Algoritmos y Estructuras de Datos III

Segundo cuatrimestre 2022

Problema de flujo de costo mínimo
Introducción a la programación lineal

Flujo de costo mínimo

- **Datos de entrada:**
  1. Un grafo dirigido $G = (N, A)$.
  2. Imbalance $b : N \to \mathbb{Z}_+$ de cada nodo.
  3. Capacidad $u : A \to \mathbb{Z}_+$ de cada arco.
  4. Costo unitario $c : A \to \mathbb{Z}_+$ para cada arco.

- **Problema:** Encontrar un flujo que respete el imbalance de cada nodo y las cotas de cada arco, con el menor costo posible.
  1. Para cada nodo $i \in N$, debemos tener $b_i = \sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji}$.
  2. La cantidad x_{ij} enviada por el arco $ij \in A$ debe cumplir $0 \leq x_{ij} \leq u_{ij}$.
  3. El costo del flujo es $C = \sum_{ij \in A} c_{ij} \, x_{ij}$.

Flujo de costo mínimo

- Realizamos las siguientes suposiciones.
  1. Todos los datos (imbalance, capacidad y costo) son enteros.
  2. Los imbalances balancean: $\sum_{i \in N} b_i = 0$.
- Cuando $b(i) = 0$ para todo $i \in N$, decimos que el flujo es una circulación.
- **Propiedad.** Toda circulación puede descomponerse en un conjunto de ciclos simples.

Flujo de costo mínimo

- **Aplicación.** Optimización de la fabricación y distribución de un producto.
  1. Tenemos dos fábricas que manufacturan un producto, cada una con una capacidad máxima por mes.
  2. De las fábricas se envía el producto a centros de distribución, cada uno con una capacidad máxima por mes. El despacho entre fábricas y centros de ditribución no es ilimitado.
  3. Tenemos varios puntos de venta, abastecidos desde los centros de distribución. Cada punto de venta tiene una demanda.
  4. Tenemos costos de fabricación y despacho.
- El problema consiste en determinar cuánto fabricar, cuánto enviar a cada centro de distribución y a cada punto de venta, de modo tal de cumplir la demanda total.

Flujo de costo mínimo

- Definimos la red residual G_x de un flujo $x : A \to \mathbb{R}_+$ reemplazando cada arco $ij \in A$ por dos arcos ij y ji.
  1. El arco ij tiene costo c_{ij} y capacidad residual $r_{ij} = u_{ij} - x_{ij}$.
  2. El arco ji tiene costo $-c_{ij}$ y capacidad residual $r_{ji} = x_{ij}$.
- La red residual consiste solamente de los arcos con capacidad residual positiva.
- **Teorema.** Una solución factible x es óptima si y sólo si la red residual G_x no contiene ningún ciclo (dirigido) de costo negativo.

Flujo de costo mínimo

Morton Klein (1926–2001)

- **Algoritmo de cancelación de ciclos** (Klein, 1967). A partir de un ciclo factible, mientras exista un ciclo negativo aumentar el flujo a lo largo de ese ciclo.

Flujo de costo mínimo

Algoritmo de cancelación de ciclos

1. Establecer un flujo x factible.
2. **Mientras** G_x contenga un ciclo negativo W **hacer**
   - Definir $\delta := \min\{r_{ij} : ij \in W\}$.
   - Aumentar δ unidades de flujo a lo largo del ciclo W y actualizar x.
3. **Fin mientras**

- ¿Cómo obtenemos el flujo inicial factible?

Flujo de costo mínimo

- **Teorema.** Si todos los imbalances y capacidades son enteros, entonces el problema de flujo de costo mínimo tiene una solución óptima entera.

- ¿Cuál es la complejidad computacional de este algoritmo?

  1. $C := \max\{c_{ij} : ij \in A\}$.
  2. $U := \max\{u_{ij} : ij \in A\}$.

- El costo del flujo inicial no puede ser superior a mCU y el costo final no puede ser inferior a cero. Luego, el algoritmo realiza a lo sumo mCU iteraciones y su complejidad total es $O(nm^2CU)$.

Flujo de costo mínimo

- Si en cada paso se selecciona un ciclo de costo promedio mínimo (y se puede hacer en $O(nm)$), entonces este algoritmo realiza a lo sumo $O(\text{mín}\{nm \log(nC), nm^2 \log n\})$ iteraciones (Goldberg y Tarjan, 1988).

- Podemos considerar también cotas inferiores $\ell : A \to \mathbb{Z}_+$ al flujo en cada arco, de modo tal que x debe cumplir $\ell_{ij} \leq x_{ij} \leq u_{ij}$ para todo $ij \in A$. Este problema también se puede resolver en tiempo polinomial.

Programación lineal

- Podemos escribir el problema de flujo de costo mínimo del siguiente modo.

$$\begin{array}{rcll} \text{mín} & \displaystyle\sum_{ij \in A} c_{ij} x_{ij} & & \\ \displaystyle\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} & = & b_i & \forall i \in N \\ x_{ij} & \geq & \ell_{ij} & \forall ij \in A \\ x_{ij} & \leq & u_{ij} & \forall ij \in A \end{array}$$

- Esta formulación es un problema de programación lineal.

Programación lineal

- **Problema:** Dados una matriz $A \in \mathbb{R}^{m \times n}$ y vectores $b \in \mathbb{R}^m$ y $c \in \mathbb{R}^n$, el problema general de programación lineal es:

$$\text{máx}\,\{c^T x : Ax \leq b,\ x \in \mathbb{R}^n\}.$$

- Es habitual suponer que $x \geq 0$, aunque estas restricciones se pueden incorporar al sistema de desigualdades $Ax \leq b$.

Programación lineal

- **Ejemplo.** Si definimos ...

$$A = \begin{pmatrix} 2 & -3 & 1 \\ 0 & 1 & 4 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad c = \begin{pmatrix} 3 \\ -1 \\ 2 \end{pmatrix}$$

- ... entonces tenemos un modelo con $n = 3$ variables y $m = 2$ restricciones.
- Escribimos el modelo de forma extendida:

$$\begin{array}{rcl} \text{máx} & & 3x_1 - x_2 + 2x_3 \\ 2x_1 - 3x_2 + x_3 & \leq & 1 \\ x_2 + 4x_3 & \leq & 2 \\ x & \geq & 0 \end{array}$$

Programación lineal

- Cuando $n = 2$, podemos resolver el problema en forma gráfica. No obstante, el problema también tiene sentido para $n > 2$.
- A pesar de que el problema original es $\max \{c^T x : Ax \leq b,\ x \in \mathbb{R}^n\}$, podemos escribir distintas variantes dentro de este esquema.
  1. Podemos minimizar en lugar de maximizar (cómo?).
  2. Podemos escribir restricciones por $\geq$ (cómo??).
  3. Podemos escribir restricciones por $=$ (cómo???).

Programación lineal

George Dantzig (1914–2005)

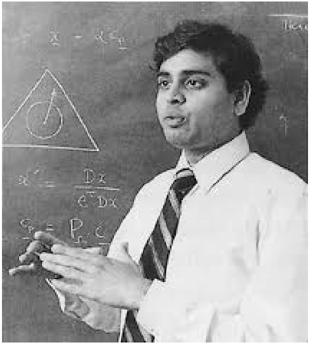
- El algoritmo más famoso para resolver programación lineal es el Método Simplex (Dantzig, 1947).
- Es un algoritmo exponencial, pero muy eficiente en la práctica.

Programación lineal

Leonid Khachiyan (1952–2005)

- Programación lineal se puede resolver en tiempo polinomial (Khachiyan, 1979), aunque con un método que no es eficiente en la práctica.

Programación lineal

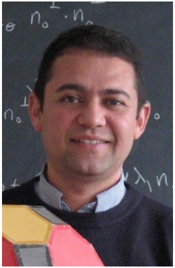


Narendra Karmarkar (1956–)

- El primer algoritmo polinomial eficiente en la práctica es el método de punto interior (Karmarkar, 1984), con complejidad $O(m^{3,5}\,L^2 \log L \log\log L)$ (L: cantidad de bits en el input).

Programación lineal

Alexander Black



Jesús De Loera

Sean Kafer

Laura Sanità

- Una variante del Método Simplex ejecuta en tiempo (casi) fuertemente polinomial si la región factible tiene vértices con extremos 0/1 (Black, De Loera, Kafer y Sanità, 2021).

Programación lineal

- Como vimos, el problema de flujo de costo mínimo se puede reducir al problema de programación lineal.

$$
\begin{array}{rcll}
\text{mín} & \displaystyle\sum_{ij\in A} c_{ij}x_{ij} & & \\
\displaystyle\sum_{j\in N^+(i)} x_{ij} - \sum_{j\in N^-(i)} x_{ji} & = & b_i & \forall i \in N \\
x_{ij} & \geq & \ell_{ij} & \forall ij \in A \\
x_{ij} & \leq & u_{ij} & \forall ij \in A
\end{array}
$$

- Las propiedades particulares de esta matriz permiten asegurar que este programa lineal tiene óptimo entero, y que el Método Simplex puede encontrarlo.

Programación lineal

- Podemos escribir también el problema de flujo máximo como un programa lineal.

$$
\begin{array}{rcll}
\text{máx} & & \displaystyle\sum_{j \in N^+(s)} x_{sj} & \\
\displaystyle\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} & = & 0 & \forall i \in N \backslash \{s, t\} \\
x_{ij} & \leq & u_{ij} & \forall ij \in A \\
x_{ij} & \geq & 0 & \forall ij \in A
\end{array}
$$

- Podemos escribir el problema de camino mínimo con programación lineal?

Programación lineal

- Podemos escribir también el problema de árbol generador mínimo como un programa lineal.

$$
\begin{array}{rcll}
\text{mín} & \displaystyle\sum_{ij \in E} c_{ij} x_{ij} & & \\
\displaystyle\sum_{ij \in E(S)} x_{ij} & \leq & |S| - 1 & \forall S \subseteq V \\
\displaystyle\sum_{ij \in E} x_{ij} & = & n - 1 & \\
x_{ij} & \geq & 0 & \forall ij \in E
\end{array}
$$

- También se puede demostrar que esta formulación tiene óptimo entero.

- Aunque es una formulación con una cantidad exponencial de restricciones, se puede resolver en tiempo polinomial.

Programación lineal

- Todos los problemas polinomiales que vimos hasta ahora se pueden reducir a programación lineal.

Programación lineal

- **Todos los problemas polinomiales** ~~que vimos hasta ahora~~ **se pueden reducir a programación lineal.**
  1. En este sentido, programación lineal es el problema polinomial más difícil (o más general). Vamos a formalizar estas ideas la próxima clase.
  2. Se conjetura que esto implica que programación lineal no se puede paralelizar en forma eficiente.
- No es el único problema con estas características, aunque posiblemente sea el problema al cual es más sencillo reducir problemas polinomiales.

Programación lineal

- Dado un problema lineal

$$\max \{c^T x : Ax \leq b,\ x \in \mathbb{R}^n_+\}, \qquad (P)$$

su problema dual es

$$\min \{y^T b : y^T A \geq c,\ y \in \mathbb{R}^m_+\}. \qquad (D)$$

- **Teorema de dualidad fuerte.** Si (P) y (D) son factibles y tienen óptimos finitos, entonces sus valores óptimos coinciden.