MA4702. Programación Lineal Mixta. 2020.
Profesor: José Soto
Escriba(s): Sebastian Bustos, Felipe Hernández y Nicolás Toro.
Fecha: 20 de marzo de 2020.

Cátedra 1

Introducción

En esta sesión se estudiarán los conceptos formales de lo que es un problema de optimización en una forma general, se definirán conceptos claves para el trabajo de problemas de optimización con diferentes restricciones y se presentan ejemplos que ilustran lo presentado.

1. Problemas y algoritmos (en optimización)

Comencemos con las definiciones esenciales:

Definición 1 (Problema de optimización e instancias)**.** Un *problema de optimización* $\mathcal{P}$ es un conjunto de *instancias*. Cada *instancia* $\mathcal{I}$ esta definida por:

- Un conjunto factible $S = \text{fact}(\mathcal{I})$.
- Una función a optimizar $f\colon S \to \mathbb{R}$.
- Un objetivo, minimizar o maximizar la función f en dicho conjunto S.

Usualmente las instancias se describen de manera implicita o compacta.

Ejemplo 1 (Árbol cubridor de peso mínimo − minimum spanning tree, MST)**.** Cada instancia del problema (MST) se describe de manera compacta indicando un grafo $G = (V, E)$ con pesos en las aristas $w\colon E \to \mathbb{R}_+$. De estos datos uno puede deducir el conjunto S de todos los árboles cubridores de G, y para cada árbol $T \in \mathcal{S}$, el valor de la función es la suma de los pesos de las aristas $w(T) = \sum_{e\in E[T]} w(e)$. El objetivo es minimizar la función de peso w:

$$\min_{T\in S}\ w(T)$$

Definición 2 (Algoritmo)**.** Un *algoritmo* para un problema de optimización $\mathcal{P}$ es un método que recibe una instancia $(S, f, \max)$ y entrega, en un número finito de pasos:

1. El óptimo en caso de existir. Es decir un elemento $\text{OPT} \in S$ tal que $f(\text{OPT}) = \max_{x\in S} f(x)$.
2. Ó bien certifica que no existe elemento óptimo. Esto puede pasar cuando:
   a) El problema es infactible ($S = \emptyset$).
   b) El problema no es acotado $\left(\max_{x\in S} f(x) = \infty\right)$.
   c) El maximo no se alcanza ($\max \neq \sup$).

Diremos que un algoritmo que siga la estructura anterior resuelve el problema de optimización.

Ejemplo 2 (Kruskal − MST)**.** Un ejemplo de un algoritmo que resuelve el problema MST es Kruskal, modificándolo previamente para que certifique que no hay solución al problema si es que el grafo es disconexo.

Es importante notar que un algoritmo **tiene** que terminar en un tiempo finito (en la practica, el tiempo de ejecución puede ser tan grande, que no hay diferencia entre esto e "infinito").

2. Programas Lineales Mixtos

Definición 3 (Conjunto lineal mixto)**.** Se dice que $S \subseteq \mathbb{Z}^E \times \mathbb{R}^C \subseteq \mathbb{R}^n$ es un conjunto lineal mixto si S pude ser descrito como intersección de un conjunto finito de desigualdades lineales. Es decir S es:

$$S := \{x \in \mathbb{Z}^E \times \mathbb{R}^C : Ax \leq b\} \subseteq \mathbb{R}^n.$$

Con $n, m \in \mathbb{N}$, $E, C \subseteq [n]$ tales que $E \cup C = [n]$, $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ y $b \in \mathbb{R}^m$.

La interpretación y nombres en la definición de un conjunto lineal mixto viene dado por:

1. Las coordenadas en E se llaman *variables enteras*.
2. Las coordenadas en C se llaman *variables continuas*.
3. Las m desigualdades de la forma $a_j^T x \leq b_j$ se llaman *restricciones*.
4. Si $E = [n]$ a S se le llama *conjunto lineal entero*.
5. Si $C = [n]$ a S se le llama *conjunto lineal puro o poliedro*.
6. Si $x \in \{0, 1\}^n$ a S se le llama *conjunto lineal binario*.

Ejemplo 3. En la Figura 1 podemos ver un conjunto lineal puro en $\mathbb{R}^2$, el cual esta determinado por las inecuaciones:

$$\begin{pmatrix} 0 \\ 0 \\ y \end{pmatrix} \leq \begin{pmatrix} y \\ x \\ c - x \end{pmatrix},$$

donde $c > 0$. Mientras que en la Figura 2 observamos el conjunto lineal mixto obtenido añadiendo la restricción adicional $x \in \mathbb{Z}$.

Es importante recordar que sin perder generalidad se puede suponer que $Ax \leq b$ incluye todas las restricciones lineales de un problema de optimización (tanto las inecuaciones como las igualdades).

En la Figura 3 podemos observar un conjunto lineal binario, que cumple la restricción $x \in \{0, 1\}^3$

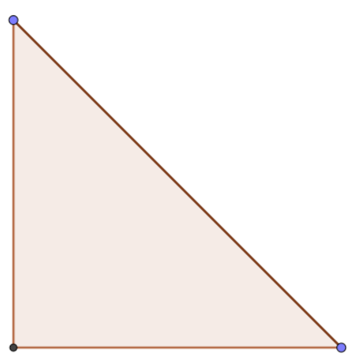

Figura 1: Conjunto lineal puro

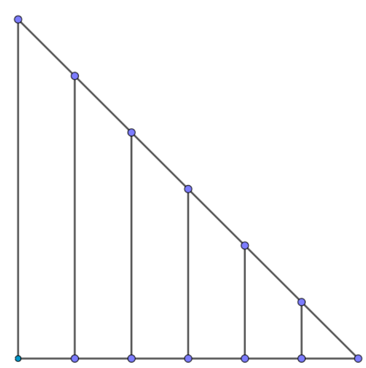

Figura 2: Conjunto lineal mixto
Eje x variable entera
Eje y variable continua

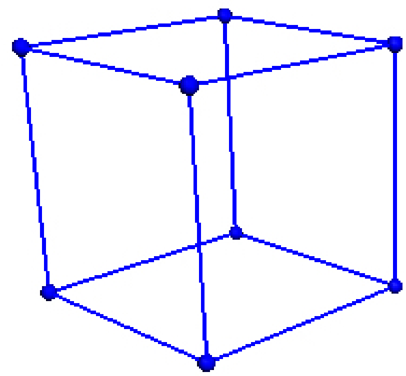

Figura 3: Conjunto lineal binario

Definición 4 (Programas lineales puros / mixtos / enteros / binarios)**.** Diremos que un problema de la forma:

$$\max\{c^t : x \in S\}$$

es un *programa lineal puro / mixto / entero / binario* si S es un conjunto lineal puro / mixto / entero / binario.

Se abreviaran como *PL / PLM / PLE / PLB* respectivamente.

Observación: A S se le conoce como dominio o conjunto factible del programa.

3. Modelos

En general los problemas de optimización que aparecen en la vida real no están escritos como un PLM. Por ende es necesario reescribir un problema para poder trabajarlo como tal.

Definición 5 (Modelo)**.** Diremos que un PLM es modelo de un problema si

- Cada solución óptima del PLM es solución óptima del problema original.
- Al menos una solución óptima del problema original es factible en el PLM

Es importante notar que a veces se relaja la primera condición y se pide que sea algún tipo de solución óptima en particular. La definición anterior da la oportunidad de que existan soluciones óptimas del problema original que no sean factibles en el PLM, en caso contrario nos encontramos frente a un modelo particular:

Definición 6 (Modelo exacto)**.** Diremos que un PLM es un modelo exacto si las soluciones factibles del problema original están en correspondencia uno a uno, de manera explicita con las soluciones factibles del PLM

Ejemplo 4 (Knapsack/ Problema de la mochila)**.** Dado $n \in \mathbb{N}$ objetos, valores $v_i \geq 0$ y tamaños $s_i \geq 0$. Dada una mochila de capacidad $B \geq 0$. Seleccionar un subconjunto a poner en la mochila maximizando el valor total.

$$\begin{aligned} \text{(Knapsack)}\ \max \quad & \sum_{i\in[n]} v_i x_i \\ \text{s.a.} \quad & \sum_{i\in[n]} s_i x_i \leq B \\ & x_i \in \{0,1\}\ \forall i \in [n], \end{aligned}$$

donde

1. $x_i = 1 \iff i$ se encuentra en la mochila
2. $\sum_{i \in [n]} s_i x_i$ es la capacidad ocupada de la mochila.
3. $\sum_{i \in [n]} v_i x_i$ es el valor de la mochila.

Observación 1: Este modelo es exacto.

Observación 2: Es importante notar que a veces, por mucho que se conozcan técnicas para resolver un problema, es conveniente escribir el problema como un PLM, lo anterior permite la libertad de agregar restricciones fácilmente a los problemas de optimización.

Ejemplo 5 (s-t Camino de largo mínimo)**.** Dado el digrafo $G = (V, \vec{E})$, nodo de origen $s \in V$, nodo destino $t \in V$. Dada además una función de largos no negativos $\ell : \vec{E} \to \mathbb{R}_+$. Determinar un camino de largo mínimo de s a t. Para resolver este problema, necesitamos caracterizar de alguna manera los s-t caminos. Veremos dos maneras de realizar esto:

I.- (Mediante cortes) Sea $x \in \{0,1\}^E$ en donde $\forall e \in E,\ x(e) = 1 \iff$ el arco e esta siendo ocupado. Recordemos la definición de corte:

Definición 7 (s-t corte)**.** Un subconjunto $U \subseteq V$ es un s-t corte si $s \in U, t \notin U$.

Tambien recordemos las siguientes definiciones:

Definición 8.

- $\delta^+(U)$ son los arcos que salen del conjunto U.
- $\delta^-(U)$ son los arcos que entran del conjunto U.

Para caracterizar que se sale de cada U s-t corte utilizaremos la siguiente ecuación:

$$x(\delta^+(U)) := \sum_{e \in \delta^+(U)} x(e) \geq 1$$

Así el problema lo podemos plantear como sigue:

$$\begin{aligned} \text{(SP-Conector)}\ \ \text{mín} \quad & \sum_{e \in E} \ell_e x_e \\ \text{s.a.} \quad & x\left(\delta^+(U)\right) \geq 1 \text{ para todo } s\text{-}t \text{ corte } U \\ & x \in \{0,1\}^E \end{aligned}$$

a) Se puede probar que si el conjunto de arcos cruza todos los cortes entonces tiene que haber un s-t camino.

b) Hay $|\mathcal{P}(V \setminus \{s,t\})| = 2^{n-2}$ cortes, y por ende 2^{n-2} restricciones.

Se dejan los siguientes ejercicios propuestos:

Ejercicio 1. Probar que para el problema s-t camino de coste mínimo y $\ell_e > 0\ \forall e$ entonces SP-Conector es un modelo. Y probar que (en general) no es un modelo exacto.

Ejercicio 2. Modificar SP-Conector para que incluso cuando ℓ pueda tomar valores nulos sea un modelo.

Es importante notar que la cantidad de restricciones crece de manera exponencial con el tamaño, luego, si bien el modelo soluciona el problema, en la practica el tiempo en que se ejecute el programa va a ser grande.

Por lo anterior es conveniente plantear el problema de otra manera.

II.- (Mediante flujos) Recordemos la definición de flujo:

Definición 9 (s-t flujo)**.** Una función $f : E \to \mathbb{R}_+$ es un s-t flujo si:

$$f\left(\delta^+(u)\right) = f\left(\delta^-(u)\right) \ \forall u \in V \backslash \{t, s\}$$

Luego el problema es posible plantearlo como:

$$\begin{aligned}
\text{(SP-Flujo)} \ \min \quad & \sum_{e \in E} \ell_e x_e \\
\text{s.a.} \quad & x(\delta^+(u)) = x(\delta^-(u)) \ \forall u \in V \backslash \{t, s\} \\
& x(\delta^+(s)) = x(\delta^-(t)) = 1 \\
& x(\delta^-(s)) = x(\delta^+(t)) = 0 \\
& x \in \{0,1\}^E
\end{aligned}$$

De esta manera existen $n+2$ restricciones lo cual es una cantidad lineal de restricciones.

Ejercicio 3. Verifique que SP-Flujo es un modelo del problema cuando $\ell_e > 0 \ \forall e$ y muestre que, en general, no es un modelo exacto.

Ejercicio 4. Modifique SP-Flujo para que sea un modelo incluso cuando ℓ pueda tomar valores nulos.

Ejercicio 5. Pruebe que SP-Flujo es equivalente a

$$\begin{aligned}
\min \quad & \sum_{e \in E} \ell_e x_e \\
\text{s.a.} \quad & x(\delta^+(u)) = x(\delta^-(u)) \ \forall u \in V \backslash \{t, s\} \\
& x(\delta^+(s)) - x(\delta^-(s)) = 1 \\
& x \in \{0,1\}^E
\end{aligned}$$

Mostraremos en el curso que el problema es posible relajarse y en vez de considerar $x \in \{0,1\}^E$ se puede transformar al problema lineal puro con la restricción $0 \leq x \leq 1$.