MA3705. Algoritmos Combinatoriales 2020.
Profesor: José Soto
Escriba(s): Vicente Cabezas y Cynthia Vega.
Fecha: 13 de noviembre de 2020.

Cátedra 17

Problemas de matching perfecto de costo mínimo.

1. Tres problemas equivalentes

1. Dado $G = (V, E)$ bipartito $w : E \to \mathbb{R}$.
   Encontrar matching M de peso $w(M)$ máximo.
2. Dado $G = (V, E)$ bipartito, $k \in \mathbb{N}$, $w : E \to \mathbb{R}$.
   Encontrar matching M con $|M| = k$ y de peso $w(M)$ máximo.
3. **(Problema de asignación)** Dado $G = (V, E)$ bipartito completo balanceado, $c : E \to \mathbb{R}$.
   Encontrar matching perfecto M de peso $c(M)$ mínimo.

Se pueden ocupar costos positivos o negativos.

Ejercicio presentable: Demostrar que $3 \Rightarrow 1$
Ejercicio desafio: Demostrar que $3 \Rightarrow 2$

1.1. Problema de asignación como programa lineal entero

Sea $G = (L \cup R; E \equiv L \times R)$ grafo bipartito completo balanceado.
$c : E \to \mathbb{R}$ función de costo.

$$\min \sum_{ij \in E} x_{ij} c_{ij}$$

<u>*Recuerdo:*</u> $\min \{ c^T x : x \in X\} = \min \{ c^T x : x \in conv(X)\}$

1.2. Birkhoff Von Neumann + Programación lineal

Optimización lineal + teorema BVN + dualidad

$$\min \sum_{ij \in E} x_{ij} c_{ij}$$

$$X = \begin{cases} \sum_{j \in \mathbb{R}} x_{ij} = 1, \ \forall i \in L \\ \sum_{i \in L} x_{ij} = 1, \ \forall j \in \mathbb{R} \\ x \in \{0, 1\}^E \end{cases}$$

Si no se considera la condición $x \in \{0, 1\}^E$ y se reemplaza por $x \geq 0$.

Luego por el teorema de Birkhoff Von Neumann, se tiene:

$$X = conv(X^M : M \text{ matching perfectos}) = \begin{cases} \sum_{j \in \mathbb{R}} x_{ij} = 1, \ \forall i \in L \\ \sum_{i \in L} x_{ij} = 1, \ \forall j \in \mathbb{R} \\ x \geq 0 \end{cases}$$

Con esto se transforma el programa entero (que podría ser difícil), por un problema lineal.

Ahora el problema es resolver este nuevo poliedro.

Para atacar este problema, se utilizan herramientas de optimización lineal, en donde la primera idea es usar dualidad.

Si se considera como primal al último X visto, tomando z_j como una variable para cada $j \in \mathbb{R}$ y y_i para cada $i \in L$. Se podría calcular la función objetivo del dual, la cual quiere maximizar de la siguiente forma:
$\max \sum\limits_{i \in L} y_i + \sum\limits_{j \in \mathbb{R}} z_j$

En el primal hay variables i y j, entonces para el dual se tiene una restricción para cada i,j posible. Con esta restricción, queda el siguente problema:

$$\begin{cases} \max \sum\limits_{i \in L} y_i + \sum\limits_{j \in \mathbb{R}} z_j \\ y_i + z_k \leq c_{ij},\ \forall i,j \in L \times \mathbb{R} \end{cases}$$

En las siguientes secciones se verá como se resuelven entos problemas.

2. Introducción a Algoritmos Primal-Dual

SIMPLEX no es un algoritmo polinomial (pero es raro que no logre soluciones rápidas).

En algunos casos, es posible encontrar soluciones de un PL de manera combinatorial:

Algoritmo Primal-Dual.

1. Encontrar solución Dual d factible inicial
2. Repetir:
   a) Obtener vector p con holgura complementaria respecto a d
   b) Si p es primal factible, se tiene
   c) Si no, usar p para encontrar un mejor d DUAL

Si se aplica esta idea general a un matching, se obtiene lo que se ve a continuación.

2.1. Holgura complementaria en problema de asignación

$$\text{(P)}=\begin{cases} \min \sum\limits_{ij \in E} x_{ij} c_{ij} \\ \sum\limits_{j \in \mathbb{R}} x_{ij} = 1,\ \forall i \in L \\ \sum\limits_{i \in L} x_{ij} = 1,\ \forall j \in \mathbb{R} \\ x \in \{0,1\}^E \end{cases} = \left.\begin{matrix} \max \sum\limits_{i \in L} y_i + \sum\limits_{j \in \mathbb{R}} z_j \\ y_i + z_k \leq c_{ij},\ \forall i,j \in L \times \mathbb{R} \end{matrix}\right\}=\text{(D)}$$

Se toma una solución primal factible, una dual factible, si satesfacen holgura complementaria, entonces todo es óptimo.

Holgura: Es la diferencia entre el lado derecho y el lado izquierdo.
Por ejemplo en $y_i + z_k \leq c_{ij}$ la holgura es $(c_{ij} - y_i - z_j)$

Holgura complementaria: Dice que la holgura es 0 o la variable asociada es 0.
Por ejemplo en el mismo caso anterior quedaría $(c_{ij} - y_i - z_j) \cdot x_{ij} = 0 \Rightarrow x_{ij} = 0 \vee (c_{ij} - y_i - z_j) = 0$

Observación: x primal factible & (y,z) dual factible & holgura complementaria $\Longrightarrow$ óptimo. Demostración propuesta (sale una 1 línea).

Lema 1. *Si (y,z) dual factible, M matching perfecto $M \subseteq E(y,z) = \{ij \in E : c_{ij} - y_i - z_j = 0\}$ entonces M es asignación óptima*

$E(y,z) = \{ij \in E : c_{ij} - y_i - z_j = 0\}$, serán llamadas las aristas ajustadas.

Como se quiere utilizar matching perfectos, se utilizará solo el dual por lo tanto el problema primal se omite.

2.2. Algoritmo Húngaro primal-dual para problema de asignación

Notación: $\sum_{i \in L} y_i = y(L)$ y $\sum_{j \in \mathbb{R}} z_j = z(R)$

Algorithm 1: Primal-Dual (Khun 1955)

Entrada $e : L \times R \to \mathbb{R}$
(y,z) solución dual-factible inicial
Repetir
  $M \leftarrow$ matching de tamaño máximo de
  $E(y,z) = \{ij \in E : c_{ij} - y_i - z_j = 0\}$
  si M perfecto entonces devolver M
  en otro caso. Encontrar dual-factible $(\hat{y}, \hat{z})$ con
  $y(L) + z(R) < \hat{y}(L) + \hat{z}(R) . (y,z) \leftarrow (\hat{y}, \hat{z})$

Entonces el dual vendría siendo: $\begin{cases} \text{máx } \ \text{y(L)+z(R)} \\ \text{y}_i + z_j \leq c_{ij}, \forall i, j \end{cases}$

El primer paso del algoritmo trata de encontrar una solución factible inicial.

Segundo paso: se encuentra un matching de tamaño máximo en las aristas ajustadas, si se tiene suerte el matching es perfecto, luego por el lema anterior es óptimo. Si no es perfecto se utiliza el matching máximo para encontrar una solución dual factible, con mejor valor $(\hat{y}, \hat{z})$. Una posible forma de solucionar esto es tomando $y_i = 0$, $\forall i \in L$, entonces necesariamente $z_j = \min_{i \in L} c_{ij} \Rightarrow y_i + z_j = z_j \leq z_j \leq c_{ij}$.

Luego, queda lo siguiente:

Algorithm 2: Primal-Dual (Khun 1955)

Entrada $e : L \times R \to \mathbb{R}$
$y \leftarrow 0, \forall j \in E, z_j \leftarrow min\{c_{ij} - y_i \; i \in L\}$
Repetir
  $M \leftarrow$ matching de tamaño máximo de
  $E(y,z) = \{ij \in E : c_{ij} - y_i - z_j = 0\}$
  si M perfecto **entonces devolver** M
  en otro caso
    $\epsilon \leftarrow min\{c_{ij} - y_i - z_j : i \in L \cap Q, j \in R \backslash Q\}$
    $y \leftarrow y + \epsilon \chi^{L \cap Q}$
    $z \leftarrow z - \epsilon \chi^{R \cap Q}$ (Q=alcanzables)
  fin

Demostración

1. $\tilde{y}, \tilde{z} = (y + \varepsilon \mathcal{X}^{L\cap Q}, z - \varepsilon \mathcal{X}^{L\cap Q})$ es dual factible.
2. Es de mayor valor que (y, z)

Del grafo bipartito inicial $D(G, M)$, se va a considerar sólo $D(E(y, z), M)$ que vendría siendo el matching máximo.

A todos los alcanzables de L se le aumenta un ϵ y a todos los alcanzables en R, se le resta un ϵ. En particular se va a usar el mayor ϵ posible tal que la solución siga siendo factible. Al observar todas las aristas que hay entre $L \cap Q$ y $R\backslash Q$ en el grafo original (no en el grafo ajustado) tales que $c_{ij} - y_i - z_j \geq 0$ (porque la solución era factible). Se analizan todos estos valores y se ocupa el mínimo, el cual no puede ser 0 ya que se sabe que $C = (L\backslash Q) \cup (R \cap Q)$ es un cubrimiento mínimo, lo cual quiere decir que todas las aristas tocan o bien a $L\backslash Q$ o bien $R \cap Q$. Entonces las aristas que van $Q \cap R$ a $R\backslash Q$ no son ajustadas. Esto implica que el mínimo es mayor estricto que 0 y ese ϵ es el que se va a utilizar.

Ahora se quiere probar que para todas las aristas del grafo se cumple que aún se tiene esta restricción. se tiene que las aristas que van de $Q \cap L$ a $R\backslash Q$ (1) (aristas de tipo uno), tenían holgura positiva, si se le suma epsilon a $Q \cap L$, entonces el lado izquierdo queda $\tilde{y}_i + \tilde{z}_i = y_i + z_j + \epsilon \leq c_{ij}$, por lo tanto, siguen siendo factibles. Luego, las aristas de tipo 2, que van de $Q \cap L$ a $Q \cap R$ (2), con esto, $\tilde{y}_i + \tilde{z}_j = y_i + \epsilon + z_j - \epsilon = y_i + z_j \leq c_{ij}$ también cumpliendo la factibilidad. Se verá ahora el tercer tipo de arista, que van de $L\backslash Q$ a $R\backslash Q$ (3), como no se alteraron, quedaron igual, entonces $\tilde{y}_i + \tilde{z}_j = y_i + z_j \leq c_{ij}$. Para finalizar se ve el cuarto tipo de arista, las que van de $L\backslash Q$ a $Q \cap R$ (línea curva en figura 1) queda lo siguiente $\tilde{y}_i + \tilde{z}_j = y_i + z_j - \epsilon \leq y_i + z_j \leq c_{ij}$. Todo esto implica que $\tilde{y}(L) + \tilde{z}(R) = y(L) = z(R) + \epsilon \cdot |Q \cap L| - \epsilon \cdot |Q \cap R|$, entonces falta probar que $\epsilon \cdot |Q \cap L| - \epsilon \cdot |Q \cap R| \geq 0$, pero se puede igualar a lo siguente $y(L) + z(R) + \epsilon \cdot \begin{pmatrix} |Q \cap L| + |L\backslash Q| \\ \text{-}|L\backslash Q| - |Q \cap R| \end{pmatrix} = \epsilon \cdot |L| - \epsilon |c|$, luego como $|c| < |L|$, ya que el cubrimiento mínimo es igual al tamaño de un matching máximo, el cual por hipótesis no es perfecto. la demostración de que es mayor a (y, z), se tiene directo de las igualdaes anteriores.

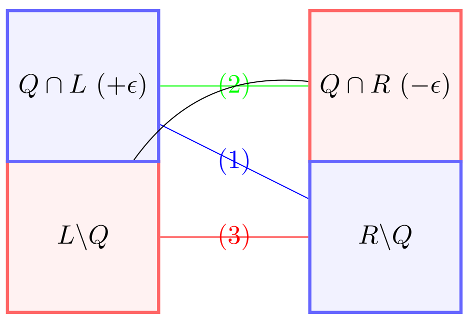

Figura 1

Correctitud

El algoritmo termina cuando encuentra un matching perfecto en las aristas ajustadas de acuerdo a la solución dual en la que está parada y el lema con el que se partio, se dice que ese matching es solución óptima del problema de simulación. Por ende si el algoritmo termina, devuelve solución óptima.

Complejidad

Lema 2. *Se tienen los siguientes:*
(1) $|M|$ no decrece.
(2) M puede crecer solo n veces.
(3) Cuando M no crece, Q aumenta.

En cada paso se va a llamar a la iteración Repetir.
Primer paso: Se quiere probar que el cardinal de M siempre aumenta, o en el peor de los casos, se mantiene. Se sabe que todas aristas del matching van de derecha a izquierda, pero ninguna va en diagonal, es decir de alcanzable a alcanzable, o inalcanzable a inalcanzable. Si $M \subseteq E(y, z)$, entonces $M \subseteq E(\tilde{y}, \tilde{z})$, luego las aristas de M siguen siguen siendo ajustadas,

es decir, al menos tiene las mismas aristas de antes (si una arista sale, quiere decir que no pertenecía al matching) (1).

Para M en cada paso se calcula el matching de tamaño máximo, ese tamaño o bien se mantiene o bien crece, y cuantas veces puede crecer es a lo más n veces (2).

ϵ se elije mirando $Q \cap L$, R y tomando el menor valor de holgura de las aristas del grafo que cruzan de $Q \cap L$ a R. Entonces al elegir ese ϵ, al menos alguna arista de esa zona se ajusta. Si $v \in Q$ sigue estando en Q, porque las aristas entre alcanzables siguen estando en el grafo. Pero como se ajustó una arista, que se llamará $e = lr$, entonces en la nueva iteración $r \in Q$, el cual es un vértice nuevo, ojo que Q no puede crecer mas de n veces tampoco (3).

Debido a lo anterior en total hay n^2 etapas o iteraciones.

3. Resumen

El algoritmo húngaro primal-dual encuentra un matching perfecto de costo mínimo en $O(n^2)$ iteraciones, cada una de ellas toma $O(n + |E(y,z)|) = O(n^2)$.
En total $O(n^4)$.
Nota: Mejores algoritmos para matching de peso máximo (no nec. completo)

Algoritmo	Complejidad
Kunh (1955)	$O(n^4)$
Iri (1960)	$O(n^2 m)$
Dinic-Kronrod (1969)	$O(n^3)$
Edmonds-Karp	$O(nm + n\log(n))$