PG_n(2) - PG(1) = 33 - 33 = 0$ en sus juegos restantes.
- 3 no gana más de $C(3) = PG_n(2) - PG(3) = 33 - 28 = 5$ en sus juegos restantes.
- 4 no gana más de $C(4) = PG_n(2) - PG(4) = 33 - 27 = 6$ en sus juegos restantes.

Sea P el conjunto de equipos sin considerar el equipo 2 y Q el conjunto de todos los posibles juegos entre los equipos en P.

$P = \{1, 3, 4\}$ $Q = \{(1, 3), (1, 4), (3, 4)\}$

Para crear la red se toma en cuenta las siguientes consideraciones:

- Crear un nodo fuente O (Todos los juegos se originan de aquí).
- Crear un nodo por cada pareja en Q. Para cada nodo (i, j) creado, agregar un arco de O a (i, j), la capacidad del arco es el número de juegos que restan por jugar entre i y j.
- Crear un nodo por cada equipo en P. Para cada nodo $(i, j) \in Q$ agregar arcos a i y a j cumpliendo la relación capacidad$((i, j) \rightarrow i)$ = capacidad$((i, j) \rightarrow j)$ =capacidad $(O \rightarrow (i, j))$.
- Crear un nodo sumidero T (Las victorias de los equipos se registran aquí). Agregar arcos desde todo nodo $i \in P$ a T, la capacidad de arco es $C(i)$.

Encontrar el flujo máximo desde O a T en la red resultante.

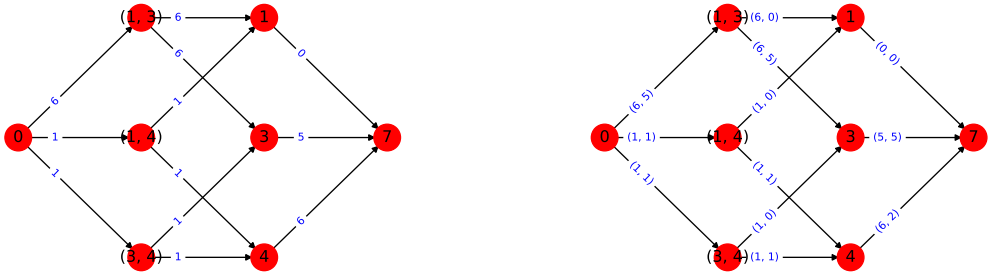
Si el valor de flujo máximo es igual al número total de juegos restantes entre los equipos en P entonces el equipo 2 aún tiene oportunidades de terminar como primero, sino ya está eliminado.

En este caso el número de juegos pendientes entre los equipos en P es 8, así que si el valor de flujo es menor a 8, entonces el equipo 2 está eliminado.

En la figura 1 se muestran las redes que representan la instancia al problema de eliminación de béisbol y la solución a dicha instancia respectivamente.

En la red de instancia se muestran las etiquetas de las capacidades de los arcos, mientras que en la red de solución se muestra la pareja ordenada (flujo permitido, flujo usado) para cada arco.

Posteriormente se encuentran los fragmentos de código 1, 2, 3, 4, 5 y 6 en donde se muestra como se configuran los parámetros de la instancia, los nodos y las conexiones entre ellos, la grafica de la instancia, guardar una imagen en formato *eps*, crear parejas y finalmente graficar la solución respectivamente.



(a) Instancia (b) Solución

Figura 1: *Instancia y solución del problema de eliminación de béisbol.*

Código 1: *Parámetros de la instancia*

```
Equipos = [i+1 for i in range(4)]
equipo = 2
cap=[[6,1,1],[0,5,6]]

P=[i for i in Equipos if i!=equipo]
Q=[(i,j) for i in P for j in P if i<j]
```

Código 2: *Crear conexiones*

```
G = nx.DiGraph()
O=0
G.add_nodes_from([O])
G.add_nodes_from(Q)
for i in range(len(Q)):
    G.add_edge(O,Q[i],capacity=cap[0][i])
G.add_nodes_from(P)
for i in range(len(Q)):
    for j in range(2):
        G.add_edge(Q[i],Q[i][j],capacity=cap[0][i])
T=len(P)+len(Q)+1
G.add_nodes_from([T])
for j in range(len(P)):
    G.add_edge(P[j],T,capacity=cap[1][j])
```

Código 3: *Graficar la instancia*

```
nx.draw_networkx_nodes(G,pos,node_shape='o',node_size=400)
nx.draw_networkx_edges(G,pos)
nx.draw_networkx_labels(G,pos)
nx.draw_networkx_edge_labels(G,pos,label_pos=0.7,font_size=8,edge_labels=nx
    .get_edge_attributes(G,'capacity'),font_color='b')
```

Código 4: *Guardar imagen en formato eps*

```
plt.axis('off')
plt.savefig('Instancia.eps')
```

Código 5: *Crear parejas de flujo permitido y usado*

```
for (i,j) in nx.get_edge_attributes(G,'capacity').keys():
    a = Capacidad[(i,j)] #Lo que puede pasar
    b = arcos[i][j]      #Lo que pasa
    dic_solcap[(i,j)]=(a,b)
```

Código 6: *Graficar la solución*

```
nx.draw_networkx_nodes(G,pos,node_shape='o',node_size=400)
nx.draw_networkx_edges(G,pos)
nx.draw_networkx_labels(G,pos)
nx.draw_networkx_edge_labels(G,pos,label_pos=0.7,font_size=8,edge_labels=
    dic_solcap,font_color='b')
```

En base a la solución mostrada en la subfigura b) de la figura 1 se puede identificar que el valor de flujo máximo es 7, el cual es menor que el número de juegos pendientes entre los equipos en P, lo que quiere decir que el equipo 2, está matemáticamente eliminado.

Referencias

[1] Python Software Foundation Versión 3.7.2. https://www.python.org/.

[2] NetworkX developers Versión 2.2. https://networkx.github.io.

[3] The Matplotlib development team Versión 3.0.2. https://matplotlib.org/.

[4] Boxley L. *An application of Maximum flow: The baseball elimination problem.* https://slideplayer.com/slide/4016790/.

[5] Wayne K. *Baseball elimination.* Princeton University.

[6] Allen P. Introduction to algorithms. University of Washington.