



UNIVERSITAS
Miguel Hernández

ANÁLISIS DE ESTABILIDAD EN OPTIMIZACIÓN LINEAL

Manuel López Romero

Tutorizado por Maria Josefa Cánovas

Departamento de Estadística,
Matemáticas e Informática

Facultad de Ciencias Sociales y
Jurídicas de Elche

4º Curso. Estadística Empresarial

Trabajo Fin de Grado

Junio 2022



Universidad Miguel Hernández de Elche

DEPARTAMENTO DE ESTADÍSTICA,
MATEMÁTICAS E INFORMÁTICA

ANÁLISIS DE
ESTABILIDAD EN
OPTIMIZACIÓN LINEAL



Trabajo Fin de Grado

Manuel López Romero
Tutorizado por Lola Cánovas

Índice general

Abstract	5
Introducción	7
1. Problemas consistentes y resolubles	9
1.1. Introducción a la Programación Lineal	9
1.1.1. Notas históricas	9
1.1.2. Planteamiento genérico de un problema de Programación Lineal	10
1.2. Consistencia de un sistema	11
1.2.1. Caracterización de la consistencia	11
1.2.2. Estudio de la consistencia para implementación en MATLAB	13
1.3. Resolubilidad de un problema	18
1.3.1. Caracterización de la resolubilidad	18
1.3.2. Estudio de la resolubilidad para implementación en MATLAB	20
2. Estabilidad del problema	25
2.1. Notación y elementos básicos	25
2.2. Estabilidad en términos de la consistencia	27
2.3. Estabilidad en términos de la resolubilidad	31

3. Aplicación en un entorno industrial	39
3.1. Planteamiento inicial	39
3.1.1. La aplicación de la estabilidad	39
3.1.2. Ejemplo numérico en un entorno de industria ali- mentaria	40
3.1.3. Modelización del problema	41
3.2. Resolución en MATLAB	43
A. Programas I	47
B. Programas II	49
Bibliografía	53



Abstract

English

In the present project, linear optimization basic properties and their associated stability are studied in three chapters. First, an analytical proposal is made for the systematized study of consistency and solvability in a linear optimization problem, to later apply it in a MATLAB-language-based original program. Secondly, the stability of the properties previously mentioned is considered so as to study it and develop an implementable system that gives measure of stability in terms of consistency and solvability of any linear optimization problem, and thus create an automatized program that checks it. The last part corresponds to a real industrial-based linear problem, its mathematical modeling and its subsequent resolution with the developed programs.

Español

En el presente proyecto se estudian las propiedades básicas de la optimización lineal y su estabilidad asociada. Se realiza una propuesta analítica para el estudio sistematizado de la consistencia y la resolubilidad en un problema de optimización lineal, para luego aplicarlo en un programa original en MATLAB. Posteriormente se considera la estabilidad de las propiedades anteriormente mencionadas para desarrollar un sistema implementable que de una medida de esta en términos de consistencia y resolubilidad, y así crear un programa automatizado que la compruebe. La última parte corresponde a un problema lineal industrial, su modelización matemática y su posterior resolución.

Introducción

La optimización lineal es una disciplina que aunque relativamente reciente, avanza constantemente y constituye un enorme campo dentro de la investigación de operaciones. Esta se puede definir como el campo de la programación matemática que se dedica a maximizar o minimizar una función lineal sobre un conjunto de soluciones factibles determinado por una colección de restricciones lineales. Para un estudio más exhaustivo, se pueden consultar varias de las fuentes que constituyen literatura base en el campo, como por ejemplo [1][2] y [7].

En el presente proyecto se estudian a través de tres capítulos las propiedades de consistencia y resolubilidad de un problema de optimización lineal, un concepto de estabilidad asociado a estas y una aplicación en un entorno industrial.

En el primer capítulo se da una definición formal de programación lineal, además de unas notas históricas que explican su desarrollo preliminar hasta su nacimiento como tal en la XX Guerra Mundial (dichas notas han sido extraídas de [4][6] y [9]). A continuación, se define la consistencia de un sistema, es decir, el carácter no vacío del conjunto de puntos que satisfacen la colección de restricciones lineales dada, para posteriormente dar una caracterización formal de la misma, y así usar dicha caracterización con el fin de implementarla en MATLAB; de este modo obtenemos un método que nos permite evaluar analítica y sistemáticamente la consistencia de un problema dado. De manera análoga, se define en el mismo capítulo la resolubilidad (informalmen-

te, el hecho de que el problema tenga solución óptima) y se procede de manera similar a la efectuada con la consistencia [5].

En el segundo capítulo, se estudian los problemas de programación lineal en relación a su estabilidad, introduciendo conceptos más generales como el espacio paramétrico y diversos subconjuntos de este, que serán de utilidad posterior para analizar la estabilidad del problema respecto a sus propiedades básicas: la consistencia y la resolubilidad. Para ello, se hace uso de resultados previos obtenidos en investigación en este área [3], los cuales servirán para hallar mediante un programa el grado de estabilidad, es decir, el margen de variación de los parámetros iniciales de un problema sin que su resolubilidad se vea afectada. Para la correcta implementación, se define un método sistemático para el estudio de la estabilidad respecto a la consistencia, y posteriormente el análogo respecto a la resolubilidad.

En el tercer y último capítulo, se plantea y resuelve un problema basado en el mundo real (inspirado en un ejercicio de [8]), concretamente en el ámbito de la industria alimentaria. Primeramente, se da el planteamiento del problema y los datos asociados al mismo, para seguidamente modelizarlo matemáticamente. Tras ello, se lleva a cabo la resolución del modelo en MATLAB, obteniendo resultados en cuanto a su consistencia, resolubilidad, valor óptimo y estabilidad, así como su interpretación.

La contribución original al proyecto supone la creación de 4 programas en MATLAB que automatizan las implementaciones vistas en los capítulos 1 y 2, las cuales pueden comprobarse en el apartado Anexos, al final del presente libro.

Capítulo 1

Problemas consistentes y resolubles

1.1. Introducción a la Programación Lineal

1.1.1. Notas históricas

La Programación Lineal (alternativamente conocida como Optimización Lineal) es el campo de la Programación Matemática que se dedica a maximizar o minimizar una función lineal sobre un conjunto de soluciones factibles determinado por una colección de restricciones lineales. Esta tiene sus precedentes directos en Joseph Fourier y Carl Friedrich Gauss, quienes hicieron aportaciones sustanciales al campo de la resolución de sistemas lineales de inecuaciones, siendo Farkas Bolyai quien posteriormente concibe un método para resolver sistemas de inecuaciones.

En sus inicios, la Programación Lineal (de aquí en adelante, PL) se dedicaba al estudio de modelos matemáticos desarrollados durante la 2ª Guerra Mundial con el fin de planificar gastos y así minimizar los costes al ejército. Fue mantenida en secreto hasta 1947, año en el cual George Dantzig publicó el algoritmo SIMPLEX y John von Neumann

desarrolló la teoría de dualidad, dando origen al inicio de la PL tal como la conocemos hoy día.

1.1.2. Planteamiento genérico de un problema de Programación Lineal

Consideramos un problema (P) de PL escrito en la forma:

$$(P) \quad \text{Min} \quad c'x$$

$$\text{s.a} \quad a'_j x \leq b_j \quad j = 1, \dots, m,$$

donde:

- x representa el vector de variables, $x \in \mathbb{R}^n$ (n es el número de variables),
- $c \in \mathbb{R}^n$, representa el vector de coeficientes asociado a la función objetivo,
- $a_j \in \mathbb{R}^n$ es el vector de coeficientes asociado a cada una de las restricciones, con $j = 1, \dots, m$,
- los vectores en \mathbb{R}^n serán considerados por defecto como vectores columna; de este modo, dado cualquier $y \in \mathbb{R}^n$, y' representa su traspuesto.

Cabe destacar que el problema de PL se presenta en forma de minimización y en términos de desigualdades 'menor o igual que' únicamente para estandarizar la notación. Nótese que este modelo no supone una pérdida de generalidad, ya que $a = b \iff a \leq b, a \geq b$ y toda desigualdad de tipo 'mayor o igual que' puede ser expresada como una desigualdad 'menor o igual que' al revertir los signos de ambos miembros ($a \geq b \iff -a \leq -b$). Además, hallar el máximo de una función f equivale a minimizar su opuesta $-f$.

Denotaremos por F al conjunto factible de (P) es decir, el conjunto de puntos en el espacio ambiente \mathbb{R}^n tales que satisfacen todas las desigualdades del sistema de restricciones asociado; formalmente:

$$F = \{x \in \mathbb{R}^n : a'_j x \leq b_j; j = 1, \dots, m\}.$$

Definimos el valor óptimo de (P) , denotado por v , como el ínfimo de los valores de la función objetivo sobre el conjunto de puntos factibles. Formalmente:

$$v := \inf \{c'x : x \in F\}.$$

El conjunto óptimo de (P) , denotado por S , queda definido como el conjunto de puntos $x \in F$ cuya imagen en la función objetivo es el valor óptimo de (P) , esto es:

$$S := \{x \in F : c'x = v\}.$$



1.2. Consistencia de un sistema

1.2.1. Caracterización de la consistencia

Sea σ el sistema de desigualdades lineales dado por:

$$\sigma = \{a'_j x \leq b_j; j = 1, \dots, m\}.$$

Decimos que un sistema σ es consistente si F es no vacío. A lo largo de esta sección caracterizaremos la consistencia de σ en términos de los propios coeficientes del sistema¹. Para ello necesitaremos hacer uso de los conceptos de *cono* y *cono convexo generado por un conjunto*.

Un conjunto no vacío $C \subset \mathbb{R}^n$ es un cono si para cualquier $x \in C$ y cualquier $\lambda \geq 0$ se satisface $\lambda x \in C$. El cono convexo generado por

¹Estos son los datos que introduciremos en el programa que chequee la consistencia en MATLAB.

un conjunto $X \subset \mathbb{R}^n$, el cual denotaremos por $\text{cone}(X)$, está formado por todas las combinaciones lineales finitas de todos los elementos de X con escalares no negativos, esto es:

$$\text{cone}(X) = \left\{ \sum_{i=1}^k \lambda_i x^i : x^i \in X, \lambda_i \geq 0, i = 1, \dots, k, k \in \mathbb{N} \right\}.$$

Nótese que distinguimos los diferentes elementos de X mediante superíndices x^i para evitar confusión respecto a las coordenadas de los mismos, que usan el subíndice x_i . Suponemos además que $\text{cone}(\emptyset) = \{0_n\}$ siendo 0_n el vector nulo de \mathbb{R}^n . En este apartado emplearemos el cono generado por el conjunto de vectores de coeficientes del sistema σ , el cual denotaremos por N ; formalmente:

$$N = \text{cone} \left\{ \begin{pmatrix} a_j \\ b_j \end{pmatrix}, j = 1, \dots, m \right\} \subset \mathbb{R}^{n+1}.$$

Teorema 1. Sea $\sigma = \{a'_j x \leq b_j; j = 1, \dots, m\}$ y N el cono generado por sus vectores de coeficientes. El sistema es **consistente** si y sólo si:

$$\begin{pmatrix} 0_n \\ -1 \end{pmatrix} \notin N.$$

El siguiente ejemplo ilustra la caracterización establecida en el teorema anterior.

Ejemplo 1. Sea el sistema de desigualdades:

$$\left. \begin{array}{l} x_1 \leq 0 \\ -x_1 \leq -1 \end{array} \right\}$$

El cono generado por sus coeficientes es $N = \text{cone} \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\}$

Es fácil observar que el conjunto factible es vacío, debido a que ningún punto satisface ambas condiciones. Mediante cálculos elementales

en las combinaciones entre desigualdades puede llegarse a un absurdo, como por ejemplo al multiplicar por $\frac{1}{2}$ en ambos miembros de las desigualdades y sumar:

$$\begin{aligned} \frac{1}{2}x_1 &\leq \frac{1}{2} * 0 \\ -\frac{1}{2}x_1 &\leq -\frac{1}{2} \end{aligned}$$

$$0 \leq -\frac{1}{2}$$

Lo cual es una contradicción, traduciéndose en que el vector $(0_n, -1)'$ (en este caso $n=1$, por lo cual solo es un 0) es obtenido como combinación lineal con escalares no negativos de los vectores de coeficientes del sistema, estando por tanto contenido en el cono generado por estos. En otros términos:

$$\begin{pmatrix} 0 \\ -1 \end{pmatrix} \in \text{cone} \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\}.$$

1.2.2. Estudio de la consistencia para implementación en MATLAB

Fundamentos en términos de distancia

El objetivo, una vez que la consistencia ha sido caracterizada, es encontrar una vía sistemática en la cual podamos determinar analíticamente si el vector $(0_n, -1)'$ pertenece al cono N . Para resolver esta cuestión, determinaremos la distancia de $(0_n, -1)'$ a N , la cual denotaremos como $d\left((0_n, -1)', N\right)$ de aquí en adelante. Dado que N es cerrado, se tiene que:

$$\begin{pmatrix} 0_n \\ -1 \end{pmatrix} \in N \iff d\left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix}, N\right) = 0.$$

Podemos, por tanto, plantear un problema de optimización basado en este concepto, asociado a los coeficientes del sistema de desigualdades

(restricciones) de nuestro problema de Programación Lineal original.

La distancia entre el vector $(0_n, -1)'$ y N se define como:

$$d\left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix}, N\right) = \min \left\{ d\left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix}, \begin{pmatrix} u \\ v \end{pmatrix}\right), \begin{pmatrix} u \\ v \end{pmatrix} \in N \right\},$$

donde *min* significa 'mínimo'. Nótese que en principio se tendría que escribir en términos de ínfimo (*inf*), sin embargo teniendo en cuenta de nuevo que N es cerrado, sabemos que la distancia se alcanza en un punto de N .

Planteamiento y desarrollo del modelo.

Tratamos entonces de encontrar la distancia entre el vector $(0_n, -1)'$ y N . Si esta es 0, sabremos que el vector está contenido en el cono generado por los vectores de coeficientes del sistema de restricciones. Para resolverlo, definimos primero las variables $u = (u_i)_{i=1, \dots, n} \in \mathbb{R}^n$ y $v \in \mathbb{R}$. El planteamiento del problema de optimización asociado a la consistencia y el cual denotaremos por P_c es:

$$(P_c) \quad \text{Min} \quad d\left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix}, \begin{pmatrix} u \\ v \end{pmatrix}\right) \\ \text{s.a} \quad \begin{pmatrix} u \\ v \end{pmatrix} \in N,$$

Para poder implementarlo correctamente es necesario expresar el problema en términos de los coeficientes y de una distancia matemática. Considerando la distancia euclídea, tenemos:

$$d\left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix}, \begin{pmatrix} u \\ v \end{pmatrix}\right) = \sqrt{(u_1 - 0)^2 + \dots + (u_n - 0)^2 + (v - (-1))^2},$$

y que coincide con:

$$\sqrt{\sum_{i=1}^n u_i^2 + (v + 1)^2}.$$

Respecto a las restricciones, la pertenencia del vector $(u, v)'$ a $N = \{(a_j, b_j)', j = 1, \dots, m\}$ puede expresarse :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \sum_{j=1}^m \lambda_j \begin{pmatrix} a_j \\ b_j \end{pmatrix}, \lambda_j \geq 0.$$

El problema queda planteado así:

$$\begin{aligned} (P_c) \quad & \text{Min} \quad \sqrt{\sum_{i=1}^n u_i^2 + (v+1)^2} \\ & \text{s.a} \quad \begin{pmatrix} u \\ v \end{pmatrix} = \sum_{j=1}^m \lambda_j \begin{pmatrix} a_j \\ b_j \end{pmatrix} \\ & \quad \lambda_j \geq 0 \end{aligned}$$

Además, al ser la función raíz cuadrada monótona y estrictamente creciente en el intervalo $[0, +\infty[$, podemos efectuar la transformación cuadrática de la función para simplificar la resolución del problema. Sabemos por conocimientos previos en materia de optimización que el problema anterior será equivalente al siguiente, es decir, sus conjuntos de soluciones óptimas coinciden, aunque puede variar su valor óptimo. Si denominamos al nuevo problema P_c^* , tenemos:

$$\begin{aligned} (P_c^*) \quad & \text{Min}_{u,v} \quad \sum_{i=1}^n u_i^2 + (v+1)^2 \\ & \text{s.a} \quad \begin{pmatrix} u \\ v \end{pmatrix} = \sum_{j=1}^m \lambda_j \begin{pmatrix} a_j \\ b_j \end{pmatrix} \\ & \quad \lambda_j \geq 0 \end{aligned}$$

Dimensionalización del modelo

Para poder implementar el problema computacionalmente primero es necesario definirlo y dimensionalizarlo correctamente. Consideremos el problema de Programación Cuadrática P_c^* en forma adaptada:

$$(P_c^*) \quad \text{Min} \quad \frac{1}{2}x'H_cx + f'_c x$$

$$\text{s.a} \quad A_c x = b_c$$

$$\lambda \geq 0_m,$$

donde:

- x es el vector de variables $x = (u, v, \lambda)'$, con $u = (u_i)_{i=1, \dots, n} \in \mathbb{R}^n$, $\lambda = (\lambda_j)_{j=1, \dots, m} \in \mathbb{R}^m$, y $v \in \mathbb{R}^n$,
- H_c es una matriz cuadrada simétrica asociada a los coeficientes de la componente no lineal de la función objetivo de P_c^* ,
- $f_c \in \mathbb{R}^n$ representa el vector de coeficientes asociado a la componente lineal de la función objetivo de P_c^* ,
- A_c y b_c son la matriz de los coeficientes y el vector de términos independientes asociados a las restricciones lineales de P_c^* .

Para la dimensionalización de H_c en el problema de consistencia, tenemos en cuenta las variables del problema, las cuales son: u_i con $i = 1, \dots, n$, v , y λ_j con $j = 1, \dots, m$. Con lo cual, el número de columnas y de filas de H_c será $n + 1 + m$, que es el número total de variables. Es destacable decir que los únicos elementos no nulos de H_c son los asociados a los cuadrados de las variables u_i y v . Formalmente la matriz quedaría:

$$H_c = \begin{pmatrix} \mathbb{I}_{n+1} & 0_{(n+1) \times m} \\ 0_{m \times (n+1)} & 0_{m \times m} \end{pmatrix} \in \mathbb{R}^{(n+1+m) \times (n+1+m)},$$

donde \mathbb{I}_k representa a la matriz identidad de orden $k \in \mathbb{N}$ y la matriz $0_{n \times s} \in \mathbb{R}^{k \times s}$ la matriz nula de orden $k \times s$.

Asimismo, la parte lineal de la función objetivo, llamada f_c , tras el desarrollo de la misma, se corresponde con el 2 en la posición de la variable v , que surge de $(v + 1)^2 = v^2 + 2v + 1$ ², y el cual está en la posición $n + 1$, siendo el resto coeficientes nulos:

$$f_c = \begin{pmatrix} 0_n \\ 2 \\ 0_m \end{pmatrix} \in \mathbb{R}^{n+1+m}.$$

La matriz de coeficientes de las restricciones, A_c , queda constituida como:

$$A_c = \left(\mathbb{I}_{(n+1)}, \begin{pmatrix} -A' \\ -b' \end{pmatrix} \right) \in \mathbb{R}^{(n+1) \times (n+1+m)},$$

donde

- A representa a la matriz de coeficientes de las restricciones del problema original.
- b representa el vector de términos independientes de las restricciones del problema original.

El vector de términos independientes (b_c) de las restricciones en el problema asociado a la consistencia será el vector nulo de dimensión $n + 1$:

$$b_c = 0_{n+1}.$$

Es útil saber a expensas del posterior coste computacional, y en términos del valor óptimo de este problema, que no es necesario efectuar la sustitución de la solución de P_c^* en la función objetivo de P_c , ya que la interpretación del valor de esta es puramente dicotómica: el valor es

²El término independiente de la función lineal, 1, no es tenido en cuenta a la hora de resolver el problema. Este será sumado posteriormente al valor obtenido por el problema.

o no es nulo. Llamemos v y v^* a los valores de los problemas P_c y P_c^* respectivamente. Sabiendo que $v^* = 0 \iff v = 0$ (ya que $0 = \sqrt{0}$) bastará el resultado que arroje P_c^* para finalizar el estudio de la consistencia.

El desarrollo del programa que chequea la consistencia en MATLAB puede ser encontrado en el apartado 'Anexo'. Este constituye parte de la contribución original al presente proyecto, y se sirve únicamente de los coeficientes de un problema de optimización lineal para dar un diagnóstico en términos de consistencia.

1.3. Resolubilidad de un problema

1.3.1. Caracterización de la resolubilidad

Sea (P) , definido en el apartado 1.1.2. el problema de PL a estudiar, decimos que este es resoluble si su conjunto de soluciones óptimas es no vacío:

$$(P) \text{ Resoluble} \iff S \neq \emptyset.$$

Paralelamente al estudio de la consistencia, necesitamos poder caracterizar la resolubilidad en términos de los coeficientes de (P) . Para ello haremos uso de un importante resultado en PL.

Teorema 2. *Sea (P) un problema de PL en la forma*

$$(P) \quad \begin{aligned} \text{Min} \quad & c'x \\ \text{s.a} \quad & a'_j x \leq b_j \quad j = 1, \dots, m \end{aligned}$$

*Se tiene que (P) es **resoluble** si y solo si este es consistente y:*

$$-c \in \text{cone} \{a_j, j = 1, \dots, m\}.$$

Ejemplo 2. Sea el problema (P):

$$(P) \quad \begin{aligned} \text{Min} \quad & x_1 + x_2 \\ \text{s.a} \quad & -x_1 + x_2 \leq 0, \\ & -x_2 \leq 0. \end{aligned}$$

El vector director asociado a la función objetivo es $c = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, mientras que el cono generado por los coeficientes es $\text{cone} \left\{ \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}$.

Es fácil ver gráficamente (gráfico 1), donde el área sombreada y las rectas discontinuas representan la región factible y las curvas de nivel de la función objetivo, respectivamente, la solución del problema, que es alcanzada en el punto $x = (0, 0)$:

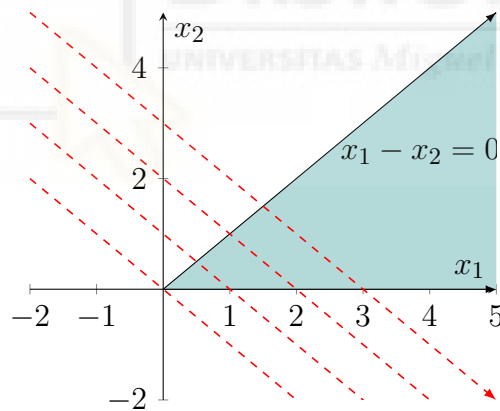


Gráfico 1. Fuente: elaboración propia.

Puede comprobarse (gráfico 2) como el opuesto del vector director de la función objetivo, $-c = (-1, -1)'$ puede ser expresado como combinación lineal con escalares no negativos de los coeficientes asociados a ambas restricciones, esto es, está contenido en su cono. En sombreado, el cono generado por los vectores de coeficientes de las restricciones:

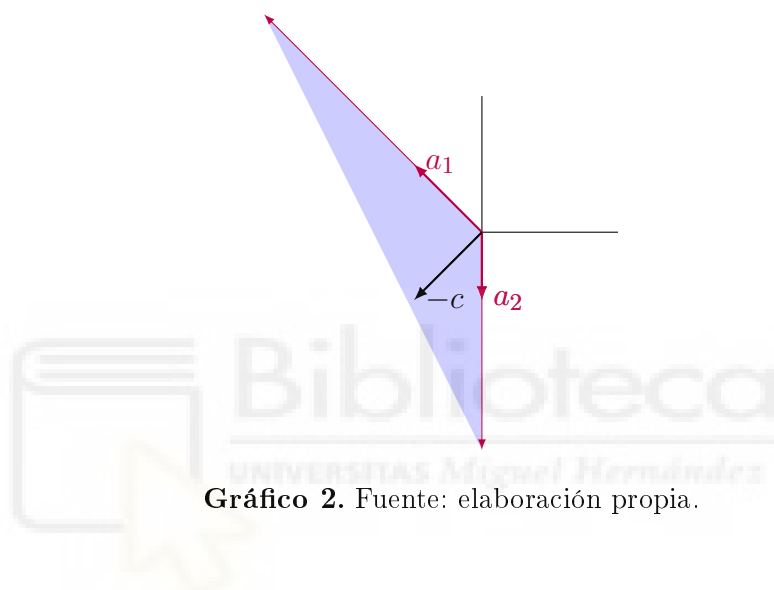


Gráfico 2. Fuente: elaboración propia.

1.3.2. Estudio de la resolubilidad para implementación en MATLAB

Planteamiento y desarrollo del modelo

En base a los mismos fundamentos que en el caso de la consistencia (véase apartado 1.3.1 y 1.3.2), buscamos la distancia mínima entre el opuesto del vector director asociado a la función objetivo y el cono generado por los coeficientes de las restricciones. El desarrollo del modelo será similar al caso estudiado en la sección 1.3, teniendo las únicas diferencias en la dimensionalización del problema y la interpretación del valor óptimo.

Así, plantearemos buscar la mínima distancia de $-c$ a cualquier punto z , que esté contenido en $M = cone \{a_j, j = 1, \dots, m\}$.

Es decir, si llamamos a este nuevo problema (P_r) :

$$(P_r) \quad \text{Min} \quad d(-c, z) \\ \text{s.a} \quad z \in M.$$

Por desarrollo similar al problema asociado a la consistencia, tenemos, considerando la distancia euclídea:

$$d(-c, z) = \sqrt{\sum_{i=1}^n (-c_i - z_i)^2} = \sqrt{\sum_{i=1}^n (c_i + z_i)^2}.$$

De manera análoga a la sección 1.3.2, expresamos la condición de $z \in M$ obligando a que la variable z sea combinación lineal de los coeficientes de las restricciones del problema original, además de efectuar la transformación cuadrática en la función objetivo y desarrollar el cuadrado para simplificar la resolución del problema³. Obtenemos así el nuevo problema (P_r^*) que será equivalente a (P_r) :

$$(P_r^*) \quad \text{Min} \quad \sum_{i=1}^n (z_i^2 + 2c_i z_i) \\ \text{s.a} \quad z_i = \sum_{j=1}^m \lambda_j a_{ij} \\ \lambda_j \geq 0$$

Dimensionalización del modelo

El problema de distancias que surge es nuevamente un problema de Programación Cuadrática en forma adaptada idéntica a la ya definida en el punto 1.3.3, cambiando todo subíndice del problema de consistencia " c " por su contraparte en el problema de resolubilidad,

³Cabe señalar que al ser c_i parámetros conocidos a priori, se pueden suprimir los términos c_i^2 del modelo al actuar estos como términos independientes, para ser posteriormente agregados al valor obtenido.

22CAPÍTULO 1. PROBLEMAS CONSISTENTES Y RESOLUBLES

" r ". En este caso, nuestro vector de variables será $x = (z, \lambda)'$, con $z = (z_i)_{i=1, \dots, n} \in \mathbb{R}^n$ y con $\lambda = (\lambda_j)_{j=1, \dots, m} \in \mathbb{R}^m$. En el caso de nuestro problema (Res^*), se expondrá a continuación por qué están constituidos sus elementos, así como sus dimensiones.

La matriz H_r contiene coeficientes no nulos tan solo en aquellos correspondientes a las variables z_i . Dado que el número de variables total es $n + m$, ya que $z \in \mathbb{R}^n$, $\lambda_j \in \mathbb{R}$ con $j = 1, \dots, m$, las dimensiones de H_r serán $(n + m) \times (n + m)$, siendo su constitución:

$$H_r = \begin{pmatrix} \mathbb{I}_n & 0_{n \times m} \\ 0_{m \times n} & 0_{m \times m} \end{pmatrix} \in \mathbb{R}^{(n+m) \times (n+m)}.$$

La componente lineal de la función objetivo, llamémosla f_r , tiene dimensiones iguales al número de variables, esto es, $n + m$, siendo:

$$f_r = \begin{pmatrix} 2c_1 \\ \vdots \\ 2c_n \\ 0_m \end{pmatrix} \in \mathbb{R}^{n+m}.$$

La matriz A_r , asociada a las restricciones, queda definida por los coeficientes de las restricciones del problema original traspuestos (es decir, A'), así como por la opuesta de la identidad de orden n , al haber un λ_j diferente en cada restricción, así:

$$A_r = \begin{pmatrix} -\mathbb{I}_n & A' \end{pmatrix} \in \mathbb{R}^{n \times (n+m)}.$$

El vector de términos independientes es el vector nulo de orden n :

$$b_r = 0_n.$$

En cuanto a la interpretación del valor entre (P_r) y (P_r^*), cabe destacar que al tratarse de una función de distancia, la interpretación será idéntica que en el caso de la consistencia: bastará interpretar el resultado que arroje (P_r^*) sin necesidad de sustitución en la función de (P_r)

y el coste computacional que conllevaría. El único cambio sustancial recae en que la distancia 0 (es decir $v = 0$) conlleva la afirmación de la resolubilidad en el problema original, al hallarse el vector en el cono estudiado, en virtud del Teorema 2.

Paralelamente y dentro de la contribución original, se ha desarrollado un programa en MATLAB que chequea la resolubilidad de un modelo de optimización lineal mediante la resolución del problema cuadrático asociado a la resolubilidad visto en esta sección.



Capítulo 2

Estabilidad del problema

2.1. Notación y elementos básicos

Problemas y espacio paramétrico

Sea Π el conjunto de todos los problemas de Programación (Optimización) Lineal con n variables y m restricciones, esto es, el conjunto formado por elementos de la forma:

$$\begin{aligned} \pi : \quad & \text{Min} \quad c'x \\ & \text{s.a} \quad a'_j x \leq b_j \quad j = 1, \dots, m, \end{aligned}$$

donde:

- $x \in \mathbb{R}^n$ representa el vector de variables.
- $c \in \mathbb{R}^n$ representa el vector de coeficientes asociado a la función objetivo.
- $a_j \in \mathbb{R}^n$ es el vector de coeficientes asociado a la j -ésima restricción, y $b_j \in \mathbb{R}$ el término independiente de la j -ésima restricción, con $j = 1, \dots, m$.

Podemos identificar el problema π con la siguiente terna $\pi \equiv (c, A, b) \in \mathbb{R}^n \times \mathbb{R}^{m \times n} \times \mathbb{R}^m$ donde A es la matriz cuya j -ésima fila es a'_j , y $b = (b_j)_{j=1, \dots, m}$. Del mismo modo, identificamos el espacio Π como:

$$\Pi \equiv \mathbb{R}^n \times \mathbb{R}^{m \times n} \times \mathbb{R}^m,$$

al cual denominaremos espacio paramétrico.

Consideramos un problema fijo, $\bar{\pi} \equiv (\bar{c}, \bar{A}, \bar{b})$, del cual se desean analizar las propiedades que se conservan en su entorno. Para ello, hemos de considerar una noción de distancia definida en el espacio paramétrico. En este caso, tomamos:

$$\begin{aligned} d: \quad \Pi \times \Pi &\longrightarrow \mathbb{R}_+ \\ (\pi_1, \pi_2) &\longrightarrow d(\pi_1, \pi_2) \end{aligned}$$

siendo $\pi_1 \equiv (c^1, A^1, b^1)$ y $\pi_2 \equiv (c^2, A^2, b^2)$ y

$$d(\pi_1, \pi_2) = \max \left\{ \|c^1 - c^2\|_1, \max_{j=1, \dots, m} \left\| \begin{pmatrix} a_j^1 \\ b_j^1 \end{pmatrix} - \begin{pmatrix} a_j^2 \\ b_j^2 \end{pmatrix} \right\|_2 \right\}.$$

Nótese que empleamos las normas $\|\cdot\|_1$ y $\|\cdot\|_2$ en \mathbb{R}^n y $\mathbb{R}^{m \times n}$, respectivamente. Recordemos que para cada una de las normas, la distancia asociada será denotada por d_1 y d_2 , respectivamente, y definiéndose como:

$$d_1(x, y) = \sum_{i=1}^p |x_i - y_i|, \quad d_2(x, y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2}.$$

Definimos asimismo los siguientes conjuntos, que serán de interés para el estudio de la estabilidad de un problema: $\Pi_c \subset \Pi$, que es el subconjunto de Π formado por los problemas consistentes, y $\Pi_r \subset \Pi$ siendo el subconjunto de Π formado por los problemas resolubles.

Denotamos por $int(\Pi_c)$ al interior de Π_c e $int(\Pi_r)$ al interior de Π_r . Recordemos que $\bar{\pi} \in int(\Pi_c)$ si existe $\epsilon > 0$ tal que para todo $\pi \in \Pi$ que verifique $d(\pi, \bar{\pi}) < \epsilon$, se tiene que $\pi \in \Pi_c$. El interior de los problemas resolubles ($int(\Pi_r)$) se define de manera análoga.

La estabilidad

La estabilidad de un problema π en optimización lineal es la medida que otorga información sobre las consecuencias en términos de factibilidad y resolubilidad que ocasionan las variaciones en los parámetros iniciales (c, A, b) . Así, a mayor margen de variación de estos parámetros sin que la consistencia y/o resolubilidad se vea afectada, diremos que existe una mayor estabilidad.

El objetivo de las siguientes secciones es el estudio de la estabilidad de un problema estudiando las características de la que esta depende: la consistencia y la resolubilidad.

Informalmente, entendemos que un problema $\bar{\pi}$ es inestable cuando existen variaciones arbitrariamente pequeñas de sus parámetros que dan lugar a cambios en sus propiedades.

2.2. Estabilidad en términos de la consistencia

Gracias al teorema 1, sabemos que un problema $\bar{\pi}$ pertenece al conjunto de los problemas consistentes ($\bar{\pi} \in \Pi_c$) si y solo si:

$$\begin{pmatrix} 0_n \\ -1 \end{pmatrix} \notin N.$$

Esta condición, aunque necesaria, no es suficiente para afirmar que $\bar{\pi}$ no es inestable con respecto a la consistencia. Para ello, debemos averiguar si este se halla en el interior del conjunto de los problemas consistentes. Cuando este se halla en la frontera de dicho conjunto, formalmente: $\bar{\pi} \in bd(\Pi_c)$, ello se traduce en que existen variaciones

arbitrariamente pequeñas del problema que ocasionan que este deje de pertenecer al conjunto de problemas consistentes.

Consideremos el conjunto siguiente:

$$H = \text{conv} \left\{ \begin{pmatrix} a_j \\ b_j \end{pmatrix}, j = 1, \dots, m \right\} + \mathbb{R}_+ \begin{pmatrix} 0_n \\ 1 \end{pmatrix},$$

donde:

- $\text{conv} \left\{ \begin{pmatrix} a_j \\ b_j \end{pmatrix}, j = 1, \dots, m \right\}$ representa la envoltura convexa

de los coeficientes del problema, alternativamente:

$$\left\{ \sum_{j=1}^m \lambda_j \begin{pmatrix} a_j \\ b_j \end{pmatrix}, \sum_{j=1}^m \lambda_j = 1, \lambda_j \geq 0, j = 1, \dots, m \right\},$$

- $\mathbb{R}_+ y := \{\mu y, \mu \geq 0\}$ para cualquier vector y .

Teorema 3. Sea $\bar{\pi} \in \Pi$. Se tiene que:

$$\bar{\pi} \in \text{int}(\Pi_c) \iff 0_n \notin H.$$

Gracias a este teorema, que relaciona el vector nulo con el conjunto H , podemos averiguar si el problema que estudiamos se encuentra en el interior del conjunto de problemas consistentes planteando un problema de distancias similar a los estudiados en las secciones 1.2 y 1.3.

Así, plantearemos encontrar la distancia mínima del vector 0_{n+1} a cualquier punto $(u, v)'$ que pertenezca a H , esto es:

$$d(0_{n+1}, H) = \min \{ \|(u, v)'\|_2 : (u, v)' \in H \},$$

que planteado como un problema de programación cuadrática que determina la distancia mínima, similar a los estudiados en las secciones anteriores y al que llamaremos $P_{\text{int}(c)}$, quedaría como:

$$P_{int(c)} \quad \text{Min} \quad d_2(0_{n+1}, (u, v)')$$

$$\text{s.a} \quad (u, v)' \in H,$$

Desarrollando la función objetivo, tenemos:

$$d_2(0_{n+1}, (u, v)') = \sqrt{\sum_{i=1}^n u_i^2 + v^2},$$

Análogamente a las secciones 1.2 y 1.3, consideramos la transformación al problema equivalente que llamaremos $P_{int(c)}^*$:

$$P_{int(c)}^* \quad \text{Min} \quad \sum_{i=1}^n u_i^2 + v^2$$

$$\text{s.a} \quad (u, v)' \in H,$$

Desarrollando la restricción de pertenencia de $(u, v)'$ a H como desigualdades lineales, tenemos el siguiente problema:

$$P_{int(c)}^* \quad \text{Min} \quad \sum_{i=1}^n u_i^2 + v^2$$

$$(i) \quad \text{s.a} \quad \begin{pmatrix} u \\ v \end{pmatrix} = \sum_{j=1}^m \lambda_j \begin{pmatrix} a_j \\ b_j \end{pmatrix} + \mu \begin{pmatrix} 0_n \\ 1 \end{pmatrix},$$

$$(ii) \quad \sum_{j=1}^m \lambda_j = 1,$$

$$(iii) \quad \lambda_j \geq 0,$$

$$(iv) \quad \mu \geq 0,$$

donde (i) obliga a $(u, v)'$ a ser una combinación lineal de los coeficientes y términos independientes con escalares no negativos (iii) del problema original, agregando un número μ real no negativo (iv) por el vector $(0_n, 1)'$ permitiendo la condición " $+\mathbb{R}_+(0_n, 1)'$ " del conjunto H . Además, se garantiza la convexidad en la suma de escalares igual a 1 (ii).

Dimensionalización del problema

El problema de distancias que obtenemos es un problema de Programación Cuadrática en forma adaptada idéntica a la ya definida en el punto 1.2.2 y 1.3.2. En este caso, nuestro vector de variables será $x = (u, v, \lambda, \mu)'$, con $u = (u_i)_{i=1, \dots, n} \in \mathbb{R}^n$, con $\lambda = (\lambda_j)_{j=1, \dots, m} \in \mathbb{R}^m$ y $v, \mu \in \mathbb{R}$. En el caso del problema $(P_{int(c)}^*)$, se expondrá a continuación por qué están constituidos sus elementos, así como las dimensiones de los mismos.

La matriz $H_{int(c)}$ tendrá elementos no nulos tan solo en aquellos asociados a las variables u_i y v , siendo el resto nulos. Esta quedaría como:

$$H_{int(c)} = \begin{pmatrix} \mathbb{I}_{n+1} & 0_{(n+1) \times (m+1)} \\ 0_{(m+1) \times (n+1)} & 0_{(m+1) \times (m+1)} \end{pmatrix} \in \mathbb{R}^{(n+m+2) \times (n+m+2)}.$$

En cuanto a la parte lineal de la función objetivo, esta será el vector nulo de orden $n + m + 2$, es decir:

$$f_{int(c)} = 0_{n+m+2}.$$

Por su parte, respecto a las restricciones, la matriz asociada a los coeficientes de las restricciones lineales, $A_{int(c)}$, resulta en:

$$A_{int(c)} = \begin{pmatrix} \mathbb{I}_{(n+1)} & \begin{pmatrix} -A' \\ -b' \end{pmatrix} & - \begin{pmatrix} 0_n \\ 1' \end{pmatrix} \\ 0'_{n+1} & 1'_m & 0 \end{pmatrix} \in \mathbb{R}^{(n+2) \times (n+m+2)}.$$

El vector de términos independientes está constituido en su totalidad por elementos nulos, excepto en su última componente, que es 1:

$$b_{int(c)} = \begin{pmatrix} 0_{n+1} \\ 1 \end{pmatrix} \in \mathbb{R}^{n+2}.$$

2.3. Estabilidad en términos de la resolubilidad

De manera similar al apartado anterior, para estudiar la estabilidad respecto a la resolubilidad, primero es necesario saber si el problema π pertenece al conjunto de problemas resolubles. Gracias al Teorema 2, sabemos que esta condición se cumple si, además de ser consistente:

$$-c \in \text{cone} \{a_j, j = 1, \dots, m\}.$$

Una vez esta condición es comprobada (mediante el algoritmo desarrollado e implementado en la sección 1.3), nuestro objetivo será medir la inestabilidad del problema con respecto a la resolubilidad. Para ello, haremos uso del siguiente resultado:

Teorema 4. *Sea $\bar{\pi} \in \Pi_r$, se tiene que la distancia a la irresolubilidad (esto es, la distancia entre el problema $\bar{\pi}$ y el conjunto de problemas no resolubles del espacio paramétrico) es:*

$$d(\bar{\pi}, \Pi \setminus \Pi_r) = \min \{d(0_{n+1}, H), d(0_n, \text{bd}(Z))\},$$

siendo $Z = \text{conv} \{c, a_j, j = 1, \dots, m\}$, es decir, la envoltura convexa del vector director de la función objetivo del problema original, y los coeficientes asociados a las restricciones lineales del mismo. Además, usamos $\text{bd}(X)$ para representar la frontera de un conjunto X .

Es fácil observar como el primer componente, $d(0_{n+1}, H)$, es la distancia obtenida en la sección inmediatamente anterior. Por ende, para averiguar la distancia a la irresolubilidad bastará con determinar la distancia del vector nulo de orden n a la frontera de Z .

Gracias a este teorema, podemos plantear un problema de optimización que determine la distancia del vector nulo a la frontera. En el

caso en el cual el vector nulo de orden n pertenece al interior de Z , esto es equivalente a afirmar que la distancia a su frontera es mayor que 0.

Nótese que $0_n \in Z$ no implica necesariamente $0_n \in \text{int}(Z)$, con lo cual no es eficaz plantear un problema que considere minimizar la norma entre el vector nulo y Z , ya que este tan solo nos informaría de la distancia al conjunto Z , mientras que lo que nos es relevante es averiguar la distancia a sus extremos.

En resumen, dada la información aportada por el Teorema 4, y sabiendo que $0_n \in \text{int}(Z) \iff d(0_n, \text{bd}(Z)) > 0$, podemos plantear un problema de optimización lineal que considere el vector nulo y trate de maximizar el radio de la bola donde el origen (el vector nulo) es el centro, sujeto a que dicha bola se halle dentro del conjunto Z .

Consideramos la maximización del radio de la bola para este fin, con lo cual un radio máximo mayor que 0 indicará que la mayor bola contenida en Z centrada en el origen presenta un radio no nulo, traduciéndose en que el origen está a una distancia no nula de la frontera, ergo encontrándose en el interior de Z . Nótese que este radio se corresponde con $d(0_n, \text{bd}(Z))$, que es el segundo término que tenemos como objetivo calcular.

Mediante los siguientes ejemplos se puede ilustrar gráficamente en \mathbb{R}^2 la idea subyacente que nos permitirá plantear el problema.

Ejemplo 3.1. Sea Z_A la envoltura convexa del siguiente conjunto $A = \{u_1 = (-2, -2), u_2 = (1, 2), u_3 = (4, 1), u_4 = (3, 3)\}$, y siendo $O = (0, 0)$. Véase el gráfico 3, donde el conjunto sombreado se corresponde con Z_A , y al trazar sucesivas bolas con centro en O (el origen), observamos como estas alcanzan una bola con radio máximo no nulo contenida en Z_A . Los círculos punteados se corresponden con bolas que no se encuentran contenidas en Z_A . Esto indica intuitivamente que O se encuentra en el interior de Z_A :

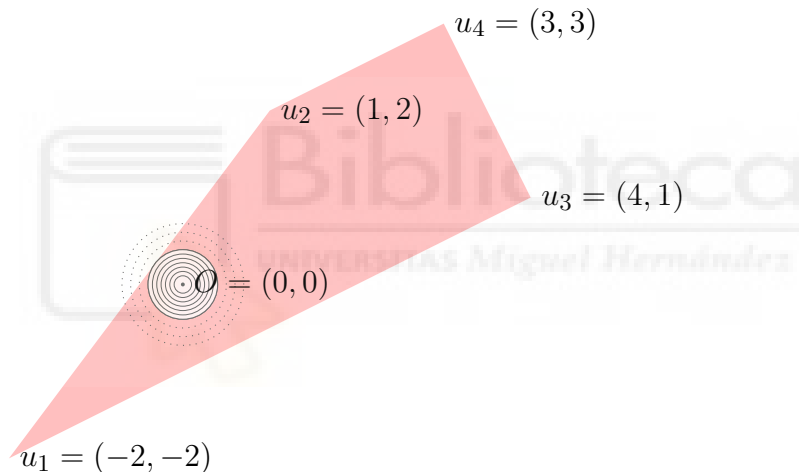


Gráfico 3. Fuente: elaboración propia.

Ejemplo 3.2. Sea Z_B la envoltura convexa del siguiente conjunto $B = \{u_1 = (-2, -2), u_2 = (4, 1), u_3 = (3, 3)\}$. En el gráfico 4 se puede ver como el radio máximo alcanzado por las bolas desde O es igual a 0, esto es, la mayor bola contenida en Z_B es el propio punto, ya que cualquier bola de radio mayor que 0 no está contenida en su totalidad en Z_B . Intuitivamente, vemos como O se halla en la frontera del conjunto Z_B :

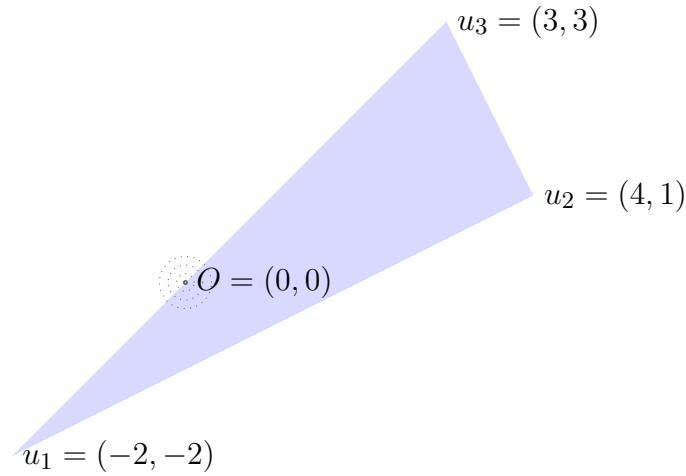


Gráfico 4. Fuente: elaboración propia.

Las bolas mostradas en los ejemplos consideran la norma euclídea, pero con una finalidad pragmática para su implementación, a la hora de diseñar el problema tendremos en cuenta la norma uno, ya definida en la sección 2.2.

Así, considerando la norma uno, y partiendo del origen, podemos plantear el siguiente problema de PL asociado a la distancia a la irresolubilidad:

$$\begin{aligned}
 P_d \quad & \text{Max } \rho \\
 (i) \quad & \text{s.a } \rho e_i = \sum_{j=1}^m \lambda_{ij}^+ a_j + \mu_i^+ \bar{c}, \quad i = 1, \dots, n, \\
 (ii) \quad & -\rho e_i = \sum_{j=1}^m \lambda_{ij}^- a_j + \mu_i^- \bar{c}, \quad i = 1, \dots, n, \\
 (iii) \quad & \sum_{j=1}^m \lambda_{ij}^+ + \mu_i^+ = 1, \quad i = 1, \dots, n, \\
 (iv) \quad & \sum_{j=1}^m \lambda_{ij}^- + \mu_i^- = 1, \quad i = 1, \dots, n, \\
 (v - viii) \quad & \lambda_{ij}^+, \mu_i^+, \lambda_{ij}^-, \mu_i^- \geq 0,
 \end{aligned}$$

donde:

- Consideramos la variable $\rho \in \mathbb{R}_+$ como el radio a maximizar.
- e_i representa el vector de la base canónica en \mathbb{R}^n cuya i -ésima componente es un 1 con $i = 1, \dots, n$.
- $\lambda_{ij}^+, \mu_i^+, \lambda_{ij}^-, \mu_i^- \in \mathbb{R}$, con $i = 1, \dots, n$ y $j = 1, \dots, m$, son los escalares que fuerzan al radio ρ por cada uno de los vectores de la base canónica a expresarse como una combinación lineal de estos por los coeficientes a_j (asociados a las restricciones) y \bar{c} (asociado a la función objetivo) del problema original (como se ve en (i) y (ii)). Cabe decir que estas combinaciones lineales cumplen la condición de convexidad ((iii) y (iv)), ergo fuerzan a ρe_i y $-\rho e_i$ (es decir, los $2n$ vértices de las bolas para cada radio ρ) a permanecer dentro del conjunto Z .

Adaptación y dimensionalización del problema

Para la correcta implementación del problema en MATLAB, consideremos primeramente el modelo de PL en forma adaptada a nuestro problema:

$$\begin{aligned}
 P_d \quad & \text{Min} \quad f'_d x \\
 \text{s.a} \quad & A_d x = b_d, \\
 & \lambda_{ij} \geq 0, \quad i = 1, \dots, n, \quad j = 1, \dots, m \\
 & \mu_i \geq 0, \quad i = 1, \dots, n,
 \end{aligned}$$

dando a continuación una descripción elemento a elemento.

Consideraremos nuestro vector de variables

$$x = (\rho, \lambda_1^+, \dots, \lambda_n^+, \mu^+, \lambda_1^-, \dots, \lambda_n^-, \mu^-),$$

siendo

- $\lambda_i^+ = (\lambda_{ij}^+)_{j=1, \dots, m} \in \mathbb{R}^m$ con $i = 1, \dots, n$,

- $\lambda_i^- = (\lambda_{ij}^-)_{j=1,\dots,m} \in \mathbb{R}^m$ con $i = 1, \dots, n$,
- $\mu^+ = (\mu_i^+)_{i=1,\dots,n} \in \mathbb{R}^n$,
- $\mu^- = (\mu_i^-)_{i=1,\dots,n} \in \mathbb{R}^n$.

La matriz de coeficientes de las restricciones A_d , debido a su gran tamaño, será subdividida en cuatro submatrices: $A_{d,e}$, $A_{d,\lambda}$, $A_{d,\mu}$ y $A_{d,conv}$ siendo cada una de ellas:

$$A_{d,e} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \\ -e_1 \\ -e_2 \\ \vdots \\ -e_n \end{pmatrix},$$

$$A_{d,\lambda} = - \left(\begin{array}{cccc|cccc} A & 0_{n \times m} & \dots & 0_{n \times m} & & & & \\ 0_{n \times m} & A & \dots & 0_{n \times m} & & & & \\ \vdots & \vdots & \ddots & \ddots & & & & \\ 0_{n \times m} & 0_{n \times m} & 0_{n \times m} & A & & & & \\ \hline & & & & A & 0_{n \times m} & \dots & 0_{n \times m} \\ 0_{n^2 \times (nm)} & & & & 0_{n \times m} & A & \dots & 0_{n \times m} \\ & & & & \vdots & \vdots & \ddots & \ddots \\ & & & & 0_{n \times m} & 0_{n \times m} & 0_{n \times m} & A \end{array} \right),$$

Capítulo 3

Aplicación en un entorno industrial

3.1. Planteamiento inicial

3.1.1. La aplicación de la estabilidad

Recordemos que la estabilidad de un problema nos otorga valiosa información sobre la medida en la cual podemos variar los parámetros iniciales del mismo sin que este pierda sus propiedades. Visto desde un punto de vista industrial, esto nos permite modelizar inmensidad de situaciones donde pueden variar elementos tales como los precios, las cantidades, los procesos técnicos, la demanda, etc. pudiendo evaluar así en qué medida los cambios en estos afectarán a las propiedades del problema fruto de la modelización que hemos llevado a cabo.

En los siguientes apartados se ejemplifica el planteamiento y posterior resolución en MATLAB con datos numéricos de un problema de producción sujeto a demanda en la industria alimentaria.

3.1.2. Ejemplo numérico en un entorno de industria alimentaria

En preparación para la temporada turística, una compañía de industria alimentaria dedicada a la elaboración de platos típicos españoles está produciendo cuatro envasados distintos: tortilla de patatas, paella alicantina, salmorejo y torrijas. Todos estos productos pasan por cuatro departamentos diferentes: limpieza de materias primas, elaboración, cocción y empaque. La compañía ha recibido pedidos para todos los productos, estipulando el contrato una penalización por los artículos no suplidos.

El objetivo de esta sección es elaborar un plan óptimo de producción para esta empresa dadas las restricciones de uso de tiempo de cada sección, el coste y las penalizaciones asociadas a cada producto, estudiando primeramente la consistencia y resolubilidad del modelo a plantear. En el caso en el que estas se cumplieran, se evaluaría la distancia a la irresolubilidad del modelo, esto es, qué tan estable es el modelo.

Los datos sobre los tiempos de cada producto con respecto a los departamentos se pueden ver en la Tabla 1:

Departamento	Tortilla	Paella	Salmorejo	Torrijas
Limpieza	0.30	0.30	0.25	0.15
Elaboración	0.25	0.35	0.30	0.10
Cocción	0.45	0.50	0.40	0.22
Empaque	0.15	0.15	0.10	0.05

Tabla 1. Fuente: elaboración propia.

La demanda, costes de penalización y beneficio (utilidad) por bien producido se pueden visualizar en la Tabla 2:

	Tortilla	Paella	Salmorejo	Torrijas
Demanda	800 uds.	750 uds.	600 uds.	500 uds.
Beneficio unitario	30€	40€	20€	10€
Penalización unitaria	15€	20€	10€	8€

Tabla 2. Fuente: elaboración propia.

Además, sabemos que la capacidad de trabajo por departamento en horas es de 1000. Con estos datos, plantearemos el modelo matemático para optimizar el beneficio en el siguiente subapartado.

3.1.3. Modelización del problema

Llamaremos a las variables del problema:

- x_1 = cantidad de tortillas
- x_2 = cantidad de paellas
- x_3 = cantidad de salmorejo
- x_4 = cantidad de torrijas

Además, se penaliza a la compañía si no cumple la demanda. El objetivo es entonces maximizar el beneficio neto, el cual definiremos como:

$$\text{Beneficio neto} = \text{Beneficio total} - \text{Penaliz. total.}$$

La función de beneficio total es $30x_1 + 40x_2 + 20x_3 + 10x_4$. Para calcular la penalización total, las restricciones de la demanda pueden escribirse así:

$$x_1 + s_1 = 800, x_2 + s_2 = 750, x_3 + s_3 = 600, x_4 + s_4 = 500,$$

$$x_i \geq 0, s_i \geq 0, i = 1, 2, 3, 4.$$

La nueva variable s_i representa la escasez en la demanda del producto i , siendo por tanto la penalización total: $15s_1 + 20s_2 + 10s_3 + 8s_4$.

42 CAPÍTULO 3. APLICACIÓN EN UN ENTORNO INDUSTRIAL

El modelo de optimización lineal quedaría planteado, entonces:

$$\begin{aligned}
 (P) \quad & \text{Max } 30x_1 + 40x_2 + 20x_3 + 10x_4 - (15s_1 + 20s_2 + 10s_3 + 8s_4) \\
 \text{s.a} \quad & 0.30x_1 + 0.30x_2 + 0.25x_3 + 0.15x_4 \leq 1000, \\
 & 0.25x_1 + 0.35x_2 + 0.30x_3 + 0.10x_4 \leq 1000, \\
 & 0.45x_1 + 0.50x_2 + 0.40x_3 + 0.22x_4 \leq 1000, \\
 & 0.15x_1 + 0.15x_2 + 0.10x_3 + 0.05x_4 \leq 1000, \\
 & x_1 + s_1 = 800, \\
 & x_2 + s_2 = 750, \\
 & x_3 + s_3 = 600, \\
 & x_4 + s_4 = 500, \\
 & x_i \geq 0, s_i \geq 0, i = 1, 2, 3, 4.
 \end{aligned}$$

Para simplificar los cálculos, expresaremos las variables s_i en términos de x_i , ya que, considerando d_i la demanda para cada producto:

$$x_i + s_i = d_i \ \& \ s_i \geq 0 \iff s_i = d_i - x_i \ \& \ x_i \leq d_i$$

Expresado en términos de minimizar y tras el cambio de variables:

$$\begin{aligned}
 (P) \quad & \text{Min } 30x_1 - 40x_2 - 20x_3 - 10x_4 + 15(800 - x_1) \\
 & \quad + 20(750 - x_2) + 10(600 - x_3) + 8(500 - x_4) \\
 \text{s.a} \quad & 0.30x_1 + 0.30x_2 + 0.25x_3 + 0.15x_4 \leq 1000, \\
 & 0.25x_1 + 0.35x_2 + 0.30x_3 + 0.10x_4 \leq 1000, \\
 & 0.45x_1 + 0.50x_2 + 0.40x_3 + 0.22x_4 \leq 1000, \\
 & 0.15x_1 + 0.15x_2 + 0.10x_3 + 0.05x_4 \leq 1000, \\
 & 0 \leq x_1 \leq 800, \\
 & 0 \leq x_2 \leq 750, \\
 & 0 \leq x_3 \leq 600, \\
 & 0 \leq x_4 \leq 500.
 \end{aligned}$$

3.2. Resolución en MATLAB

Primeramente estudiaremos la consistencia del problema mediante la función 'Consistencia' desarrollada originalmente en este proyecto. Esta tiene como valores de entrada la matriz de coeficientes y términos independientes asociados a las restricciones. Así:

```
>>>consistencia
Introduzca la matriz de coeficientes A:
[0.3 0.3 0.25 0.15;
0.25 0.35 0.3 0.1;
0.45 0.5 0.4 0.22;
0.15 0.15 0.1 0.05;
1 0 0 0;
0 1 0 0;
0 0 1 0;
0 0 0 1;
-1 0 0 0;
0 -1 0 0;
0 0 -1 0;
0 0 0 -1]
Introduzca el vector de terminos indep. b columna:
[1000,1000,1000,1000,800,750,600,500,0,0,0,0]'
```

Del cual obtenemos como resultado:

```
v =
1.0000
```

Al tener que $v(P_c^*) > 0$, si recordamos la sección 1.2.2., esto nos informa de que el problema es consistente, esto es, su conjunto factible es no vacío, pudiendo estudiarse ahora su resolubilidad.

A continuación, usaremos la función 'Resolubilidad' implementada en MATLAB, con parámetros de entrada la matriz de coeficientes y el vector de la función objetivo:

```

>> resolubilidad
Introduzca el vector de la funcion objetivo:
[-45 -60 -30 -18]
Introduzca la matriz de coeficientes A:
[0.3 0.3 0.25 0.15;
0.25 0.35 0.3 0.1;
0.45 0.5 0.4 0.22;
0.15 0.15 0.1 0.05;
1 0 0 0;
0 1 0 0;
0 0 1 0;
0 0 0 1;
-1 0 0 0;
0 -1 0 0;
0 0 -1 0;
0 0 0 -1]

```

De este programa, obtenemos 0 como resultado, traducíéndose en que el problema sí posee la propiedad de resolubilidad, ya que $v(P_r^*) = 0$ implicaba esta condición.

Una vez la consistencia y la resolubilidad han sido garantizadas, podemos proceder a resolver el problema de optimización lineal en sí mismo mediante la instrucción *linprog*. Con la cual, obtenemos una solución óptima de $x_1 = 800, x_2 = 750, x_3 = 387.5, x_4 = 500$. La solución satisface toda la demanda de Tortilla de Patatas, Paella Alicantina y Torrijas pero considera una merma de 212.5 litros de Salmorejo (al restar la demanda satisfecha a la demanda total de Salmorejo), representando un coste de penalización de 2130€. El valor óptimo del problema es en este caso 64625€.

Por último, vamos a calcular la distancia a la irresolubilidad del problema.

Primero estudiaremos si este problema se halla en el interior del conjunto de problemas consistentes. Usaremos la función 'Interior-Consistencia' implementada en MATLAB basada en el estudio de la estabilidad visto

en la sección 2.2:

```
>> interior_consistencia
Introduzca la matriz de coeficientes A:
[0.3 0.3 0.25 0.15;
 0.25 0.35 0.3 0.1;
 0.45 0.5 0.4 0.22;
 0.15 0.15 0.1 0.05;
 1 0 0 0;
 0 1 0 0;
 0 0 1 0;
 0 0 0 1;
 -1 0 0 0;
 0 -1 0 0;
 0 0 -1 0;
 0 0 0 -1]
Introduzca el vector de terminos indep. b columna:
[1000,1000,1000,1000,800,750,600,500,0,0,0,0]'
```

Obtenemos el siguiente resultado:

```
v =
2.5000e+05
```

Este es indicador de que la distancia de 0_{n+1} al conjunto H es positiva, lo cual nos dice que el problema sí se halla en el interior de los resolubles.

Por último, veamos si el problema se halla en el interior del conjunto de problemas resolubles mediante la función análoga a este caso, la cual es llamada 'Interior-Resolubles', y está basada en el problema visto en el apartado 2.3.

```
>> interior_resolubles
Introduzca el vector de la funcion objetivo:
[-45 -60 -30 -18]
```

Introduzca la matriz de coeficientes A:

```
[0.3 0.3 0.25 0.15;
 0.25 0.35 0.3 0.1;
 0.45 0.5 0.4 0.22;
 0.15 0.15 0.1 0.05;
 1 0 0 0;
 0 1 0 0;
 0 0 1 0;
 0 0 0 1;
 -1 0 0 0;
 0 -1 0 0;
 0 0 -1 0;
 0 0 0 -1]
```

En el cual obtenemos:

```
valor =
0.1324
```

De este resultado deducimos que este también se encuentra en el interior del conjunto de problemas resolubles, al ser este valor mayor que 0.

La distancia a la irresolubilidad es por tanto, el mínimo de estos dos últimos valores, el cual es 0.1324. Al ser esta distancia positiva, podemos decir que el problema no es inestable, o lo que es lo mismo, cualesquiera perturbaciones del objetivo y los coeficientes de las restricciones (tiempos de procesado, en la demanda de los diferentes alimentos, etc.) menores que 0.1324 (medidas con las normas introducidas en 2.1) no modifican su resolubilidad.

Apéndice A

Programas I

En este apartado se pueden encontrar los programas correspondientes con las funciones asociadas al capítulo 2.

Función para el estudio de la consistencia

```
function consistencia ()
A=input('Introduzca la matriz de coeficientes A: ');
b=input('Introduzca el vector de terminos indep. b
columna: ');
[m,n]=size(A);

Ab=[A';b'];

H=[eye(n+1),zeros(n+1,m);
zeros(m,n+1+m)];
f=[zeros(1,n),2,zeros(1,m)];

Aeq=[eye(n+1),-Ab]
beq=zeros(1,n+1)

LB=[-inf(1,n+1),zeros(1,m)];

[x,v]=quadprog(2*H,f,[],[],Aeq,beq,LB,[]);
v=v+1
end
```

Función para el estudio de la resolubilidad

```
function resolubilidad()
f_Problema=input('Introduzca el vector de la funcion
    objetivo: ');
A=input('Introduzca la matriz de coeficientes A: ');

[m,n]=size(A)

H_1=[eye(n);zeros(m,n)];

H_2=[zeros(n+m,m)];

H=[H_1,H_2]

f=[2*f_Problema';zeros(m,1)]

Aeq=[-eye(n),A']
beq=zeros(n,1)

LB=[-inf(1,n),zeros(1,m)]

[x,v]=quadprog(2*H,f,[],[],Aeq,beq,LB,[])
v=v+sumsqr(f_Problema)

end
```

Apéndice B

Programas II

En este apartado se pueden encontrar los programas correspondientes con las funciones asociadas al capítulo 2.

Función para el estudio de la estabilidad en términos de la consistencia

```
function interior_consistencia ()
A=input('Introduzca la matriz de coeficientes A: ');
b=input('Introduzca el vector de terminos indep. b
columna: ');
[m,n]=size(A);

Ab=[A';b'];
H=[eye(n+1),zeros(n+1,m+1);
zeros(m+1,n+1+m+1)]
f=[zeros(1,n),0,zeros(1,m+1)]
Aeq=[eye(n+1),-Ab,-[zeros(n,1);1];
zeros(1,n+1),ones(1,m),0]
beq=[zeros(n+1,1);1]
LB=[-inf(1,n+1),zeros(1,m+1)];

[x,v]=quadprog(2*H,f,[],[],Aeq,beq,LB,[]);
v
end
```

**Función para el estudio de la estabilidad
en términos de la resolubilidad**

```

function interior_resolubles()
f_Problema=input('Introduzca el vector de la funcion
    objetivo: ');
A=input('Introduzca la matriz de coeficientes A: ');
[m,n]=size(A)
Aeq=zeros(n^2*2,1);
col_1=[];
ident=eye(n)
Aeq_lambda=zeros((n^2)*2,2*m*n);
A=A'

for p=1:2*n
if p<=n
if mod(p,n)~=0
col_1=[col_1;ident(:,mod(p,n))];
end
if mod(p,n)==0
col_1=[col_1;ident(:,n)];
end
else
if mod(p,n)~=0
col_1=[col_1;-ident(:,mod(p,n))];
end
if mod(p,n)==0
col_1=[col_1;-ident(:,n)];
end
end
end

for w=1:2*m*n

if mod(w,m)~=0
Aeq_lambda(ceil(w/m)+(n-1)*(ceil(w/m)-1):ceil(w/m)+(n-1)*(ceil(w/m)-1)+n-1,w)=A(:,mod(w,m));
end

if mod(w,m)==0

```

```

Aeq_lambda(ceil(w/m)+(n-1)*(ceil(w/m)-1):ceil(w/m)+(n-1)*(ceil(w/m)-1)+n-1,w)=A(:,m);
end
end

Aeq(:,1)=col_1;

Amu=zeros(n^2,n);

for i=1:n
Amu((i-1)*n+1:i*n,i)=f_Problema';
end

Amu_def=zeros(2*n^2,2*n);
Amu_def(1:n^2,1:n)=Amu;
Amu_def(n^2+1:2*n^2,n+1:2*n)=Amu;
Aeq=[Aeq,Aeq_lambda,-Amu_def]

Aconvex1=zeros(n,m*n);

for i=1:n
Aconvex1(i,(i-1)*m+1:m*i)=ones(1,m);
end

Aconvex=zeros(2*n,2*m*n);
Aconvex(1:n,1:m*n)=Aconvex1;
Aconvex(n+1:2*n,m*n+1:2*m*n)=Aconvex1;
Aconvexmu=eye(2*n);
Aconvex=[Aconvex,Aconvexmu];

Aeq=[Aeq;zeros(2*n,1),Aconvex]

beq=[zeros(2*n^2,1);ones(2*n,1)]
f=[1,zeros(1,2*m*n+2*n)];
LB=[-inf,zeros(1,2*m*n+2*n)];

[x,v]=linprog(-f,[],[],Aeq,beq,LB,[])
valor=-v
end

```


Bibliografía

- [1] Bazaraa, M. S., Jarvis, J. J., & Sherali, H. D. (2009). Linear programming and network flows: Bazaraa/linear. John Wiley & Sons, Inc.
- [2] Bertsimas, D. (1997). Introduction to linear optimization. Athena Scientific.
- [3] Cánovas, M. J., López, M. A., Parra, J., & Toledo, F. J. (2006). Distance to solvability/unsolvability in linear optimization. *SIAM Journal on Optimization: A Publication of the Society for Industrial and Applied Mathematics*, 16(3), 629–649. <https://doi.org/10.1137/040612981>
- [4] Galindo, J. (2016, May 27). Dantzig: pensamiento positivo y programación lineal. Ediciones EL PAÍS S.L.
- [5] Goberna, M.A. Jornet, V. Puente, R. (2004). Optimización Lineal. Teoría, Métodos y Modelos. McGraw-Hill.
- [6] Historia de la programación lineal. (2012, May 7). Optimus. <https://inveoperaciones.wordpress.com/2012/05/07/historia-de-la-programacion-lineal/>
- [7] Luenberger, D. G., & Ye, Y. (2021). Linear and nonlinear programming (5th ed.). Springer Nature.
- [8] Taha. (1998). Investigación De Operaciones. Prentice Hall & IBD.
- [9] Wikipedia contributors. (n.d.). Programación lineal. Wikipedia, The Free Encyclopedia.