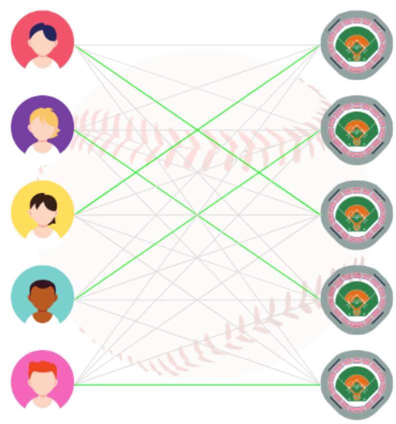
Diseño y Análisis de Algoritmos

Proyecto # 1: La Pelota



Laura Victoria Riera Pérez
Marié del Valle Reyes

Cuarto año. Ciencias de la Computación.
Facultad de Matemática y Computación, Universidad de La Habana, Cuba

3 de abril de 2023

I. Repositorio del proyecto

https://github.com/computer-science-crows/algorithms-design-and-analysis

II. Definición inicial del problema

Para un campeonato de pelota, el manager debe elegir de un conjunto de n personas, a su equipo de p jugadores, y a k espectadores especiales para que suban la moral del equipo. De cada persona i, el manager conoce el valor que aporta a la moral del equipo a_i y el valor que aporta siendo situado en la posición j, $s_{i,j}$. Determine una alineación entre jugadores en el campo y espectadores de forma que el equipo tenga la mayor cantidad de valor acumulado posible.

III. Definición en términos matemático - computacionales

I. Preliminares

Definición 1. *Grafo bipartito*

Definición 2. *Grafo bipartito completo*

Definición 3. *Para un grafo no dirigido $G = (V, E)$, un emparejamiento es un subconjunto de aristas $M \in E$ tal que cada vértice en V tiene al menos una a arista incidente en M.*

Definición 4. *Otras definiciones*

Definición 5. *Camino M-alternativo*

Definición 6. *Camino M-aumentativo*

II. Problema de asignación

IV. Posibles soluciones investigadas

V. Línea de pensamiento

I. Max flow min cut para mayor emparejamiento

No maximiza segun costos de aristas
Se decidio implementar el Hungarian por ser la solucion existente que resuelve lo pensado. referencia al introduction

VI. Hungarian algorithm

Dado un grafo bipartito completo ponderado $G = (V, E)$, donde $V = L \cup R$. Se asume que los vértices de los conjuntos L y R contienen n vértice cada uno, por tanto el grafo contiene n^2 aristas. Para $l \in L$ y $r \in R$, se denota el peso de la arista (l, r) como $w(l, r)$, lo cual representa ganancia de emparejar el vértice l con el vértice r.

El objetivo es encontrar el emparejamiento perfecto $M*$ cuyas aristas tengan el peso máximo total de todos los emparejamientos perfectos posibles.

Sea $w(M) = \sum_{(l,r) \in M} w(l, r)$ el peso total de las aristas en el emparejamiento M, se quiere encontrar el emparejamiento perfecto M^* tal que,

$$w(M*) = \max\{w(M) : M \text{ es un emparejamiento perfecto}\}$$

.

A encontrar un emparejamiento perfecto de peso máximo se le llama **problema de asignación**. Una solución del problema de asignación es un emparejamiento perfecto que máximice el costo total.

Aunque se pueden enumerar los $n!$ emparejamientos perfectos pra resolver este problema, existe un algoritmo llamado **algoritmo Húngaro** que lo resuelve más rápido. En vez de trabajar con un grafo bipartito completo G, el algoritmo Húngaro trabaja con un subgrafo de G llamado **subgrafo de igualdad**. El subgrafo de igualdad cambia en el tiempo y tiene la propiedad que cualquier emparejamiento perfecto en el subgrafo de igualdad es también una solución óptima del problema de asignación.

El subgrafo de igualdad depende de asignar un atributo h a cada vértice. El atributo h se llama **etiqueta** del vértice.

Se dice que h es un **etiquetado de vértice factible** de G si $l.h + r.h \geq w(l, r)$ para todo $l \in L$ y $r \in R$.

Un etiquetado de vértice factible siempre existe, como el **etiquetado de vértice por defecto** dado por

$$l.h = \max\{w(l,r) : r \in R\} \qquad \text{para todo } l \in R, \tag{1}$$
$$r.h = 0 \qquad \text{para todo } r \in R \tag{2}$$

Dado un eqiquetado de vértice factible h, el **subgrafo de igualdad** $G_h = (V, E_h)$ de G consiste de los mismos vértice de G y el subconjunto de aristas $E_h = \{(l,r) \in E : l.h + r.h = w(l,r)\}$.

Teorema 1. *Sea $G = (V, E)$, donde $V = L \cup R$, un grafo bipartito completo donde cada arista $(l,r) \in E$ tiene peso $w(l,r)$. Sea h un etiquetado de vértice factible de G y G_h el subgrafo de igualdad de G. Si G_h contiene un emparejamiento perfecto M^*, entonce M^* es una solución óptima del problema de asignación G.*

Demostración. Si G_h tiene un emparejamiento perfecto M^*, entonces debido a que G_h y G tienen el mismo conjunto de vértices, M^* es también un emparejamiento perfecto en G. Debido a que cada arista de M^* pertenece a G_h y cada vértice tiene exactamente una arista incidente del emparejamiento perfecto, entonces se tiene

$$\begin{aligned}
w(M*) &= \sum_{(l,r)\in M*} w(l,r) && (3)\\
&= \sum_{(l,r)\in M^*} (l.h + r.h) \qquad \text{(porque todas las aristas de } M^* \text{ pertenecen a } G_h) && (4)\\
&= \sum_{l\in L} l.h + \sum_{r\in R} r.h \qquad \text{(porque } M^* \text{ es un emparejamiento perfecto)} && (5)\\
& && (6)
\end{aligned}$$

Sea M un emparejamiento perfecto cualquiera de G, se tiene

$$\begin{aligned}
w(M) &= \sum_{(l,r)\in M} w(l,r) && (7)\\
&\leq \sum_{(l,r)\in M} (l.h + r.h) \qquad \text{(porque } h \text{ es un etiquetado de vértice factible)} && (8)\\
&= \sum_{l\in L} l.h + \sum_{r\in R} r.h \qquad \text{(porque } M \text{ es un emparejamiento perfecto)} && (9)
\end{aligned}$$

Entonces se tiene

$$w(M) \leq \sum_{l\in L} l.h + \sum_{r\in R} r.h = w(M^*), \tag{10}$$

por tanto M^* es un emparejamiento perfecto de máximo costo en G. ■

El objetivo ahora sería encontrar un emparejamiento perfecto en un subgrafo de igualdad.

Si se elige cualquier conjunto de etiquetas de vértices que defina un subgrafo de igualdad, entonces un emparejamiento de cardinalidad máxima en este subgrafo tiene un valor total de a lo sumo la suma de las etiquetas de los vértices. Si el conjunto de etiquetas de vértices es el 'correcto', entonces tendrá un valor total igual a $w(M^*)$, y un emparejamiento de cardinalidad máxima es el subgrafo de igualdad es también un emparejamiento perfecto de máximo peso.

Explicación del algoritmo Húngaro

El algoritmo Húngaro repetidamente modifica el emparejamiento y las etiquetas de vértices en orden de alcanzar su objetivo.

El algoritmo Húngaro empieza con un etiquetado de vértice factible h y cualquier emparejamiento M en el subgrafo de igualdad G_h. Repetidamente encuentra un camino M-aumentativo P en G_h y utilizando el Lema 25.1 , actualiza el emparejamiento para que sea la diferencia simétrica de M y P, incrementando así el tamaño del emparejamiento. Mientras haya algún subgrafo de igualdad que contenga un camino M- aumentativo, el tamaño del emparejamiento puede incrementar, hasta que un emparejamiento perfecto se logre.

Escribir lema

Caracteristicas del algoritmo

- El etiquetado de vértices inicial del algoritmo es el etiquetado de vértice por defecto de 1.
- El emparejamiento de G_h inicial puede ser cualquier emparejamiento. En la resolución de este problema se utilizó un algoritmo de emparejamiento maximal greedy.
- Para encontrar un camino M - aumentativo en G_h, se utiliza una variante de BFS.
- Si la búsqueda de un camino M-aumentativo falla, se debe actualizar el etiquetado de vértice factible para adicionar a G_h al menos una arista nueva.

Lema 1. *Sea h un etiquetado de vértice factible en el grafo bipartito completo G con el grafo de igualdad G_h, y se M un emparejamiento para G_h y F el bosque construído a partir de una Búsqueda Primero a lo Ancho (en inglés, BFS) sobre el subgrafo de igualdad G_h. Entonces, la etiqueta h',*

$$v.h' = \begin{cases} v.h - \delta & si\ v \in F_l, \\ v.h + \delta & si\ v \in F_r, \\ v.h & e.o.c \end{cases} \tag{11}$$

donde

$$\delta = min\{l.h + r.l - w(l,r) : l \in F_l, r \in F_r\} \tag{12}$$

con $F_l = L \cap V_f$ y $F_r = R \cap V_f$ son vértices del bosque F que pertenecen a L y a R, respectivamente, es una etiqueta de vértice factible para G con las siguientes propiedades:

1. *Si (u,v) es una arista de bosque F para G_h, entonces (u,v) $\in E_{h'}$.*
2. *Si (l,r) pertenece al emparejamiento M para G_h, entonces (l,r) $\in E_{h'}$.*
3. *Existen vértices $l \in F_l$ y $r \in R - F_r$ tales que (l,r) $\notin E_h$, pero (l,r) $\in E_{h'}$.*

Demostración. Primero se demuestra que h' es un etiquetado de vértice factible para G. Debido a que h es un etiquetado de vértice factible, se tiene $l.h + r.h \geq w(l,r)$ para todo $l \in L$ y $r \in R$. Para que h' no sea un etiquetado de vértice factible, entonces se necesitaría que $l.h' + r.h' < l.h + r.h$ para algún $l \in L$ y $r \in R$. La única forma en que esto pudiera ocurrir sería para algún $l \in F_l$ y $r \in R - F_r$. En esta instancia, la cantidad de decrecimiento es igual a δ, entonces $l.h' + r.h' = l.h - \delta + r.h$. Por ecuación 12, se tiene que $l.h - \delta + r.h \geq w(l,r)$ para cualquier $l \in F_l$ y $r \in R - F_r$, por tanto $l.h' + r.h' \geq w(l,r)$. Para cualquier otra arista, se tiene $l.h' + r.h' \geq l.h + r.h \geq w(l,r)$. Por tanto, h' es un etiquetado de vértice factible.

Ahora se mostrará la veracidad de las propiedades:

1. Si $l \in F_l$ y $r \in F_r$, entonces se tiene $l.h' + r.h' = l.h + r,h$ debido a que δ se adiciona a la etiqueta de l y se substrae de la etiqueta de r. entonces, si una arista pertenece a F para el grafo G_h, también pertenece a $G_{h'}$.

2. Se afirma que para el momentp en que el algoritmo Húngaro computa el nuevo etiquetado de vértice factible h', para toda arista $(l, r) \in M$, se tiene que $l \in F_l$ si y solo sí $r \in F_r$.

   Para demostrar por qué, se considera el vértice saturado r y la arista $(l, r) \in M$.

   Primero se supone que $r \in F_r$, entonces la búsqueda encuentra r y lo pone en la cola. Cuando r se remueve de la cola, l es descubierto, entonces $l \in F_l$.

   Luego se supone que $r \notin F_r$, por tanto r no se ha descubierto. Se demostrará que $l \notin F_l$. La única arista en G_h que entra l es (r, l), y dado que r no se ha descubierto, la búsqueda no ha tomado esta arista; si $l \in F_l$, no es por la arista (r, l). La única otra forma que un vértice en L puede estar en F_l es si es raíz de la búsqueda, pero solo vértices no saturados de L son raíces y l está saturado. Por tanto, $l \notin F_l$ y la afirmación se cumple.

   Se conoce que para $l \in F_l$ y $r \in F_r$ se cumple $l.h' + r.h' = l.h + r.h$. En caso contrario, cuando $l \in L - F_l$ y $r \in R - F_r$, se tiene que $l.h' = l.h$ y $r.h' = r.h$, entonces $l.h' + r.h' = l.h + r.h$. Por tanto, si la arista (l, r) está en el emparejamiento M para el grafo G_h, entonces $(l, r) \in E_{h'}$.

3. Sea (l, r) una arista que no pertenece a E_h, tal que $l \in F_l$, $r \in R - F_r$ y $\delta = l.h + r.h - w(l, r)$. Entonces, por definición de δ, existe al menos una de esas arista. Luego, se tiene

$$
\begin{aligned}
l.h' + r.h' &= l.h - \delta + r.h \\
&= l.h - (l.h + r.h - w(l, r)) + r.h \\
&= w(l, r)
\end{aligned}
$$

   y por tanto $(l, r) \in E_{h'}$. Dado que $(l, r) \notin E_h$, no está en el emparejamiento M, entonces en $E_{h'}$ debe estar dirigida de L a R. Por tanto, $(l, r) \in E_{h'}$.

arreglar esto porque nuestro grafo no es dirigido

■

I. Correctitud

II. Complejidad Temporal

III. Complejidad Espacial

VII. Generador de casos de prueba

VIII. Tester

IX. Otras soluciones y demostraciones

X. Comparación de soluciones implementadas

Referencias

[1] Cormen, Thomas H. y otros. *Introduction to Algorithms*. The MIT Press. 4ta Edición. Cambridge, Massachusetts. 2022.