Investigación Operativa

Álvaro García Tenorio [1]

15 de julio de 2017

[1] alvgar14@ucm.es

Índice general

Prefacio

Estas notas son una transcripción (libremente adaptada) de las clases de la asignatura "*Investigación Operativa*", impartidas por María Inés Sobrón Fernández en el curso 2016–2017 a los cursos de cuarto y tercero de los dobles grados de Matemáticas – Física e Ingeniería Informática – Matemáticas (respectivamente) en la facultad de Ciencias Matemáticas de la Universidad Complutense de Madrid (UCM).

Cualquier aportación o sugerencia de mejora es siempre bienvenida.

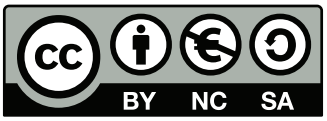
Requisitos previos

Para comprender estas notas en su totalidad es necesario tener soltura a la hora de trabajar con bases de espacios vectoriales de dimensión finita y comprender bien las aplicaciones lineales. También es bastante recomendable recordar algunos aspectos del cálculo diferencial en varias variables, no obstante, el texto es bastante autocontenido en ese aspecto.

Agradecimientos

La existencia de estas notas es debida a la amabilidad de Clara Rodríguez Núñez, quien me cedió sus apuntes tomados durante el curso, en los cuales se basa el núcleo de este texto.

Licencia



Parte I

Programación lineal

Capítulo 1

Fundamentos teóricos

1.1. Planteamiento del problema

De ahora en adelante consideraremos fijadas las bases canónicas tanto de $\mathbb{R}^n$ como de $\mathbb{R}$, a no ser que se especifique lo contrario.

Antes de plantear formalmente el problema de la programación lineal, introduzcamos unas definiciones.

Definición 1.1.1 (Vector no negativo)**.** Un vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ se dice ***no negativo*** si todas sus componentes son, como su propio nombre indica, no negativas. Es decir, $x_i \geq 0$ para todo $i \in \{1, \ldots, n\}$. Si un vector x es no negativo escribiremos $x \geq 0$.

Al hilo de esta definición, vamos a generalizar ese concepto que lleva presente en nuestras vidas desde que aquel docente de primaria nos dibujó dos rectas perpendiculares en la pizarra, los cuadrantes.

Definición 1.1.2 (2^n–ante positivo)**.** Definimos el 2^n***–ante positivo*** de $\mathbb{R}^n$ como el conjunto de los vectores no negativos de $\mathbb{R}^n$.

Se deja como ejercicio al lector justificar el nombre de 2^n–ante.

Dicho esto, ya podemos formular el problema de la programación lineal.

Problema 1.1 (Formulación general)**.** La disciplina de la ***programación lineal*** estudia procedimientos (implementables en ordenador) para calcular los extremos absolutos de una aplicación lineal $f : \mathbb{R}^n \to \mathbb{R}$ cuando restringimos su dominio a una variedad afín de $\mathbb{R}^n$ cortada con el 2^n–ante positivo.

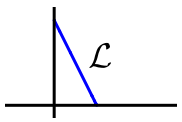


Figura 1.1: Ilustración del problema de programación lineal.

En el caso de la ilustración 1.1 suponemos que f es una aplicación lineal definida sobre todo el plano a la cual restringimos el dominio a la variedad afín $\mathcal{L}$ cortada con el cuadrante positivo (línea azul). Nuestro deber es encontrar los puntos (si los hay) sobre la línea azul en los cuales f alcanza un máximo o un mínimo absoluto.

En esta sección nos dedicaremos simplemente a formular de diversas maneras el problema de la programación lineal, observando que todas son equivalentes entre sí. Comencemos pues, sin más dilación nuestro viaje por lo desconocido, un viaje del que probablemente no regresaremos.

A continuación vemos una sencilla caracterización de las aplicaciones lineales $f : \mathbb{R}^n \to \mathbb{R}$ a partir de su expresión analítica.

Observación 1.1.1 (Expresión analítica)**.** Consideremos una aplicación lineal $f : \mathbb{R}^n \to \mathbb{R}$.

Por ser f una aplicación lineal, tendrá una única matriz asociada C. Como salta a la vista, C será una matriz fila con n columnas.

Entonces, dado un vector $x \in \mathbb{R}^n$ se tiene que

$$f(x) = f(x_1, \dots, x_n) = CX = \begin{pmatrix} c_1 & \cdots & c_n \end{pmatrix} \begin{pmatrix} x_1 \\ \cdots \\ x_n \end{pmatrix} = c_1 x_1 + \cdots + c_n x_n \tag{1.1}$$

De esta forma se concluye que toda aplicación lineal $f : \mathbb{R}^n \to \mathbb{R}$ tiene una expresión analítica de la forma de la ecuación (1.1). Recíprocamente, toda función $g : \mathbb{R}^n \to \mathbb{R}$ con una expresión analítica como la de (1.1) es una ecuación lineal, para demostrarlo, basta con leer como los árabes la ecuación (1.1). ♢

Antes de continuar es importante hacer un inciso acerca de la notación.

Observación 1.1.2 (Notación matricial)**.** Siempre que escribamos matrices fila o columna lo haremos con letras minúsculas. Asimismo, siempre que estemos escribiendo una matriz fila añadiremos el símbolo de trasposición, para dejarlo claro implícitamente. De esta manera, a las funciones lineales las denotaremos por

$$f(x_1 \dots, x_n) = c^t x$$

donde c^t es la matriz asociada a f y x es el vector columna con las coordenadas de x respecto de la base canónica (nótese el pequeño abuso de notación). ♢

1.1.1. Forma estándar y formas canónicas

Sea $f(x_1, \dots, x_n) = c_1 x_1 + \cdots + c_n x_n = c^t x$ una función lineal a la que a partir de ahora llamaremos ***función objetivo***. A la matriz c se la denomina ***vector de coeficientes de la función objetivo***, mientras que x recibe el nombre de ***vector de variables de decisión***.

Consideremos asimismo la variedad afín de $\mathbb{R}^n$ dada por el conjunto de soluciones al sistema no necesariamente homogéneo $Ax = b$. Escrito de forma desarrollada

$$\begin{cases} a_{11}x_1 & +\cdots+ & a_{1n}x_n & = b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{1m}x_1 & +\cdots+ & a_{mn}x_n & = b_m \end{cases} \qquad \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

A la matriz $A \in \mathfrak{M}_{m\times n}(\mathbb{R})$ usualmente la llamaremos ***matriz de coeficientes de las restricciones***, además, a cada una de las ecuaciones del sistema las llamaremos ***restricciones***. Asimismo el vector b será conocido como ***vector de términos independientes***.

Expongamos ahora la llamada "*formulación estándar*" del problema de la programación lineal.

Problema 1.2 (Forma estándar)**.** El problema de programación lineal en ***forma estándar*** consiste en hallar los vectores $x \geq 0$ tales que sean solución de $Ax = b$ y además sean mínimos absolutos de la función f. Usualmente se escribe de forma sintética (aunque no lo usaremos mucho)

$$\begin{array}{rc} & \min c^t x \\ \text{Sujeto a:} & Ax = b, \qquad x \geq 0 \end{array}$$

Nótese que la formulación del problema en forma estándar es exactamente la misma que la formulación general 1.1 con la única diferencia de que ya no buscamos extremos absolutos en general, sino únicamente mínimos.

La formulación estándar del problema de programación lineal puede parecer muy inflexible, en el sentido de que, a simple vista, no parece ser demasiado útil para modelizar problemas reales. Es esta aparente rigidez la que lleva a plantearse otras formulaciones más laxas del problema de programación lineal. Las llamadas "*formulaciones canónicas*" que exponemos a continuación.

Problema 1.3 (Primera forma canónica)**.** Esta nueva formulación a la que llamamos ***primera forma canónica*** plantea el problema de hallar los vectores $x \geq 0$ tales que satisfagan la ecuación $Ax \leq b$ y que además sean máximos absolutos de la función objetivo f. Escrito de forma compacta

$$\begin{aligned} &\max c^t x \\ \text{Sujeto a:} \quad &Ax \leq b, \qquad x \geq 0 \end{aligned}$$

Problema 1.4 (Segunda forma canónica)**.** El problema de programación lineal en ***segunda forma canónica*** consiste en encontrar los vectores $x \geq 0$ tales que satisfagan la ecuación $Ax \geq b$ y que además sean mínimos absolutos de la función objetivo f. En resumen

$$\begin{aligned} &\min c^t x \\ \text{Sujeto a:} \quad &Ax \geq b, \qquad x \geq 0 \end{aligned}$$

Una vez vistas todas las formulaciones hagamos una pequeña reflexión. Sería la apoteosis del tedio tener que demostrar resultados teóricos para cada una de las formulaciones. No obstante, no hay que dejar nunca de confiar en los dioses olímpicos, pues a veces son benévolos con nosotros. Esta benevolencia consiste en que todas los formulaciones son equivalentes en el sentido que veremos a continuación.

Maximización y minimización

Si se nos plantea el problema de hallar el máximo absoluto de una función objetivo f podemos transformar este problema en uno equivalente de forma que el objetivo de el nuevo problema sea hallar el mínimo de otra función objetivo g.

La utilidad de este hecho es que podemos transformar un problema de programación lineal en primera forma canónica (que surgen de forma natural a la hora de modelizar problemas reales) en un problema más parecido a la formulación estándar.

Lema 1.1.1 (Maximización–minimización)**.** *Sea A un conjunto arbitrario y una función $f : A \to \mathbb{R}$ que alcanza el máximo y el mínimo. Entonces se cumple*

$$\max f = -\min(-f)$$

Demostración. Sea ξ el máximo de f, es decir, el único número de $f(A)$ que verifica que $f(x) \leq \xi$ para todo $x \in A$. Consideremos ahora la función $g \equiv -f$, que, por las propiedades de los números reales alcanza el mínimo, al que llamaremos η.

Evidentemente η verifica que $g(x) \geq \eta$ para todo $x \in A$, o equivalentemente $-g(x) \leq -\eta$ para todo $x \in A$. Pero $-g(x) = f(x)$, luego $f(x) \leq -\eta$ para todo $x \in A$. Por unicidad del máximo $-\eta = \xi$, como queríamos demostrar. ■

Observación 1.1.3 (Aplicación)**.** Si tenemos un problema de programación lineal formulado en términos de maximización bastará con cambiar la función objetivo f por la función objetivo $-f$ para obtener un problema equivalente en términos de minimización cuyas soluciones deben ser cambiadas de signo. ♢

Igualdades y desigualdades

Una vez visto el apartado anterior, vamos a aprender a trasformar restricciones de tipo desigualdad a restricciones de tipo igualdad, con lo cual ya podremos transformar cualquier problema planteado en alguna forma canónica a un problema equivalente planteado en forma estándar.

Lema 1.1.2 (Variables de holgura)**.** *Sea una restricción de la forma*

$$a_{i1}x_1 + \cdots + a_{in}x_n \overset{(\geq)}{\leq} b_i \tag{1.2}$$

entonces la restricción

$$a_{i1}x_1 + \cdots + a_{in}x_n \overset{(-)}{+} x_i^h = b_i \tag{1.3}$$

es equivalente, en el sentido de que hay una biyección "natural" entre los conjuntos de vectores ***no negativos*** *que verifican cada una de las restricciones.*

A la variable x_i^h añadida en la restricción (1.3) *se la denomina* ***variable de holgura****.*

Demostración. Sean los conjuntos $\mathcal{S}$ y $\mathcal{S}'$ de vectores que verifican las restricciones (1.2) y (1.3) respectivamente. Consideremos la aplicación

$$\begin{array}{rc} \varphi: & \mathcal{S} \to \mathcal{S}' \\ & (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, z) \end{array}$$

donde $z = b_i - a_{i1}x_1 + \cdots + a_{in}x_n$. Veamos que es una biyección.

La inyectividad es clara por definición, si dos vectores compartieran imagen todas sus componentes coinciden.

En cuanto a la sobreyectividad, dado un vector $(x_1, \ldots, x_n, x_{n+1})$ que verifica la restricción (1.3), es claro que el vector resultante de eliminar la última componente, $(x_1, \ldots, x_n)$, cumple la restricción (1.2).

La demostración para el caso $\geq$ es totalmente simétrica. ■

Veamos que utilidad tiene lo que acabamos de demostrar.

Observación 1.1.4 (Aplicación)**.** Si a la hora de modelizar un problema nos aparecen restricciones de tipo $\geq$ o $\leq$ basta con sustituirlas por restricciones equivalentes añadiendo variables de holgura.

Además, debemos cambiar la función objetivo f, extendiéndola de la siguiente manera

$$\begin{array}{rc} \widehat{f}: & \mathbb{R}^{n+m} \to \mathbb{R} \\ & (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}) \mapsto f(x_1, \ldots, x_n) \end{array}$$

donde n es el número total de restricciones originales y m el número de restricciones de tipo $\geq$ o $\leq$. De esta forma obtenemos un problema equivalente cuya solución será el vector solución del nuevo problema quitando las componentes correspondientes a variables de holgura (¡compruébese!). ♢

Con lo que sabemos hasta el momento podemos transformar cualquier problema escrito en cualquiera de las formas canónicas en un problema en forma estándar.

Positividad y negatividad

Otro caso que se presenta de manera usual es que al modelizar un problema concreto sea necesario permitir que algunas componentes de los vectores candidatos a ser solución sean negativas. Esta es una situación que no contempla la formulación estándar, sin embargo, con un simple truco podemos meter esta clase de problemas dentro de nuestro marco de actuación.

Observación 1.1.5 (Truco)**.** Si tenemos un problema en el que la componente i–ésima de los vectores puede ser negativa tomamos todas las restricciones

$$a_{j1}x_1 + \cdots + a_{ji}x_i + \cdots + a_{jn}x_n = b_j$$

que involucren a la componente x_i y las sustituimos por las restricciones

$$a_{j1}x_1 + \cdots + a_{ji}(x_i^+ - x_i^-) + \cdots + a_{jn}x_n = b_j$$

con x_i^+ y x_i^- no negativos.

Es claro que si un vector $(x_1, \ldots, x_i, \ldots, x_n)$ (donde x_i es negativo) verifica las restricciones originales, entonces el vector $(x_1, \ldots, 0, -x_i, \ldots, x_n)$ (por ejemplo) verifica las restricciones nuevas. Análogamente si x_i es positivo. Asimismo, si el vector $(x_1, \ldots, x_i^+, x_i^-, \ldots, x_n)$ cumple las restricciones nuevas, el vector $(x_1, \ldots, x_i^+ - x_i^-, \ldots, x_n)$ cumplirá las viejas, pudiendo concluir así que las restricciones viejas y nuevas son equivalentes en un sentido un poco más laxo que en el apartado anterior, ya que únicamente se tiene la sobreyectividad (pero esto es irrelevante).

De esta forma, si cambiamos la función objetivo f por la función

$$\begin{array}{rc} \overline{f}: & \mathbb{R}^{n+1} \to \mathbb{R} \\ & (x_1 \ldots, x_i^+, x_i^-, \ldots, x_n) \mapsto f(x_1, \ldots, x_i^+ - x_i^-, \ldots, x_n) \end{array}$$

obtenemos un problema equivalente cuya solución será el vector $(x_1, \ldots, x_i^+ - x_i^-, \ldots, x_n)$ donde $(x_1 \ldots, x_i^+, x_i^-, \ldots, x_n)$ es el vector solución del problema con las restricciones nuevas. ♢

Hechas todas estas disquisiciones iniciales cabe mencionar que a la hora de modelizar problemas, siempre que tanto la función objetivo como las restricciones sean lineales, podremos realizar transformaciones (las vistas en esta sección) para que el problema quede planteado en forma estándar, pudiendo ser así resuelto mediante el uso de toda la artillería teórica que veremos a continuación.

1.1.2. Terminología y casuística

Dado un problema de programación lineal en forma estándar que tiene a A como matriz de coeficientes de las restricciones y a b como vector de términos independientes. Exponemos las siguientes definiciones.

Definición 1.1.3 (Soluciones)**.** Llamamos ***solución*** del problema P a cualquier vector $x \in \mathbb{R}^n$ (no necesariamente no negativo) que verifique $Ax = b$.

Sacando un poco de punta al asunto, diremos que si $x \in \mathbb{R}^n$ es solución del problema y además $x \geq 0$ entonces x es una ***solución factible***.

Llamaremos ***región factible*** al conjunto de todas las soluciones factibles del problema. Normalmente la denotaremos con la letra $\mathcal{S}$.

Una vez resuelto el problema se pueden dar las siguientes situaciones

- La región factible posee al menos un punto en el cual la función objetivo alcanza el mínimo. En esta situación diremos que el problema posee ***solución óptima***.
- La región factible del problema es el conjunto vacío. En este caso se dirá que el problema es ***infactible***.
- La función objetivo alcanza valores tan bajos como queramos en la región factible. En este caso se dice que el problema posee una solución ***no acotada***.

Teniendo en cuenta estas pequeñas observaciones podemos empezar a estudiar cómo son las regiones factibles, lo cual nos será muy útil para resolver el problema de programación lineal.

1.2. Programación lineal y conjuntos convexos

En esta sección demostraremos que las regiones factibles de un problema de programación lineal en forma estándar son siempre conjuntos convexos.

Asimismo, estudiaremos en profundidad las propiedades de estos conjuntos, explotándolas para obtener un procedimiento robusto con el cual resolver estos problemas.

1.2.1. Conjuntos convexos. Definición y propiedades

Comenzamos definiendo el concepto de conjunto convexo.

Definición 1.2.1 (Convexos)**.** Sea $\mathcal{S}$ un subconjunto de $\mathbb{R}^n$, se dice ***convexo*** si para cada par de puntos $x, y \in \mathcal{S}$ el segmento que tiene por extremos a x e y está contenido en $\mathcal{S}$. Es decir

$$\lambda x + (1 - \lambda)y \in \mathcal{S}$$

para todo $\lambda \in [0, 1]$. Nótese que el conjunto vacío se considera convexo.

Una cuestión importante a destacar es que la intersección arbitraria de conjuntos convexos es convexo, tal y como muestra el siguiente lema.

Lema 1.2.1 (Intersección)**.** *Sea $\mathcal{F}$ una familia de conjuntos convexos. La intersección de la familia es un conjunto convexo.*

Demostración. Sean x e y dos puntos de la intersección de la familia, luego x e y están en todos los conjuntos de la familia. Como estos conjuntos son convexos, el segmento que une x e y estará contenido en todos los conjuntos de la familia simultáneamente, luego también en la intersección de la familia. ■

Un concepto por el que merece la pena pasar es el de "envoltura convexa" de un conjunto $\mathcal{S}$ de $\mathbb{R}^n$, que viene a ser el menor conjunto convexo que contiene a $\mathcal{S}$.

Esta idea es especialmente recurrente en las matemáticas, por ejemplo, en álgebra lineal teníamos el subespacio generado por un subconjunto, en teoría de grupos el subgrupo generado por un subconjunto, en topología tenemos la adherencia de un conjunto,... La lista es interminable.

Observación 1.2.1 (Justificación)**.** Es fácil darse cuenta de que, como la intersección arbitraria de convexos es convexa, el menor conjunto convexo que contiene a uno dado (al que llamaremos $\mathcal{S}$) puede ser construido de la siguiente manera

$$\mathrm{Conv}(\mathcal{S}) := \bigcap_{\mathcal{F} \supset \mathcal{S}} \mathcal{F}$$

En efecto, es un conjunto que contiene a $\mathcal{S}$, ya que todos los conjuntos de la familia a intersecar contienen a $\mathcal{S}$, además, es el menor de ellos, ya que, de haber uno más pequeño, pertenecería a la familia que se está intersecando, lo cual es absurdo (¡compruébese!).

De esta forma queda justificada la existencia del menor conjunto convexo que contiene a $\mathcal{S}$, no obstante, la construcción que hemos realizado nos da escasa información acerca de los elementos de la envoltura convexa. ♢

Veamos a continuación una caracterización útil de la envoltura convexa de un conjunto, para la cual necesitamos introducir una definición.

Definición 1.2.2 (Combinaciones convexas)**.** Sea $\{z_1, \ldots, z_r\}$ un conjunto finito de puntos. Diremos que z es ***combinación lineal convexa*** de dichos puntos si

$$z = \lambda_1 z_1 + \cdots + \lambda_r z_r$$

donde $\lambda_i > 0$ para $i \in \{1, \ldots, r\}$ y se cumple que $\sum_{i=1}^{r} \lambda_i = 1$.

Proposición 1.2.2 (Caracterización)**.** *Dado un conjunto $\mathcal{S}$, su envoltura convexa es el conjunto de puntos que son una combinación lineal convexa de puntos de $\mathcal{S}$.*

Demostración. Llamemos $\mathcal{H}(\mathcal{S})$ al conjunto de puntos que son combinación lineal convexa de puntos de $\mathcal{S}$. Veamos en primer lugar que $\mathcal{H}(\mathcal{S})$ contiene a $\mathcal{S}$. Esto es claro, ya que todo punto x de $\mathcal{S}$ es combinación lineal convexa de puntos de $\mathcal{S}$, en efecto, $x = 1 \cdot x$.

Veamos ahora que $\mathcal{H}(\mathcal{S})$ es convexo. ■

1.2.2. Puntos extremos

1.2.3. Direcciones extremas

1.3. Teoremas fundamentales

1.4. Algoritmo del Símplex

1.4.1. Fundamentos teóricos

1.4.2. Recapitulación

1.5. Ejercicios resueltos

Ejercicio 1.1. contenidos...

Solución 1.1. contenidos... □

Capítulo 2

Algoritmo del Símplex

2.1. Implementación del algoritmo del símplex

2.2. Método de las dos fases

2.3. Método de las penalizaciones

2.4. Regla lexicográfica

2.5. Regla de Bland

Capítulo 3

Dualidad

3.1. Planteamiento del problema dual

3.2. Relaciones de dualidad

Parte II

Programación entera

Capítulo 4

Modelización

4.1. Modelización con variables binarias

4.2. Restricciones disyuntivas

4.3. Problemas con costes fijos

Capítulo 5

Ramificación y acotación

Parte III

Programación no lineal

Parte IV

Anexos

Índice de términos