

19
24.



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ACATLAN**

**INSTRUMENTACION COMPUTACIONAL DE LOS
METODOS DE ESCALA AFIN PRIMAL Y DUAL
PARA LA SOLUCION DE PROBLEMAS DE
PROGRAMACION LINEAL**

T E S I S

**QUE PARA OBTENER EL TITULO DE
LICENCIADO EN MATEMATICAS
APLICADAS Y COMPUTACION
P R E S E N T A N :
GUSTAVO ERNESTO FLORES VAZQUEZ
SOFIA LOPEZ AGUILERA**



1997.

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "ACATLAN"
DIVISION DE MATEMATICAS E INGENIERIA
PROGRAMA DE MATEMATICAS APLICADAS Y COMPUTACION

SRITA. SOFIA LOPEZ AGUILERA
SR. GUSTAVO E. FLORES VAZQUEZ
Alumnos de la carrera de Matemáticas Aplicadas y Computación.
Presente.

De acuerdo a su solicitud presentada con fecha 30 de mayo de 1996, me complace notificarle que esta Jefatura tuvo a bien asignarles el siguiente tema de Tesis: "INSTRUMENTACION COMPUTACIONAL DE LOS METODOS DE ESCALA AFIN PRIMAL Y DUAL PARA LA SOLUCION DE PROBLEMAS DE PROGRAMACION LINEAL", el cual se desarrollará como sigue:

INTRODUCCION

- CAP. I Planteamiento del Problema de Programación Lineal
 - CAP. II Método de escala proyectiva de Karmarkar
 - CAP. III Métodos de escala Afín
 - CAP. IV Aspectos necesarios a considerar en la instrumentación computacional de los métodos de escala Afín.
 - CAP. V Instrumentación computacional de los métodos de escala Afín y evaluación de resultados
- CONCLUSIONES**
APENDICES
BIBLIOGRAFIA

Asimismo, fue designado como Asesor de Tesis el DR. JESUS AGUSTIN CANO GARCÉS

Ruego a ustedes tomar nota que en cumplimiento de lo especificado en la Ley de Profesiones, deberán presentar servicio social durante un tiempo mínimo de seis meses como requisito básico para sustentar examen profesional, así como de la disposición de la Coordinación de la Administración Escolar en el sentido de que se imprima en lugar visible de los ejemplares de la Tesis el título del trabajo realizado. Esta comunicación deberá imprimirse en el interior de la misma.

A T E N T A M E N T E
"POR MI RAZA HABLARA EL ESPIRITU"
Acatlán, Edo. Mex. febrero 8 de 1997


MAT. MAXYA ELIZONDO CORTES
Encargada del Programa de Matemáticas Aplicadas y Computación

COLEP. ACATLAN



SEJUTURA DEL PROGRAMA DE
ACTUARIA Y MATEMATICAS
APLICADAS Y COMPUTACION

A nuestros padres

**Gustavo Luis Flores Ávila
Ivonne Vázquez García**

**Anastacio López Segura
María Aguilera Baca**

**Por su gran apoyo, ánimo y confianza que nos brindaron siempre durante
nuestros estudios.**

A nuestros hermanos

**Alma Ruth, Luis Antonio, Jorge Omar y Diana Luz Flores Vázquez
José Minervino y Ma. del Rocío López Aguilera**

A nuestros hijos

Emmanuel y Karen

**Que son y serán siempre el principal motivo para que alcancemos nuestras
metas.**

INDICE

Introducción	v
Capítulo I. Planteamiento del Problema de Programación Lineal	1
I.1 Antecedentes de la Programación Lineal	2
I.2 El Problema de Programación Lineal	5
I.2.1 Presentación Informal de PPL	5
I.2.2 Poliedros	6
I.2.3 Conjuntos Afines y Conjuntos Convexos	9
I.2.4 Puntos Extremos y Soluciones Básicas Factibles	12
I.2.5 Soluciones Básicas Factibles Degeneradas	16
I.2.6 Definición del Problema de Programación Lineal	17
I.3 El Problema de Programación Lineal Dual	20
I.3.1 Definición del PPL Dual	20
I.3.2 Propiedades de los PPL Primal y Dual	21
I.3.3 Holguras Complementarias	24
I.3.4 Condiciones de Optimalidad de Karush-Kuhn-Tucker	26
I.3.5 El Vector Dual	28
Capítulo II. Método de Escala Proyectiva de Karmarkar	30
II.1 Ideas Básicas del Algoritmo de Karmarkar	30
II.2 La Forma Canónica de Karmarkar	32
II.2.1 Obtención de un Politopo	32
II.2.2 Condiciones de Karmarkar	33
II.2.3 Interpretación Geométrica	34
II.3 Transformación de un PPL a la Forma Canónica de Karmarkar	36
II.3.1 Combinación de los Problemas Primal y Dual	36

II.3.2 Obtención de un Punto Interior Inicial y de una Función Objetivo con Valor Mínimo Cero	37
II.3.3 Transformación del PPL	39
II.4 Algoritmo de Karmarkar	44
II.4.1 Transformación de un Simplejo a un Simplejo	44
II.4.2 Obtención de la Mejor Dirección de Movimiento	48
II.4.2.1 Dirección en un Poliedro	48
II.4.2.2 Dirección de Mayor Descenso	49
II.4.2.3 Proyección del Gradiente para la Obtención de la Dirección Factible de Mayor Descenso	50
II.4.3 Optimización Sobre una Bola y Transformación Inversa de un Simplejo a un Simplejo	55
II.4.4 Verificación de Factibilidad y Chequeo de Paro	59
II.5 Esquema de Purificación	65
II.6 Presentación Formal del Método de Escala Proyectiva de Karmarkar	66
Capítulo III. El Método de Escala Afín	69
III.1 El Método de Escala Afín Primal	70
III.1.1 Ideas Básicas del Método de Escala Afín Primal	71
III.1.2 Transformación de Escala Afín Sobre el Ortante No Negativo	73
III.1.3 Optimización Sobre una Bola	75
III.1.4 Dirección Factible de Mayor Descenso	77
III.1.5 El Algoritmo de Escala Afín Primal	81
III.1.6 Convergencia del Algoritmo de Escala Afín Primal	88
III.2 El Método de Escala Afín Dual	92
III.2.1 Ideas Básicas del Método de Escala Afín Dual	93
III.2.2 Transformación de Escala Afín Sobre la Variante Dual	94

III.2.3 Obtención de la Dirección Factible de Mayor Ascenso	95
III.2.4 Selección de Tamaño de Paso	100
III.2.5 El Algoritmo de Escala Afín Dual	101
Capítulo IV. Aspectos Numéricos a Considerar en la Instrumentación Computacional de los Métodos de Escala Afín	104
IV.1 Obtención de una Solución Interior Inicial	104
IV.1.1 Obtención de una Solución Interior Inicial Factible para el Método de Escala Afín Primal	105
IV.1.2 Obtención de una Solución Interior Inicial Factible para el Método de Escala Afín Dual	106
IV.2 Selección de Método de Paro	108
IV.2.1 Condiciones de KKT como Regla de Paro para el Método de Escala Afín Primal	108
IV.2.2 Condiciones de KKT como Regla de Paro para el Método de Escala Afín Dual	110
IV.2.3 Detección de No Acotación	112
IV.2.4 Análisis del Vector Dirección para Detección de Función Objetivo Constante y Selección de Tamaño de Paso	113
IV.2.5 Evaluación de la Variable Artificial	114
IV.3 Factorizaciones Matriciales para el Cálculo de la Dirección	115
IV.3.1 Los Sistemas de Ecuaciones Lineales en los Vectores Dirección	115
IV.3.2 Factorización Simétrica Indefinida	117
IV.3.3 Factorización de Cholesky	122
IV.3.4 Factorización QR	125
IV.3.5 Método de los Mínimos Cuadrados con Factorización QR	128

ÍNDICE

Capítulo V. Instrumentación Computacional de los Métodos de Escala Afín y Evaluación de Resultados	134
V.1 Descripción de los Programas Instrumentados	134
V.1.1 Características del Software y del Hardware	134
V.1.2 Descripción del Programa Correspondiente al Método de Escala Afín Primal	136
V.1.3 Descripción del Programa Correspondiente al Método de Escala Afín Dual	141
V.2 Descripción de los Problemas Prueba a Evaluar	148
V.2.1 Problemas Prueba, Grupo 1	149
V.2.2 Problemas Prueba, Grupo 2	153
V.3 Evaluación y Comparación de Resultados	156
Conclusiones	167
Apéndice A. Conceptos de Álgebra Lineal	169
Apéndice B. Conceptos de Programación No Lineal	182
Apéndice C. Manual del Usuario	190
Apéndice D. Código Correspondiente al Método de Escala Afín Primal	206
Apéndice E. Código Correspondiente al Método de Escala Afín Dual	224
Glosario	246
Bibliografía	248

INTRODUCCIÓN

Una de las ramas de las Matemáticas Aplicadas que se ha utilizado con mucho éxito en la asignación óptima de recursos limitados entre actividades competitivas, es la programación lineal.

La programación lineal se estableció como una disciplina de gran utilidad después de que G. B. Dantzig inventó el Método Simplex en 1947, para la fuerza aérea de los Estados Unidos, con el fin de resolver problemas de planificación y coordinación entre varios proyectos así como el uso eficaz de los recursos disponibles.

El Método Simplex resultó ser tan importante que se convirtió en la principal herramienta para la solución de problemas de programación lineal y en el tema casi único de los cursos universitarios relacionados con esta área.

Pero de acuerdo a los resultados reportados por Dantzig y corroborados en otros experimentos, el Simplex parece resolver problemas prácticos con matrices de $m \times n$ en un número de iteraciones que aumenta proporcionalmente a m y muy poco con incrementos de n . Además, para cierta clase de problemas (desarrollados por Klee y Minty [KyM]), el número de iteraciones requeridas por el Simplex para solucionarlos crece exponencialmente con el tamaño del problema.

Estos hechos motivaron el desarrollo de nuevos algoritmos para la solución de problemas de programación lineal como el algoritmo de

Khachiyan [Kha] y el algoritmo de Karmarkar [Kar]. Ambos algoritmos proponen procedimientos completamente distintos al Simplex para solucionar problemas de programación lineal; además dichos algoritmos, desde un punto de vista teórico, resultaron ser más eficientes que el Simplex, pero no así en la práctica.

Sin embargo, del algoritmo de Karmarkar, se generaron diversas variantes que se consideran realmente competitivas con el Simplex, desde un punto de vista práctico, tales como los métodos de escala afín. Dado el impacto que han tenido esta nueva clase de algoritmos en diversas áreas de las Matemáticas Aplicadas se considera importante el conocimiento más a fondo de este nuevo concepto.

Los objetivos de este trabajo son presentar y analizar el Método de Escala Projectiva de Karmarkar (como se le conoció posteriormente al algoritmo de Karmarkar), el Método de Escala Afín Primal y el Método de Escala Afín Dual (dos de las variantes del algoritmo de Karmarkar); analizar también las técnicas numéricas que permiten la instrumentación computacional de los métodos de escala afín y evaluar las instrumentaciones sobre un conjunto de problemas prueba que con los que es posible efectuar comparaciones entre ambos métodos y con los resultados reportados en la literatura, con respecto al método de escala projectiva.

Para alcanzar estos objetivos, se estructuró este trabajo en cinco capítulos.

En el primer capítulo se describen los fundamentos matemáticos de la programación lineal necesarios para el estudio de los métodos considerados en este trabajo (conocidos también en forma genérica como métodos de puntos interiores).

En el segundo se analiza el método de escala proyectiva de Karmarkar para la solución de problemas de programación lineal en base a los fundamentos considerados en el objetivo anterior.

En el tercero se analizan el método de escala afín primal y el método de escala afín dual, como variantes del algoritmo de Karmarkar, en base a los fundamentos del primer capítulo y del capítulo anterior.

En el cuarto se complementan los métodos de escala afín primal y dual con técnicas numéricas adecuadas para su instrumentación computacional, tales como : obtención de una solución interior inicial, obtención de la mejor forma de paro, y factorizaciones matriciales para el cálculo de las direcciones de búsqueda consideradas en los métodos.

Finalmente en el quinto capítulo se describen las instrumentaciones computacionales de los métodos de escala afín primal y dual en base a los análisis que se efectuaron y a las técnicas numéricas seleccionadas para tal fin; además, se muestran los resultados de las pruebas efectuadas con dichas instrumentaciones y se comparan con los resultados reportados en la bibliografía.

INTRODUCCIÓN

Cabe aclarar que los programas instrumentados se consideran como experimentales y no como un sistema. Pero se proponen como base para el desarrollo de paquetes computacionales eficientes y robustos.

Queremos agradecer al Dr. Agustín Cano Garcés por su confianza y apoyo en la realización de esta tesis, así como sus consejos, paciencia y valiosa ayuda en la revisión de la misma. Queremos agradecer también al Dr. Miguel Bravo Díaz, al Dr. Isidro Romero, al Dr. Gilberto Calvillo, al Mat. Antonio Pérez Ortiz y al Mat. Cesar Hernández por su colaboración durante el desarrollo de esta obra.

CAPÍTULO I

**PLANTEAMIENTO DEL
PROBLEMA DE PROGRAMACIÓN LINEAL**

En muchos aspectos de la vida real se requiere resolver el problema de asignar recursos limitados entre actividades competitivas de la mejor manera posible, es decir, en forma óptima.

Este problema puede ser solucionado a través de técnicas de programación matemática, en especial, por la programación lineal.

La programación lineal, que consiste básicamente en la aplicación del álgebra lineal a la asignación de recursos, puede ser vista como parte de un gran desarrollo revolucionario, el cual ha dado a la humanidad la capacidad de establecer metas generales y diseñar un camino para la toma de decisiones detallada, a fin de un mejor logro de las metas cuando se encaran situaciones prácticas de gran complejidad.

Las herramientas para realizar esto son : la formulación de problemas del mundo real en términos matemáticos detallados (modelación), técnicas para la solución de modelos (algoritmos) y tecnología para la ejecución de los pasos del algoritmo (computadoras y software).

Los problemas del mundo real de gran complejidad suelen tener miles de variables y de restricciones en su modelo matemático. Para la solución

de este tipo de problemas surgieron los métodos de puntos interiores y, con los nuevos avances tecnológicos, han resultado ser una herramienta poderosa.

Dos de estos métodos de puntos interiores son el algoritmo de escala afín primal y el algoritmo de escala afín dual, los cuales, si bien no han demostrado ser eficientes teóricamente, son algoritmos que se han utilizado exitosamente en la solución de problemas prácticos.

Dado que la aceptación de los métodos de puntos interiores para la solución de programas lineales (problemas de programación lineal) no es debida a sus ventajas teóricas, sino a su capacidad de solucionar problemas de gran complejidad en forma competitiva, se considera de suma importancia el analizar estos algoritmos para posibles mejoras e implementarlos para la solución de problemas de programación lineal. En este capítulo se presentan antecedentes históricos y fundamentos matemáticos de la programación lineal que permiten realizar un análisis adecuado de estos algoritmos.

I.1 ANTECEDENTES DE LA PROGRAMACIÓN LINEAL

La programación lineal (PL) estudia el problema de maximizar o minimizar una función lineal en presencia de igualdades y/o desigualdades lineales, conocidas como restricciones. Esta técnica recibió un gran impulso con el desarrollo del Método Simplex en 1947, así como por el advenimiento de las computadoras [Dan].

El Método Simplex se convirtió en el procedimiento de mayor uso para la solución de problemas de programación lineal (PPL). Este método recorre los vértices de la región factible (poliedro de solución) del PPL hasta alcanzar la solución óptima buscada.

Debido a que el Método Simplex alcanza la solución óptima al recorrer los vértices del poliedro, se observó que para cierta clase de problemas (desarrollados por Víctor Klee y George Minty [KyM]) el Simplex se ejecutaba de manera ineficiente, ya que el número de iteraciones requeridas para obtener la solución de los problemas crece exponencialmente de acuerdo al tamaño de éstos. Este hecho motivó la búsqueda de nuevos procedimientos para la solución de PPL.

En 1979, el matemático soviético L. G. Khachiyan [Kha] presentó un algoritmo que resuelve programas lineales y que es completamente distinto al Método Simplex. El algoritmo de Khachiyan posee dos propiedades importantes: Bajo ciertos criterios teóricos es más eficiente que el Método Simplex, y además proporciona información sobre la dificultad computacional de los programas lineales que es valiosa y que antes de Khachiyan no se había logrado probar. Así, el número de iteraciones requeridas por el método elipsoidal [Sou] (como posteriormente se conoció al algoritmo de Khachiyan) crece polinomialmente de acuerdo al tamaño del problema. Pero este método se mostró muy poco práctico, dado que presentaba una lenta convergencia a la solución óptima y un gran uso de memoria de computadora.

En 1984, un nuevo algoritmo matemático fue propuesto para la solución de problemas de programación lineal. El algoritmo fue inventado por el matemático hindú Narendra K. Karmarkar para los laboratorios AT&T en los Estados Unidos [Kar]. El Algoritmo de Karmarkar para Programación Lineal fue desarrollado como resultado de la búsqueda de un algoritmo que fuera eficiente teóricamente como el Método Elipsoidal, pero proporcionara facilidades prácticas de implementación como el Método Simplex.

A diferencia del Método Simplex que obtiene el punto óptimo (aquel que proporciona el valor máximo o mínimo del problema en cuestión) analizando las soluciones factibles en los vértices del poliedro de solución, el Algoritmo de Karmarkar analiza las soluciones factibles contenidas en el interior del poliedro hasta alcanzar la solución óptima (la cual se encuentra en la frontera del poliedro solución), por lo que es conocido como un método de puntos interiores, el cual empieza a ser empleado en la solución de problemas de programación lineal en gran escala.

Debido a las investigaciones sobre el algoritmo de Karmarkar, se han propuesto distintas variantes de éste, que se clasifican en tres principales grupos: i) algoritmos proyectivos [Kar], ii) algoritmos de escala afín [Bar] y iii) algoritmos de seguimiento de trayectorias [GMS]. Los algoritmos de la clase i) y iii) se muestran teóricamente eficientes, lo cual no ha podido ser probado para los de la clase ii), sin embargo, estos algoritmos, han mostrado un excelente comportamiento en la solución de programas lineales en la práctica. Por lo que este trabajo está enfocado a esta clase de algoritmos.

I.2 EL PROBLEMA DE PROGRAMACIÓN LINEAL.

En esta sección se define el problema de programación lineal y se describen sus principales propiedades matemáticas con el fin de establecer los conceptos necesarios para el desarrollo de los Métodos de Escala Afin. En primer término se realiza una presentación informal del PPL, posteriormente se proporcionan las definiciones y teoremas necesarios que lo fundamentan así como sus principales características.

I.2.1. Presentación Informal del PPL.

La programación lineal utiliza un modelo matemático para dar solución a problemas de la siguiente naturaleza general: se tiene alguna cantidad (tal como un costo o tiempo) que es una función lineal de cierto número de variables. Se requiere, a su vez, que estas variables satisfagan un sistema de igualdades o desigualdades lineales. Es necesario hallar aquellos valores no negativos de las variables que hagan máxima o bien mínima a la cantidad dada. El adjetivo lineal significa que todas las funciones matemáticas del modelo deben ser lineales. En este caso, la palabra programación es un sinónimo de planeación. Así, la PL trata la planeación de las actividades para obtener un resultado de la mejor forma posible, entre todas las alternativas de solución. Cualquier problema cuyo modelo matemático se ajuste al formato general del modelo de programación lineal es un problema de programación lineal.

Dado que el PPL presenta la necesidad de encontrar los puntos en una región en los cuales alguna función alcanza un valor máximo o mínimo,

resulta natural hacer la pregunta: ¿ Por qué no podemos usar los bien conocidos métodos clásicos del cálculo para hallar estos valores ? La respuesta es, que estos métodos clásicos requieren que cada una de las derivadas parciales de la función sean igualadas a cero. Esto debido, a que en los puntos máximos y mínimos de una función, la pendiente de la tangente, en dichos puntos, debe ser cero. Como la función es lineal, las derivadas parciales tendrán valores constantes, lo cual hará imposible igualar estas derivadas a cero y poder obtener los puntos críticos. Este hecho fue precisamente el que motivó la creación de nuevos métodos para abordar los problemas relacionados con funciones lineales y la programación lineal es uno de estos métodos.

Ahora se expondrán los conceptos necesarios para la definición formal del problema de programación lineal.

I.2.2. Poliedros.

Definición 1.1. Un *poliedro* es un conjunto definido por la intersección de un número finito de semiespacios¹ cerrados (sistema de desigualdades), es decir, un poliedro es de la forma $P = \{ x \in \mathbb{R}^n \mid Ax \leq b \}$ donde A es una matriz de $m \times n$, cuyo i -ésimo renglón es A_i , y b es un vector en \mathbb{R}^m , cuyo i -ésimo elemento es b_i . En este caso, cada una de las desigualdades $S_i = \{ x \in \mathbb{R}^n \mid A_i x \leq b_i \}$, $i = 1, \dots, m$, determina un semiespacio cerrado en \mathbb{R}^n con los hiperplanos² $H_i = \{ x \in \mathbb{R}^n \mid A_i x = b_i \}$, $i = 1, \dots, m$, que definen a P .

¹Ver definición de semiespacio en el Apéndice A.

²Ver definición de hiperplano en el Apéndice A.

Si existe por lo menos un punto $x^0 \in \mathbb{R}^n$ que satisfaga todas las desigualdades del sistema, se dice que el sistema es *consistente* y el poliedro asociado P es *no-vacío* ($P \neq \emptyset$). Si ningún punto en \mathbb{R}^n satisface las desigualdades del sistema, se dice que el sistema es *inconsistente* y el poliedro asociado P es *vacío* ($P = \emptyset$).

Se dice que $P \neq \emptyset$ es un *poliedro acotado* cuando existen $x^0 \in \mathbb{R}^n$ y $r > 0$ tales que $P \subset B(x^0, r)$, donde B es una bola abierta³ con centro en el vector x^0 y radio r . Un poliedro no vacío y acotado es conocido como un *politopo*.

La representación algebraica de un poliedro adquiere mayor flexibilidad con el siguiente teorema (consultar [Sou]), que es considerado como una generalización de la definición 1.1.

Teorema 1.1. Todo poliedro de la forma $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ se puede representar de cualquiera de las siguientes formas:

- i) $P = \{x \in \mathbb{R}^n \mid Cx \leq d, x \geq 0\}$
- ii) $P = \{x \in \mathbb{R}^r \mid Ex \geq f, x \geq 0\}$
- iii) $P = \{x \in \mathbb{R}^s \mid Gx = v, x \geq 0\}$

donde $n \leq q, r, s$.

La comprobación de este teorema puede ser efectuada en tres partes:

³Ver definición de bola abierta en el Apéndice A.

I) Sea $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ un poliedro, donde A es una matriz $m \times n$, o equivalentemente:

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b_i, i = 1, \dots, m\} \quad (I.1)$$

En (I.1) se puede sustituir cada componente x_j ($j = 1, 2, \dots, n$) de $x \in P$ por $x'_j - x_0$ donde $x'_j - x_0 \geq 0$. x_0 es la misma variable para cada x_j reemplazada. Así, el poliedro (I.1) se representa como $P = \{x \in \mathbb{R}^q \mid Cx \leq d, x \geq 0\}$ donde $C_i = [A_i \mid -\sum_{j=1}^n a_{ij}]$ con $i=1, \dots, m$, $d = b$, $x = [(x')^t, x_0]^t$ y $q = n+1$.

II) Representétese el poliedro $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ como $P = \{x \in \mathbb{R}^q \mid Cx \leq d, x \geq 0\}$ de acuerdo a lo indicado en I). Obsérvese que cada semiespacio $C_i x \leq d_i \Leftrightarrow -C_i x \geq -d_i \forall i = 1, \dots, m$, por lo cual $P = \{x \in \mathbb{R}^r \mid Ex \geq f, x \geq 0\}$ es válido y $E = -C$, $f = -d$ y $r = q$.

III) Expreséese nuevamente el poliedro $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ como $P = \{x \in \mathbb{R}^q \mid Cx \leq d, x \geq 0\}$ o equivalentemente como $P = \{x \in \mathbb{R}^q \mid C_i x \leq d_i, i = 1, \dots, m, x \geq 0\}$, tal como se indica en la primera parte de esta comprobación. Se observa que las desigualdades que componen al poliedro $C_i x \leq d_i$ pueden ser representadas como ecuaciones sumando la *variable de holgura* h_i en cada desigualdad, obteniéndose así $C_i x + h_i = d_i$, con $x \geq 0$ y $h \geq 0$. Similarmente se observa que las desigualdades que conforman al poliedro $E_i x \geq f_i$ pueden ser representadas como ecuaciones sustrayendo la variable h_i en cada desigualdad, obteniéndose $E_i x - h_i = f_i$ con $x \geq 0$ y $h \geq 0$.

Por lo tanto $P = \{x \in \mathbb{R}^s \mid Gx = v, x \geq 0\}$ es válido con $G = [C \mid I]$,⁴ $v = d$, $x = [x^t, h^t]^t$ o $G = [E \mid -I]$, $v = f$ y $x = [x^t, h^t]^t$; en los dos casos $s = q + m$. Se dice que un poliedro está en la *forma estándar* si todas las restricciones son ecuaciones y todas las variables son no-negativas. Así, es factible pasar de las representaciones I y II a la III (es decir, representar en la forma estándar a I y II) e inversamente, ya que $G_i x = v_i \Leftrightarrow G_i x \leq v_i$ y $G_i x \geq v_i$; obteniéndose las restricciones $C_i x \leq d_i$ al multiplicar por -1 a las $G_i x \geq v_i$; o las restricciones $E_i x \geq f_i$ al multiplicar por -1 a las $G_i x \leq v_i$.

Dado que $G_i x = v_i \Leftrightarrow G_i x \leq v_i$ y $G_i x \geq v_i$, obsérvese, que si ninguna desigualdad es multiplicada por -1 , se tendría un poliedro con restricciones de los dos tipos. Aún más, si en un poliedro $P = \{x \in \mathbb{R}^n \mid Gx = v, x \geq 0\}$ sólo algunas restricciones fueran reemplazadas por sus equivalentes $G_i x \leq v_i$ y $G_i x \geq v_i$, entonces el poliedro estaría conformado por restricciones de los dos tipos de desigualdades y por ecuaciones. De lo anterior se concluye que un poliedro puede estar representado como $P = \{x \in \mathbb{R}^n \mid A_1 x \leq b_1, A_2 x \geq b_2, A_3 x = b_3, x \geq 0\}$ donde A_1, A_2 y A_3 son matrices reales y b_1, b_2 y b_3 son vectores columna reales.

1.2.3 Conjuntos Afines y Conjuntos Convexos.

Para un estudio más detallado de los poliedros y los politopos se requiere la consideración de algunas propiedades básicas de los conjuntos afines y convexos [Baz] y [Fan].

⁴Ver definición de matriz identidad en el Apéndice A.

Dados k vectores $x_1, x_2, \dots, x_k \in \mathbb{R}^n$ y k escalares $\lambda_1, \lambda_2, \dots, \lambda_k \in \mathbb{R}$, la expresión $\lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_k x_k$ es conocida como una *combinación lineal*. Si $\lambda_1 + \lambda_2 + \dots + \lambda_k = 1$ se dice que la combinación lineal es una *combinación afín*. Si además $\lambda_1, \lambda_2, \dots, \lambda_k \geq 0$, se dice que la combinación lineal es una *combinación convexa*.

Desde el punto de vista geométrico, para cualesquiera x_1 y $x_2 \in \mathbb{R}^n$ y λ_1 y $\lambda_2 \in \mathbb{R}$, todas las posibles combinaciones convexas $\lambda_1 x_1 + \lambda_2 x_2$ con $\lambda_1 + \lambda_2 = 1$ y $\lambda_1, \lambda_2 \geq 0$ describen a un segmento rectilíneo entre x_1 y x_2 . Si ahora sólo consideramos la condición $\lambda_1 + \lambda_2 = 1$ para $\lambda_1 x_1 + \lambda_2 x_2$ y obtenemos todas las posibles combinaciones afines para x_1 y x_2 , se describirá a la recta completa que pasa por ambos puntos.

Como puede observarse una combinación convexa es una combinación afín, pero una combinación afín llega a ser convexa sólo cuando $x_1 = x_2$, es decir cuando geoméricamente se tiene un punto (caso degenerado).

De acuerdo a lo discutido anteriormente para un conjunto no vacío $S \subset \mathbb{R}^n$, diremos que S es *convexo* si para cualesquiera x_1 y $x_2 \in S$ todas sus posibles combinaciones convexas se encuentran contenidas en S . S es *afín* si contiene cualquier combinación afín de cualesquiera x_1 y $x_2 \in S$.

Así, observamos que los conjuntos afines son convexos pero los conjuntos convexos no necesariamente son afines. Además, la intersección de una colección (finita o infinita) de conjuntos afines es vacía o afín. Similarmente, la intersección de una colección (finita o infinita) de conjuntos convexos es vacía o convexa.

Como podemos notar los hiperplanos son conjuntos afines (y por tanto convexos) pero los semiespacios cerrados son convexos únicamente. El conjunto solución de un sistema finito de ecuaciones lineales $\{x \in \mathbb{R}^n \mid Ax = b\}$ es afín (y por tanto convexo), pero el poliedro $P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ es convexo únicamente. En general, de acuerdo a la definición 1.1 y a lo expuesto hasta estos momentos, todo poliedro es un conjunto convexo.

Dado un conjunto $S \subset \mathbb{R}^n$ y $x \in S$, se dice que x es un *punto interior* de S si existe un escalar $\epsilon > 0$ tal que la bola abierta $B(x, \epsilon) \subset S$. De otra forma x es un punto frontera de S .

Para un conjunto convexo $S \subset \mathbb{R}^n$, una propiedad geométrica importante es debida al siguiente teorema de separación:

Teorema 1.2 (de Separación). Sea $S \subset \mathbb{R}^n$ convexo y x un punto frontera de S . Entonces existe un hiperplano H que contiene a x , con S contenido en alguno de los semiespacios generados por H .

Para una prueba de este teorema consultar [MMT].

Basándonos en este teorema, podemos definir al *hiperplano de soporte* H como el hiperplano tal que:

- i) La intersección de H y S es no vacía.
- ii) S está totalmente contenido en alguno de los semiespacios generados por H .

En general, para un poliedro P y un hiperplano de soporte H , el conjunto $F = P \cap H$ es llamado una *cara* de P . Si F es un vértice de P entonces se tiene una cara de dimensión cero. Una arista de P es una cara de dimensión uno. Si la dimensión de poliedro P es d , donde $0 \leq d \leq n$, entonces las caras de P cuya dimensión es $d-1$ son llamadas *facetas*. Se dice que P es un *poliedro simple* cuando en cada una de sus vértices se intersectan exactamente d de los hiperplanos que definen a P . Si P es un poliedro simple y acotado se dice que P es un *simplejo*⁵.

I.2.4 Puntos Extremos y Soluciones Básicas Factibles.

Otro aspecto importante a considerar es el de los puntos extremos de un poliedro, los cuales son definidos a continuación:

Definición I.2. Sea S un conjunto convexo. Se dice que $x \in S$ es un punto extremo, si este no puede ser representado como una combinación lineal de cualesquiera otros dos puntos distintos de S .

En otras palabras, un punto extremo es aquel punto que no está dentro del segmento de línea que conecta a otros dos puntos diferentes del conjunto convexo. Para el caso de los poliedros, los puntos extremos son los vértices de éstos.

En especial caracterizaremos a los puntos extremos del poliedro $P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$. Debe recordarse del teorema I.1, que cualquier poliedro puede ser representado en esta forma (forma estándar).

⁵Ver propiedades del simplejo en el Apéndice A.

Supongamos que A es una matriz de $m \times n$ con $m \leq n$. Denotemos la j -ésima columna de A por A^j , donde $j = 1, 2, \dots, n$. Entonces para cada punto $x = (x_1, x_2, \dots, x_n)^t \in P$ tenemos

$$x_1 A^1 + x_2 A^2 + \dots + x_n A^n = b$$

Por lo tanto, nombraremos a la columna A^j como la columna correspondiente a la j -ésima componente x_j de x , para $j = 1, 2, \dots, n$.

A partir de esta afirmación podemos enunciar el siguiente teorema:

Teorema 1.3. Un punto $x \in P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ es un punto extremo de P si y sólo si las columnas de A correspondientes a las componentes positivas de x son linealmente independientes⁶ (l.l.).

Una prueba de este teorema puede consultarse en [Fan] y [MMT].

Cabe hacer notar que los puntos extremos son entidades geométricas, las cuales tienen asociadas a sus equivalentes algebraicas conocidas como soluciones básicas factibles. Cuando estos dos conceptos son ligados tenemos las herramientas algebraicas para solucionar problemas de PL guiados por la intuición geométrica.

Del teorema 1.3, se observa que si se tiene igual número de columnas l.l. que de ecuaciones, dichas columnas pueden ser agrupadas en una

⁶Ver definición de independencia lineal en el Apéndice A.

matriz llamada básica y las columnas restantes en otra llamada no básica, tal y como se indica en la siguiente definición.

Definición 1.3. Considere el conjunto de soluciones no negativas del sistema $Ax = b$, es decir

$$P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\} \quad (1.2)$$

donde $A \in \mathbb{R}^{m \times n}$ y $b \in \mathbb{R}^m$. Supóngase que el sistema posee soluciones y que las ecuaciones redundantes⁷ en $Ax = b$ han sido eliminadas. De este modo, puede suponerse sin pérdida de generalidad que la matriz A es de rango completo, es decir, A tiene m renglones y m columnas linealmente independientes. Reordenense las columnas del sistema de tal forma que $A=[B \mid N]$, donde B es una matriz invertible $m \times m$ llamada matriz básica o base⁸ y N es una matriz $m \times (n-m)$ llamada no - básica. Similarmente, al reordenarse las columnas vemos que $x = [x_B^t, x_N^t]^t$ donde x_B representa a las variable básicas y x_N representa a las variables no básicas. Así el sistema de ecuaciones $Ax = b$ se reinterpreta como $[B \mid N][x_B^t, x_N^t]^t = b$, es decir, $Bx_B + Nx_N = b$. Dado que B es una matriz invertible se puede resolver x_B en términos de x_N premultiplicando por B^{-1} para obtener:

$$x_B = B^{-1}b - B^{-1}Nx_N \quad (1.3)$$

⁷ Son aquellas ecuaciones que al ser omitidas no modifican el poliedro.

⁸ Ver definición de base en el Apéndice A.

Si en (I.3), el punto $x_N = 0$ entonces $x_B = B^{-1}b$ y se obtiene una solución básica asociada a B en (I.2). Si $x_B \geq 0$, se dice que x es una solución básica factible de (I.2). Esta es no degenerada si $x_B > 0$, y degenerada en caso de que al menos un componente de x_B sea cero.

Si la matriz A tiene rango completo, se tienen

$$C(n,m) = \frac{n!}{m!(n-m)!}$$

formas diferentes de seleccionar m columnas linealmente independientes de las n columnas de A , es decir, se tienen a lo más $C(n,m)$ soluciones básicas.

Los siguientes corolarios son consecuencia directa del teorema I.3.

Corolario I.3.1. Un punto $x \in P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ es un punto extremo de P si y sólo si x es una solución básica factible correspondiente a alguna base B .

Debe observarse que cualquier solución básica factible es una solución básica, por lo que tenemos el siguiente corolario.

Corolario I.3.2. Un poliedro representado en la forma estándar, contiene a lo más $C(n,m)$ puntos extremos.

I.2.5. Soluciones Básicas Factibles Degeneradas.

En la definición I.3 se mencionó que si al menos una componente de x_B es cero, la solución básica factible es degenerada. Este hecho es muy importante y para poder explicarlo retomaremos el corolario I.3.1. En él, se habla de la correspondencia entre las soluciones básicas factibles y los puntos extremos.

Debemos notar aquí que esta correspondencia no es uno a uno. Para cada solución básica factible existe un único punto extremo en P , pero cada punto extremo de P puede tener asociada más de una solución básica factible.

Esto es debido a que en alguna cara de dimensión cero (vértice) del poliedro se intersectan más de los m hiperplanos que definen al poliedro, es decir, no se tiene un poliedro simple.

Basándonos en esta observación, para que un poliedro sea no degenerado (simple), en cada uno de sus vértices (puntos extremos) deben intersectarse exactamente m de los hiperplanos que lo definen.

Algebraicamente, una solución básica factible es no degenerada si tiene exactamente m variables básicas positivas. En caso contrario la solución básica factible tiene al menos $n-m+1$ elementos cero y se dice que ésta es degenerada.

Un PPL es no degenerado si todas las soluciones básicas factibles son no degeneradas. En este caso, existe una correspondencia uno a uno entre los puntos extremos y las soluciones básicas factibles.

1.2.6 Definición del Problema de Programación Lineal.

En esta sección se dará la definición del Problema de Programación Lineal, así como la representación en su forma estándar y en su forma canónica y, posteriormente, se enunciará el teorema fundamental de la PL.

Definición 1.4. Considérense las matrices reales A_1 , A_2 y A_3 y, los vectores columna b_1 , b_2 , b_3 y c con componentes reales, todos ellos con dimensiones adecuadas. El Problema de Programación Lineal se define como:

$$\text{Min } z = c^t x$$

s. a

$$A_1 x \leq b_1 \quad (1.4)$$

$$A_2 x \geq b_2 \quad (1.5)$$

$$A_3 x = b_3 \quad (1.6)$$

$$x \geq 0 \quad (1.7)$$

Así, la programación lineal trata el problema de optimizar una función lineal en presencia de desigualdades y ecuaciones lineales, es decir, encontrar el vector x^* que satisface (1.4), (1.5), (1.6) y (1.7) y, para el cual la *función objetivo* $z = c^t x$ obtiene su valor mínimo. Si se denota por P el poliedro (región factible) formado por las soluciones en \mathbb{R}^n de las desigualdades (1.4), (1.5) y (1.7) y de las ecuaciones (1.6), en donde cada punto $x \in P$ es llamado *solución factible*, entonces, el problema se puede expresar como: encontrar $x^* \in P$ tal que $c^t x^* \leq c^t x \quad \forall x \in P$. A x^* se le llama la *solución óptima* de P y a $z^* = c^t x^*$ el valor objetivo óptimo de P . Si

existen dos o más soluciones óptimas, éstas son llamadas *alternativas*, es decir, el problema tiene *soluciones factibles óptimas múltiples*.

Sin embargo, el problema así planteado puede no tener solución, debido a que el sistema de desigualdades y ecuaciones es inconsistente (infactible) y el poliedro $P = \emptyset$; o bien, si $P \neq \emptyset$ (el sistema es consistente, es decir, tiene soluciones factibles) y no acotado hacia abajo.

Puede observarse que $c^T x^* \leq c^T x \quad \forall x \in P \Leftrightarrow -c^T x^* \geq -c^T x \quad \forall x \in P$, de tal forma que todo problema de programación lineal puede ser formulado como un problema de minimización o maximización indistintamente ya que $\text{Min } z = - \text{Max } -z$. Además como resultado del Teorema I.1, se sabe que la región factible P puede ser representada en múltiples formas que equivalen a un mismo problema de programación lineal.

En especial, un problema de programación lineal de minimización se puede representar en su forma estándar como

$$\begin{aligned} \text{Min } z &= c^T x \\ \text{s. a} \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

y, en la forma canónica como

$$\begin{aligned} \text{Min } z &= c^T x \\ \text{s. a} \\ Ax &\geq b \\ x &\geq 0. \end{aligned}$$

Dado lo expuesto hasta estos momentos, enunciaremos ahora el teorema fundamental de la programación lineal (su demostración puede ser consultada en [Fan]).

Teorema 1.4 (Teorema Fundamental de la Programación lineal). Para un problema de programación lineal consistente en su forma estándar con región factible P , la función objetivo $z = c^T x$ sobre P cumple uno y solo uno de los siguientes casos:

- 1) Alcanza el valor mínimo en al menos uno de los puntos extremos de P .
- 2) No alcanza el valor mínimo dado que $z = c^T x$ no está acotada hacia abajo.

Es importante hacer notar que el teorema anterior no excluye la posibilidad de tener una solución óptima en un punto no extremo. Este simplemente dice que entre todas las soluciones óptimas para un PPL dado, al menos una de éstas se encuentra en un punto extremo.

1.3 EL PROBLEMA DE PROGRAMACIÓN LINEAL DUAL.

Otro aspecto importante del problema de programación lineal lo constituye el problema dual, así como las propiedades que lo relacionan con el PPL original.

1.3.1 Definición del PPL Dual.

Definición 1.5. Considere el problema de programación lineal de minimización en su forma canónica; al cual llamaremos el PPL primal (P). Entonces el problema dual (D) asociado a (P) es :

<p>(P)</p> <p>Min $z = c^t x$</p> <p>s. a</p> <p>$Ax \geq b$</p> <p>$x \geq 0$</p>	<p>(D)</p> <p>Max $w = b^t y$</p> <p>s. a</p> <p>$A^t y \leq c$</p> <p>$y \geq 0.$</p>
-----------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------

La dualidad puede caracterizarse a través de la siguiente afirmación: Para todo problema de minimización de programación lineal existe un problema asociado de maximización; y a la inversa, para todo problema de maximización de programación lineal existe un problema asociado de minimización.

Se puede verificar que las soluciones de los problemas (P) y (D) están en \mathbb{R}^n y \mathbb{R}^m respectivamente. Se observa que m , n , A , b y c definen a ambos problemas.

I.3.2 Propiedades de los PPL Primal y Dual.

El problema dual tiene varias propiedades relacionadas con el primal, las cuales se enumeran a continuación (consultar [Baz] y [Fan]).

Propiedad 1: Si x^* e y^* son soluciones factibles de los problemas primal y dual respectivamente entonces, se cumple que $c^T x^* \geq b^T y^*$.

Propiedad 2: Dados el problema primal y su dual se cumple uno y sólo uno de los siguientes casos:

- i) Ambos problemas poseen soluciones óptimas x^* e y^* , en cuyo caso $c^T x^* = b^T y^*$, es decir el valor objetivo óptimo es el mismo para los dos.
- ii) Uno de los problemas es no acotado y el otro no posee soluciones factibles.
- iii) Ninguno de los dos problemas posee soluciones factibles.

Propiedad 3: Considérense los problemas primal (P) y dual (D). Entonces (P) y (D) poseen una solución óptima si y sólo si el siguiente sistema posee una solución:

$$Ax \geq b$$

$$x \geq 0$$

$$A^t y \leq c$$

$$y \geq 0$$

$$c^t x = b^t y.$$

La propiedad 1 establece que el valor de la función objetivo en el problema de minimización es una cota superior del valor de la función objetivo en el problema de maximización. Esto nos permite obtener un intervalo cuyo límite inferior es el valor de la función objetivo del PPL de maximización y cuyo límite superior es el valor de la función objetivo del PPL de minimización. Dicho intervalo (en inglés, *duality gap*) contendrá, en caso de existir, al valor de la función objetivo óptimo para ambos problemas.

En la propiedad 2 debe observarse que la dualidad no es completamente simétrica. Lo más que puede decirse es que

Tabla 1.1. Conclusiones de la Propiedad 2 (consultar [Baz]).

(P) ÓPTIMO	\Rightarrow	(D) ÓPTIMO.
(P) NO ACOTADO	\Rightarrow	(D) NO FACTIBLE.
(D) NO ACOTADO	\Rightarrow	(P) NO FACTIBLE.
(P) NO FACTIBLE	\Rightarrow	(D) O NO ACOTADO O NO FACTIBLE.
(D) NO FACTIBLE	\Rightarrow	(P) O NO ACOTADO O NO FACTIBLE.

Además en el primer caso de la propiedad 2 el valor óptimo para las funciones objetivo de los dos problemas es igual, por lo que ambos PPL

poseen soluciones básicas óptimas y estas soluciones básicas son complementarias. Cabe hacer notar que algunos autores refieren a esta propiedad como el teorema fundamental o teorema fuerte de dualidad (ver [Baz] o [Fan] por ejemplo).

Es importante observar también que el sistema correspondiente a la propiedad 3 tiene solución sólo si (P) y (D) tienen soluciones óptimas. Este sistema se utilizará para la obtención de un PPL en la forma canónica de Karmarkar en el siguiente capítulo.

Un resumen de la relación existente entre los problemas primal y dual es presentado en la siguiente tabla (consultar [Baz]):

Tabla 1.2. Relaciones entre los PPL primal y dual.

Problema de Minimización		Problema de Maximización
<i>Si las variables son :</i>		<i>Las restricciones asociadas son :</i>
≥ 0	\Rightarrow	\leq
≤ 0	\Rightarrow	\geq
No restringidas	\Rightarrow	=
<i>Si las restricciones son :</i>		<i>Las variables asociadas son :</i>
\geq	\Rightarrow	≥ 0
\leq	\Rightarrow	≤ 0
=	\Rightarrow	No restringidas

Se puede utilizar esta tabla para obtener el problema lineal dual de cualquier PPL.

I.3.3 Holguras Complementarias.

Otra forma de caracterizar la relación entre los problemas primal y dual es a través de la propiedad conocida como *holguras complementarias*. Para ello consideremos nuevamente a los problemas (P) y (D) descritos en la definición I.5.

Para el problema primal, definimos

$$u = Ax - b \geq 0,$$

como el vector de holguras primal. Para el problema dual, definimos

$$v = c - A^t y \geq 0,$$

como el vector de holguras dual (observe que estos vectores contienen a las variables de holgura descritas en la tercera parte de la comprobación del teorema I.1, necesarias para la obtención de PPL en su forma estándar). Note que $u \in \mathbb{R}^m$ y $v \in \mathbb{R}^n$. Además, para cualquier solución factible primal x y cualquier solución factible dual y , tenemos

$$0 \leq v^t x + u^t y,$$

$$0 \leq (c^t - y^t A)x + y^t (Ax - b),$$

$$0 \leq c^t x - b^t y.$$

Por lo tanto, la cantidad de $v^t x + u^t y$ es igual al intervalo dual entre la solución factible primal x y la solución factible dual y , es decir

$$0 \leq v^t x + u^t y = c^t x - b^t y. \quad (1.8)$$

La expresión (1.8) es conocida como la *ecuación de dualidad*. El intervalo dual se anula si y sólo si,

$$v^t x = 0 \quad \text{y} \quad u^t y = 0. \quad (1.9)$$

En este caso, x es una solución óptima primal e y es una solución óptima dual. Como los vectores x , v , u e y son no negativos, la primera ecuación de (1.9) requiere que, si $v_j > 0$ entonces $x_j = 0$ o que, si $x_j > 0$ entonces $v_j = 0$ para $j = 1, \dots, n$; la segunda ecuación requiere que, si $u_i > 0$ entonces $y_i = 0$ o que, si $y_i > 0$ entonces $u_i = 0$ para $i = 1, \dots, m$. Por lo que las ecuaciones de (1.9) son conocidas como *las condiciones de holguras complementarias*. Este importante resultado es sumariado en el siguiente teorema cuya demostración puede consultarse en [MyO]:

Teorema 1.5 (Teorema de Holguras Complementarias). Sea x una solución factible primal y sea y una solución factible dual para un par de programas lineales (P) y (D). Entonces x e y son soluciones óptimas de (P) y (D) respectivamente, si y sólo si las condiciones de holguras complementarias

$$\begin{aligned} \text{i) } & v_j = (c - A^t y)_j = 0 \text{ y } x_j > 0 \text{ o} \\ & v_j = (c - A^t y)_j > 0 \text{ y } x_j = 0, \text{ para } j = 1, \dots, n; \end{aligned}$$

y

$$\begin{aligned} \text{ii) } & u_i = (Ax - b)_i = 0 \text{ e } y_i > 0 \text{ o} \\ & u_i = (Ax - b)_i > 0 \text{ e } y_i = 0, \text{ para } i = 1, \dots, m. \end{aligned}$$

se satisfacen.

El teorema de holguras complementarias también se puede enunciar como sigue: en caso de optimalidad, "si una variable en uno de los problemas es positiva, entonces la restricción correspondiente en el otro problema es sin holgura"; y "si una restricción en uno de los problemas es con holgura, entonces la variable correspondiente en el otro problema debe ser cero". Una vez conocidas las condiciones de holguras complementarias podemos exponer las condiciones de optimalidad necesarias para verificar si una solución factible es óptima.

I.3.4 Condiciones de Optimalidad de Karush - Kuhn - Tucker.

Antes de enunciar las condiciones de optimalidad, debemos observar que para el PPL de minimización en forma estándar

$$\begin{aligned} \text{Min } z &= c^T x \\ \text{s. a} & \quad (I.10) \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

y su correspondiente PPL dual (obtenido haciendo uso de la tabla 1.1)

$$\begin{aligned} \text{Max } w &= b'y \\ \text{s. a} \\ A'y &\leq c, \end{aligned} \quad (\text{I.11})$$

la condición de holgura complementaria $u'y = 0$ automáticamente se satisface (debido a que las restricciones explícitas de (I.10) están en igualdad). Por tanto, las condiciones de holguras complementarias para los PL (I.10) y (I.11) se simplifican a $v'x = 0$.

Ahora estamos en posibilidades de enunciar las condiciones necesarias y suficientes para que un PPL tenga solución óptima.

Teorema I.6. (Condiciones de Optimalidad de Karush - Kuhn - Tucker para PPL). Dado el problema de programación lineal (I.10); el vector x es solución óptima de este problema si y sólo si existen vectores v e y tales que

- i) $Ax = b, \quad x \geq 0$ (factibilidad primal)
- ii) $A'y + v = c, \quad v \geq 0.$ (factibilidad dual)
- iii) $v'x = 0$ (holguras complementarias)

en este caso, y es una solución óptima para el problema (I.11).

La condición i) se refiere al hecho de que el vector óptimo x sea factible; lo mismo sucede en la condición ii) para los vectores v e y . La condición iii) indica que $c'x = b'y$, como se vio anteriormente; por tanto y

debe ser solución óptima del problema dual. La demostración de este teorema puede ser consultada en [Baz].

Como puede observarse los teoremas y condiciones enunciados anteriormente, son consecuencia de las propiedades expuestas en 1.3.2, las cuales relacionan a los PPL primal y dual.

1.3.5 El Vector Dual.

Para concluir esta sección obtendremos la representación del *vector dual* y, con respecto a la definición de las variables básicas en el problema primal.

Considere el PPL (1.10). Por la definición 1.3, las columnas de la matriz A pueden reordenarse de tal forma que $A = [B | N]$; las variables como $x = [x_B, x_N]^t$ y el vector de coeficientes de la f.o. como $c = [c_B, c_N]^t$.

De este modo (1.10) se reescribe de la siguiente manera:

$$\begin{aligned} \text{Min } z &= c_B^t x_B + c_N^t x_N \\ \text{s. a} \\ Bx_B + Nx_N &= b \quad (1.12) \\ x_B, x_N &\geq 0 \end{aligned}$$

Si $x_N = 0$, entonces en (1.12), $b = Bx_B$. De la propiedad 1 de dualidad, sabemos que $c^t x \geq b^t y$, equivalentemente

$$c_B^t x_B \geq y^t b \quad (I.13)$$

Sustituyendo b en (I.13) tenemos $c_B^t x_B \geq y^t B x_B$. Si analizamos el caso de optimalidad, por la propiedad 2 de dualidad se debe cumplir que $c_B^t x_B = y^t B x_B$; para que esta desigualdad se satisfaga es necesario que

$$c_B^t = y^t B.$$

Dado que B es no singular, se sigue que

$$y = (B^t)^{-1} c_B. \quad (I.14)$$

Así (I.14) expresa la relación entre las variables duales y la base primal. En este caso y es conocido como el *vector dual*.

Hasta este momento se ha expuesto la teoría necesaria para trabajar con problemas de programación lineal. En el siguiente capítulo se presentará el algoritmo de escala proyectiva de Karmarkar para la solución de problemas de programación lineal.

CAPÍTULO II

MÉTODO DE ESCALA PROYECTIVA DE KARMARKAR

En este capítulo se expondrán los nuevos conceptos para solucionar PPL que aparecieron con el Algoritmo de Karmarkar [Kar], con lo cual se tendrá una base más sólida para el estudio de los Métodos de Escala Afín en el siguiente capítulo. Para una consulta adicional acerca de este algoritmo refiérase a [Baz], [Cal], [Fan], [Her], [Hoo] y [RyS].

II.1 IDEAS BÁSICAS DEL ALGORITMO DE KARMARKAR.

Posterior a que los ejemplos de Klee y Minty [KyM] mostraron que el Método Simplex debía visitar casi todos los puntos extremos de la región factible antes de llegar a detenerse (por lo que en el caso de poliedros con miles de puntos extremos sería inadecuado), surgió la necesidad de buscar nuevas opciones para la solución de este problema.

Una idea alternativa fue la de moverse a través de la región factible a lo largo de una ruta corta, a fin de reducir el número total de iteraciones. Sin embargo el recorrido sobre los puntos interiores del poliedro de soluciones, requiere la consideración de todas las direcciones factibles para un mejor movimiento en cada iteración. Es decir, este nuevo concepto reduce el número de iteraciones requeridas pero aumenta el número de cálculos en cada iteración.

En general, no es una tarea sencilla el identificar la mejor dirección de movimiento entre todas las direcciones factibles, a partir de un punto interior del poliedro de soluciones. Sin embargo Karmarkar observó dos aspectos fundamentales, suponiendo que la región factible es un politopo.

1. Si la solución interior actual está cerca del centro del politopo, entonces tiene sentido moverse en la dirección de mayor descenso de la función objetivo (f.o.) para alcanzar un valor mínimo.

2. Sin cambiar el problema en una forma esencial, una transformación puede ser aplicada sobre el PPL, tal que la solución interior actual esté colocada cerca del centro del poliedro de soluciones transformado.

La primera idea se ilustra en la figura 2.1. Dado que x^1 está cercano al centro del politopo, podemos mejorar la solución substancialmente al movernos en la dirección de mayor descenso. Pero si nos movemos a partir del punto x^2 el cual no está cercano al centro del politopo, llegaremos a estar fuera de la región factible antes de que alguna mejora sea realizada.

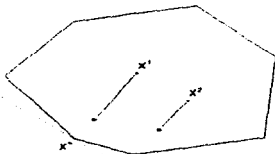


Figura 2.1

Con estas ideas en mente, la estrategia básica del Algoritmo de Karmarkar consiste en tomar una solución interior, transformar el PPL de manera que dicha solución este cercana al centro del politopo en el PPL transformado, y entonces moverse a partir de este punto en la dirección de mayor descenso, teniendo cuidado de que la nueva solución obtenida permanezca en el interior de la región transformada. A continuación aplicar la transformación inversa que mapea la nueva solución (mejorada) de regreso al poliedro de soluciones original como un nuevo punto interior. Este proceso debe ser repetido hasta que el óptimo sea obtenido con la exactitud deseada.

II.2 LA FORMA CANÓNICA DE KARMARKAR.

En la sección anterior se mencionó que las observaciones hechas por Karmarkar eran válidas para una región factible acotada, por lo que es necesario que el PPL sea transformado para obtener dicha región.

II.2.1. Obtención de un Polítopo.

Dado que el poliedro asociado a la región factible de un PPL no siempre es acotado, Karmarkar decidió transformar el PPL de forma tal, que en el nuevo planteamiento la región factible fuera invariablemente un politopo.

Para ello, ideó la siguiente representación de un PPL :

$$\begin{aligned} & \text{Min } \bar{c}^t \bar{x} \\ & \text{s. a} \\ & \bar{x} \in \Omega \cap \Delta^n \end{aligned} \quad (\text{II.1})$$

donde $\bar{c}, \bar{x} \in \mathbb{R}^{n+1}$, $\Omega = \{\bar{x} \mid \bar{A}\bar{x} = 0\}$. \bar{A} es una matriz $m \times (n+1)$ dimensional con componentes reales y rango completo; Ω representa el subespacio nulo de la matriz de restricciones y, $\Delta^n = \{\bar{x} \mid e^t \bar{x} = 1, \bar{x} \geq 0\}$ es un simplejo n dimensional donde $e \in \mathbb{R}^{n+1}$ y $e^t = (1, 1, \dots, 1)$.

En este caso la región factible está formada por la intersección de un sistema de ecuaciones lineales homogéneo y un simplejo n -dimensional contenido en \mathbb{R}^{n+1} . Debido a sus características, el simplejo Δ^n es un poliedro acotado y al intersectarlo con Ω la región resultante es también acotada. Así, la región factible de (II.1) es un politopo. Por otro lado una solución factible \bar{x} de (II.1) es considerada una solución interior si todos sus componentes son estrictamente positivos, es decir $\bar{x} > 0$.

II.2.2. Condiciones de Karmarkar.

Karmarkar impone además dos condiciones al problema (II.1):

- II.1.1) El centro $a_0 = e/(n+1)$ del simplejo Δ^n es una solución interior inicial.
- II.1.2) El valor objetivo óptimo del problema (II.1) es cero.

El problema (II.1) junto con II.1.1) y II.1.2) conforman lo que en este trabajo conoceremos como un PPL en la forma canónica de Karmarkar

(FCK). Este planteamiento junto con las condiciones puede parecer muy restrictivo, pero como veremos en la sección II.3 cualquier PPL puede ser puesto en la FCK.

II.2.3. Interpretación Geométrica.

Para poder interpretar geoméricamente que sucede con un PPL que es transformado a la FCK, consideremos la siguiente región factible (con puntos extremos $(0,0)$, $(0,r)$ y $(f,0)$) contenida en el ortante⁹ positivo de 2 dimensiones \mathbb{R}^2 .

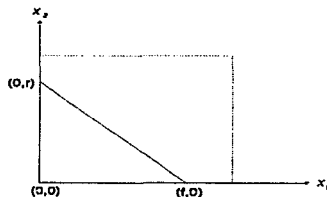


Figura 2.2

Como puede observarse en la fig. 2.2, cualquier función que esté en \mathbb{R}^2 está acotada por los ejes X_1 y X_2 (hacia abajo y hacia la izquierda respectivamente), pero puede crecer sin límite hacia arriba y/o a la derecha (línea punteada).

⁹Ver definición de ortante en el Apéndice A.

Al transformar \mathbb{R}_+^2 y la región factible como se requiere en la forma canónica de Karmarkar, se obtiene una región como la siguiente:

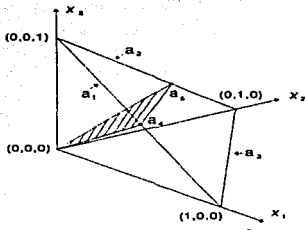


Figura 2.3

Ahora tenemos un simplejo de 2 dimensiones, es decir un plano intersectado con otro plano que pasa por el origen (de \mathbb{R}_+^3), generando como región factible a un segmento de recta (que se encuentra entre los puntos a_4 y a_5) en \mathbb{R}^3 .

En Δ^2 (fig. 2.3), el segmento de recta a_1 representa al eje x_1 , el segmento a_2 al eje x_2 y a_3 a la línea al infinito (es decir, a las líneas punteadas en la fig. 2.2). Así, el ortante \mathbb{R}_+^2 ha quedado compactado en una región triangular acotada contenida en \mathbb{R}^3 . Observe además que los puntos extremos $(0,0)$, $(f,0)$ y $(0,r)$ de la región factible en \mathbb{R}_+^2 han sido transformados en los puntos $(0,0,1)$, $a_4 = (f,0,1-f)$ y $a_5 = (0,r,1-r)$ respectivamente, con $0 \leq f \leq 1$ y $0 \leq r \leq 1$. La transformación de los puntos de \mathbb{R}_+^2 a $\Delta^2 \subset \mathbb{R}_+^3$ será tratada en la siguiente sección.

II.3. TRANSFORMACIÓN DE UN PPL A LA FORMA CANÓNICA DE KARMARKAR.

En esta sección veremos ahora como transformar un PPL a la FCK, es decir, la intersección de un simplejo con un sistema de ecuaciones homogéneo, satisfaciendo las condiciones impuestas.

II.3.1. Combinación de los Problemas Primal y Dual.

Considere el siguiente PPL en su forma canónica:

$$\begin{aligned}
 \text{Min } z &= c^t x \\
 \text{s. a} \\
 Ax &\geq b && \text{(II.2)} \\
 x &\geq 0.
 \end{aligned}$$

Como se vió en el capítulo anterior, cualquier PPL puede ser expresado en esta forma. Procederemos entonces a transformar este problema en la FCK.

De acuerdo a la definición I.5, el PPL dual asociado a (II.2) es

$$\begin{aligned}
 \text{Max } w &= b^t y \\
 \text{s. a} \\
 A^t y &\leq c && \text{(II.3)} \\
 y &\geq 0.
 \end{aligned}$$

Por la propiedad 3 de dualidad (sección I.3.2), tanto el problema (II.2) como (II.3), tienen solución óptima sólo si

$$\begin{aligned} Ax &\geq b \\ x &\geq 0 \\ A^t y &\leq c \\ y &\geq 0 \\ c^t x - b^t y &= 0. \end{aligned} \quad (\text{II.4})$$

También, por dicha propiedad, se sabe que (II.4) tiene solución si los PPL (II.2) y (II.3) tiene solución óptima.

Introduciendo variables de holgura ($u \in \mathbb{R}^m$ y $v \in \mathbb{R}^n$) al sistema anterior, se obtiene:

$$\begin{aligned} Ax - u &= b \\ A^t y + v &= c \\ c^t x - b^t y &= 0 \\ x \geq 0, y \geq 0, u \geq 0, v \geq 0; \end{aligned}$$

el cual es un sistema de ecuaciones en el primer ortante que al ser resuelto proporciona las soluciones óptimas de (II.2) y (II.3) respectivamente.

II.3.2 Obtención de un Punto Interior Inicial y de una Función Objetivo con Valor Mínimo Cero.

Ahora crearemos un punto interior inicial, para ello considérense los siguientes vectores x_0 , y_0 , u_0 y v_0 con componentes positivas reales y de

tamaño adecuado, y a la variable artificial $\lambda \geq 0$. Entonces postulemos el siguiente problema:

$$\begin{aligned} & \text{Min } \lambda \\ & \text{s. a} \\ & Ax - u + \lambda(b - Ax_0 + u_0) = b \\ & A^t y + v + \lambda(c - A^t y_0 - v_0) = c \quad (\text{II.5}) \\ & c^t x - b^t y + \lambda(-c^t x_0 + b^t y_0) = 0 \\ & x, y, u, v, \lambda \geq 0. \end{aligned}$$

Como puede observarse este es un PPL en la forma estándar, y si hacemos $x = x_0$, $y = y_0$, $u = u_0$, $v = v_0$ y $\lambda = 1$, obtenemos una solución interior inicial. Además la función objetivo de este PPL es cero si (II.4) tiene solución.

Por conveniencia reescribiremos (II.5). Para ello observe que se tienen n variables de decisión y m de holgura del problema primal, m variables de decisión y n de holgura del dual y tanto del PPL primal como del dual una variable artificial. Es decir, (II.5) tiene $n_1 = 2n + 2m + 1$ variables en total. En cuanto a las restricciones se tienen m del primal, n del dual y una para la optimalidad de las f.o de los PPL (II.2) y (II.3). Por lo que en este caso (II.5) tiene $m_1 = m + n + 1$ restricciones.

Así pues considérense a $q, h \in \mathbb{R}^{n_1}$ donde $q = (0, 0, \dots, 0, 1)^t$ y $h = (x^t, y^t, u^t, v^t, \lambda)^t$, además considérense a $G \in \mathbb{R}^{m_1 \times n_1}$ y $r \in \mathbb{R}^{m_1}$ donde $r = (b^t, c^t, 0)^t$. En este caso la matriz G , a partir de (II.5), tiene la siguiente estructura:

$$G = \left[\begin{array}{c|c|c|c|c} A_{m \times n} & 0_{m \times m} & -I_{m \times m} & 0_{m \times n} & (b - Ax_0 + u_0)_{m \times 1} \\ \hline 0_{n \times n} & A^t_{n \times m} & 0_{n \times m} & I_{n \times n} & (c - A^t y_0 - v_0)_{n \times 1} \\ \hline c^t_{1 \times n} & -b^t_{1 \times m} & 0_{1 \times m} & 0_{1 \times n} & (-c^t x_0 + b^t y_0)_{1 \times 1} \end{array} \right]$$

donde 0_{ij} es una matriz cuyos elementos son ceros en los i renglones y j columnas.

Dado lo anterior, el PPL (II.5) será rescrito como:

$$\begin{aligned} \text{Min } q^t h \\ \text{s. a} \\ Gh = r \quad (\text{II.6}) \\ h \geq 0. \end{aligned}$$

En este caso $h = \underline{a}$, donde $\underline{a} = (x_0^t, y_0^t, u_0^t, v_0^t, 1)^t$, es una solución interior inicial conocida y el valor mínimo de la f.o. en (II.6) es cero, si el problema (II.2) tiene solución óptima.

II.3.3 Transformación del PPL

Procederemos ahora a transformar el PPL (II.6) para obtener la intersección del simplejo Δ^n con el sistema homogéneo Ω . Para llevar a cabo esto, se utilizará una transformación proyectiva, la cual se define a continuación.

Definición II.1. Sea $T: \mathbb{R}_+^n \rightarrow \Delta^n \subset \mathbb{R}^{n+1}$ una transformación proyectiva tal que:

$$T(h) = \begin{cases} \bar{h}_k = \frac{h_k \underline{a}_k}{\sum_{j=1}^n \frac{h_j}{\underline{a}_j} + 1} & \text{donde } k = 1, \dots, n \\ y \\ \bar{h}_{n+1} = \frac{1}{\sum_{j=1}^n \frac{h_j}{\underline{a}_j} + 1} \end{cases} \quad (11.7)$$

donde $\bar{h} \in \Delta^n$; $h, \underline{a} \in \mathbb{R}_+^n$ y \underline{a} es una solución interior.

Esta transformación proyectiva tiene las siguientes propiedades (las demostraciones de cada una de estas propiedades pueden ser consultadas en [Her]):

- i) T mapea a \mathbb{R}_+^n en Δ^n .
- ii) T es uno a uno y T^{-1} está dada por

$$T^{-1}(\bar{h}_k) = h_k = \frac{\underline{a}_k \bar{h}_k}{\bar{h}_{n+1}} \quad \text{para } k = 1, \dots, n$$
- iii) T mapea a la solución interior \underline{a} al centro de Δ^n .
- iv) T mapea los puntos extremos de \mathbb{R}_+^n a las caras $\bar{h}_k = 0$ de Δ^n , $k=1, \dots, n$.
- v) T mapea los puntos infinitos de \mathbb{R}_+^n en la cara $\bar{h}_{n+1} = 0$ de Δ^n .
- vi) T mapea el sistema de ecuaciones $Gh = r$ a un sistema homogéneo, debido a la componente adicional \bar{h}_{n+1} , la cual es usada para eliminar los términos constantes.

En nuestro caso, dado que estamos interesados en aplicar la transformación (II.7) al PPL (II.6), $\bar{h} \in \mathbb{R}^{n+1}$ por lo que T^{-1} se escribe como :

$$T^{-1}(\bar{h}_k) = h_k = \frac{a_k \bar{h}_k}{h_{n+1}} \quad \text{donde } k = 1, \dots, n_1. \quad (\text{II.8})$$

Antes de aplicar la transformación, reescribiremos el problema (II.6) como sigue:

$$\text{Min } (0, 0, \dots, 0, 1)(h_1, h_2, \dots, h_{n_1})^t$$

s. a

$$\sum_{j=1}^{n_1} a_{ij} h_j = \bar{q}_i; \quad i = 1, \dots, m_1$$

$$h_j \geq 0; \quad j = 1, \dots, n_1.$$

Aplicando a este problema la transformación (II.8) (para que el PPL quede en términos de $\bar{h} \in \Delta^{n_1}$) tenemos :

$$\text{Min } \frac{a_{n_1+1} \bar{h}_{n_1+1}}{h_{n_1+1}}$$

s. a

$$\left(\frac{1}{h_{n_1+1}} \right) \sum_{j=1}^{n_1} a_{ij} \bar{h}_j = \bar{q}_i; \quad i = 1, \dots, m_1 \quad (\text{II.9})$$

$$\sum_{j=1}^{n_1+1} \bar{h}_j = 1$$

$$\bar{h}_j \geq 0; \quad j = 1, \dots, n_1+1.$$

Pero (II.9) resulta ser un problema de programación no lineal (debido a que la función objetivo y las restricciones son no lineales). Considerando que el PPL (II.6) alcanza su valor mínimo en cero, (II.9) tiene la misma solución que el siguiente PPL.

$$\begin{aligned} & \text{Min } \underline{a}_{n1} \bar{h}_{n1} \\ & \text{s. a} \\ & \sum_{j=1}^{n1} \underline{a}_{ij} \bar{h}_j - \bar{h}_{n1+r_i} = 0 ; \quad i = 1, \dots, m1 \quad (\text{II.10}) \\ & \sum_{j=1}^{n1+1} \bar{h}_j = 1 \\ & \bar{h}_j \geq 0 ; j = 1, \dots, n1+1. \end{aligned}$$

Sea $n2 = n1+1$. Consideremos ahora a los vectores $\bar{c}, \bar{x} \in \mathbb{R}^{n2}$ y a la matriz $\bar{A} \in \mathbb{R}^{m1 \times n2}$ tales que :

$$\begin{aligned} \bar{x} &= \bar{h}, \\ \bar{c} &= (0, \dots, 0, \underline{a}_{n1}, 0)^t, \\ \bar{A} &= [GL \mid -r], \end{aligned}$$

donde $L = \text{diag}(\underline{a}_1, \dots, \underline{a}_{n1})$ es una matriz diagonal¹⁰ de dimensión $n1 \times n1$. Por lo que la matriz \bar{A} está formada por el producto de las matrices G y L , y por una columna adicional correspondiente al vector $-r$.

Así el problema (II.10) puede reescribirse como:

¹⁰Ver definición de matriz diagonal en el Apéndice A.

$$\begin{aligned}
 & \text{Min } \bar{c}'\bar{x} \\
 & \text{s. a} \\
 & \quad \bar{A}\bar{x} = 0 \qquad \qquad \qquad (\text{II.11}) \\
 & \quad e'\bar{x} = 1 \\
 & \quad \bar{x} \geq 0.
 \end{aligned}$$

Como puede observarse (II.11) es ya un PPL en la FCK, ya que $\Omega = \{\bar{x} \mid \bar{A}\bar{x} = 0\}$, $\Delta^1 = \{\bar{x} \mid e'\bar{x} = 1, \bar{x} \geq 0\}$, $e \in \mathbb{R}^{n^2}$ y además, las condiciones II.1.1) y II.1.2) se satisfacen, pues la solución interior inicial \underline{a} es enviada al centro de Δ^1 (propiedad 3 de T), obteniéndose $\underline{a}_0 = e/n^2$ y, el valor mínimo de la función objetivo es cero.

Note que si el valor óptimo de $\bar{c}'\bar{x}$ no fuera cero, entonces el problema (II.9) tampoco tendría solución óptima en cero y el PPL (II.2) sería infactible o no acotado.

Debido a la transformación proyectiva T y a sus propiedades, la explicación geométrica de la sección II.2.3 es válida, pues se verifica como los puntos extremos son enviados a las caras del simplejo (a excepción del vértice (0,0) que es enviado al vértice (0,0,1)) y los puntos al infinito son enviados a otra de las caras. Los puntos $(f, 0, 1-f)$ y $(0, r, 1-r)$ son obtenidos al aplicar T a los puntos $(f, 0)$ y $(0, r)$.

Observe que si el valor óptimo alcanzado en el problema (II.11) es cero, se obtienen las soluciones óptimas primales y duales de los PPL (II.2) y (II.3) respectivamente.

Cabe aclarar que existen otras formas de transformar un PPL a la FCK; para ello puede consultar [Baz] y [Fan]. Ahora estamos en condiciones de aplicar las ideas básicas de Karmarkar sobre el PPL (II.11). Esto lo llevaremos a cabo en la siguiente sección.

II.4 ALGORITMO DE KARMARKAR.

En esta sección describiremos el Algoritmo de Karmarkar o Algoritmo de Escala Proyectiva de Karmarkar (escala proyectiva debido a las transformaciones proyectivas que cambian la escala de las variables del problema), para solucionar el PPL (II.11) obtenido en el apartado anterior. Para facilitar la explicación utilizaremos a las letras m y n en lugar de m_1 y n_1 , respectivamente. Pero no debe olvidarse que para poder aplicar el algoritmo a dicho PPL, debemos considerar nuevamente las dimensiones m_1 , n_1 y n_2 en vez de m , n y $n+1$.

II.4.1 Transformación de un Simplejo a un Simplejo.

El Algoritmo de Karmarkar genera una secuencia de puntos interiores $\bar{x}^{(0)}$, $\bar{x}^{(1)}$, ..., $\bar{x}^{(k)}$ en cada iteración (el superíndice entre paréntesis sobre las \bar{x} indica la iteración en que fue obtenido el punto), para encontrar la solución óptima del PPL que está en la FCK.

En la sección anterior al transformar el PPL en la FCK, se calculó también una solución interior inicial (a_0 el centro del simplejo) a la que llamaremos $\bar{x}^{(0)}$. Como hemos mencionado, al aplicar este algoritmo sobre el PPL obtenido se genera una secuencia de puntos interiores, los cuales

como es de esperarse no están centrados en la región factible (a excepción de $\bar{x}^{(0)}$). Cada punto interior es encontrado a partir de otro punto interior, por lo que se requiere (de acuerdo a la segunda idea básica de Karmarkar) que el problema sea modificado de forma tal, que cada punto (desde del cual se va a obtener el nuevo punto) se encuentre cercano al centro del poliedro de soluciones. Así, para cada solución interior, el problema debe ser distorsionado de manera que dicha solución quede centrada sobre la región factible.

Debido a que el PPL actualmente está contenido en un simplejo, se puede utilizar una transformación proyectiva, para un punto interior dado, que transforme al problema de forma tal que el punto quede ubicado en el centro del simplejo. Esta transformación modifica a la región factible sobre sí misma (es decir, va de un simplejo a un simplejo) y se define a continuación.

Definición II.2. Sea \bar{x} un punto interior de Δ^n , es decir, $\bar{x}_i > 0$ para $i=1, \dots, n+1$ y $\sum_{i=1}^{n+1} \bar{x}_i = 1$. Sea también $D = \text{diag}(\bar{x}_1, \dots, \bar{x}_{n+1})$ una matriz diagonal de dimensión $(n+1)(n+1)$. Obsérvese que la matriz D es no singular¹¹ y su inversa D^{-1} es también una matriz diagonal, cuyos elementos en la diagonal principal son $1/\bar{x}_i$ para $i = 1, \dots, n+1$.

Así la transformación proyectiva $U: \Delta^n \rightarrow \Delta^n$ es:

$$\bar{y} = U(\bar{x}) = \frac{D^{-1}\bar{x}}{e^t D^{-1}\bar{x}} \quad \text{donde } \bar{x} \in \Delta^n \text{ original, y } e^t = (1, \dots, 1)^t \in \mathbb{R}^{n+1}.$$

¹¹Ver matriz no singular en el Apéndice A.

Puede notarse que $D^{-1}\bar{x}$ es un vector columna $(n+1)$ -dimensional y $e^t D^{-1}\bar{x}$ es un escalar igual a la suma de los elementos en el vector $D^{-1}\bar{x}$. Por lo tanto, los elementos de U son normalizados. Así, $\bar{y} \in \Delta^n$, donde \bar{y} es ahora el centro del Δ^n transformado.

Esta transformación tiene las siguientes propiedades (para una ejemplificación de éstas, consultar [Fan]):

- i) U mapea a Δ^n en Δ^n si \bar{x} es un punto interior de Δ^n .
- ii) U mapea a \bar{x} al centro $e/(n+1)$ de Δ^n .
- iii) U es un vértice de Δ^n si \bar{x} es un vértice.
- iv) U está sobre la frontera de Δ^n si \bar{x} está sobre la frontera.
- v) U es un punto interior de Δ^n si \bar{x} es un punto interior.
- vi) U es uno a uno y U^{-1} está dada por:

$$\bar{x} = U^{-1}(\bar{y}) = \frac{D\bar{y}}{e^t D\bar{y}}, \quad \text{donde } \bar{y} \in \Delta^n \text{ transformado.}$$

Aplicando la transformación proyectiva U^{-1} al problema (II.11), para que el PPL quede en términos de $\bar{y} \in \Delta^n$, tenemos:

$$\begin{aligned} & \text{Min } \frac{c^t D\bar{y}}{e^t D\bar{y}} \\ & \text{s. a} \\ & \frac{A D\bar{y}}{e^t D\bar{y}} = 0 \quad (\text{II.12}) \\ & e^t \bar{y} = 1 \\ & \bar{y} \geq 0. \end{aligned}$$

Pero dado que el valor mínimo de (II.11) es cero; el problema (II.12) tiene la misma solución que el siguiente PPL:

$$\begin{aligned} \text{Min } \bar{c}^t D \bar{y} \\ \text{s. a} \\ \bar{A} D \bar{y} = 0 \\ e^t \bar{y} = 1 \\ \bar{y} \geq 0. \end{aligned} \quad (\text{II.13})$$

Como puede observarse la región factible de (II.13) está formada nuevamente por la intersección de un simplejo con un sistema de ecuaciones homogéneo. Además, por la propiedad ii) de la transformación U, $\bar{y} = e/(n+1)$ es una solución factible.

Así cada solución interior $\bar{x}^{(k)}$ generada por el Algoritmo de Karmarkar en la k-ésima iteración, es utilizada por la transformación U para transformar el PPL (II.11) en el PPL (II.13).

Por conveniencia y facilidad de manejo para las siguientes secciones, (II.13) se reescribirá como:

$$\begin{aligned} \text{Min } c_k^t \bar{y} \\ \text{s. a} \\ P_k \bar{y} = \bar{b} \\ \bar{y} \geq 0, \end{aligned} \quad (\text{II.14})$$

donde $c_k^t = \bar{c}^t D$, $P_k = \begin{bmatrix} \bar{A} D \\ e^t \end{bmatrix}$ y $\bar{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

El procedimiento descrito en esta sección satisface la segunda idea básica de Karmarkar enunciada en el apartado 2.1. En la siguiente sección trataremos con la primera idea básica.

II.4.2 Obtención de la Mejor Dirección de Movimiento.

En la sección anterior se mostró que cada punto generado por el Algoritmo de Karmarkar es colocado cerca del centro de la región factible (transformada). De acuerdo a la primera idea básica (sección II.1), para cada uno de estos puntos, tiene sentido ahora moverse en la dirección de mayor descenso de la función objetivo, para obtener una mejor solución interior factible en el PPL (II.14).

II.4.2.1 Dirección en un Poliedro.

Definición II.3. Una *dirección* en un poliedro P es un vector $d \in \mathbb{R}^n$ distinto de cero, tal que para cada $x_0 \in P$ el rayo $\{x \in \mathbb{R}^n \mid x = x_0 + \alpha d, \alpha \geq 0\}$ está contenido en P .

En este caso x_0 es el vértice del rayo y α un escalar no negativo, que puede considerarse como el tamaño de paso que se toma sobre d .

Si $P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ entonces un rayo en P debe satisfacer las condiciones $A(x_0 + \alpha d) = b$ y $(x_0 + \alpha d) \geq 0$, o lo que es lo mismo:

$$y \quad Ax_0 + \alpha Ad = b \quad (\text{II.15})$$

$$x_0 + \alpha d \geq 0. \quad (\text{II.16})$$

Como se debe cumplir que $x_0 \in P$, entonces $Ax_0 = b$ y $x_0 \geq 0$. Sustituyendo esto último en (II.15) tenemos: $b + \alpha Ad = b$. Despejando a b y posteriormente a α , concluimos que $Ad = 0$.

Por otro lado, dado que $x_0 \geq 0$ y $\alpha \geq 0$, (II.16) se cumple con $d \neq 0$ siempre y cuando se preserve la no negatividad del rayo. Así d es una dirección de $P \neq \emptyset$ si sus componentes son distintas de cero (al menos una) y, es solución del sistema homogéneo de ecuaciones, es decir $d \neq 0$ y $Ad = 0$ respectivamente.

En el caso del PPL (II.14), \bar{d} es una dirección del poliedro asociado a la región factible si $\bar{d} \neq 0$ y $P_k \bar{d} = 0$. Así, estamos interesados en encontrar un vector dirección que sea solución al sistema $[D\bar{A}^t \mid e]^t \bar{d} = 0$ y tenga al menos un elemento distinto de cero.

II.4.2.2 Dirección de Mayor Descenso.

Como se vio en el capítulo 1, la solución al problema (II.14) se obtiene al encontrar el valor mínimo de la función objetivo, $c_k^t \bar{y}$, sobre el poliedro (siempre y cuando sea no vacío) $P_k \bar{y} = \bar{b}$ y $\bar{y} \geq 0$. Así pues estamos interesados en encontrar, para cada punto interior factible en que se evaluó la f.o., la dirección de mayor descenso (ya que cada punto se encuentra cercano al centro del poliedro de soluciones) que nos permita obtener más rápidamente el mínimo.

De los conceptos de Cálculo¹² sabemos que la dirección en la que una función disminuye más rápidamente puede ser obtenida a través del gradiente¹³ de dicha función, ya que el gradiente determina la mejor dirección (la cual es perpendicular al contorno de la función) para poder alcanzar el óptimo en el menor número de iteraciones posibles. En nuestro caso estamos interesados en obtener el gradiente de la función objetivo¹⁴ la cual, dada la linealidad de la función, está formado por los coeficientes de ésta. En este caso el gradiente es $c_k^T = \bar{c}^T D$.

De esta forma, $\bar{d} = c_k$ (la cual varía dependiendo del punto interior con el que se construye D) por lo que, de la sección anterior, \bar{d} debe ser solución del sistema de ecuaciones homogéneo formado por las restricciones de (II.14). Cabe aclarar, que para alguna iteración dada, \bar{d} si puede valer cero, lo que indicaría que la solución factible actual es óptima.

Por otro lado, dado que requerimos obtener la dirección en que disminuye más rápidamente la función objetivo, debemos tomar el negativo del gradiente ($-\bar{d}$), ya que este indica la dirección de descenso acelerado de la f.o. para alcanzar el mínimo.

II.4.2.3 Proyección del Gradiente para la Obtención de la Dirección Factible de Mayor Descenso.

Un aspecto importante a tener en cuenta es que no basta con tomar los coeficientes de la f.o. como componentes de \bar{d} , ya que éste vector no es

¹²Ver filosofía de los métodos de gradientes en el Apéndice B.

¹³Ver definición de gradiente en el Apéndice B.

¹⁴Ver gradiente de un hiperplano en el Apéndice B.

solución del sistema de ecuaciones homogéneo constituido por las restricciones del problema. Para ello observe que $\bar{d} = D\bar{c}$, y que se debe satisfacer el sistema $A\bar{d} = 0$, $e^T\bar{d} = 0$ y $\bar{d} \geq 0$. \bar{c} es un vector, para nuestro caso, con todos sus elementos cero, a excepción del n -ésimo que tiene coeficiente a_n (correspondiente a λ). Al multiplicarlo por la matriz diagonal D , la cual tiene elementos en la diagonal principal mayores que cero, obtenemos nuevamente un vector (\bar{d}) con todos sus elementos cero a excepción del n -ésimo (el cual es mayor que cero). Por otro lado una de las restricciones $e^T\bar{d} = 0$, exige que la suma de los elementos de \bar{d} sea cero, lo que no sucede con nuestro vector por lo que no soluciona al sistema homogéneo. Así pues debemos obtener una representación del vector dirección que efectivamente sea solución de dicho sistema. Para ello utilizaremos las siguientes definiciones, (consultar [GMW]).

Definición II.4. Sea S un conjunto de vectores en \mathbb{R}^m . Diremos que S es un subespacio de \mathbb{R}^m si, para cualesquiera escalares α y β ,

$$x, y \in S \text{ implica que } \alpha x + \beta y \in S.$$

Esta propiedad inmediatamente implica que cualquier subespacio debe contener al vector cero (haciendo $\alpha = \beta = 0$).

Definición II.5. Sea S un subespacio de \mathbb{R}^m . El complemento ortogonal de S , denotado por S^\perp , es definido como el conjunto de todos los vectores en \mathbb{R}^m que son ortogonales a los vectores en S , i.e.,

$$S^\perp = \{l \in \mathbb{R}^m \mid l^T s = 0 \text{ para toda } s \in S\}.$$

Puede verificarse que S^\perp es un subespacio de \mathbb{R}^m . Además, se cumple: $S \cup S^\perp = \mathbb{R}^m$ y $S \cap S^\perp = \{0\}$.

Debido a que ambos subespacios contienen únicamente al vector cero en común, cualquier vector b de tamaño m puede ser representado como $b = b_S + b_{S^\perp}$, donde $b_S \in S$ y $b_{S^\perp} \in S^\perp$.

Definición II.6. El conjunto de todos los vectores en \mathbb{R}^m que son combinaciones lineales de las columnas de una matriz A de $m \times n$ es llamado el subespacio rango de A , y se denota por $\text{rango}(A)$, y es tal que:

$$b_A \in \text{rango}(A) \text{ si y sólo si } \exists \text{ una } x \in \mathbb{R}^n \text{ tal que } b_A = Ax.$$

Definición II.7. El conjunto de todos los vectores en \mathbb{R}^n que son combinaciones lineales de los renglones (transpuestos) de A definen un subespacio de vectores expresable como $A^t v$ para algún vector $v \in \mathbb{R}^m$. Este subespacio es llamado el subespacio rango de A^t , y se denota por $\text{rango}(A^t)$ y es tal que:

$$b_{A^t} \in \text{rango}(A^t) \text{ si y sólo si } \exists \text{ una } v \in \mathbb{R}^m \text{ tal que } b_{A^t} = A^t v$$

Definición II.8. Para una matriz A de $m \times n$, el conjunto de todos los vectores en \mathbb{R}^n ortogonales a los vectores en $\text{rango}(A^t)$ es llamado el subespacio nulo de A , y es denotado por $\text{null}(A)$. Sus elementos se definen como sigue:

$$z \in \text{null}(A) \text{ si y sólo si } Az = 0.$$

Aplicando estas definiciones al PPL (II.14) vemos que :

i) El vector \bar{b} \in rango(P_K) si y sólo si $\exists \bar{y} \in \mathbb{R}^{n+1}$ tal que $\bar{b} = P_K \bar{y}$.

ii) Un vector $\bar{c}_R \in$ rango(P_K^t) si y sólo si $\exists \pi \in \mathbb{R}^{m+1}$ tal que $\bar{c}_R = P_K^t \pi$. (II.17)

iii) Un vector $\bar{c}_N \in$ null(P_K) si y sólo si $P_K \bar{c}_N = 0$. (II.18)

Cualquier vector en \mathbb{R}^{n+1} puede ser escrito como una combinación lineal de los vectores que están en el rango(P_K^t) y en el null(P_K), y además se verifica que $\text{rango}(P_K^t) \cap \text{null}(P_K) = \{0\}$. Consecuentemente diremos que el rango(P_K^t) y el null(P_K) son *complementos ortogonales* uno del otro, por lo que la siguiente representación de cualquier vector $\bar{d} \in \mathbb{R}^{n+1}$ es única:

$$\bar{d} = \bar{c}_R + \bar{c}_N, \text{ donde } \bar{c}_R \in \text{rango}(P_K^t) \text{ y } \bar{c}_N \in \text{null}(P_K) \quad (\text{II.19})$$

Se verifica además que los vectores \bar{c}_R y \bar{c}_N satisfacen $\bar{c}_R^t \bar{c}_N = 0$.

Como podemos observar, en (II.18), $\bar{c}_N \in \text{null}(P_K)$ lo que indica que $P_K \bar{c}_N = 0$. Por lo que estamos interesados en encontrar al vector \bar{c}_N , que es la representación de \bar{d} en el subespacio nulo de P_K , ya que dicho vector si es solución del sistema de ecuaciones homogéneo como se requiere en la sección II.4.2.1.

En primera instancia de (II.19) vemos que, $\bar{c}_R = \bar{d} - \bar{c}_N$. Por otro lado en la sección II.4.2.2 se concluyó que $\bar{d} = c_K$. Sustituyendo esta igualdad en la ecuación anterior tenemos : $\bar{c}_R = c_K - \bar{c}_N$. Además de (II.17)

sabemos que $\bar{q}_k = P_k^t \pi$. Igualando estas últimas dos expresiones resulta que:

$$c_k - \bar{q}_k = P_k^t \pi. \quad (\text{II.20})$$

Premultiplicando ambos lados de (II.20) por P_k obtenemos

$$P_k c_k - P_k \bar{q}_k = P_k P_k^t \pi. \quad (\text{II.21})$$

Dado que en (II.18) se indicó que $P_k \bar{q}_k = 0$, (II.21) se reescribe como

$$P_k c_k = P_k P_k^t \pi. \quad (\text{II.22})$$

Debido a que \bar{A} es una matriz de rango completo y a que $\bar{y} > 0$, pues ésta es una solución interior factible, entonces la matriz $P_k P_k^t$ es invertible.

Premultiplicando $(P_k P_k^t)^{-1}$ en ambos lados de (II.22) tenemos:

$$\pi = (P_k P_k^t)^{-1} P_k c_k. \quad (\text{II.23})$$

Sustituyendo (II.23) en (II.20) se tiene:

$$c_k - \bar{q}_k = P_k^t (P_k P_k^t)^{-1} P_k c_k. \quad (\text{II.24})$$

Despejando \bar{q}_k en (II.24):

$$\bar{q}_k = c_k - P_k^t (P_k P_k^t)^{-1} P_k c_k. \quad (\text{II.25})$$

Factorizando c_k en (II.25) obtenemos la dirección buscada:

$$\bar{q}_k = [I - P_k^t (P_k P_k^t)^{-1} P_k] c_k. \quad (\text{II.26})$$

Al efectuar las operaciones que se encuentran entre los paréntesis cuadrados de (II.26) se obtiene la matriz de proyección $M_p = I - P_k^t (P_k P_k^t)^{-1} P_k$, que es conocida como el *proyector* sobre el subespacio nulo de P_k y que al multiplicarla con el vector $\bar{d} = c_k$ nos proporciona únicamente al vector \bar{q}_1 , la porción de \bar{d} que cae en $\text{null}(P_k)$ y que además es la dirección factible buscada. Consecuentemente M_p aniquila a \bar{q}_2 la porción de \bar{d} que no cae en $\text{null}(P_k)$. Si \bar{d} está contenida ya en $\text{null}(P_k)$, M_p proyecta a \bar{d} sobre sí mismo. Esta matriz tiene las siguientes propiedades:

- i) Cualquier vector en $\text{null}(P_k)$ puede ser escrito como una combinación lineal de las columnas de M_p ;
- ii) $M_p^t = M_p$ (M_p es simétrica);
- iii) $M_p^2 = M_p$ (La matriz M tal que $M^2 = M$ es llamada idempotente).

II.4.3. Optimización sobre una Bola y Transformación Inversa de un Simplejo a un Simplejo.

Como se explicó en la sección II.1., dado que ha sido calculada una solución interior cercana al centro del politopo, debemos movernos sobre la dirección de mayor descenso (obtenida en el apartado anterior) para generar una nueva solución, pero teniendo cuidado de que ésta permanezca en el interior de la región transformada (conservando así la factibilidad).

Para lograr esto Karmarkar decidió remplazar al simplejo Δ^n por la bola más grande inscrita¹⁵ en él con centro en $\bar{y}_0 = e/(n+1)$ y radio

¹⁵Ver bola inscrita en un simplejo en el Apéndice A.

$r = \frac{1}{\sqrt{n(n+1)}}$ (es decir con la bola cerrada¹⁶ $B_c(\bar{y}_0, r)$). Con esto aseguramos que la nueva región (formada por la intersección entre la bola y el subespacio nulo $\bar{A}D\bar{y} = 0$) contenga solamente puntos factibles. Por tanto retomando el problema (II.13), este puede interpretarse como:

$$\begin{aligned} & \text{Min } \bar{c}'D\bar{y} \\ & \text{s. a} \\ & \quad \bar{A}D\bar{y} = 0 \quad (\text{II.27}) \\ & \quad \bar{y} \in B_c(\bar{y}_0, r) \end{aligned}$$

Para absorber errores de redondeo Karmarkar decidió tomar una bola más pequeña $B_c(\bar{y}_0, \alpha r)$, donde $0 < \alpha < 1$, en vez de $B_c(\bar{y}_0, r)$. Además, dado que $\bar{y}_0 = e/(n+1)$ es solución del sistema $\bar{A}D\bar{y} = 0$, entonces la intersección de este sistema con la bola $B_c(\bar{y}_0, \alpha r)$ es otra bola $B_c(\bar{y}_0, \alpha r)$ de dimensión menor, en donde el centro y el radio permanecen iguales. Así (II.27) se reescribe como:

$$\begin{aligned} & \text{Min } \bar{c}'D\bar{y} \\ & \text{s. a} \\ & \quad \bar{y} \in B_c(\bar{y}_0, \alpha r) \quad (\text{II.28}) \end{aligned}$$

Pero este problema es trivial, ya que estamos obteniendo el mínimo de una función lineal sobre una bola, cuya solución puede obtenerse en una iteración, utilizando la ecuación:

$$\bar{y}^* = \bar{y}_0 - \frac{\alpha r \bar{c}_1}{\|\bar{c}_1\|} \quad (\text{II.29})$$

¹⁶Ver definición de bola cerrada en el Apéndice A.

donde \bar{q}_k se calcula como se explicó en la sección anterior, y al ser dividida ésta entre su norma, se obtiene un vector dirección normalizado que apunta a una solución interior mejorada dentro de la bola.

Así dado un punto conocido de partida \bar{y}_0 (el centro de la bola), debemos desplazarnos sobre la dirección de mayor descenso $(-\bar{q}_k/\|\bar{q}_k\|)$ una longitud αr , para obtener un nuevo punto \bar{y}^* , sobre la frontera de la bola $B_c(\bar{y}_0, \alpha r)$, el cual es solución óptima al PPL (II.28).

Como podemos observar en el problema (II.28), por cada punto interior (calculado en la k -ésima iteración), se obtiene un nuevo problema que es solucionado utilizando la ecuación (II.29), la cual es conocida también como la *ecuación de descenso acelerado*¹⁷. Cada una de estas soluciones óptimas \bar{y}^* , debe ser transformada de nuevo a la región factible del PPL (II.11), para obtener una mejor solución interior factible $\bar{x}^{(k+1)}$ que se aproxime a la solución óptima de dicho problema.

Ahora, procederemos a dar una breve ilustración de los conceptos expuestos acerca del algoritmo de Karmarkar, pero desde el punto de vista geométrico. Para facilitar dicha explicación consideraremos el caso de Δ^2 .

La fig. 2.4 ejemplifica la transformación de Δ^2 sobre sí mismo. Para ello imagínese dos simplejos separados, de distintos tamaños y orientaciones tales que las líneas que los unen, y que pasan por cada uno de sus vértices, se intersectan en un punto común θ . Además la imagen de $\bar{x}^{(k)}$ en el primer simplejo (izquierdo) es \bar{y}_0 en el segundo simplejo

¹⁷Ver método de descenso acelerado en el Apéndice B.

(derecho). A continuación el segundo simplejo es remplazado por la bola más grande inscrita en él (con centro en \bar{y}_0 y radio r), y ésta, por una bola con centro en \bar{y}_0 y radio αr que es más pequeña.

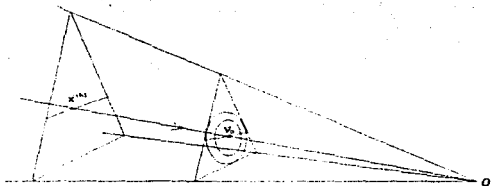


Figura 2.4

En la fig. 2.5 se ilustra el proceso de optimización sobre la bola $B(\bar{y}_0, r)$, contenida en el simplejo transformado (derecho) y en el que se obtiene el punto \bar{y}^* , al aplicar (11.29). También se ilustra la transformación inversa que envía a \bar{y}^* al simplejo original (izquierdo), obteniéndose así $\bar{x}^{(k+1)}$. Geométricamente esto puede realizarse tomando a los puntos \bar{y}^* y θ y, trazando una línea que intersecta al simplejo original en el nuevo punto $\bar{x}^{(k+1)}$.

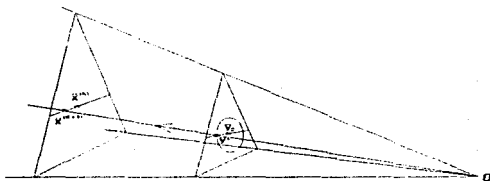


Figura 2.5

Algebraicamente la transformación inversa de \bar{y}^* es efectuada utilizando la propiedad vi) de la transformación U (sección II.4.1). Así el nuevo punto se obtiene como:

$$\bar{x}^{(k+1)} = \frac{D\bar{y}^*}{e^*D\bar{y}^*} \quad (\text{II.30})$$

donde $\bar{x}^{(k+1)}$ está en la región factible del PPL (II.11) y \bar{y}^* en la región del PPL (II.28).

II.4.4 Verificación de Factibilidad y Chequeo de Paro.

Otro aspecto importante a considerar es que las transformaciones proyectivas no preservan la linealidad de la función objetivo. Esto lo podemos constatar en las secciones II.3.3 y II.4.1, pues al aplicar las transformaciones proyectivas T y U a los PPL (II.6) y (II.11) (cuyas f.o. son lineales) respectivamente, obtenemos los PPL transformados (II.9) y (II.12) cuyas f.o. son no lineales, ya que han sido transformadas en funciones fraccionales con funciones lineales en el numerador y el

denominador. Pero en esas secciones establecimos que, dado que la f.o. debe alcanzar un valor mínimo de cero, podemos tomar como estimación únicamente al numerador de la función, con lo que se obtienen los problemas (II.10) y (II.13).

El problema es pues, cómo medir los progresos del algoritmo si la f.o. en la región transformada es sólo una estimación de la f.o. original. Es decir, cómo medir la ganancia de estando en el punto $\bar{x}^{(k)}$ ir al punto $\bar{x}^{(k+1)}$.

Karmarkar observó que aún cuando las funciones lineales (correspondientes a las f.o.) no son invariantes bajo la transformación proyectiva, los cocientes de funciones lineales sí lo son, ya que éstos son transformados en cocientes de funciones lineales. Por tanto, propuso que la f.o. lineal $\bar{c}^t \bar{x}$ debe tener asociada una función $f(\bar{x})$, expresada en términos de logaritmos naturales de cocientes de funciones lineales, con el fin de medir el progreso del algoritmo. La función asociada se define como :

$$f(\bar{x}) = \sum_i \ln \left\{ \frac{\bar{c}^t \bar{x}}{\bar{x}_i} \right\} \quad (\text{II.31})$$

$f(\bar{x})$ es conocida como la función potencial y tiene las siguientes propiedades :

- i) Cualquier cantidad deseada de reducción en el valor de $\bar{c}^t \bar{x}$ puede ser lograda por una reducción en el valor de $f(\bar{x})$.
- ii) $f(\bar{x})$ es mapeado en una función de la misma forma por la transformación proyectiva U descrita en la sección II.4.1

Así pues, el objetivo es comprobar que en cada iteración el algoritmo hace disminuir a $f(\bar{x})$ en al menos una constante δ que depende del parámetro α . Si no se observa la mejora esperada, i.e., si $f(\bar{x}^{(k+1)}) > f(\bar{x}^{(k)}) - \delta$, entonces debemos detenernos y concluir que el valor mínimo de la f.o. en el PPL (II.11) debe ser estrictamente positivo. En el problema original, (II.2), esta situación corresponde al caso en que se detecta infactibilidad o no acotación de la f.o.

El siguiente teorema establece que si la reducción en la función potencial se satisface entonces el algoritmo converge (para una demostración de este teorema consultar [Cal] y [Kar]).

Teorema II.1. Sea $\bar{x}^{(0)}, \bar{x}^{(1)}, \dots, \bar{x}^{(k)}, \dots$ la sucesión de puntos generados por el Algoritmo de Karmarkar. Si

$$f(\bar{x}^{(k+1)}) \leq f(\bar{x}^{(k)}) - \delta$$

durante $\frac{(n+1)(q+\ln(n+1))}{\delta}$ iteraciones, entonces el algoritmo encuentra un punto factible \bar{x} tal que

$$\frac{\bar{c}^t \bar{x}}{\bar{c}^t a_0} \leq 2^{-q} \quad (\text{II.32})$$

donde q es un entero positivo grande.

Para probar este teorema supongamos que el algoritmo se ha ejecutado para m iteraciones y que $\bar{c}^t \bar{x}^{(m)} > 0$, entonces para cada m tenemos

$$f(\bar{x}^{(m)}) \leq f(\bar{x}^{(m-1)}) - \delta \leq f(\bar{x}^{(m-2)}) - 2\delta \leq \dots \leq f(\bar{x}^{(0)}) - m\delta$$

Por transitividad y utilizando (II.31) vemos que:

$$\sum_{i=1}^{n+1} \text{Ln} \left(\frac{\bar{c}^i \bar{x}^{(m)}}{\bar{x}_i^{(m)}} \right) \leq \sum_{i=1}^{n+1} \text{Ln} \left(\frac{\bar{c}^i \bar{x}^{(0)}}{\bar{x}_i^{(0)}} \right) - m\delta.$$

Aplicando reglas de logaritmos y como $\bar{x}^{(0)} = a_0$ tenemos:

$$\sum_{i=1}^{n+1} [\text{Ln}(\bar{c}^i \bar{x}^{(m)}) - \text{Ln}(\bar{x}_i^{(m)})] \leq \sum_{i=1}^{n+1} [\text{Ln}(\bar{c}^i a_0) - \text{Ln}(\gamma_{n+1}^i)] - m\delta.$$

Si ahora aplicamos las Σ sobre cada término resulta que:

$$(n+1)\text{Ln}(\bar{c}^i \bar{x}^{(m)}) - \sum_{i=1}^{n+1} \text{Ln}(\bar{x}_i^{(m)}) \leq (n+1)[\text{Ln}(\bar{c}^i a_0) + \text{Ln}(n+1)] - m\delta.$$

Dividiendo entre $(n+1)$, reordenando y aplicando reglas de logaritmos vemos que:

$$\text{Ln} \left(\frac{\bar{c}^i \bar{x}^{(m)}}{\bar{c}^i a_0} \right) \leq \text{Ln}(n+1) + \left(\frac{-1}{n+1} \right) \left(\sum_{i=1}^{n+1} \text{Ln}(\bar{x}_i^{(m)}) \right) - \frac{m\delta}{n+1}.$$

Debido a que $\bar{x}^{(m)}$ está en el interior de Δ^n , todas sus componentes son $0 < \bar{x}_i^{(m)} < 1$ por lo que el término $\sum \text{Ln}(\bar{x}_i^{(m)})$ es negativo, y al sumarlo con $\frac{-m\delta}{n+1}$ sigue siendo un negativo, por lo que sólo tomamos a esta última constante. Entonces

$$\text{Ln} \left(\frac{\bar{c}^i \bar{x}^{(m)}}{\bar{c}^i a_0} \right) \leq \text{Ln}(n+1) - \frac{m\delta}{n+1}. \quad (\text{II.33})$$

Considere ahora a $m = \frac{(n+1)(q + \text{Ln}(n+1))}{\delta}$, que nos proporciona el número de iteraciones del Algoritmo de Karmarkar que necesitamos calcular. Sustituyendo a m en (II.33) tenemos

$$\text{Ln} \left(\frac{\bar{c}^i \bar{x}^{(m)}}{\bar{c}^i a_0} \right) \leq -q.$$

Exponenciando ambos lados y teniendo en cuenta que $e^{-q} \leq 2^{-q}$ para $q \geq 0$ resulta

$$\frac{\bar{c}^T \bar{x}(m)}{\bar{c}^T a_0} \leq e^{-q} \leq 2^{-q}.$$

Con lo que concluimos la prueba de este teorema.

Cabe hacer mención que una forma de paro del método es por medio de (II.32). Es decir, el algoritmo se detiene en la iteración k si $\frac{\bar{c}^T \bar{x}(k)}{\bar{c}^T a_0} \leq 2^{-q}$.

Debe notarse también que la constante q debe ser tal que $2^{-q} < \epsilon$ para una tolerancia dada $\epsilon > 0$; q también puede ser seleccionada como el tamaño del problema o un múltiplo de este (consultar [Baz], [Kar], [Fan]).

A continuación se enuncian los teoremas (cuya demostración puede consultarse en [Kar]) que garantizan que el algoritmo se comporta satisfaciendo las hipótesis del teorema II.1. Para ello requerimos la transformación f' de la función potencial $f(\bar{x})$, la cual se define como

$$f'(\bar{y}) = \sum_{j=1}^{n+1} \text{Ln} \left(\frac{\bar{c}^T \bar{D} \bar{y}}{\bar{x}_j \bar{y}_j} \right) = \sum_{j=1}^{n+1} \text{Ln} \left(\frac{\bar{c}^T \bar{D} \bar{y}}{\bar{y}_j} \right) - \sum_{j=1}^{n+1} \text{Ln}(\bar{x}_j).$$

En primer término, estamos interesados en optimizar $f'(\bar{y})$ sobre $B_c(\bar{V}_0, \alpha r)$. Por lo que es necesaria la existencia de un punto que logre una reducción constante en la función potencial. El siguiente teorema garantiza la existencia de tal punto.

Teorema II.2. Existe un punto $\bar{y}^* \in B'_c(\bar{y}_0, \alpha r)$ tal que

$$f'(\bar{y}^*) \leq f'(\bar{y}_0) - \delta$$

donde δ es una constante.

En segunda instancia se requiere que la minimización de $f'(\bar{y})$ pueda ser aproximada por la minimización de la función lineal $\bar{c}'D\bar{y}$. Esto es debido a que la optimización se realiza sobre una bola con centro en \bar{y}_0 y radio αr , como lo indica el siguiente teorema.

Teorema II.3. Sea \bar{y}^* el punto que minimiza $\bar{c}'D\bar{y}$ sobre $B'_c(\bar{y}_0, \alpha r)$. Entonces $f'(\bar{y}^*) \leq f'(\bar{y}_0) - \delta$ donde δ es una constante positiva que depende de α . Para $\alpha = 1/4$ podemos tomar $\delta = 1/8$.

Finalmente debemos transformar a \bar{y}^* de regreso al simplejo original, para obtener el nuevo punto $\bar{x}^{(k+1)}$ y así medir el progreso del algoritmo. Esto se verifica en el siguiente teorema.

Teorema II.4. Si $\bar{x}^{(k)} = U^{-1}(\bar{y}_0)$ y $\bar{x}^{(k+1)} = U^{-1}(\bar{y}^*)$, \bar{y}^* como en el teorema II.3, entonces

$$f(\bar{x}^{(k+1)}) \leq f(\bar{x}^{(k)}) - \delta.$$

Aunque teóricamente es adecuado tomar a $\alpha = 1/4$ y a $\delta \geq 1/8$, en la práctica α puede tomar valores cercanos a uno. En particular si $\alpha \approx 1/4$ y $\alpha \in (0, 1)$, se sugiere calcular a δ como (consultar [Cál], [Kar]):

$$\delta = \alpha - \frac{\alpha^2}{2} - \frac{\alpha^2(n+1)}{n(1-\alpha\sqrt{(n+1)/n})}$$

II.5 ESQUEMA DE PURIFICACIÓN.

Note que los puntos $\bar{x}^{(0)}, \bar{x}^{(1)}, \dots, \bar{x}^{(l)}, \dots$ generados por el algoritmo en j iteraciones, permanecen siempre en el interior de la región factible (debido al esquema de centrado); al igual que el punto $\bar{x}^{(k)}$, que satisface la condición $\frac{\bar{c}^t \bar{x}^{(k)}}{\bar{c}^t a_0} \leq 2^{-q}$. Por lo que podemos obtener, a partir de $\bar{x}^{(k)}$, un punto extremo exacto y verificar que este sea solución óptima.

Para encontrar dicha solución básica factible, la cual proporciona un valor objetivo al menos tan bueno como el de $\bar{c}^t \bar{x}^{(k)}$, tendremos que identificar a las variables básicas y no básicas por medio del método conocido como esquema de purificación (consultar [Baz], [Fan]).

Observe que el PPL en la FCK tiene $m+n+2$ ($m+1+n+1$ en el PPL (II.11)) restricciones, incluyendo las restricciones explícitas y de no negatividad, en total. Si $n+1$ restricciones linealmente independientes se satisfacen en igualdad dado que se obtuvo $\bar{x}^{(k)}$ (i.e. si se cumple $\bar{A} \bar{x}^{(k)} = 0$, $e^t \bar{x}^{(k)} = 1$ y $n-m$ elementos de $\bar{x}^{(k)}$ son iguales a cero), entonces ésta es ya una solución básica factible. De otra forma existe una dirección $d \neq 0$ en el subespacio nulo de las restricciones (las cuales son menos de $n+1$) que cumplen la igualdad. Si $\bar{c}^t d < 0$ entonces debemos desplazarnos sobre la dirección d , en caso contrario sobre $-d$, hasta que alguna restricción bloquee el movimiento por consideraciones de factibilidad. Debido a que la región factible del PPL en la FCK es acotada, podemos siempre encontrar una nueva solución con al menos una restricción adicional, satisfaciendo la igualdad ($\bar{x}_j = 0$). El valor objetivo en esta nueva solución es al menos tan bueno como $\bar{c}^t \bar{x}^{(k)}$. Repitiendo este proceso

sucesivamente, una solución básica factible \bar{x}^* puede ser eventualmente identificada tal que $\bar{c}^T \bar{x}^* = 0$. Este proceso involucra a lo más $n - m$ pasos, ya que iniciamos con $m+1$ restricciones explícitas linealmente independientes en el PPL que está en la FCK. Así al menos una restricción linealmente independiente, que satisface la igualdad, se agrega a este conjunto en cada paso. Note además que si $\bar{c}^T \bar{x}^* = 0$, el sistema (II.4) tiene solución y \bar{x}^* proporciona las soluciones óptimas de los PPL (II.2) y (II.3), es decir las soluciones óptimas primal y dual.

II.6 PRESENTACIÓN FORMAL DEL MÉTODO DE ESCALA PROYECTIVA DE KARMARKAR.

Ahora pongamos juntos los conceptos descritos en este capítulo para dar una presentación formal del Método de Escala Proyectiva de Karmarkar para solucionar Problemas de Programación Lineal. Debe recordarse que en primer lugar el PPL debe ser puesto en la forma canónica de Karmarkar, una vez representado el problema en esta forma aplicar el algoritmo de Karmarkar hasta encontrar una solución interior cercana a la óptima (en caso de que exista), para finalmente aplicar el esquema de purificación y obtener la solución óptima.

Método de Escala Proyectiva de Karmarkar.

- I) Transformar el PPL a la *Forma Canónica de Karmarkar* como se describe en la sección II.3.
- II) Una vez que el PPL está en la FCK efectuar los siguientes pasos correspondientes al *Algoritmo de Karmarkar*.

i.- Inicialización.

Hágase: $k = 0$, $\bar{x}^{(0)} = a_0$.

Fijese a $\alpha \in (0,1)$ y a q como un entero positivo grande.

Calcúlese: $r = (n(n+1))^{-1/2}$.

$$f(\bar{x}_0) = (n+1) [\text{Ln}(\bar{c}^t \bar{x}^{(0)}) + \text{Ln}(n+1)].$$

Si $\alpha = 1/4$ entonces $\delta \geq 1/8$,

$$\text{No, entonces calcular } \delta = \alpha - \frac{\alpha^2}{2} - \frac{\alpha^2(n+1)}{n(1-\alpha)\sqrt{(n+1)/n}}.$$

ii.- Transformación de un simplejo a un simplejo:

Sea $D_k = \text{diag}(\bar{x}_1^{(k)}, \dots, \bar{x}_{n+1}^{(k)})$ y calcúlese $c_k^t = \bar{c}^t D_k$,

$$P_k = \begin{bmatrix} \bar{A} D_k \\ e^t \end{bmatrix}.$$

iii.- Obtención de la mejor dirección de movimiento:

$$\bar{d}_k = [I - P_k^t (P_k P_k^t)^{-1} P_k] c_k.$$

iv.- Optimización sobre una bola:

$$\text{Sea } \bar{y}_0 = a_0 \text{ entonces calcule } \bar{y}^* = \bar{y}_0 - \frac{\bar{c}^t \bar{d}_k}{\| \bar{d}_k \|^2} \bar{d}_k.$$

v.- Transformación inversa de un simplejo a un simplejo:

$$\text{Calcular } \bar{x}^{(k+1)} = \frac{D_k \bar{y}^*}{e^t D_k}.$$

vi.- Verificación de factibilidad:

$$\text{Calcúlese } f(\bar{x}^{(k+1)}) = \sum_{i=1}^{n+1} \ln \left(\frac{\bar{c}_i \bar{x}^{(k+1)}}{\bar{x}_i^{(k+1)}} \right).$$

Si $f(\bar{x}^{(k+1)}) > f(\bar{x}^{(k)}) - \delta$ entonces el algoritmo termina, pues el valor óptimo de $\bar{c}^t \bar{x}$ es positivo. Esto implica que el PPL original es infactible o su función objetivo no está acotada.

No, ir al paso vii.

vii.- Chequeo de paro:

Si $\frac{\bar{c}^t \bar{x}^{(k)}}{\bar{c}^t a_0} \leq 2^{-q}$ entonces el algoritmo termina, ir a iii).

No, entonces hacer $k = k+1$, ir al paso ii.

iii) Calcular una solución básica factible óptima por medio del *esquema de purificación* descrito en la sección II.5.

Como hemos visto a través de este capítulo el método proyectivo de Karmarkar proporciona un nuevo enfoque para la solución de PPL. Aunque en principio Karmarkar dijo que su algoritmo soluciona los PPL en menos tiempo que el método simplex, esto no ha podido ser probado por otros investigadores, dada la negativa por parte del autor de presentar los trucos de la implantación. En cambio esto ha repercutido en una gran cantidad de investigaciones que han generado variantes del algoritmo que parecen ser realmente eficaces en la solución de PPL, sobre todo en gran escala. Una de tales variantes se expone a continuación, de acuerdo a lo discutido en la sección I.1.

CAPÍTULO III.

MÉTODOS DE ESCALA AFÍN

Desde su introducción en 1984, el método de escala proyectiva de Karmarkar ha estimulado una gran cantidad de trabajos de investigación en el campo de la programación lineal; por lo que surgieron nuevas interpretaciones, implementaciones computacionales y variantes del método. Una de las variantes que acaparó la atención de varios investigadores fue la de escala afín, conocida así por la utilización de transformaciones afines en lugar de las transformaciones proyectivas, dando como resultado que la estructura simplejo-homogéneo en el algoritmo original fuera relajada.

La variante de escala afín tiene las siguientes ventajas sobre el método original. En primer lugar, el valor mínimo de la función objetivo no tiene que ser conocido de antemano. En segundo lugar, esta variante se aplica directamente a un PPL en su forma estándar y produce una secuencia de valores objetivo que decrecen en forma monótona.

El método de escala afín básico fue primero descubierto por el matemático ruso I. I. Dikin en 1967 [Dik]. Posteriormente, en 1985 el método fue independientemente redescubierto por Barnes [Bar]; Vanderbei, Meketon y Freedman [VMF]; y Cavalier y Soyster [CyS]. Ellos propusieron usar el algoritmo de escala afín (primal) para solucionar PPL en su forma estándar, y establecieron pruebas de convergencia del algoritmo. Un algoritmo similar, el método de escala afín dual, fue diseñado e implementado por Adler,

Karmarkar, Resende y Veiga [AKR] para solucionar programas lineales duales en forma de desigualdades en 1989. Comparados con las engorrosas transformaciones proyectivas, las implementaciones de los algoritmos de escala afín primal y dual son bastante directas. Ambos algoritmos están actualmente sujetos a una gran cantidad de experimentaciones y han exhibido resultados promisorios; consultar [Arb], [MSS], [MyM], [PQV], [Van], [VQV].

En este capítulo analizaremos las variantes de escala afín primal y dual basándonos en los conceptos presentados en los capítulos anteriores.

III.1 EL MÉTODO DE ESCALA AFÍN PRIMAL.

Como se mencionó anteriormente, este algoritmo se aplica directamente sobre un PPL en su forma estándar. Así pues consideremos el siguiente PPL:

$$\begin{aligned} \text{Min } z &= c^T x \\ \text{s. a} & \quad (III.1) \\ Ax &= b \\ x &\geq 0, \end{aligned}$$

donde $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ y $A \in \mathbb{R}^{m \times n}$ es de rango completo por renglones con $m \leq n$.

En este caso la región factible está dada por

$$P = \{x \in \mathbb{R}^n \mid Ax = b \text{ y } x \geq 0\}, \quad (\text{III.2})$$

por lo que procederemos a definir el interior relativo de P .

Definición III.1. El *interior relativo* de P , con respecto al subespacio afín $\{x \in \mathbb{R}^n \mid Ax = b\}$ está dado por:

$$P^0 = \{x \in \mathbb{R}^n \mid Ax = b \text{ y } x > 0\}.$$

Un vector x n -dimensional es llamado un *punto interior relativo factible* o *solución interior factible* del PPL (III.1) si $x \in P^0$. Para fines de exposición se supondrá que $P^0 \neq \emptyset$.

III.1.1 Ideas Básicas del Método de Escala Afín Primal.

Al ser el método de escala afín una variante del método proyectivo, éste sigue las ideas originales que utilizó Karmarkar para el diseño de su algoritmo (sección II.1). Pero como se hizo notar en la sección II.2, estas ideas son válidas para una región factible acotada, por lo que debemos obtener una representación de (III.1) que cumpla con dicha condición.

Como puede observarse, la complejidad y la no trivialidad de (III.1), estriba en la existencia de las desigualdades. Si la condición de no negatividad pudiera sustituirse por una restricción (una sola) que la estimara adecuadamente, se podría encontrar una buena aproximación a la solución óptima con poco esfuerzo. Para ello suponga que el PPL (III.1)

tiene una solución interior factible $x^k = (x_1^k, \dots, x_n^k)^t$. Entonces el primer orlante $x \geq 0$ de (III.1) se puede sustituir por el elipsoide

$$E = \{x \in \mathbb{R}^n \mid \sum_{i=1}^n (x_i - x_i^k)^2 / (x_i^k)^2 \leq r^2, 0 < r < 1\} \quad (\text{III.3})$$

que está contenida en el interior de \mathbb{R}_+^n . Para verificar este hecho, observe que si $x_j \leq 0$ para alguna j , entonces:

$$\sum_{i=1}^n (x_i - x_i^k)^2 / (x_i^k)^2 \geq (x_j - x_j^k)^2 / (x_j^k)^2 \geq 1 \geq r^2. \quad (\text{III.4})$$

En (III.4), la primera desigualdad se cumple, dado que la suma se efectúa sobre números positivos, por lo que da como resultado un valor positivo; la segunda desigualdad podemos verificarla si desarrollamos el binomio del numerador y observamos que el término $(x_j^k)^2$ es sumado con valores no negativos; por último, la tercera desigualdad se cumple dada la condición impuesta sobre r en (III.3).

Esto implica que podemos obtener una solución factible de (III.1), satisfaciendo $c^t x^{k+1} < c^t x^k$ al solucionar el siguiente problema auxiliar

$$\begin{aligned} \text{Min } z &= c^t x \\ \text{s. a} \\ Ax &= b \\ \sum_{i=1}^n (x_i - x_i^k)^2 / (x_i^k)^2 &\leq r^2 \\ 0 < r &< 1. \end{aligned} \quad (\text{III.5})$$

Note que al sustituir las restricciones $x \geq 0$, por el elipsoide E la región factible estará formada por la intersección de $Ax = b$ con E , lo que dará como resultado un elipsoide E' de dimensión menor contenido en (III.2). Dado que es posible encontrar una bola abierta que contenga a E' y, dada la suposición de que $P^0 \neq \emptyset$, podemos concluir que el poliedro de soluciones de (III.5) es un politopo. Por lo que las ideas básicas de Karmarkar pueden ser aplicadas a este PPL, es decir, para una solución interior factible dada:

1*) Si dicha solución está cercana al centro del politopo, entonces tiene sentido moverse en la dirección de mayor descenso de la f.o. para alcanzar el valor mínimo;

2*) En caso contrario, sin cambiar el PPL en su forma esencial, una transformación apropiada puede ser aplicada al problema, tal que la solución interior factible sea puesta cercana al centro del politopo transformado.

III.1.2 Transformación de Escala Afín Sobre el Ortante No Negativo.

Para llevar a cabo la segunda idea básica, citada anteriormente, procederemos a transformar el PPL (III.5). La transformación a utilizar en este caso, es la que se define a continuación.

Definición III.2. Sea $T_k : \mathbb{R}_+^n \rightarrow \mathbb{R}_+^n$ la transformación de escala afín tal que

$$T_k(x) = D_k^{-1}x = w,$$

donde $D_k^{-1} = \text{diag}(1/x_1^k, \dots, 1/x_n^k)$ y, las $x_i^k > 0$ son las componentes de la solución interior factible $x^k \in \mathbb{R}_+^n$.

Esta transformación tiene las siguientes propiedades:

- i) T_k mapea a \mathbb{R}_+^n en \mathbb{R}_+^n si x^k es una solución interior factible de \mathbb{R}_+^n .
- ii) $T_k(x^k) = e$.
- iii) $T_k(x)$ es un vértice de \mathbb{R}_+^n si x es un vértice.
- iv) $T_k(x)$ está sobre la frontera de \mathbb{R}_+^n si x está sobre la frontera.
- v) $T_k(x)$ es una solución interior factible de \mathbb{R}_+^n si x es una solución interior factible.
- vi) $T_k(x)$ es uno a uno y T_k^{-1} está dada por
 $T_k^{-1}(w) = D_k w$ para cada w que está en \mathbb{R}_+^n ; y $D_k = \text{diag}(x_1^k, \dots, x_n^k)$.

Antes de efectuar la transformación de (III.5), note que el elipsoide definido en (III.3) puede representarse en términos matriciales como: $E = \{x \in \mathbb{R}^n \mid \|D_k^{-1}(x - x^k)\| \leq r\}$. Así el PPL (III.5) se reescribe como:

$$\begin{aligned} \text{Min } z &= c^1 x \\ \text{s. a} \\ Ax &= b & \text{(III.6)} \\ \|D_k^{-1}(x - x^k)\| &\leq r \\ 0 < r < 1. \end{aligned}$$

Para llevar a cabo la transformación de escala afín sobre (III.6) debe observarse que $w = D_k^{-1}x$ y $e = D_k^{-1}x^k$.

Aplicando T_k^{-1} sobre (III.6), para que el PPL quede en términos de $w \in \mathbb{R}^n$, tenemos:

$$\begin{aligned} & \text{Min } (c^k)^T w \\ & \text{s. a} \\ & A_k w = b \\ & \|w - e\| \leq r \\ & 0 < r < 1, \end{aligned} \tag{III.7}$$

donde $c^k = D_k c$ y $A_k = A D_k$.

III.1.3 Optimización Sobre una Bola.

Como podemos observar en (III.7), el elipsoide E de (III.6) ha sido transformado en una bola cerrada con centro en e y radio r . Además, similar a (II.27) y (II.28), el vector $e = D_k^{-1} x^k$ es solución del sistema $A_k w = b$, por lo que la intersección de este sistema con la bola $B_c(e, r)$, es otra bola de dimensión menor $B'_c(e, r)$ con el mismo centro y radio. Así el punto óptimo de (III.7) puede ser obtenido a través del siguiente PPL:

$$\begin{aligned} & \text{Min } (c^k)^T w \\ & \text{s. a} \\ & w \in B'_c(e, r). \end{aligned} \tag{III.8}$$

Análogo a lo expuesto en la sección II.4.3, el procedimiento de solución del PPL (III.8) es trivial, ya que estamos obteniendo el mínimo de la

función lineal $(c^T D_k)w$ sobre la bola $B_c(e, r)$, cuya solución puede obtenerse en una sola iteración, utilizando la ecuación de descenso acelerado

$$w^{k+1} = e - r(d_w^k / \|d_w^k\|). \quad (\text{III.9})$$

Como en (II.29), el vector óptimo de (III.8), w^{k+1} , permanece en el interior de la región factible de (III.7), si d_w^k se encuentra en el subespacio nulo de la matriz A_k y $r \in (0, 1)$. Además su imagen inversa $x^{k+1} = T_k^{-1}(w^{k+1}) = D_k w^{k+1}$ es una nueva solución interior de (III.5).

En la figura 3.1 se ilustran los conceptos expuestos hasta este momento.

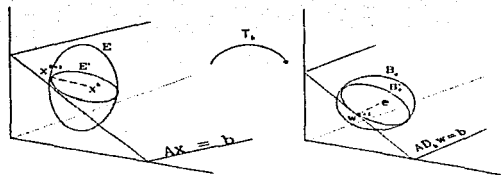


Figura 3.1

En esta figura se observa que, al aplicar la transformación T_k al ortante no negativo del lado izquierdo, se modifican tanto la región factible $(Ax=b$ en $AD_k w=b)$ como los elipsoides (E en B_c y E' en B'_c) para quedar centrados con respecto a e , en el ortante del lado derecho; y a partir de este punto se efectúa un desplazamiento $-rd_w^k / \|d_w^k\|$ unidades para obtener una solución interior factible w^{k+1} , sobre la frontera de la bola.

Ahora, estamos en condiciones de enunciar *la estrategia a seguir por el algoritmo de escala afín primal*: Dada una solución interior factible inicial, aplicar la transformación de escala afín al espacio de soluciones de tal forma que la solución interior actual sea enviada a e , el centro de la bola en el espacio transformado; y entonces, movernos una distancia r sobre la dirección de mayor descenso (la cual es un vector unitario que nos permite permanecer en la bola), para obtener el óptimo w^{k+1} . Posteriormente tomar T_k^{-1} que mapea la solución w^{k+1} de regreso al espacio solución original, como una nueva solución interior mejorada. Repetir este proceso hasta que las condiciones de paro, consideradas para el método en el espacio original, se satisfagan.

III.1.4 Dirección Factible de Mayor Descenso.

Dado que la estrategia de adoptar la ecuación (III.9) aplica para solucionar al problema (III.8), estamos interesados en proyectar el gradiente c^k sobre el subespacio nulo de A_k , para crear una buena dirección d_w^k que nos permita obtener un valor mejorado de la f.o. en el espacio transformado.

De (II.26), la mejor dirección, proyectada sobre el subespacio nulo de la matriz A_k , es

$$d_w^k = [I - A_k^t(A_k A_k^t)^{-1}A_k]c^k, \quad (\text{III.10})$$

cuya matriz de proyección es

$$M_p = I - A_k^t(A_k A_k^t)^{-1}A_k. \quad (\text{III.11})$$

Por lo que $d_w^k = M_p(c^k)$. Es fácil verificar también que $A_k d_w^k = 0$.

La figura 3.2 ilustra la proyección del vector dirección.

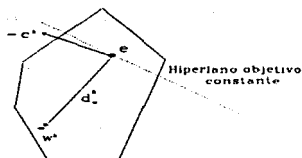


Figura 3.2.

Así pues al sustituir (III.10) en (III.9), y considerando que $re(0,1)$, obtenemos la solución óptima $w^{k+1} > 0$ del PPL (III.8), el cual es equivalente a (III.7).

De acuerdo a la estrategia citada en la sección III.1.3, debemos mapear a la solución w^{k+1} , que se encuentra en el espacio transformado, al espacio original, para obtener la nueva solución mejorada x^{k+1} correspondiente al problema (III.5). Esto puede lograrse al aplicar la transformación inversa T_k^{-1} sobre w^{k+1} . En otras palabras:

$$x^{k+1} = T_k^{-1}(w^{k+1}) = D_k w^{k+1}. \quad (\text{III.12})$$

Sustituyendo (III.9) en (III.12)

$$x^{k+1} = x^k - r D_k d_w^k. \quad (\text{III.13})$$

donde $r' = r/\|d_w^k\|$ y $x^k = D_k e$. Sustituyendo (III.10) en (III.13) tenemos

$$x^{k+1} = x^k - r' D_k [I - A_k^t (A_k A_k^t)^{-1} A_k] c^k. \quad (\text{III.14})$$

Dado que $A_k = A D_k$ y $c^k = D_k c$ (III.14) se escribe como

$$x^{k+1} = x^k - r' D_k [I - D_k A^t (A D_k^2 A^t)^{-1} A D_k] D_k c;$$

$$x^{k+1} = x^k - r' D_k^2 [c - A^t (A D_k^2 A^t)^{-1} A D_k^2 c];$$

ya que $r' = r/\|d_w^k\|$ entonces $r' = r/\|D_k [c - A^t (A D_k^2 A^t)^{-1} A D_k^2 c]\|$ por lo que la ecuación anterior se reescribe como:

$$x^{k+1} = x^k - r D_k^2 [c - A^t \lambda^k] / \|D_k [c - A^t \lambda^k]\|; \quad (\text{III.15})$$

donde

$$\lambda^k = (A D_k^2 A^t)^{-1} A D_k^2 c. \quad (\text{III.16})$$

Esto significa que la dirección de movimiento en el espacio original es:

$$\frac{-d_k^k}{\|d_k^k\|} = \frac{-D_k^2 [c - A^t \lambda^k]}{\|D_k [c - A^t \lambda^k]\|}$$

y la longitud de paso es r ; mientras que en el espacio transformado:

$$\frac{-d_w^k}{\|d_w^k\|} = \frac{-D_k [c - A^t \lambda^k]}{\|D_k [c - A^t \lambda^k]\|}$$

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Entonces tenemos que $\frac{-d_k^k}{\|d_k^k\|} = \frac{-D_k d_w^k}{\|d_w^k\|}$.

Dados los resultados anteriores, observamos que

$$x^{k+1} = x^k - r' D_k d_w^k \quad \text{donde } r' = \frac{r}{\|d_w^k\|},$$

entonces

$$c^t x^{k+1} = c^t x^k - r' c^t D_k d_w^k. \quad (\text{III.17})$$

Como $c^t D_k = (c^k)^t$ y $d_w^k = M_p c^k$, de (III.17) tenemos:

$$c^t x^{k+1} = c^t x^k - r' (c^k)^t M_p c^k. \quad (\text{III.18})$$

Considerando las propiedades de la matriz de proyección (sección II.4.2.3):

$$\begin{aligned} c^t x^{k+1} &= c^t x^k - r' (c^k)^t M_p M_p c^k; \\ c^t x^{k+1} &= c^t x^k - r' (d_w^k)^t (d_w^k); \\ c^t x^{k+1} &= c^t x^k - r' \|d_w^k\|^2; \\ c^t x^{k+1} &= c^t x^k - r' \|D_k [c - A^t \lambda^k]\|. \end{aligned} \quad (\text{III.19})$$

Esto implica que x^{k+1} es, de hecho, una solución mejorada y ésta es alcanzada cuando x^k y λ^k son obtenidas como se indica en (III.15) y (III.16) respectivamente; y cuando la dirección de movimiento $d_w^k \neq 0$. Estos resultados sugieren un algoritmo iterativo para encontrar la solución de (III.1), el cual se enuncia formalmente en la siguiente sección.

III.1.5 El Algoritmo de Escala Afín Primal.

Las ecuaciones (III.15) y (III.16) de la sección anterior, pueden ser usadas para obtener un algoritmo que proporcione la solución del PPL (III.5). Este algoritmo se enuncia formalmente a continuación:

Algoritmo A. Sea $x^0 > 0$, una solución interior tal que $Ax^0 = b$. En general, si x^k es conocido definimos a la matriz

$$D_k = \text{diag}(x_1^k, \dots, x_n^k)$$

y calculamos a la solución interior factible mejorada, $x^{k+1} > 0$, por la fórmula

$$x^{k+1} = x^k - r \frac{D_k^2 [c - A^T \lambda^k]}{\|D_k [c - A^T \lambda^k]\|}, \quad k = 0, 1, 2, \dots$$

donde $\lambda^k = (AD_k^2 A^T)^{-1} AD_k^2 c$ y $\|d_w^k\| \neq 0$.

Cabe mencionar, que el Algoritmo A es de hecho la variante de escala afín, diseñada por Earl R. Barnes, en el Centro de Investigación T. J. Watson de I.B.M., consultar [Bar].

Procederemos ahora a realizar algunas modificaciones sobre el algoritmo A para facilitar la instrumentación computacional del método de escala afín primal. Para este propósito haremos algunas observaciones y enunciaremos algunos lemas que generarán una versión instrumentable del algoritmo.

Como primera observación tenemos que, si $d_w^k \leq 0$, en (III.9), entonces $\alpha_k = r$ puede ser cualquier número positivo entre 0 y 1. De otra forma, si $(d_w^k)_i > 0$ para alguna i , entonces el tamaño de paso es tan pequeño como $\frac{1}{(d_w^k)_i}$.

Por tanto podemos seleccionar a $r \in (0,1)$ y aplicar la siguiente prueba de mínimo cociente

$$\alpha_k = \min_i \left\{ \frac{r}{(d_w^k)_i} \mid (d_w^k)_i > 0 \right\}. \quad (\text{III.20})$$

Para determinar una longitud de paso apropiada que garantice la positividad de w^{k+1} , se recomienda tomar una r cercana a 1. Debemos notar, que si en un momento dado decidimos aplicar directamente el algoritmo de escala afín al problema (III.1), relajando de esta forma la utilización de la bola, el α_k propuesta anteriormente nos permite mantener la factibilidad. Así pues apliquemos directamente este método al problema (III.1) y relajemos la utilización de la bola.

Esta modificación nos lleva a la consideración de que la región factible puede no ser un polítopo, por lo que la definición del tamaño de paso, (III.20) es trascendental para la utilización de la dirección factible discutida en la sección III.1.4. Note también que \mathbb{R}_+^n no tiene un centro real; sin embargo, para un $x \in \mathbb{R}_+^n$ la transformación de escala afín puede ser aplicada, enviando este punto al vector e (debido a la propiedad ii de T_k), el cual es equidistante a cada una de las paredes del ortante no-negativo. Dando como resultado, una vez más, que la región factible sea

distorsionada. Bajo esta consideración, la segunda idea de III.1.1 se sigue aplicando. Este concepto se ilustra en la figura 3.3.

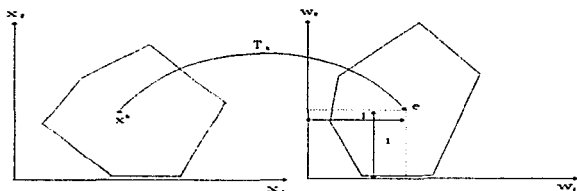


Figura 3.3

Debido a la conclusión anterior, la primera idea de la sección III.1.1 también sigue aplicando, al tomar a la dirección factible de mayor descenso (III.10) junto con el tamaño de paso α_k . Por último puede verificarse que al no normalizar el vector dirección en el Algoritmo A, la bola del PPL (III.7) es relajada.

Como segunda observación tenemos que, si x^k es actualmente un vértice entonces (III.16) puede reescribirse como:

$$\lambda^k = (BD_B^2 B^t)^{-1} B D_B^2 c_B, \quad (\text{III.21})$$

donde $D_B^2 = \text{diag}[(x_1^k)^2, \dots, (x_m^k)^2]$ y, B y c_B son tal como se definieron en el capítulo 1. Simplificando (III.21), vemos que

$$\lambda^k = (B^{-1} D_B^{-2} B^{-1}) B D_B^2 c_B;$$

$$\lambda^k = (B^t)^{-1}c_B. \quad (\text{III.22})$$

Este vector fue definido en la sección 1.3.5 como el vector dual. Por lo tanto diremos que λ^k es el estimado dual (correspondiente a la solución primal x^k) en el algoritmo de escala afín primal. Además, al sustituir (III.22) en el vector de holguras dual definido en la sección 1.3.3

$$v^k = c - A^t \lambda^k,$$

obtenemos $v^k = c - A^t (B^t)^{-1} c_B$, el cual es, el vector de costos reducidos utilizado en el Método Simplex (consultar bibliografía de PL acerca del Método Simplex). Por lo que v^k será el vector de costos reducidos asociado con x^k en el algoritmo de escala afín.

Note que cuando $v^k \geq 0$, el estimado dual λ^k es una solución factible y $(x^k)^t v^k = e^t D_k v^k$ es equivalente al intervalo dual (dado que (III.1) es un PPL en la forma estándar y, como vimos en la sección 1.3.4, la condición $(u^k)^t \lambda^k = 0$ automáticamente se satisface) correspondiente a las soluciones factibles x^k y λ^k , es decir

$$c^t x^k - b^t \lambda^k = e^t D_k v^k.$$

En el caso de que $e^t D_k v^k = 0$, $v^k \geq 0$ y considerando que x^k es una solución factible, las condiciones de KKT (teorema 1.6) se satisfacen y concluimos que x^k es una solución óptima primal y λ^k es solución óptima dual.

A continuación enunciamos algunos lemas necesarios para instrumentar correctamente el algoritmo, las demostraciones de estos lemas pueden consultarse en [Fan].

Lema III.1. Si existe un $x^k \in P^0$ con $d_w^k < 0$, entonces el PPL (III.1) es no acotado.

Como podemos observar si $d_w^k < 0$ entonces $D_k[c-A^t\lambda^k] < 0$; dado que D_k^{-1} existe, tenemos que $c-A^t\lambda^k < 0$. Si ahora obtenemos el PPL dual de (III.1) con λ^k como variables, concluimos, considerando lo anterior, que el problema dual es infactible; y de acuerdo a la tabla I.1 el PPL primal (III.1) es no acotado.

Lema III.2. Si existe un $x^k \in P^0$ con $d_w^k = 0$, entonces la solución factible x^k del PPL (III.1) es óptima.

Note que si $d_w^k = 0$ entonces $D_k[c-A^t\lambda^k] = 0$. Como D_k^{-1} existe, reducimos la ecuación de la dirección a $c-A^t\lambda^k = 0$ y observamos que la factibilidad dual permanece. Ahora bien, para $x^k \in P^0$ la factibilidad primal se satisface así como la condición de holgura complementaria, por lo que las condiciones de KKT se cumplen; y de acuerdo al teorema (I.6) concluimos que, efectivamente, cualquier $x^k \in P^0$ con $d_w^k = 0$ es solución óptima del PPL (III.1). Además dado que las condiciones de holguras complementarias se satisfacen tenemos que $c^t x^k = b^t \lambda^k$ y, como $b^t \lambda^k$ no depende de x^k , se dice que la f.o. primal permanece constante sobre P .

Lema III.3. Si el PPL (III.1) es acotado hacia abajo, y su función objetivo no es constante, entonces la sucesión $\{c^t x^k \mid k = 1, 2, \dots\}$ es monótona decreciente.

Consideremos la ecuación (III.19); dado que estamos proponiendo que el vector dirección no este normalizado y dado que el tamaño de paso es α_k , entonces (III.19) se reescribe como:

$$c^t x^{k+1} = c^t x^k - \alpha_k \|D_k [c - A^t \lambda^k]\|^2. \quad (\text{III.23})$$

Ahora bien, como (III.1) es acotado, entonces del lema (III.1), $d_w^k < 0$; además como la f.o. no debe ser constante, entonces del lema (III.2), $d_w^k \neq 0$; por lo que concluimos que, o $d_w^k > 0$ o sus componentes son positivas y no positivas. En ambos casos $\|d_w^k\|^2 > 0$, por lo que el escalor $\alpha_k \|d_w^k\|^2$ disminuye siempre a $c^t x^k$ en (III.23). Así dadas las condiciones anteriores si $x^{k+1} < x^k$ en

$$x^{k+1} = x^k - \alpha_k D_k d_w^k; \quad (\text{III.24})$$

y además $c^t x^{k+1} < c^t x^k$, entonces $c^t x$ es una función monótona decreciente.

Una vez realizadas las observaciones pertinentes y enunciados los lemas que se consideran adecuados para facilitar la instrumentación del método, procederemos a describir una segunda versión del algoritmo de escala afín primal.

Algoritmo B.

1. Inicialización:

Hacer $k = 0$; asignar un $r \in (0,1)$ y un $\varepsilon > 0$ dependiendo de la precisión requerida; y obtener un $x^0 > 0$ tal que $Ax^0 = b$.

2. Cálculo del vector estimado dual:

$$\lambda^k = (AD_k^2 A^t)^{-1} AD_k^2 c \quad \text{donde } D_k = \text{diag}(x_1^k, \dots, x_n^k).$$

3. Cálculo del vector de costos reducidos:

$$v^k = c - A^t \lambda^k$$

4. Chequeo de las condiciones de optimalidad:

Si $v^k \geq 0$ y $e^t D_k v^k \leq c$, entonces detenerse. x^k es una solución óptima primal y λ^k es la solución óptima dual. En caso contrario, ir a 5.

5. Obtención de la dirección de movimiento:

$$d_w^k = D_k v^k.$$

6. Chequeo de no acotación o valor objetivo constante:

Si $d_w^k < 0$, entonces detenerse. El problema es no acotado. Si $d_w^k = 0$, entonces detenerse. El valor de la f. o. es constante, por lo que x^k es una solución óptima primal. En caso de que alguna de estas condiciones no se cumpla, ir a 7.

7. Cálculo de la longitud de paso:

Si $d_w^k \leq 0$, entonces $\alpha_k = r$.

No, entonces $\alpha_k = \min_i \left\{ \frac{r}{(d_w^k)_i} \mid (d_w^k)_i > 0 \right\}$.

8. Trasládase a la nueva solución mejorada:

Calcular

$$x^{k+1} = x^k - \alpha_k D_k d_w^k;$$

hacer $k = k+1$ e ir a 2.

III.1.6 Convergencia del Algoritmo de Escala Afín Primal.

En esta sección enunciaremos los lemas y teoremas que aseguran que la sucesión $\{x^k\}$ generada por el algoritmo B (sin considerar el paso 6), converge a una solución óptima del PPL (III.1). Para este propósito se realizarán las siguientes suposiciones:

- 1*) El PPL bajo consideración tiene región factible acotada y no vacía.
- 2*) El PPL no tiene soluciones degeneradas tanto en el planteamiento dual como en el primal.

La primera suposición excluye la posibilidad de terminar el algoritmo de escala afín con no acotación. Además, estas dos suposiciones implican que: la matriz AD_k es de rango completo para cualquier $x^k \in P$; y que el vector v^k tiene a lo más m ceros (debido a la condición de holguras complementarias) para cualquier $\lambda^k \in \mathbb{R}^m$.

Ahora, enunciaremos los lemas que nos permitirán, posteriormente, exponer los teoremas relacionados con la convergencia del algoritmo de escala afín.

Lema III.4. Cuando el algoritmo de escala afín es aplicado al PPL (III.1), el $\lim_{k \rightarrow \infty} D_k v^k = 0$.

Este lema nos indica que, bajo las suposiciones anteriores, el algoritmo de escala afín primal alcanzará, a la larga, la solución óptima del PPL en consideración; para ello observe que, en (III.23) $\lim_{k \rightarrow \infty} c^T x^{k+1} = \lim_{k \rightarrow \infty} c^T x^k$, ya que el $\lim_{k \rightarrow \infty} \alpha_k^* = \lim_{k \rightarrow \infty} D_k v^k = 0$ y por tanto $\alpha_k \|d_w^k\|^2 = 0$. Por otro lado debe observarse que si, a la larga, $D_k v^k = 0$, entonces las condiciones de holguras complementarias correspondientes al PPL (III.1) se satisfacen y podemos concluir, de acuerdo a las suposiciones anteriores, que v^k es la solución óptima dual.

La conclusión obtenida anteriormente nos permite definir a $C \subset P$ como el conjunto en el cual las condiciones de holguras complementarias prevalecen, es decir

$$C = \{x^k \in P \mid D_k v^k = 0\};$$

además, podemos definir a $D \subset P$ como el conjunto en el cual las condiciones de factibilidad dual permanecen, es decir

$$D = \{x^k \in P \mid v^k \geq 0\}.$$

Considerando los conjuntos definidos anteriormente procederemos a enunciar el siguiente lema.

Lema III.5. Cualquier $x \in C \cap D$ es solución óptima del PPL (III.1).

Este lema nos indica que x es solución óptima si satisface a ambos conjuntos, pero también indica que x debe satisfacer las condiciones de optimalidad de KKT. Para ello observe que si $x \in C$ o si $x \in D$, entonces $x \in P$, por lo que la factibilidad primal se cumple. Si $x \in D$ se satisface la factibilidad dual y si $x \in C$ se satisfacen las condiciones de holguras complementarias.

Una vez examinados los resultados de los lemas anteriores, estamos en posibilidades de enunciar los teoremas que nos aseguran la convergencia de las soluciones factibles $\{x^k\}$, generadas por el algoritmo B, a la solución óptima x^* del PPL (III.1).

Teorema III.1. Si $\{x^k\}$ converge, entonces $x^* = \lim_{k \rightarrow \infty} x^k$ es una solución óptima del PPL (III.1).

Observe que cuando $\{x^k\}$ converge a x^* , x^* debe ser una solución factible primal. Pero qué sucedería si suponemos que x^* no es una solución óptima primal.

Dado que v^k es una función continua de x en x^k , sabemos que $v^* = \lim_{k \rightarrow \infty} v^k$ se encuentra definido. Asimismo el lema III.4 implica que

$$D^*v^* = \lim_{k \rightarrow \infty} D^k v^k = 0.$$

Por tanto podemos concluir que $x^* \in C$. Pero por nuestra conjetura y por el lema III.5, debemos deducir que $x^* \notin D$. En consecuencia, existe al menos un índice j tal que $v_j^* < 0$. Sin embargo, como $x^* \in C$, entonces $x_j^* = 0$. Además, considerando nuevamente la continuidad de v^k , existe un entero K tal que para cualquier $k \geq K$ se tiene que $\{v_j^k\} < 0$. Por otro lado, de (III.24) tenemos que:

$$x_j^{k+1} = x_j^k - \alpha_k (x_j^k)^2 v_j^k.$$

Como $(x_j^k)^2 v_j^k < 0$ y $\alpha_k > 0$, tenemos que $x_j^{k+1} > x_j^k > 0$, $\forall k \geq K$, lo cual contradice el hecho de que $x_j^k \rightarrow x_j^* = 0$. Por lo tanto nuestra suposición es errónea y podemos concluir que x^* es una solución óptima primal.

Para que el teorema III.1 tenga validez, la sucesión $\{x^k\}$ debe ser convergente. Esto lo podemos verificar en el siguiente teorema.

Teorema III.2. La sucesión $\{x^k\}$ generada por el algoritmo B es convergente.

Bajo las suposiciones enunciadas al principio de este subcapítulo, tenemos que la región factible del PPL (III.1) es no vacía, cerrada¹⁸ y acotada. Por lo que la sucesión $\{x^k\}$ tiene al menos un punto de acumulación¹⁹ en P , digamos x^* . De hecho, x^* es el único punto de acumulación de $\{x^k\}$ y por tanto este llega a ser el límite de $\{x^k\}$.

¹⁸Ver definición de conjunto cerrado en el Apéndice A.

¹⁹Ver definición de punto de acumulación en el Apéndice A.

Las demostraciones de los lemas y teoremas correspondientes a la convergencia del algoritmo B pueden ser consultadas en [Fan]. Los resultados acerca de la convergencia del algoritmo A pueden ser consultados en [Bar]. Resultados acerca de la convergencia bajo degeneración, del algoritmo de escala afín pueden ser consultados en [BCJ].

III.2 EL MÉTODO DE ESCALA AFÍN DUAL.

Consideremos nuevamente el PPL (1.11)

$$\begin{aligned} & \text{Max } b^t \lambda \\ & \text{s. a} \\ & A^t \lambda \leq c \qquad \qquad \qquad \text{(III.25)} \\ & \lambda \text{ irrestricta;} \end{aligned}$$

el cual es el PPL dual de (III.1). Si agregamos variables de holgura a (III.25) obtenemos el siguiente PPL:

$$\begin{aligned} & \text{Max } b^t \lambda \\ & \text{s. a} \\ & A^t \lambda + v = c \qquad \qquad \qquad \text{(III.26)} \\ & v \geq 0 \\ & \lambda \text{ irrestricta.} \end{aligned}$$

Para solucionar (III.26) el método de escala afín es modificado, obteniéndose así una nueva variante conocida como el método de

escala afín dual. En este caso A^t es una matriz $n \times m$ de rango completo con $n \geq m$.

III.2.1 Ideas Básicas del Método de Escala Afín Dual.

La variante de escala afín dual, inicia con una solución interior factible dual y se mueve hacia el óptimo a través del poliedro de soluciones, incrementando progresivamente el valor de la f.o. y conservando la factibilidad dual durante el proceso.

Note que el problema (III.26) contiene tanto a las variables irrestrictas, $\lambda \in \mathbb{R}^m$, como a las variables no negativas, $v \in \mathbb{R}^n$. En este caso, consideraremos a $(\lambda; v)$ como una solución interior factible dual si $A^t \lambda + v = c$ y $v > 0$. Note también que no tiene sentido cambiar la escala de las variables λ , debido a que ellas son irrestrictas, pero las variables v serán tratadas como las variables x del problema primal.

Así pues, suponiendo que la región factible del PPL (III.26) es no vacía y acotada, las ideas básicas citadas en la sección III.1.1 pueden ser nuevamente consideradas y adecuadas para esta variante:

1*) Si la solución interior factible dual es equidistante a las restricciones $v \geq 0$, entonces tiene sentido moverse en la dirección de mayor ascenso de la f.o. para alcanzar el valor máximo;

2*) En caso contrario, sin cambiar el PPL (III.26) en su forma esencial, una transformación apropiada puede ser aplicada a dicho problema, tal que

la solución interior factible dual equidistante de cada una de las restricciones $v \geq 0$.

III.2.2 Transformación de Escala Afín Sobre el Ortante No Negativo Para la Variante Dual.

La transformación empleada por el método de escala afín dual, es la misma que utiliza el método de escala afín primal (definición III.2) para la consecución de la segunda idea básica, por lo que tiene las mismas propiedades. Pero en este caso, el operador de la transformación se define como $V_k^{-1} = \text{diag}(1/v_1^k, \dots, 1/v_n^k)$. Por lo que de la propiedad ii), correspondiente a la definición III.2, $V_k^{-1}v^k = e$ y, cualquier variable v es transformada en una nueva variable $u \geq 0$ tal que

$$u = V_k^{-1}v^k. \quad (\text{III.27})$$

Además, por la propiedad vi) correspondiente a la definición III.2 resulta que

$$v = V_k u. \quad (\text{III.28})$$

donde $V_k = \text{diag}(v^k)$.

Cuando la transformación (III.27) es aplicada sobre las variables v^k , el espacio solución es transformado de tal manera que las variables λ^k son equidistantes a todos los hiperplanos asociados a los semiespacios cerrados, cuya intersección genera el poliedro factible transformado:

$$\{\lambda \in \mathbb{R}^m \mid V_k^{-1}A^T \lambda \leq V_k^{-1}c\}.$$

Reescribiendo el PPL (III.26) en términos de las variables de holgura escaladas, tenemos:

$$\begin{aligned} & \text{Max } b^T \lambda \\ & \text{s. a} \\ & A^T \lambda + V_k u = c \\ & u \geq 0. \end{aligned}$$

El conjunto de soluciones factibles de (III.26) es denotado por:

$$P' = \{ \lambda \in \mathbb{R}^m \mid A^T \lambda + v = c, \forall v \in \mathbb{R}_+^n \},$$

y el conjunto de soluciones factibles con respecto a las variables de holgura escaladas es:

$$\hat{P} = \{ u \in \mathbb{R}_+^n \mid A^T \lambda + V_k u = c, \forall \lambda \in P' \}.$$

III.2.3 Obtención de la Dirección Factible de Mayor Ascenso.

Dado que el problema en consideración es uno de maximización, tendremos que recurrir a la ecuación de ascenso acelerado²⁰, la cual requiere el conocimiento de la dirección factible de mayor ascenso. En nuestro caso, la determinación de esta dirección estará dividida en dos partes. Una parte correspondiente a las variables no negativas y la otra a las variables no restringidas satisfaciendo a los conjuntos P' y \hat{P} respectivamente. Por consiguiente, aunque en la sección anterior no se

²⁰Véase método de ascenso acelerado en el Apéndice B.

establecieron condiciones para las variables no restringidas, se impondrán condiciones sobre su dirección.

Suponga que en la k -ésima iteración, obtenemos una solución factible dual $(\lambda^k; v^k)$ tal que $A\lambda^k + v^k = c$ y $v^k > 0$. Debemos encontrar una buena dirección de movimiento $(d_\lambda^k; d_v^k)$ y una longitud de paso apropiada $\beta_k > 0$, tal que la nueva solución interior factible dual $(\lambda^{k+1}; v^{k+1})$ sea generada por

$$\lambda^{k+1} = \lambda^k + \beta_k d_\lambda^k \quad (\text{III.29})$$

y

$$v^{k+1} = v^k + \beta_k d_v^k; \quad (\text{III.30})$$

y satisfaga a

$$A\lambda^{k+1} + v^{k+1} = c, \quad (\text{III.31})$$

$$v^{k+1} > 0, \quad (\text{III.32})$$

$$b\lambda^{k+1} \geq b\lambda^k. \quad (\text{III.33})$$

Sustituyendo (III.29) y (III.30) en (III.31) y recordando que $A\lambda^k + v^k = c$, obtenemos uno de los requerimientos para la dirección de movimiento, es decir

$$A\lambda^k + \beta_k A^t d_\lambda^k + v^k + \beta_k d_v^k = c;$$

$$\beta_k A^t d_\lambda^k + \beta_k d_v^k = 0;$$

$$A^t d_\lambda^k + d_v^k = 0. \quad (\text{III.34})$$

Con el fin de obtener el mejor valor de la f.o., sustituiremos (III.29) en (III.33), con lo que obtendremos otro de los requerimientos para la dirección de movimiento:

$$b^t \lambda^k + \beta_k b^t c_\lambda^k \geq b^t \lambda^k;$$

$$\beta_k b^t c_\lambda^k \geq 0;$$

$$b^t c_\lambda^k \geq 0.$$

(III.35)

Así pues, el método de escala afín dual puede ser utilizado para solucionar (III.26) siempre y cuando podamos encontrar soluciones interiores factibles duales (λ^k, v^k) tales que (III.31) y (III.32) se cumplan, para $k = 0, 1, 2, \dots$; y direcciones de movimiento $(c_\lambda^k; c_v^k)$ que satisfagan a (III.34) y (III.35), además de un tamaño de paso adecuado. Ahora bien, para que (III.32) se cumpla, el vector v^k , en cada iteración, debe ser enviado a $e = (1, \dots, 1)^t \in \mathbb{R}^n$ en el espacio transformado, de tal manera que la distancia a cada una de las paredes de \mathbb{R}_+^n sea conocida. De esta forma, cualquier movimiento dentro de una distancia unitaria, preservará el requerimiento de positividad.

Hasta este momento hemos obtenido a (III.34) y (III.35) como las condiciones que debe satisfacer la dirección en el espacio original P . Ahora estamos interesados en conocer las características que debe tener dicha dirección en el espacio transformado \hat{P} . Para ello consideremos a c_λ^k como el vector dirección de mayor ascenso de la f.o. en dicho espacio. Similar a lo analizado en el método de escala afín primal, el vector dirección en \hat{P} guarda relación con la dirección en P , pero en este caso la correspondencia es con las direcciones de las variables no negativas y no restringidas.

En primera instancia examinemos la analogía entre c_λ^k y c_v^k . Como sabemos esta última dirección corresponde a las variables v^k , las cuales

deben ser enviadas al vector $e \in \hat{P}$, por lo que requieren la aplicación de la transformación (III.27) discutida en la sección III.2.2. En consecuencia si aplicamos la transformación inversa (III.28) sobre d_k^* su representación en P' será:

$$d_k^* = V_k d_k \quad (\text{III.36})$$

Ahora examinemos la analogía entre d_k^* y d_λ^* . Con el fin de obtener una buena dirección de movimiento en \hat{P} , la condición (III.34) implica que: Si $A^t d_\lambda^* + d_0^* = 0$ entonces $A^t d_k^* + V_k d_k^* = 0$; como V_k es no singular vemos que

$$V_k^{-1} A^t d_k^* + d_0^* = 0;$$

sumando $-d_0^*$ en ambos lados de la ecuación anterior tenemos

$$V_k^{-1} A^t d_k^* = -d_0^*.$$

multiplicando ambos lados por AV_k^{-1} resulta

$$AV_k^{-2} A^t d_k^* = -AV_k^{-1} d_0^*.$$

Asumiendo que A es de rango completo por renglones obtenemos

$$d_k^* = -(AV_k^{-2} A^t)^{-1} AV_k^{-1} d_0^*.$$

Si definimos a $Q_k = (AV_k^{-2} A^t)^{-1} AV_k^{-1}$, la ecuación anterior se reescribe como

$$d_k^* = -Q_k d_0^* \quad (\text{III.37})$$

La ecuación (III.37) indica que d_k^* actualmente esta determinada por d_0^* en el espacio transformado. Así pues, si logramos encontrar una dirección apropiada d_0^* tal que (III.35) se satisfaga, entonces habremos

encontrado la dirección buscada. Es decir, dado que d_k^* está determinada por (III.37), debemos encontrar un vector, tal que su producto interno con d_k^* sea no negativo. Observe que si realizamos el producto interno de un vector consigo mismo, obtendremos la norma cuadrada de dicho vector, la cual siempre es positiva o cero en caso de que todas las componentes del vector sean cero. Bajo esta noción proponemos a

$$d_k^* = -Q_k^t b. \quad (\text{III.38})$$

Entonces tenemos:

$$b^t d_k^* = -b^t Q_k d_k^* = b^t Q_k Q_k^t b = \|b^t Q_k\|^2 \geq 0.$$

Sustituyendo (III.38) en (III.37) obtenemos:

$$d_k^* = Q_k Q_k^t b. \quad (\text{III.39})$$

Como vimos anteriormente

$$Q_k = (AV_k^{-2}A^t)^{-1}AV_k^{-1}, \quad (\text{III.40})$$

además $Q_k^t = (AV_k^{-1})^t[(AV_k^{-2}A^t)^{-1}]^t$, o lo que es lo mismo:

$$Q_k^t = V_k^{-1}A^t(AV_k^{-2}A^t)^{-1}. \quad (\text{III.41})$$

Sustituyendo ahora (III.40) y (III.41) en (III.39) resulta que

$$\begin{aligned} d_k^* &= (AV_k^{-2}A^t)^{-1}(AV_k^{-2}A^t)(AV_k^{-2}A^t)^{-1}b; \\ d_k^* &= (AV_k^{-2}A^t)^{-1}b; \end{aligned} \quad (\text{III.42})$$

que es la dirección de movimiento correspondiente a las variables no restringidas.

Consecuentemente, sustituyendo a (III.42) en (III.34) y, posteriormente despejando a d_k^* , obtendremos la dirección de movimiento correspondiente a las variables no negativas:

$$\begin{aligned} A^t d_k^* + d_k^* &= 0; \\ A^t (AV_k^{-2} A^t)^{-1} b + d_k^* &= 0; \\ d_k^* &= -A^t (AV_k^{-2} A^t)^{-1} b. \end{aligned} \quad (\text{III.43})$$

III.2.4 Selección de Tamaño de Paso.

Una vez que la dirección de movimiento ($d_k^*; d_k^*$) es conocida, la longitud de paso β_k es obtenida de acuerdo a los requerimientos de positividad de v^{k+1} , tal y como se hizo para x^{k+1} en el algoritmo de escala afín primal, es decir:

1°. Si $d_k^* = 0$, entonces (III.26) tiene un valor objetivo constante en su región factible y $(\lambda^k; v^k)$ forman una solución óptima dual.

2°. Si $d_k^* \geq 0$, pero $d_k^* = 0$, entonces el problema (III.26) es no acotado.

3°. Si $d_k^* > 0$, entonces $\beta_k = r$:

$$\text{No, entonces: } \beta_k = \min \left\{ \frac{rv^k}{-(d_k^*)_i} \mid (d_k^*)_i < 0 \right\}; \quad 0 < r < 1.$$

Note que, similar a la forma en que definimos el estimado dual en el algoritmo de escala afín primal, podemos definir a

$$x^k = -V_k^{-2} d_k^* \quad (\text{III.44}),$$

entonces $Ax^k = -AV_k^{-2} d_k^*$. Como $d_k^* = -A^t d_k$ tenemos que $Ax^k = (AV_k^{-2} A^t) d_k$; además de (III.42), $d_k^* = (AV_k^{-2} A^t)^{-1} b$, con lo que concluimos que $Ax^k = b$.

Por tanto, x^k puede ser considerada como el "estimado primal" en el algoritmo de escala afín dual. Cuando $x^k \geq 0$, entonces x^k es una solución factible primal, con un intervalo dual $c^t x^k - b^t \lambda^k$. Además, si $c^t x^k - b^t \lambda^k = 0$, entonces (λ^k, v^k) debe ser una solución óptima dual y x^k una solución óptima primal.

III.2.5 El Algoritmo de Escala Afín Dual.

Basándonos en los conceptos discutidos en las secciones previas, procederemos a enunciar formalmente el algoritmo de escala afín dual para solucionar PPL con una estructura como la de (III.26).

1. Inicialización:

Hacer $k = 0$: asignar un $r \in (0, 1)$ y un $\epsilon > 0$ dependiendo de la precisión requerida; encontrar una solución inicial (λ^0, v^0) tal que $A^t \lambda^0 + v^0 = c$ y $v^0 > 0$.

2. Obtención de la dirección de movimiento:

Hágase $V_k^{-1} = \text{diag}(1/v_1^k, \dots, 1/v_n^k)$. Calcular

$$d_k^* = (AV_k^{-2} A^t)^{-1} b \quad \text{y} \quad d_k = -A^t d_k^*.$$

3. Chequeo de no acotación o valor objetivo constante:
 Si $d_k^k \geq 0$ y $d_k^k \neq 0$ entonces detenerse. El PPL dual es no acotado. Si $d_k^k = 0$, entonces detenerse. El valor de la f. o. es constante, por lo que $(\lambda^k; v^k)$ es una solución óptima dual. En caso de que alguna de estas condiciones no se cumpla, ir a 4.
4. Cálculo del estimado primal:

$$x^k = -V_k^{-2} d_k^k.$$
5. Chequeo de las condiciones de optimalidad:
 Si $x^k \geq 0$ y $c^k x^k - b^k \lambda^k \leq \epsilon$, entonces detenerse. $(\lambda^k; v^k)$ es una solución óptima dual y x^k es la solución óptima primal. En caso contrario, ir a 6.
6. Cálculo de la longitud de paso:
 Si $d_k^k > 0$, entonces $\beta_k = r$.
 No, entonces $\beta_k = \min \left\{ \frac{r v^k}{-(d_k^k)_i} \mid (d_k^k)_i < 0 \right\}$.
7. Trasiadarse a la nueva solución mejorada:
 Calcular

$$\lambda^{k+1} = \lambda^k + \beta_k d_k^k$$
 y
$$v^{k+1} = v^k + \beta_k d_k^k.$$
 Hacer $k = k+1$ e ir a 2.

Considerando el hecho de que el algoritmo de escala afín dual es equivalente al algoritmo de escala afín primal, sólo que el primero

aplicado a la formulación dual del segundo, las propiedades de convergencia para ambas variantes son similares, por lo que no se mencionará aquí la correspondiente a dualidad.

En el siguiente capítulo analizaremos las técnicas numéricas que se consideran adecuadas para la instrumentación computacional de las variantes examinadas del método de escala afín.

CAPÍTULO IV

ASPECTOS NUMÉRICOS A CONSIDERAR EN LA INSTRUMENTACIÓN COMPUTACIONAL DE LOS MÉTODOS DE ESCALA AFÍN

Para que la instrumentación computacional de los algoritmos considerados en el capítulo anterior (en específico, el algoritmo B de la sección III.1.5 y el algoritmo de la sección III.2.5) pueda ser efectuada correctamente procederemos, en este capítulo a describir los mecanismos de inicio y de verificación de paro que se consideran adecuados tanto para el método de escala afín primal como para el método de escala afín dual. También se explicará cómo se detectará la no acotación para ambos algoritmos desde el punto de vista de la instrumentación, así como si la variable artificial ha alcanzado o no la nulidad. Además, para ambos algoritmos, se describen cuatro métodos de factorización matricial, que permiten resolver el sistema de ecuaciones lineales que se presenta en el cálculo del vector dirección.

IV.1 OBTENCIÓN DE UNA SOLUCIÓN INTERIOR INICIAL

En esta sección se expondrán, los métodos seleccionados para la obtención de una solución interior inicial factible, tanto en el método de escala afín primal como en el método de escala afín dual.

IV.1.1 Obtención de una Solución Interior Inicial factible para el Método de Escala Afín Primal.

El mecanismo considerado en este trabajo para encontrar una solución interior inicial en la instrumentación computacional del método de escala afín primal, requiere aumentar una variable artificial x_a , asociada con un número positivo M , lo suficientemente grande, al PPL (III.1). Así el vector $(1, \dots, 1)^t \in \mathbb{R}^{n+1}$ es una solución interior inicial factible del siguiente PPL:

$$\begin{aligned} & \text{Min } c^t x + M x_a \\ & \text{s. a} \\ & [A \mid b - Ae] \begin{bmatrix} x \\ x_a \end{bmatrix} = b \quad (\text{IV.1}) \\ & x \geq 0, x_a \geq 0 \end{aligned}$$

donde $e = (1, \dots, 1)^t \in \mathbb{R}^n$.

El algoritmo B, descrito en el capítulo anterior, puede ser ahora aplicado al problema (IV.1), el cual tiene $n+1$ variables y cuya solución interior inicial es ya conocida. Al aplicar dicho algoritmo a (IV.1), con una M lo suficientemente grande y considerando que el problema es factible, obtendremos una solución óptima ó concluiremos que el problema es no acotado. En caso de obtenerse una solución óptima (x^*, x_a^*) para (IV.1) con $x_a^* = 0$, entonces x^* es una solución óptima para el PPL (III.1). En caso de detectarse no acotación en (IV.1) con $x_a^* = 0$, entonces el PPL (III.1) es también no acotado. Para ambos casos, si $x_a^* \neq 0$ entonces (III.1) es infactible.

IV.1.2 Obtención de una Solución Interior Inicial Factible para el Método de Escala Afin Dual.

En el caso del PPL (III.26) requerimos encontrar una solución interior factible (λ^0, v^0) tal que $A^t \lambda^0 + v^0 = c$ y $v^0 > 0$. Si en (III.26) $c > 0$, entonces podríamos seleccionar $\lambda^0 = 0$ y $v^0 = c$ como una solución interior inicial para el algoritmo de escala afin dual. Pero como esto no siempre sucede, necesitamos agregar algunas otras condiciones que nos permitan obtener una solución interior factible inicial para el algoritmo descrito en la sección III.2.5.

Al igual que en la sección anterior, debemos utilizar una variable artificial, en este caso λ_a , asociada con un número positivo M lo suficientemente grande. Pero en esta ocasión λ_a debe ser sustraída de la f. o. y de cada una de las restricciones con $c_i \leq 0$. De esta forma, de (III.26), se obtiene el siguiente PPL aumentado:

$$\begin{aligned} \text{Max } & b^t \lambda - M \lambda_a \\ \text{s. a} & \\ & A^t \lambda - q \lambda_a + v = c \quad (\text{IV.2}) \\ & v \geq 0 \\ & \lambda, \lambda_a \text{ irrestrictas.} \end{aligned}$$

donde $q \in \mathbb{R}^n$ es un vector columna cuya i -ésima componente $i = 1, \dots, n$, está definida por

$$q_i = \begin{cases} 1 & \text{si } q \leq 0 \\ 0 & \text{si } q > 0 \end{cases}$$

Ahora requerimos definir a $\bar{c} = \max |c_i|$ y seleccionar un parámetro $\theta > 1$. Así, podemos inicializar a $\lambda_a = \theta \bar{c}$; además, dado que λ es irrestricta entonces haremos $\lambda = 0$, con lo que $v = c + q\lambda_a$.

Considerando los resultados anteriores, se puede verificar que el vector $(0; \theta \bar{c}; c + q\theta \bar{c})^t$ es una solución interior factible inicial para el PPL (IV.2), ya que $c + q\lambda_a > 0$. Por lo que el algoritmo de la sección (III.2.5) puede ser aplicado a (IV.2) con la solución interior inicial factible obtenida.

Observe que λ_a inicia con $\theta \bar{c} > 0$ y que su coeficiente en la f. o., M , es un número muy grande, sustraído de los coeficientes de la f. o. original. Dado que la presencia de dicho valor es poco atractiva para la f. o., el algoritmo de escala afín dual obligará a la variable artificial a decrementarse hasta llegar a ser no positiva (debido a que λ_a es irrestricta). Esto, siempre y cuando el PPL (III.26) sea factible. Cuando λ_a se aproxima o llega a cruzar a cero en la k -ésima iteración, podemos tomar a $\hat{\lambda} = \lambda^k$ y $\hat{v} = v^k - q\lambda_a$ como una solución interior inicial factible para el PPL dual (III.26). Si λ_a no se aproxima o cruza cero, entonces podemos concluir que (III.26) es infactible. Por último debemos mencionar que en el caso de que $q = 0$, entonces en la f.o. de (IV.2), $M = 0$, con lo que el algoritmo de la sección (III.2.5) puede ser aplicado directamente al PPL dual original.

IV.2 SELECCIÓN DE MÉTODO DE PARO.

Tanto el algoritmo de escala afín primal (algoritmo B) como el algoritmo de escala afín dual incluyen, aunque de forma implícita, las condiciones de optimalidad de KKT (teorema 1.6) para verificar, en cada iteración, si una solución es o no óptima, para el algoritmo correspondiente. Es decir, las condiciones de factibilidad primal, factibilidad dual y holguras complementarias son consideradas como métodos de paro.

Sin embargo, para nuestra instrumentación computacional, estas condiciones deben ser rescritas para poder manejar las dificultades numéricas debidas a limitaciones en la exactitud de la computadora [Fan].

Por otro lado, en la sección III.1.5 y en la sección III.2.4 se especifica que los algoritmos pueden parar al detectar no acotación o función objetivo constante. Por lo que en este subcapítulo se detallan también los métodos seleccionados para tales casos y se indica como fueron instrumentados.

IV.2.1 Condiciones de KKT como Reglas de Paro para el Método de Escala Afín Primal.

Sea x^k una solución interior factible obtenida al aplicar el algoritmo de escala afín primal. Las condiciones de KKT, requieren:

1) Factibilidad primal

$$Ax^k = b, \quad x^k \geq 0.$$

En nuestra instrumentación la factibilidad primal será medida por

$$f_p = \frac{\|Ax^k - b\|}{\|b\| + 1} \text{ con } x^k \geq 0. \quad (\text{IV.3})$$

Así, si $x^k \geq 0$ y f_p es suficientemente pequeño, aceptaremos que x^k es una solución factible primal. La adición del 1 en el denominador de (IV.3) es para asegurar estabilidad numérica en el cálculo.

2) Factibilidad dual.

La factibilidad dual requiere que el vector de costos v^k sea no negativo, es decir, $v^k = c - A^T \lambda^k \geq 0$, donde λ^k es el estimado dual definido en la ecuación (III.16).

En nuestra instrumentación la factibilidad dual será medida por

$$f_d = \frac{\|\hat{v}^k\|}{\|\hat{c}\| + 1}.$$

donde \hat{v}^k es un vector cuyos elementos son los elementos de v^k , tales que $v_i^k < 0$ y, \hat{c} es un vector constituido por los coeficientes de la f.o. correspondientes a las componentes de \hat{v}^k . Cuando f_d sea lo suficientemente pequeña, aceptaremos que λ^k satisface la factibilidad dual.

3) Holguras complementarias.

Las condiciones de holguras complementarias requieren que

$$(x^k)^t v^k = 0.$$

Dado que $c^t x^k - b^t \lambda^k = e^t D_k v^k$ donde $D_k = \text{diag}(x^k)$, x^k es primal factible y λ^k es dual factible, en nuestra instrumentación utilizaremos

$$h_c = \frac{|c^t x^k - b^t \lambda^k|}{|c^t x^k| + 1}$$

para medir las condiciones de holguras complementarias. Si $f_p \leq \epsilon_1$, $f_d \leq \epsilon_2$ y $h_c \leq \epsilon_3$ con $\epsilon_1 > 0$, $\epsilon_2 > 0$ y $\epsilon_3 > 0$ como tolerancias dadas, lo suficientemente pequeñas, pararemos el método con x^k como solución óptima.

IV.2.2 Condiciones de KKT como Reglas de Paro para el Método de Escala Afín Dual.

Similar a lo discutido anteriormente para el método de escala afín primal, en esta sección utilizaremos las condiciones de KKT para generar las reglas de paro del método de escala afín dual.

Sea (λ^k, v^k) una solución interior factible obtenida al aplicar el algoritmo de escala afín dual, entonces se requiere checar:

1) Factibilidad primal.

Sustituyendo (III.43) en (III.44), tenemos que el estimado primal está dado por

$$x^k = -V_k^{-2} d_k^* = V_k^{-2} A^t (A V_k^{-2} A^t)^{-1} b. \quad (\text{IV.4})$$

Se puede verificar que las restricciones explícitas $Ax = b$ son satisfechas para cualquier x^k que esté definido de acuerdo a (IV.4); además, si $x^k \geq 0$, entonces esta solución debe ser factible primal; para medir dicha factibilidad utilizaremos nuevamente a:

$$f_p = \frac{\|Ax^k - b\|}{\|b\| + 1} \text{ con } x^k \geq 0.$$

2) Factibilidad dual.

Para medir la factibilidad dual de (IV.2), debemos verificar que $v^k \geq 0$ y que $A^t \lambda^k - q \lambda_s^k + v^k = c$. Por lo que en nuestra instrumentación computacional utilizaremos la siguiente regla que nos permite efectuar tal medición:

$$f_d = \frac{\|c - A^t \lambda^k - v^k + q \lambda_s^k\|}{\|c\| + 1} \text{ con } v^k \geq 0.$$

3) Holguras complementarias.

Una vez que se ha verificado la factibilidad primal de x^k y la factibilidad dual de λ^k , entonces la condición de holguras complementarias es probada por

$$h_c = \frac{|c^t x^k - b^t \lambda^k|}{|b^t \lambda^k| + 1}$$

Así pues, similar a lo discutido en el caso del método de escala afín primal, si $f_p \leq \epsilon_1$, $f_d \leq \epsilon_2$ y $h_c \leq \epsilon_3$ con $\epsilon_1 > 0$, $\epsilon_2 \geq 0$ y $\epsilon_3 > 0$ como tolerancias dadas lo suficientemente pequeñas, pararemos el método con λ^k como solución óptima.

IV.2.3 Detección de No Acotación.

Otra de las formas de paro que se tiene en los métodos instrumentados es cuando se detecta no acotación en el PPL primal o dual. En el capítulo III, tanto para el algoritmo de escala afín primal (sección III.1.5) como para el algoritmo de escala afín dual (sección III.2.4) se proporcionaron reglas que permitían la detección de no acotación para la f.o. en base al vector dirección. Pero dichas condiciones están incompletas desde el punto de vista computacional por lo que es necesario agregar otra condición más.

En su artículo [AKR] recomiendan que la f.o. sea considerada como no acotada cuando éste exceda a una cota dada. Por otro lado, [MyM] indican que el PPL puede ser determinado como no acotado si el tamaño de paso excede a una cota dada. En nuestras primeras experimentaciones con las instrumentaciones computacionales, observamos que el tamaño de paso se veía afectado por el condicionamiento de la matriz, y que éste podía tomar valores grandes sin que necesariamente en el problema a solucionar, la f.o. fuera no acotada. Por lo que se decidió tomar la primera recomendación comparando el valor absoluto de la f.o. (a partir de que la variable artificial es cero) contra una cota proporcionada.

IV.2.4 Análisis del Vector Dirección para detección de Función Objetivo Constante y Selección de Tamaño de Paso.

Además de los casos citados anteriormente, los algoritmos de escala afín primal y dual paran su ejecución cuando detectan función objetivo constante. En los métodos a instrumentar el vector dirección es utilizado para checar el valor objetivo constante. En ambos algoritmos, si el vector dirección es igual al vector cero, se considera que el valor de la f.o. es constante y los métodos se detienen. Para realizar tal verificación, en nuestra instrumentación y evitar en lo posible los errores numéricos, decidimos tomar la norma 2 de dichos vectores. Si las normas son menores a $\epsilon > 0$, donde ϵ es un valor muy pequeño, las instrumentaciones detectan f.o. constante. Esto es debido a que el único vector con norma euclidiana cero es el vector cero.

Por otro lado, el vector dirección es utilizado para el cálculo de la longitud de paso de cada método. Así, en la versión primal, si algún elemento del vector dirección es mayor que cero (menor que cero en el caso dual), éste es utilizado para la prueba de mínimo cociente. Como observamos en las primeras evaluaciones de nuestras instrumentaciones, es posible obtener tamaños de paso muy grandes debido a que los elementos considerados en la prueba citada, son mayores que cero pero muy pequeños (y sin que aún se haya detectado f.o. constante). Por lo que en vez de comparar contra el cero, compararemos contra una $\epsilon > 0$ muy pequeña (contra una $\epsilon < 0$ en el caso dual).

IV.2.5 Evaluación de la Variable Artificial.

Aunque en esta sección no tratamos con un método de paro en específico, ésta es considerada para ser incluida en este subcapítulo ya que la evaluación de la variable artificial debe ser efectuada precisamente una vez que los algoritmos paren por cualesquiera de los métodos citados anteriormente. Ambas instrumentaciones, utilizan como mecanismo para la obtención de la solución interior inicial la M grande, como se expuso en la sección IV.1. En dicha sección se pide evaluar para ambos algoritmos, si la variable artificial en el problema aumentado es igual a cero (o cruza cero en el caso dual). Como vimos al principio de este capítulo, errores numéricos de redondeo pueden afectar dicha evaluación ya que desde el punto de vista computacional la variable artificial no llega a ser cero exactamente. Lo que puede ocasionar que una vez que cualesquiera de los algoritmos pare, detecte infactibilidad sin que esto sea cierto (inclusive en el caso primal pueda alcanzar la solución óptima y decir que el PPL es infactible dado que la variable artificial no es exactamente cero). Para evitar este problema determinaremos que el PPL es infactible, cuando las condiciones de paro sean satisfechas (con optimalidad o no acotación) y $|x_a| > \epsilon$, en el caso primal, o $\lambda_a > \epsilon$, en el caso dual. Además, si en la instrumentación computacional del método de escala afín dual no se detecta que λ_a no se aproxima a ϵ o llega a ser negativa, se considera al PPL dual como infactible.

IV.3 FACTORIZACIONES MATRICIALES PARA EL CÁLCULO DE LA DIRECCIÓN.

En esta sección se describirán las técnicas de descomposición matricial para resolver el sistema de ecuaciones implícito en la dirección de búsqueda, tanto del método de escala afín primal como del método de escala afín dual. Para ello, en primer término, se explicarán cuales son los sistemas de ecuaciones a resolver en cada método.

IV.3.1 Los Sistemas de Ecuaciones Lineales en los Vectores Dirección.

Como podemos observar en la ecuación (III.10), correspondiente a la mejor dirección factible primal, y en las ecuaciones (III.42) y (III.43), correspondientes a la mejor dirección factible dual, se requiere el cálculo de la inversa de una matriz, a saber, $(A_k A_k^t)^{-1} = (AD_k^2 A^t)^{-1}$ en el caso primal y $(A V_k^{-2} A^t)^{-1}$ en el caso dual.

En específico, en el paso 2 del algoritmo B, se requiere calcular

$$\lambda^k = (AD_k^2 A^t)^{-1} AD_k^2 c \quad (IV.5)$$

donde $D_k = \text{diag}(x^k)$; $x^k > 0$; $c, x^k \in \mathbb{R}^n$ y $A \in \mathbb{R}^{m \times n}$ es una matriz de rango completo por renglones; y, en el paso 2 del algoritmo de escala afín dual (sección (III.2.5)), debemos obtener

$$d_k^* = (A V_k^{-2} A^t)^{-1} b \quad (IV.6)$$

y

$$d_v^k = -A^1 d_x^k$$

donde $V_k^{-1} = \text{diag}\left(\frac{1}{v_1^k}, \dots, \frac{1}{v_n^k}\right)$, $v_1^k, \dots, v_n^k > 0$; $b \in \mathbb{R}^m$ y $A \in \mathbb{R}^{m \times n}$ de rango completo por renglones.

Dado que las matrices de cambio de escala D_k y V_k^{-1} , correspondientes al algoritmo primal y dual respectivamente, son matrices diagonales cuyos elementos en la diagonal principal son positivos, dichas matrices son no singulares y de rango completo por renglones. Además, al multiplicarlas por A , que es una matriz de rango completo por renglones, el producto es una matriz de rango completo por renglones²¹. Por otro lado, considerando que el producto de una matriz de $m \times n$ de rango completo por renglones con su transpuesta, es una matriz simétrica de $m \times m$ de rango completo por renglones,²² es decir es invertible, tanto (IV.5) como (IV.6) pueden reescribirse como

$$(AD_k^2 A^1) \lambda^k = AD_k^2 c \quad (\text{IV.7})$$

y

$$(AV_k^{-2} A^1) d_x^k = b \quad (\text{IV.8})$$

Como podemos notar (IV.7) y (IV.8) son ahora sistemas de ecuaciones lineales, con λ^k y d_x^k como incógnitas, respectivamente. Observe además que (IV.7) y (IV.8) pueden escribirse en forma general como:

²¹Ver rango de una matriz en el Apéndice A.

²²Ver teorema A2 en el Apéndice A.

$$Mu = v, \quad (IV.9)$$

donde $M = (A \ominus A^t)$. En este caso vemos que \ominus , u y v dependen del algoritmo que está siendo utilizado. Además, para ambos casos \ominus es una matriz diagonal de $n \times n$ que cambia en cada iteración.

De acuerdo a lo reportado en [HyM], [JMS], [PQV] y [Sou], es precisamente la solución de (IV.9) el paso que ocupa entre el 80 y 90 por ciento del tiempo computacional. Es por esta razón que el sistema (IV.9) debe ser solucionado de forma eficiente. Para este propósito consideraremos los siguientes métodos numéricos para la solución de sistemas de ecuaciones lineales: factorización simétrica indefinida, factorización de Cholesky, factorización QR y mínimos cuadrados con factorización QR.

IV.3.2 Factorización Simétrica Indefinida.

Como vimos en la sección anterior, la matriz M , correspondiente al sistema (IV.9), es simétrica; por lo que es necesario encontrar una factorización que retenga la simetría. Debemos tener en cuenta que si durante la factorización de M , se requiere algún intercambio de renglones o de columnas (aunque sea uno), la simetría de M se pierde. La única forma de evitar esta pérdida, es efectuando los mismos intercambios de renglones que de columnas, es decir, si los renglones i y j son intercambiados, las columnas i y j deben ser intercambiadas.

Consideremos a la matriz de permutación²³ P . El producto PM genera una matriz en la que se ha realizado un intercambio de renglones en M . Para cumplir con el requerimiento citado anteriormente, debemos obtener PMP^t , ya que al multiplicar P^t a la derecha de PM se realizan los mismos intercambios de columnas que de renglones en M , conservándose así la simetría en PMP^t .

Si además M es no singular, una permutación P existe, tal que el elemento $(1,1)$ de PMP^t es no cero ó el bloque²⁴ de 2×2 superior izquierdo de PMP^t es no singular. Esta propiedad garantiza la existencia de la siguiente factorización

$$PMP^t = LBL^t \quad (IV.10)$$

donde P es una permutación, L es una matriz triangular inferior unitaria y B es una matriz simétrica con diagonal en bloques de dimensión 1×1 ó 2×2 . La factorización (IV.10) es conocida como *la factorización simétrica indefinida*.

Así pues, los pivotes que pueden utilizarse para efectuar dicha factorización, pueden ser de 1×1 ó de 2×2 .

Por el momento, supongamos que no se requieren de permutaciones, y que la matriz M , puede descomponerse tal que

²³Ver definición de matriz de permutación en el Apéndice A.

²⁴Ver definición de bloque de una matriz en el Apéndice A.

$$\begin{pmatrix} M_{11} & M_{21}^i \\ M_{21} & M_{22} \end{pmatrix} = LBL^t. \quad (\text{IV.11})$$

Si seleccionamos un pivote de 1×1 , quiere decir que M_{11} es un escalar y que puede ser utilizado para realizar la eliminación gaussiana por renglones (consultar [GMW]).

Cuando un pivote de 2×2 es seleccionado, M , B y L en (IV.11) deben satisfacer:

$$\begin{pmatrix} M_{11} & M_{21}^i \\ M_{21} & M_{22} \end{pmatrix} = \begin{pmatrix} I_2 & 0 \\ L_{21} & I_{n-2} \end{pmatrix} \begin{pmatrix} B_{11} & 0 \\ 0 & C \end{pmatrix} \begin{pmatrix} I_2 & L_{21}^t \\ 0 & I_{n-2} \end{pmatrix}$$

donde $I_2 = I \in \mathbb{R}^{2 \times 2}$, $I_{n-2} = I \in \mathbb{R}^{(n-2) \times (n-2)}$, $B_{11} \in \mathbb{R}^{2 \times 2}$, y $L_{21} \in \mathbb{R}^{(n-2) \times 2}$.

Efectuando operaciones vemos que

$$B_{11} = M_{11} \quad \text{y} \quad L_{21} = M_{21} B_{11}^{-1}$$

donde

$$B_{11}^{-1} = \frac{1}{m_{11}m_{22} - m_{21}^2} \begin{pmatrix} m_{22} & -m_{21} \\ -m_{21} & m_{11} \end{pmatrix}$$

Estos pasos son repetidos, para la secuencia de matrices restantes cada vez más pequeñas, las cuales son obtenidas al acortar en dimensión por 1 ó 2 a la matriz actual de cada paso, dependiendo de la dimensión del pivote. Por último, cuando la matriz restante C es de 1×1 su valor puede ser obtenido a través de la matriz que le antecede de la siguiente forma:

$$C = M_{22} - L_{21}M_{11}^{-1}$$

Como puede observarse, la selección del pivote a utilizar así como su dimensión es muy importante. Para que dicha selección pueda realizarse adecuadamente y en forma estable se tiene el siguiente algoritmo de pivoteo parcial en la factorización LBL¹:

ALGORITMO DE PIVOTEO PARCIAL

1. Determinar el elemento de mayor valor absoluto fuera de la diagonal principal en la primera columna, denotar al renglón correspondiente por r y a dicho elemento por m_{1r} .
2. Si $|m_{1r}| \geq \tau|m_{11}|$ donde $0 < \tau < 1$, no efectuar intercambios y usar m_{11} como un pivote de 1×1 . Se recomienda tomar $\tau = 0.64$ [GMW].
3. De otra forma, buscar en la columna r el elemento de mayor valor absoluto fuera de la diagonal principal (no incluyendo $m_{1r} = m_{r1}$); denótese a este elemento por β_r .
4. Si $|m_{1r}\beta_r| \geq \tau m_{11}^2$, no ejecutar intercambios, y usar a m_{11} como un pivote de 1×1 .
5. Si $|m_{rr}| \geq \tau|\beta_r|$, seleccionar la permutación P para intercambiar los renglones 1 y r , y usar un pivote de 1×1 (en este caso, m_{rr} será el elemento pivote).

6. Finalmente, si una de las condiciones anteriores no se satisface, seleccionar la permutación P para intercambiar los renglones 2 y r , y usar un pivote de 2×2 . En este caso, el pivote será:

$$\begin{pmatrix} m_{11} & m_{r1} \\ m_{r1} & m_{rr} \end{pmatrix}.$$

El algoritmo anterior debe ser usado en cada paso de la factorización LBL^t para asegurar estabilidad numérica, para que el pivote de 2×2 esté bien condicionado²⁵, y para poder controlar el crecimiento de las matrices reducidas. Observe también que $P_j P_{j-1} \dots P_1 = P$, donde j es el número de intercambios de renglones utilizados durante el proceso de factorización.

Para concluir, una vez que M ha sido factorizada en LBL^t , el sistema (IV.9) puede resolverse de la siguiente forma:

Sea $\bar{u} = Pu$ y $\bar{v} = Pv$, el sistema $Mu = v$ es equivalente a

$$LBL^t \bar{u} = \bar{v}, \quad (\text{IV.12})$$

donde \bar{u} y \bar{v} son versiones "perturbadas" de u y v . Para resolver (IV.12), agrupamos las matrices y solucionamos sucesivamente.

$$Ly = \bar{v}, \quad Bz = y, \quad L^t \bar{u} = z \text{ y tomamos } u = P^t \bar{u}.$$

²⁵Ver número de condición en el Apéndice A.

Los sistemas $Ly = \bar{v}$ y $L^t\bar{u} = z$ involucran matrices triangulares inferiores, y son fáciles de resolver. El sistema $Bz = y$ es solucionado en bloques. Particionando z e y en bloques conformables con los bloques de B , tenemos

$$\begin{pmatrix} B_{11} & & & 0 \\ & B_{22} & & \\ & & \dots & \\ 0 & & & B_{kk} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{pmatrix},$$

donde z_i e y_i son escalares si B_{ii} es de un sólo elemento, y vectores de dimensión 2 si B_{ii} es de 2×2 . Cada bloque en z puede ser calculado directamente de la relación $B_{ii}z_i = y_i$.

Para una revisión más amplia de este método, consultar [GMW] y [DMB].

IV.3.3 Factorización de Cholesky.

Bajo la consideración de que la matriz A es de rango completo, de que la matriz diagonal tiene entradas positivas y de que el producto de la matriz A y la matriz diagonal produce también una matriz de rango completo, la matriz M del sistema (IV.9) resulta ser una matriz definida positiva simétrica (consultar [Kar], [JRM], [Sau]). La más famosa y extremadamente útil factorización para este tipo de matrices especiales, es conocida como la *factorización de Cholesky*.

Esta factorización descompone a la matriz M en el producto LL^t , donde L es una matriz triangular inferior. A continuación mostraremos como lograr esta factorización para cuando $M \in \mathbb{R}^{3 \times 3}$, para después generalizar este resultado.

Sea $M \in \mathbb{R}^{3 \times 3}$ una matriz definida positiva simétrica tal que:

$$M = \begin{pmatrix} m_{11} & m_{21} & m_{31} \\ m_{21} & m_{22} & m_{32} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}.$$

Sea L una matriz triangular inferior dada por:

$$L = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix}.$$

Si $M = LL^t$ tenemos que

$$\begin{pmatrix} m_{11} & m_{21} & m_{31} \\ m_{21} & m_{22} & m_{32} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{pmatrix}.$$

Por lo que los valores correspondientes para la matriz L son los siguientes:

$$m_{11} = l_{11}^2 \Rightarrow l_{11} = \sqrt{m_{11}};$$

$$m_{21} = l_{21}l_{11} \Rightarrow l_{21} = \frac{m_{21}}{l_{11}};$$

$$m_{22} = l_{21}^2 + l_{22}^2 \Rightarrow l_{22} = \sqrt{m_{22} - l_{21}^2};$$

$$m_{31} = l_{31}l_{11} \Rightarrow l_{31} = \frac{m_{31}}{l_{11}};$$

$$m_{32} = l_{31}l_{21} + l_{32}l_{22} \Rightarrow l_{32} = \frac{m_{32} - l_{31}l_{21}}{l_{22}} \text{ y}$$

$$m_{33} = l_{31}^2 + l_{32}^2 + l_{33}^2 \Rightarrow l_{33} = \sqrt{m_{33} - l_{31}^2 - l_{32}^2}.$$

Estos resultados pueden ser generalizados para obtener un algoritmo que efectúe la factorización de Cholesky sobre cualquier matriz $M \in \mathbb{R}^{n \times n}$ definida positiva simétrica. Este algoritmo de descomposición matricial será descrito a continuación:

Sea $M \in \mathbb{R}^{n \times n}$ una matriz definida positiva simétrica y sea $L \in \mathbb{R}^{n \times n}$ una matriz triangular inferior, si $M = LL^t$, entonces tenemos que:

$$l_{11} = \sqrt{m_{11}};$$

$$l_{j1} = \frac{m_{j1}}{l_{11}}, \text{ para } j = 2, \dots, n;$$

$$l_{ii} = \sqrt{m_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}, \text{ para } i = 2, \dots, n-1;$$

$$l_{ji} = \frac{1}{l_{ii}} \left[m_{ji} - \sum_{k=1}^{i-1} l_{jk}l_{ik} \right], \text{ para } i = 2, \dots, n-1 \text{ y } j = i+1, \dots, n; \text{ y}$$

$$l_{nn} = \sqrt{m_{nn} - \sum_{k=1}^{n-1} l_{nk}^2}$$

Por medio de este algoritmo, la matriz de coeficientes del sistema (IV.9), $Mu = v$, puede ser factorizada de forma tal que:

$$(LL^T)u = v, \quad (IV.13)$$

Para la solución de (IV.13) agrupamos las matrices y solucionamos sucesivamente los siguientes sistemas:

$$Ly = v \text{ y } L^T u = y, \quad (IV.14)$$

Los sistemas en (IV.14) involucran matrices triangulares inferiores y superiores, respectivamente, por lo que es más sencillo resolver estos sistemas que (IV.13).

IV.3.4 Factorización QR.

Esta técnica consiste en descomponer a una matriz A con vectores columna linealmente independientes en el producto de una matriz Q , con vectores columna ortonormales y una matriz R invertible, triangular superior. La aplicación de dicha factorización a un sistema de ecuaciones facilita su solución.

Considerando que la matriz M del sistema (IV.9) es cuadrada, la exposición del método para obtener la factorización QR de M , está enfocada para el caso en que M es cuadrada; aunque éste puede ser

utilizado en la factorización de sistemas con más ecuaciones que variables, sin problema alguno. Por ejemplo si $A \in \mathbb{R}^{m \times n}$ y $m > n$ entonces $A = (Q_1 Q_2)(R^T 0)^t = Q_1 R$, donde Q se particiona como $Q = [Q_1 Q_2]$ y Q_1 tiene el mismo número de columnas que A (consultar [DMB]).

El método más utilizado para encontrar la factorización de una matriz M , es conocido como *transformación de Householder*. En este método se debe construir una clase especial de matrices que son simultáneamente simétricas, elementales y ortogonales. Para cualquier vector u distinto de cero, la correspondiente *transformación de Householder* (o matriz de Householder) es una matriz de la forma:

$$H = I - \frac{2uu^t}{\|u\|_2^2} = I - \frac{uu^t}{\beta}, \quad \text{con } \beta = \frac{1}{2}\|u\|_2^2, \quad (\text{IV.15})$$

donde el vector u es llamado el *vector de Householder*.

Las matrices de Householder tienen dos propiedades importantes para construir una matriz ortogonal que reduce a M a la forma triangular. La primera, para cualesquiera dos vectores distintos de cero, de norma euclidiana igual, existe una matriz de Householder que transforma a un vector en el otro. Es decir, debemos buscar un vector u de Householder tal que

$$Hx = \left(I - \frac{uu^t}{\beta} \right) x = y, \quad (\text{IV.16})$$

donde x e y son vectores tales que $\|x\|_2 = \|y\|_2$ y $x \neq y$. De (IV.16), $Hx = y$ si el vector $u \neq 0$ satisface

$$\left(-\frac{u^t x}{\beta} \right) u = y - x,$$

de forma tal que u es un múltiplo de $y-x$ (donde $y-x$ es distinto de cero, dado que $x \neq y$).

En segundo lugar, cualquier vector transformado por H tiene una forma especial. Aplicando H al vector z obtenemos:

$$Hz = \left(I - \frac{uu^t}{\beta} \right) z = z - \left(\frac{u^t z}{\beta} \right) u,$$

de esta manera, H_z es la diferencia entre el vector original z y un múltiplo del vector u de Householder.

De acuerdo a lo explicado anteriormente, para cualquier matriz $M \in \mathbb{R}^{m \times m}$ no singular es posible construir una secuencia de $m-1$ matrices de Householder, tales que

$$H_{m-1} \cdots H_2 H_1 M = R, \quad (\text{IV.17})$$

donde R es una matriz triangular superior, no singular de $m \times m$.

En (IV.17) la matriz de Householder H_1 , es construída tal que, en la primera columna de M , las componentes 2 a m son transformadas a

ceró. Una vez efectuada esta operación, la matriz de Householder H_2 , es construída tal que, en la segunda columna de H_1M , las componentes 3 a m son transformadas a cero; y así sucesivamente hasta que la matriz de Householder H_{m-1} es construída tal que, en la columna $m-1$ de la matriz $H_{m-2} \cdots H_2H_1M$ el elemento m es transformado a cero. Para que de esta forma $H_{m-1} \cdots H_2H_1M$ sea una matriz triangular superior.

Sea Q^t una matriz ortogonal $m \times m$ tal que

$$Q^t = H_{m-1} \cdots H_2H_1 \text{ así que } Q = H_1 \cdots H_{m-1}.$$

Cualesquiera de las siguientes dos formas es llamada la factorización QR de M :

$$Q^tM = R \quad \text{ó} \quad M = QR.$$

Una vez que la factorización QR es conocida, la solución de (IV.9) puede ser obtenida observando que $Mu = QRu = v$. Lo que equivale a resolver el sistema triangular superior

$$Ru = Q^tv.$$

IV.3.5 Método de los Mínimos Cuadrados con Factorización QR.

Como vimos en la sección II.4.2.2, la dirección en la que la f.o. de un PPL disminuye (aumenta) más rápidamente, es obtenida a través del gradiente de dicha función. Además, tal dirección es factible si es solución

del conjunto homogéneo formado por las restricciones del PPL. Dado que el gradiente de la f.o. no siempre se encuentra en el subespacio nulo de la matriz de restricciones, en la sección II.4.2.3, se dijo que debía obtenerse una representación del vector dirección, que efectivamente sea solución al sistema homogéneo. Para lograr tal representación, el gradiente de la f.o. es premultiplicado por una matriz de proyección, la cual generará únicamente la porción del vector dirección que cae en el subespacio nulo, consiguiéndose así la dirección factible de mayor descenso (ascenso).

En el caso del método de escala afín primal, en la sección III.1.4, se indicó que debemos proyectar el gradiente $c^k = D_k c$ sobre el subespacio nulo de $A_k = AD_k$ para crear la dirección factible de mayor descenso d_w^k . De la sección II.4.2.3 sabemos que el complemento ortogonal de $\text{null}(A_k)$ es el rango(A_k^t) y que, si $d_w^k \in \text{null}(A_k)$ entonces $A_k d_w^k = 0$ ó si $c^k \in \text{rango}(A_k^t)$, entonces $A_k^t \lambda = c^k$, donde $\lambda \in \mathbb{R}^m$. Considerando que el sistema $A_k^t \lambda = c^k$ puede ser sobredeterminado (es decir, el número de ecuaciones es mayor que el número de variables en el sistema), entonces debemos encontrar un vector que minimice la distancia (o error) entre $A_k^t \lambda$ y c^k . Dicho vector puede obtenerse a través del siguiente problema de *mínimos cuadrados*:

$$\text{Min}_{\lambda} \|c^k - A_k^t \lambda\|^2 \quad (\text{IV.18})$$

Pero resulta que el vector $c^k - A_k^t \hat{\lambda}$ se encuentra en el complemento ortogonal del rango(A_k^t). Así si obtenemos un $\hat{\lambda}$ (residual óptimo) que solucione a (IV.18) entonces tendremos que

$$c^k - A_k^t \hat{\lambda} = d_w^k. \quad (\text{IV.19})$$

Considerando que $(A_k^t)^t(c^k - A_k^t \hat{\lambda}) = 0$ se puede verificar que $\hat{\lambda} = (A_k A_k^t)^{-1} A_k c^k$. Si ahora calculamos la factorización QR de la matriz A_k^t tendremos una forma efectiva de conseguir el residuo del problema (IV.18). De acuerdo a [DMB] y a [MyM], para cualquier matriz $\bar{M} \in \mathbb{R}^{m \times n}$, de rango completo y con $m \geq n$, es posible obtener una factorización QR, dada por las matrices ortonormales Q_1 y Q_2 (en donde $Q_2^t Q_1 = 0$) y por la matriz triangular superior no singular R, tal que $\bar{M} = (Q_1 Q_2)(R^t)^t = Q_1 R$. Así, dado que $\bar{M} = A_k^t$, entonces $A_k^t = Q_1 R$ y $\hat{\lambda}$ puede reescribirse como:

$$\begin{aligned}\hat{\lambda} &= (R^t Q_1^t Q_1 R)^{-1} R^t Q_1^t c^k \\ \hat{\lambda} &= R^{-1} Q_1^t c^k.\end{aligned}\tag{IV.20}$$

Sustituyendo (IV.20) en (IV.19) tenemos:

$$\begin{aligned}c_w^k &= c^k - Q_1 R (R^{-1} Q_1^t c^k) \\ c_w^k &= (I - Q_1 Q_1^t) c^k.\end{aligned}\tag{IV.21}$$

Obteniéndose así una nueva forma de calcular el vector dirección sobre $\text{null}(A_k)$.

Para el caso del PPL dual, consideremos el problema (III.26). Como se describió en el capítulo III, una transformación V_k^{-1} puede ser aplicada sobre (III.26) para obtener el siguiente PPL.

$$\begin{aligned}\text{Max } & b^t \lambda \\ \text{s. a } & \\ & V_k^{-1} A^t \lambda + u = V_k^{-1} c \\ & u \geq 0.\end{aligned}\tag{IV.22}$$

Ahora bien, dado que $V_k^{-1}A^t$ es de rango completo, la descomposición descrita anteriormente puede ser aplicada a esta matriz, es decir $V_k^{-1}A^t = Q_1 R$. De (IV.22), observamos que λ puede ser expresada en términos de u , de la siguiente manera:

$$\begin{aligned} (Q_1 R)\lambda &= V_k^{-1}c - u, \quad u \geq 0. \\ R\lambda &= Q_1^t V_k^{-1}c - Q_1^t u, \quad u \geq 0. \\ \lambda &= R^{-1}Q_1^t V_k^{-1}c - R^{-1}Q_1^t u, \quad u \geq 0. \end{aligned} \quad (IV.23)$$

Así, al sustituir (IV.23) en la f.o. de (IV.22), dicha función queda únicamente en términos de u . Para que la región factible de (IV.22) sea también expresada en términos de u , nos basaremos en el hecho de que $Q_2^t Q_1 = 0$, entonces en la región factible tenemos:

$$\begin{aligned} Q_1 R \lambda + u &= V_k^{-1}c, \quad u \geq 0. \\ Q_2^t Q_1 R \lambda + Q_2^t u &= Q_2^t V_k^{-1}c, \quad u \geq 0. \\ Q_2^t u &= Q_2^t V_k^{-1}c, \quad u \geq 0. \end{aligned} \quad (IV.24)$$

Con (IV.23) y (IV.24) es posible ahora obtener un PPL equivalente al PPL (IV.22):

$$\begin{aligned} \text{Max } b^t R^{-1} Q_1^t V_k^{-1}c - b^t R^{-1} Q_1^t u \\ \text{s. a} \\ Q_2^t u &= Q_2^t V_k^{-1}c \\ u &\geq 0. \end{aligned} \quad (IV.25)$$

Al igual que en el caso del método de escala afín primal, requerimos encontrar la proporción del gradiente de la f.o. que cae en el subespacio nulo de la matriz de restricciones. Como se explicó para dicho método, este vector puede obtenerse al resolver el siguiente problema de *mínimos cuadrados*:

$$\text{Min}_y \|Q_2 y - Q_1(R^{-1})^t b\|_2^2, \quad (\text{IV.26})$$

donde $-Q_1(R^{-1})^t b$ es el gradiente de la f.o. y $Q_2 y$ genera el complemento ortogonal de $\text{null}(Q_2^t)$. Ahora podemos encontrar y el óptimo de (IV.26) tal que

$$Q_2 \hat{y} - Q_1(R^{-1})^t b = d_0^*. \quad (\text{IV.27})$$

Una vez más, el residual óptimo puede ser obtenido al considerar que $(Q_2)^t [Q_2 \hat{y} - Q_1(R^{-1})^t b] = 0$, y al efectuar operaciones tenemos que $\hat{y} = 0$. Al sustituir este resultado en (IV.27), la dirección de las variables de holgura escaladas, es:

$$d_0^* = -Q_1(R^{-1})^t b. \quad (\text{IV.28})$$

Ahora bien, para el caso del método de escala afín dual, estamos interesados en encontrar una equivalencia de d_k^* a partir de (IV.28). En el capítulo III en la ecuación (III.37), vimos que la relación entre d_k^* y d_0^* está dada por:

$$d_k^* = -(A V_k^{-2} A^t)^{-1} (A V_k^{-1}) d_0^*$$

Nuevamente, si $V_k^{-1}A^t = Q_k R$ y d_k^t como en (IV.28), entonces

$$d_k^t = (R^t R)^{-1} b. \quad (\text{IV.29})$$

Dado que R es una matriz triangular superior no singular, entonces (IV.29) se puede reescribir como

$$(R^t R) d_k^t = b. \quad (\text{IV.30})$$

La solución del sistema (IV.30) puede ser obtenida al resolver sucesivamente los siguientes sistemas:

$$R^t z = b \text{ y } R d_k^t = z. \quad (\text{IV.31})$$

Como los sistemas en (IV.31) involucran matrices triangulares inferiores y superiores respectivamente, es más sencillo obtener las soluciones en (IV.31) que la solución en (IV.30). Una vez obtenidas d_k^t , d_k^x y x^k se obtienen como se indica en el algoritmo expuesto en la sección III.2.5.

En el siguiente capítulo se da una descripción de la instrumentación computacional de los métodos de escala afín primal y dual, en donde se incluye la utilización de las técnicas computacionales introducidas en este capítulo.

CAPÍTULO V.

INSTRUMENTACIÓN COMPUTACIONAL DE LOS MÉTODOS DE ESCALA AFÍN Y EVALUACIÓN DE RESULTADOS

En este capítulo se describen las características de los códigos correspondientes a los métodos de escala afín primal y dual que han sido instrumentados. Para evaluar dichas instrumentaciones se seleccionaron un conjunto de problemas prueba los cuales también son descritos. Para concluir este capítulo se reportan los resultados obtenidos de las evaluaciones con ambos códigos y se comparan con los resultados de experimentos computacionales referidos en la literatura.

V.1 DESCRIPCIÓN DE LOS PROGRAMAS INSTRUMENTADOS.

En esta sección se proporcionan las características del software y del hardware considerados y de cada uno de los programas instrumentados.

V.1.1 Características del Software y del Hardware.

Como hemos visto en los capítulos anteriores, en especial en el capítulo IV, las operaciones matriciales son fundamentales para ambos algoritmos, por lo que se considera importante que tales operaciones se

efectúen en forma óptima. Actualmente es posible trabajar con software para cómputo científico numérico que permite realizar operaciones de álgebra lineal numérica de manera eficiente, tal como el IMSL (consultar [IMSL]). El IMSL es un conjunto de subrutinas escritas en Fortran, especialmente diseñadas para ser utilizadas en tres principales áreas: Matemáticas Aplicadas en general (Math/Library), Estadística (Stat/Library) y Funciones Especiales (Sfun/Library).

Para nuestro caso, se seleccionó a la biblioteca IMSL Math/Library dada la disponibilidad que teníamos de ésta y de sus características, como la inclusión de las Subrutinas de Álgebra Lineal Básicas (BLAS por su abreviación en inglés), que representan el estado del arte en álgebra lineal numérica. Con el conjunto de subrutinas BLAS es posible resolver los sistemas lineales encontrados en los vectores dirección, a través de los métodos propuestos en el capítulo IV. Además, la biblioteca seleccionada permite efectuar operaciones matriciales básicas, con lo que esperamos que las instrumentaciones realizadas sean más exactas y confiables.

Como hemos mencionado, las subrutinas de IMSL se encuentran escritas en Fortran, por lo que decidimos realizar las codificaciones de los métodos de escala afín en dicho lenguaje. La versión de IMSL con la que contamos para nuestra instrumentación computacional es la 1.1 (1989), pero sólo tuvimos disponibilidad de los códigos objeto. Debido a esta limitación fue necesario trabajar con el compilador Microsoft Fortran versión 5.0 de 1989. Este compilador permite la programación estructurada del Fortran 77, por lo que los programas pudieron ser escritos en forma modular.

Por otro lado, las instrumentaciones computacionales y las experimentaciones fueron efectuadas en una computadora personal 486 SX a 25 mhz., con 6 mb de memoria RAM y un disco duro de 129 mb. Se utilizó además un sistema operativo MSDOS versión 6.0 y un procesador DX 33 mhz. como coprocesador matemático.

Procederemos ahora a la descripción de cada uno de los programas instrumentados.

V.1.2 Descripción del Programa Correspondiente al Método de Escala Afín Primal.

La instrumentación computacional del método de escala afín primal recibe como entrada a un PPL como el de la definición I.4, pero cuya f.o. pueda estar también en maximización. El PPL proporcionado es puesto en la forma estándar como en (III.1), y posteriormente transformado a un PPL de la M-grande (como se indica en la sección IV.1.1). El algoritmo de escala afín primal es aplicado sobre este último PPL y se detiene cuando se cumplen cualesquiera de los métodos de paro descritos en las secciones IV.2.1, IV.2.3 y IV.2.4, indicando si se ha alcanzado solución óptima, no acotación o infactibilidad como se indica en la sección IV.2.5.

A continuación se listan las principales subrutinas que conforman a esta instrumentación indicándose también que subrutinas de IMSL fueron utilizadas.

1. **LECTAM.** Procedimiento para leer el número de variables y de restricciones del PPL a capturar.

2. **CARACTER.** Procedimiento para leer las características del PPL a capturar (si es de maximización o de minimización o si las restricciones son \leq , \geq ó $=$).

3. **LECTURA.** Procedimiento para capturar los datos del PPL a resolver: coeficientes de la f.o., matriz de restricciones y vector del lado derecho. La lectura de la matriz de restricciones puede hacerse capturando todos los elementos o sólo aquellos distintos de cero.

4. **IMPRESO1.** Procedimiento que imprime en pantalla el PPL capturado y el PPL de la M-grande a resolver por el algoritmo de escala afin primal.

5. **CORREGIR.** Procedimiento para corregir (en caso de requerirse) los coeficientes del PPL proporcionado.

6. **FORMEST.** Procedimiento para transformar el PPL proporcionado a la forma estándar, es decir, a un PPL de la forma (III.1).

7. **PARAMETER.** Procedimiento para leer los parámetros con los que se ejecutará el programa ó tomarlos por default. Como se vio en el capítulo IV, para que la instrumentación pueda ser efectuada correctamente, se requiere de los siguientes parámetros: el valor de la M-grande, cuyo valor por default es 10^4 ; el factor de tamaño de paso, cuyo valor por default es 0.9999; la cota sobre la f.o., cuyo valor por default es 10^6 ; la cota para

factibilidad primal, cuyo valor por default es 10^{-5} ; la cota para factibilidad dual, cuyo valor por default es 10^{-5} ; la cota para holguras complementarias, cuyo valor por default es 10^{-5} ; la cota para la variable artificial, cuyo valor por default es 10^{-9} ; y la cota sobre el vector dirección, cuyo valor por default es 10^{-5} .

8. OSIFI. Procedimiento que aplica el método de la M-grande al PPL en la forma estándar para obtener una solución interior factible inicial. Además se obtiene la transpuesta de la matriz de restricciones. Las subrutinas de IMSL que se utilizan en este procedimiento son:

- 8.1 *SSET*: Inicializa el vector e con la constante 1.
- 8.2 *MURRV*: Multiplica la matriz A por el vector e .
- 8.3 *TRNRR*: Transpone la matriz A .

9. SELMET. Procedimiento para seleccionar el método de factorización matricial con el que se resolverá el sistema de ecuaciones implícito en la dirección de movimiento. Como vimos en el capítulo anterior, se tienen 4 posibles alternativas: factorización UDU^t , factorización de Cholesky, factorización QR y Mínimos Cuadrados. La subrutina de IMSL que se utiliza en este procedimiento es :

- 9.1 *SSET*: Inicializa el vector x con la constante 1.

10. IMPRESO2. En caso de requerirse el punto interior obtenido en cada iteración, es impreso en pantalla (no se despliega tiempo de ejecución).

11. CVED. Procedimiento para calcular el vector estimado dual. Esto de acuerdo al paso 2 del algoritmo B en el capítulo III. Las subrutinas del IMSL que se utilizan en este procedimiento son:

11.1 *MRRRR*: Multiplica la matriz AD^2 por la matriz A^t .

11.2 *MURRV*: Multiplica la matriz AD^2 por el vector c .

y de acuerdo a *SELMET*, alguna de las siguientes:

11.3 *LSLSF*: Resuelve el sistema $(AD^2A^t)\lambda=AD^2c$ por factorización UDU^t .

11.4 *LSADS*: Resuelve el sistema $(AD^2A^t)\lambda=AD^2c$ por factorización de Cholesky.

11.5 *LSQRR*: Resuelve el sistema $(AD^2A^t)\lambda=AD^2c$ por factorización QR.

12. CVCR. Procedimiento para el cálculo del vector de costos reducidos de acuerdo al paso 3 del algoritmo B en el capítulo III. La subrutina del IMSL que se utiliza en este procedimiento es:

12.1 *MURRV*: Multiplica la matriz A^t por el vector λ .

13. VEDVCRDW. Procedimiento para calcular por el método de los mínimos cuadrados el vector estimado dual, el vector de costos reducidos y la dirección de movimiento de acuerdo a los pasos 2 y 3 del algoritmo B en el capítulo III. Las subrutinas del IMSL que se utilizan en este procedimiento son:

13.1 *SHPROD*: Efectúa el producto elemento a elemento de los vectores x y c para obtener otro vector.

13.2 *LSQRR*: Resuelve el problema de mínimos cuadrados lineales obtenido en (IV.18).

14. PARO. Procedimiento que verifica la factibilidad primal, la factibilidad dual y las condiciones de holguras complementarias para decidir si el método debe detenerse, de acuerdo a lo indicado en la sección IV.2.1 del capítulo anterior y en el paso 4 del algoritmo B. Las subrutinas y funciones de IMSL que se utilizan en este procedimiento son:

- 14.1 *ISMIN*: Obtiene el índice del elemento mínimo en x y en $v=c-A^Tb$.
- 14.2 *MURRV*: Multiplica la matriz A por el vector x .
- 14.3 *SAXPY*: Efectúa la operación $Ax-b$.
- 14.4 *SNRM2*: Obtiene la norma euclidiana de un vector. En este caso de $Ax-b$, b , \hat{v} y \hat{c} .

15. DMNAOC. Procedimiento para obtener la dirección de movimiento y efectuar el chequeo de no acotación y de valor objetivo constante, tal y como se especifica en los pasos 5 y 6 del algoritmo B, junto con las condiciones obtenidas en las secciones IV.2.3 y IV.2.4. Las subrutinas y funciones de IMSL que se utilizan en este procedimiento son:

- 15.1 *SHPROD*: Efectúa el producto elemento a elemento de los vectores x y v para obtener el vector dw .
- 15.2 *SNRM2*: Obtiene la norma euclidiana del vector dw .
- 15.3 *ISMIN*: Obtiene el índice del elemento mínimo en el vector dw .
- 15.4 *ISMAX*: Obtiene el índice del elemento máximo en el vector dw .

16. LPOSM. Procedimiento para calcular la longitud de paso y trasladarse a la nueva solución mejorada como es indicado en los pasos 7 y 8 del algoritmo B y en la sección IV.2.4. Las subrutinas y funciones de IMSL que se utilizan en este procedimiento son:

- 16.1 *ISMIN*: Obtiene el índice del elemento mínimo en el vector $tem = r/dw_i$ donde $dw_i > \epsilon$.
- 16.2 *SHPROD*: Efectúa el producto elemento a elemento de los vectores x y dw para obtener otro vector.

16.3 SAXPY: Efectúa la operación $x - \alpha D w$.

17. IMPRESO3. Procedimiento para desplegar en pantalla los resultados obtenidos para el PPL proporcionado. En caso de que se detenga el algoritmo y la variable artificial sea distinta de cero, entonces como se indica en IV.1.1 se despliega que el PPL proporcionado es infactible. En caso de que la condición de no acotación en el paso 6 del algoritmo B ó la descrita en la sección IV.2.3 se satisfaga, se despliega que la f.o. del PPL proporcionado es no acotada. En caso de que no se cumpla cualesquiera de las condiciones anteriores se despliega la solución óptima alcanzada, el número de iteraciones, el tiempo de CPU que se utilizó para resolver el PPL (en caso de proceder), el vector de costos reducidos asociado a la solución óptima y el vector estimado dual.

Finalmente, se utiliza la función *SECNDS* para obtener el tiempo utilizado en CPU del método de escala afín primal para la solución del PPL proporcionado. El tiempo es medido al obtener la forma estándar, al obtener la solución interior inicial y al aplicar el algoritmo al PPL a resolver.

V.1.3 Descripción del Programa Correspondiente al Método de Escala Afín Dual.

De la misma forma que en el método de escala afín primal en la instrumentación computacional del método de escala afín dual, se recibe como entrada a un PPL como el de la definición 1.4, donde la f.o. puede estar también en maximización. El PPL proporcionado es llevado a la forma estándar. Posteriormente, de dicho problema, se obtiene el dual y se

transforma en un PPL de la M-grande como en (IV.2). El algoritmo de la sección III.2.5 se aplica a este PPL hasta que la variable artificial se aproxima o cruza cero como se indica en la sección IV.2.5 (en caso contrario, se detiene el método indicando que el PPL proporcionado es no acotado). La variable artificial es eliminada del PPL y la solución interior factible inicial es obtenida como se explica en la sección IV.1.2. El algoritmo de escala afín dual es nuevamente aplicado hasta que alguna de las condiciones de paro, descritas en las secciones IV.2.2, IV.2.3 y IV.2.4, se satisfaga, indicando si se ha alcanzado solución óptima, no acotación o infactibilidad como se indica en la sección IV.2.5.

A continuación se listan las subrutinas principales que conforman a esta instrumentación, especificando las subrutinas de IMSL que han sido utilizadas.

1. **LECTAM.** Procedimiento para leer el número de variables y de restricciones del PPL a capturar.

2. **CARACTER.** Procedimiento para leer las características del PPL a capturar (si es de maximización o de minimización o si las restricciones son \leq , \geq ó $=$).

3. **LECTURA.** Procedimiento para capturar los datos del PPL a resolver: coeficientes de la f.o., matriz de restricciones y vector del lado derecho. La lectura de la matriz de restricciones puede hacerse capturando todos los elementos o sólo aquellos distintos de cero.

4. **IMPRESO1.** Procedimiento que imprime en pantalla el PPL capturado a resolver por el algoritmo de escala afín dual.

5. **CORREGIR.** Procedimiento para corregir (en caso de requerirse) los coeficientes del PPL proporcionado.

6. **FORMEST.** Procedimiento para llevar el PPL proporcionado a la forma estándar.

7. **PARAMETER.** Procedimiento para leer los parámetros con los que se ejecutará el programa ó tomarlos por default. Como se vio en el capítulo IV, para que la instrumentación pueda ser efectuada correctamente, se requiere de los siguientes parámetros: el valor de la M-grande, cuyo valor por default es 10^4 ; el factor de tamaño de paso, cuyo valor por default es 0.9999; la cota sobre la f.o., cuyo valor por default es 10^6 ; la cota para factibilidad primal, cuyo valor por default es 10^{-5} ; la cota para factibilidad dual, cuyo valor por default es 10^{-5} ; la cota para holguras complementarias, cuyo valor por default es 10^{-5} ; la cota sobre el vector dirección, cuyo valor por default es 10^{-5} ; y, el parámetro para calcular la variable artificial, cuyo valor por default es 2.

8. **OSIFID.** Procedimiento que obtiene del PPL en la forma estándar su dual, y que aplica el método de la M-grande para obtener una solución interior factible inicial. Las subrutinas y funciones de IMSL que se utilizan en este procedimiento son:

8.1 **TRNRR:** Transpone la matriz A.

8.2 **SSET:** Inicializa el vector y con la constante 0.

- 8.3 *ISMIN*: Obtiene el Índice del elemento mínimo en c .
- 8.4 *ISAMAX*: Obtiene el Índice del elemento máximo en c .
- 8.5 *SCOPY*: Copia el vector c al vector v .

9. **IMPRESO4**. Procedimiento que imprime en pantalla el PPL de la M-grande dual a resolver.

10. **SELMET**. Procedimiento para seleccionar el método de factorización matricial con el que se resolverán los sistemas de ecuaciones implícitos en las direcciones de movimiento. Como vimos en la sección IV.3, se tienen 4 posibles alternativas: factorización UDU^t , factorización de Cholesky, factorización QR y Mínimos Cuadrados.

11. **IMPRESO2**. En caso de requerirse el punto interior dual obtenido en cada iteración, es impreso en pantalla (no se despliega tiempo de ejecución).

12. **DIRSMOVS**. Procedimiento para obtener la dirección de movimiento de las variables irrestrictas y de las variables restringidas. Esto de acuerdo al paso 2 del algoritmo de escala afín dual. Las subrutinas de IMSL que se utilizan en este procedimiento son:

- 12.1 *MXTXF*: Multiplica la matriz AV^{-1} por VA^t .
- 12.2 *MURRV*: Multiplica la matriz A por el vector dy .
- 12.3 *SSCAL*: Multiplica por -1 al vector $A^t dy$.

y de acuerdo a *SELMET*, alguna de las siguientes:

- 12.4 *LSLSF*: Resuelve el sistema $(AV^{-2}A^t)dy = b$ por factorización UDU^t .

12.5 *LSADS*: Resuelve el sistema $(AV^{-2}A^T)dy = b$ por factorización de Cholesky.

12.6 *LSQRR*: Resuelve el sistema $(AV^{-2}A^T)dy = b$ por factorización QR.

13. **MINCUAD**. Procedimiento que obtiene la dirección de movimiento de las variables irrestrictas y de las variables restringidas utilizando mínimos cuadrados, de acuerdo al paso 2 del algoritmo dual y a (IV.31). Las subrutinas del IMSL que se utilizan en este procedimiento son:

13.1 *ISSET*: Inicializa IPVT con la constante 0.

13.2 *LQRRR*: Calcula la descomposición QR.

13.3 *LSLRT*: Resuelve un sistema triangular inferior real de ecuaciones lineales obtenido en (IV.31).

13.4 *LSLRT*: Resuelve un sistema triangular superior real de ecuaciones lineales obtenido en (IV.31)

13.5 *MURRV*: Multiplica la matriz A^T por el vector dy .

13.6 *SSCAL*: Efectúa la siguiente operación $dv = -dv$.

14. **CVEP**. Procedimiento para calcular el vector estimado primal de acuerdo al paso 4 del algoritmo de escala afín dual.

15. **PARO**. Procedimiento que verifica la factibilidad primal, la factibilidad dual y la condición de holguras complementarias para decidir si el método debe detenerse, de acuerdo a lo indicado en la sección IV.2.2 del capítulo anterior y del paso 5 del algoritmo dual. Las subrutinas y funciones del IMSL que se utilizan en este procedimiento son:

15.1 *MURRV*: Multiplica la matriz A^T por el vector y .

15.2 *SAXPY*: Efectúa las operaciones $c \cdot (A^T y - q)_{i_3} + v$ y $Ax - b$.

15.3 *SNRM2*: Obtiene la norma euclidiana de los vectores $Ax - b$, $c \cdot (A^T y - q)_{i_3} + v$, b y c .

15.4 *SDOT*: Calcula el producto punto de c y x .

16. NAFOC. Procedimiento para efectuar el chequeo de no acotación y de valor objetivo constante en el PPL dual, como se especificó en el paso 3 del algoritmo dual, junto con las consideraciones hechas en las secciones IV.2.3 y IV.2.4. Las subrutinas y funciones de IMSL que se utilizan en este procedimiento son:

- 16.1 *ISMIN*: Obtiene el índice del elemento mínimo en dv .
- 16.2 *ISMAX*: Obtiene el índice del elemento máximo en dv .
- 16.3 *SNRM2*: Obtiene la norma euclídiana del vector dv .
- 16.4 *SDOT*: Calcula el producto punto de b e y .

17. LPOSM. Procedimiento para calcular la longitud de paso y trasladarse a la nueva solución mejorada como se indicó en los pasos 6 y 7 del algoritmo dual y en la sección IV.2.4. La subrutina y la función de IMSL que se utilizan en este procedimiento son:

- 17.1 *ISMIN*: Obtiene el índice del elemento mínimo en el vector $tem = rv / (-dv_i)$ donde $dv_i < 0$.
- 17.2 *SAXPY*: Efectúa las operaciones $y + \beta dy$ y $v + \beta dv$.

18. CHECVART. Procedimiento que verifica si la variable artificial se aproxima o cruza cero, como se indicó en la sección IV.2.5. Una vez que se detecta que la variable artificial cumple con esta condición, se hace $N=N-1$ y la variable artificial se iguala a cero, como bandera; además se obtiene la solución interior inicial como se indica en la sección IV.1.2. La subrutina de IMSL que se utiliza en este procedimiento es:

- 18.1 *SAXPY*. Efectúa la operación $A^t y + v$.

19. **IMPRESO3.** Procedimiento para desplegar en pantalla los resultados obtenidos para el PPL proporcionado. En caso de que se detenga el algoritmo y la variable artificial sea distinta de cero, entonces como se indica en IV.1.2 se despliega que la f.o. del PPL proporcionado es no acotada. En caso de que la condición de no acotación en el paso 3 del algoritmo dual ó la descrita en la sección IV.2.3 se satisfaga, se despliega que el PPL proporcionado es infactible. En caso de que no se cumpla cualesquiera de las condiciones anteriores se despliega la solución óptima dual alcanzada, el número de iteraciones, el tiempo de CPU que se utilizó para resolver el PPL, (en caso de proceder) el vector de estimado primal y el valor de la función objetivo primal.

Por último, también se usó la función *SECNDS* para obtener el tiempo utilizado en CPU del método de escala afín dual para la solución del PPL proporcionado. El tiempo es medido al obtener la forma estándar, al obtener el PPL dual y la solución interior inicial y, al aplicar el algoritmo al PPL dual a resolver.

Los listados correspondientes a las instrumentaciones aquí descritas se encuentran en el apéndice D. Las instrucciones que deben tenerse en cuenta para poder ejecutar cada programa se encuentran en el apéndice C. En la siguiente sección describiremos los PPL prueba.

V.2 DESCRIPCIÓN DE LOS PROBLEMAS PRUEBA A EVALUAR.

En esta sección presentaremos el conjunto de problemas prueba, seleccionados para validar los resultados de las implementaciones computacionales de los métodos de escala afín primal y dual. Dicho conjunto está constituido por un grupo de problemas propuestos por Colmenares [Col] y el PPL conocido como problema de Hilbert, lo cual nos permitirá comparar los resultados de los experimentos computacionales con los referidos en [ABR], [HBA] y [Col]. Se incluyen también otros problemas adicionales que nos permitirán efectuar comparaciones entre ambos métodos.

Cabe hacer mención que los resultados reportados en la literatura fueron obtenidos con el método de escala proyectiva de Karmarkar (capítulo II) en una PC de menor capacidad que la utilizada por nosotros (a excepción de los reportados en [Col] donde se utilizó una Amdahl System 470 Modelo V77-B ((4)) y una IBM 3033 ((6)).

Cada uno de los problemas prueba seleccionados fueron solucionados también utilizando el método simplex. El único paquete que tuvimos disponible, para efectuar las pruebas, fue el Micromanager Software, el cual no proporciona tiempo de obtención de la solución de los problemas de programación lineal.

V.2.1 Problemas Prueba, Grupo 1.

El primer grupo de problemas prueba es:

PROBLEMA 1.

$$\begin{aligned} & \text{Max } 30x_1 + 20x_2 \\ & \text{s. a} \\ & \quad x_1 + x_2 \geq 1 \\ & \quad -x_1 + x_2 \leq 1 \\ & \quad 3x_1 + 2x_2 \leq 6 \\ & \quad x_1 - 2x_2 \leq 1 \\ & \quad x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Este PPL tiene soluciones óptimas múltiples, un punto extremo degenerado en la forma estándar y la matriz AA^t está mal condicionada. La solución de este problema por el método simplex ocupó 3 iteraciones y la solución óptima alcanzada fue $x^* = (1.75, 0.38, 1.13, 2.38, 0, 0)^t$ con $z^* = 60$. El otro punto extremos óptimo es $x^* = (0.80, 1.80, 1.60, 0, 2, 0)^t$.

PROBLEMA 2.

$$\begin{aligned} & \text{Min } -x_1 + 7x_2 + x_3 + 2x_4 \\ & \text{s. a} \\ & x_1 + x_2 + x_3 + x_4 \leq 1 \\ & 0.5x_1 - 5.5x_2 - 2.5x_3 + 9x_4 \leq 0 \\ & 0.5x_1 - 1.5x_2 - 0.5x_3 + x_4 \leq 0 \\ & x_i \geq 0 \text{ para } i = 1, \dots, 4. \end{aligned}$$

Este PPL presenta degeneración, mal condicionamiento en la matriz de coeficientes y soluciones óptimas múltiples. La solución óptima degenerada alcanzada por el método simplex en 8 iteraciones es $x^* = (0, 0, 0, 0, 1, 0, 0)^t$ y la solución alternativa alcanzada es $x^* = (0.5, 0, 0.5, 0, 0, 1, 0)^t$ ambas con $z^* = 0$.

PROBLEMA 3.

$$\begin{aligned} & \text{Min } 2x_1 - x_2 \\ & \text{s. a} \\ & x_1 + x_2 \geq -10 \\ & -10x_1 + x_2 \leq 10 \\ & -4x_1 + x_2 \leq 20 \\ & x_1 + 4x_2 \geq -2 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Para este problema la función objetivo es no acotada. Esta condición fue detectada también por el método simplex.

PROBLEMA 4.

$$\begin{aligned} & \text{Min } 10x_1 + 3x_2 \\ \text{s. a } & x_1 + x_2 \geq 20 \\ & x_1 \leq 6 \\ & x_1 \geq 2 \\ & x_2 \leq -12 \\ & x_2 \geq 1 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Como podemos observar las restricciones 4 y 5 de este problema son inconsistentes, por lo que el poliedro asociado es vacío y el PPL es infactible. Sin embargo en este caso, el método simplex utilizado no detecta la infactibilidad y proporciona una solución errónea en una de las opciones de ejecución y, en la otra opción, el método diverge.

PROBLEMA 5.

$$\begin{aligned} & \text{Min } -0.4x_1 - 0.4x_2 + 1.8x_3 \\ \text{s. a } & 1.5x_1 - 6.4x_2 + 4.8x_3 \leq 0 \\ & 0.2x_1 - 1.8x_2 + 0.6x_3 \leq 0 \\ & 0.4x_1 - 1.6x_2 + 0.2x_3 \leq 0 \\ & x_2 \leq 1 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

En este problema la matriz AA^t está mal condicionada, es decir, la matriz es casi singular. La solución óptima alcanzada en 4 iteraciones por el método simplex es $x^* = (4, 1, 0, 0, 4, 1, 0, 0)^t$ con $z^* = -2$.

PROBLEMA 6. (Problema de Hilbert).

$$\begin{aligned} \text{Max } z &= c^t x \\ \text{s. a.} \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$

donde

$$c_i = \frac{2}{i+1} + \sum_{j=2}^n \frac{1}{i+j}, \quad 1 \leq i \leq n$$

$$b_i = \sum_{j=1}^n \frac{1}{i+j}$$

$$A = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{(n+1)} \\ \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{(n+2)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{(n+1)} & \frac{1}{(n+2)} & \cdots & \frac{1}{2n} \end{bmatrix}$$

Una matriz como la definida en A es conocida como matriz de Hilbert. Este problema tiene como solución óptima $x^* = (1, 1, \dots, 1)^t$. En nuestro caso se evaluó este problema con $N = 3$ y $N = 5$. Para $N = 3$ el método simplex alcanzó la solución óptima $x^* = (1, 1, 1)^t$ con $z^* = 3.57$ en tres iteraciones. Para $N = 5$, en 5 iteraciones, el método simplex alcanza la solución óptima $x^* = (1, 1, 1, 1, 1)^t$ con $z^* = 6.27$.

V.2.2 Problemas Prueba, Grupo 2.

Estos problemas fueron obtenidos de textos de Programación Lineal con el fin de probar los métodos de escala afín en problemas clásicos.

PROBLEMA 7 (Propuesto por E.M.L Beale en 1955 [Fan]).

$$\begin{aligned} \text{Min } & -0.75x_4 + 20x_5 - 0.5x_6 + 6x_7 \\ \text{s. a } & \\ & x_1 + 0.25x_4 - 8x_5 - x_6 + 9x_7 = 0 \\ & x_2 + 0.5x_4 - 12x_5 - 0.5x_6 + 3x_7 = 0 \\ & x_3 + x_6 = 1 \\ & x_i \geq 0 \text{ para } i = 1, \dots, 7. \end{aligned}$$

Este PPL tiene soluciones degeneradas, por lo que el método simplex puede ser atrapado en un ciclo. Al intentar solucionar este problema utilizando el paquete Micromanager software, en cualesquiera de sus dos opciones, pudimos verificar que el método simplex disponible efectuaba una gran cantidad de operaciones sin poder alcanzar la solución óptima, la cual está dada por $x^* = (0.75, 0, 0, 1, 0, 1, 0)^t$ con $z^* = -1.75$.

PROBLEMA 8. (Problema de Mezcla de Fluidos [DyB]).

$$\text{Min } 7.2x_1 + 7.2x_2 + 4.35x_3 + 4.35x_4 + 3.8x_5 + 3.8x_6 + 4.3x_7 + 4.3x_8$$

s. a

$$x_1 + x_2 \leq 700$$

$$x_3 + x_4 \leq 600$$

$$x_5 + x_6 \leq 900$$

$$x_7 + x_8 \leq 500$$

$$x_1 + x_3 + x_5 + x_7 = 1300$$

$$x_2 + x_4 + x_6 + x_8 = 800$$

$$-2x_1 - 0.5x_3 - 3x_5 + 11x_7 \leq 0$$

$$-2x_2 - 0.5x_4 - 3x_6 + 11x_8 \leq 0$$

$$8x_1 - 3x_3 - 10x_5 + 10x_7 \geq 0$$

$$8x_2 - 6x_4 - 13x_6 + 8x_8 \geq 0$$

$$x_i \geq 0, \text{ para } i = 1, \dots, 8$$

Como podemos observar, este problema tiene más variables y restricciones que cualesquiera de los anteriores, además de que se tienen más restricciones que variables. Al solucionar este problema por el método simplex, la solución óptima alcanzada en 8 iteraciones fue: $x^* = (324.14, 348.72, 600, 0, 227.59, 304.76, 148.28, 146.52)^t$ con $z^* = 10745.11$.

PROBLEMA 9. (Problema de Planeación de la Producción [DyB]).

$$\begin{aligned} \text{Min } & 0.8x_1 + x_2 + 2.05x_3 + 2.2x_4 + 3.05x_5 + 3.2x_6 + 0.8x_7 + \\ & x_8 + 2.05x_9 + 2.2x_{10} + 0.8x_{11} + x_{12} + 1.6x_{13} + 2x_{14} + \\ & 2.85x_{15} + 3.2x_{16} + 3.85x_{17} + 4.2x_{18} + 1.6x_{19} + 2x_{20} + \\ & 2.85x_{21} + 3.2x_{22} + 1.6x_{23} + 2x_{24} \end{aligned}$$

s. a

$$0.2x_1 + 0.25x_2 + 0.2x_3 + 0.25x_4 + 0.2x_5 + 0.25x_6 \leq 100$$

$$0.2x_7 + 0.25x_8 + 0.2x_9 + 0.25x_{10} \leq 100$$

$$0.2x_{11} + 0.25x_{12} \leq 100$$

$$0.2x_{13} + 0.25x_{14} + 0.2x_{15} + 0.25x_{16} + 0.2x_{17} + 0.25x_{18} \leq 40$$

$$0.2x_{19} + 0.25x_{20} + 0.2x_{21} + 0.25x_{22} \leq 40$$

$$0.2x_{23} + 0.25x_{24} \leq 40$$

$$x_1 + x_{13} \geq 225$$

$$x_2 + x_{14} \geq 200$$

$$x_3 + x_7 + x_{15} + x_{19} \geq 375$$

$$x_4 + x_8 + x_{16} + x_{20} \geq 260$$

$$x_5 + x_9 + x_{11} + x_{17} + x_{21} + x_{23} \geq 450$$

$$x_6 + x_{10} + x_{12} + x_{18} + x_{22} + x_{24} \geq 300$$

$$x_i \geq 0, \text{ para } i = 1, \dots, 24$$

Este es un problema típico de programación lineal, el cual, como podemos ver, tiene el doble de variables que de restricciones, además de ser un problema esparsos pues tiene varios coeficientes con valor cero. La solución óptima alcanzada por el método simplex en 14 iteraciones es : $x^* = (225, 120, 0, 0, 0, 100, 375, 100, 0, 0, 450, 40, 0, 80, 0, 0, 0, 0, 0, 160, 0, 0, 0, 160)^t$ con $z^* = 2220$.

V.3 EVALUACIÓN Y COMPARACIÓN DE RESULTADOS.

En esta sección se proporcionan los resultados obtenidos de las instrumentaciones computacionales de los métodos de escala afín primal y dual, permitiéndonos de esta forma realizar comparaciones con otras versiones del método reportados en la literatura. En cada tabla se muestra la solución óptima (en caso de encontrarse), el número de iteraciones y el tiempo de CPU utilizado (en segundos). En caso de que se indique lo contrario se utilizan los parámetros por default.

PROBLEMAS PRUEBA, GRUPO 1.

PROBLEMA 1.

Resuelto por el Método de Escala Afín Primal.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	59.99920	59.99875	59.99916	59.99889
ITER.	3	3	3	3
SEG.	(0.051)	(0.113)	(0.051)	(0.063)

Resuelto por el Método de Escala Afín Dual.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*		59.99974	60.00038	60.00196
ITER.	5	5	5
SEG.		(0.500)	(0.109)	(0.063)

Como podemos observar en las tablas correspondientes al problema 1, sólo el método de escala afín dual utilizando factorización UDU^t no alcanzó la solución óptima. La versión primal en cualesquiera de los casos utiliza 3 iteraciones y la dual 5 iteraciones. Ambos métodos alcanzaron la solución óptima en menos de un segundo. En [Col] se indica que la solución se alcanzó en 14 iteraciones y 1.826 segundos. En [ABR] la solución se alcanzó en 12 iteraciones y 5 segundos.

PROBLEMA 2.

Resuelto por el Método de Escala Afín Primal.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	0.00029	0.00029	0.00029	0.00029
ITER.	9	9	9	9
SEG.	(0.109)	(0.160)	(0.109)	(0.109)

Resuelto por el Método de Escala Afín Dual.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	0.00045	0.00456	0.00033	0.00356
ITER.	4	4	4	4
SEG.	(0.325)	(0.291)	(0.105)	(0.105)

Para este problema, en la versión primal se modificaron los parámetros $M = 10^5$, $\alpha = 0.993$ y en la versión dual se modificó el parámetro $M = 10^5$. Como podemos observar tanto la versión primal como la dual alcanzan la solución óptima para todos los casos en 9 y 4 iteraciones respectivamente y en menos de un segundo. En [Col] se indica que la solución se alcanzó en

16 iteraciones en 1.477 segundos. En [ABR] la solución se alcanza en 12 iteraciones y 5 segundos.

PROBLEMA 3.

Resuelto por el Método de Escala Afín Primal.

UDU ^t	CHOLESKY	QR	MIN. CUAD.
N. A.	N. A.	N. A.	N. A.

Resuelto por el Método de Escala Afín Dual.

UDU ^t	CHOLESKY	QR	MIN. CUAD.
N. A.	N. A.	N. A.	N. A.

Para este problema, en todos los casos, las versiones primal y dual detectan la no acotación. En [Col] y [ABR] se indica también la no acotación. En [HBA] se indica que se llegó a un valor erróneo con los métodos de factorización utilizados en ese artículo y sólo se detectó no acotación con mínimos cuadrados. En [HBA] reportan también no haber detectado no acotación con un método con barrera.

PROBLEMA 4.

Resuelto por el Método de Escala Afín Primal.

UDU ^t	CHOLESKY	QR	MIN. CUAD.
SOL. INF.	SOL. INF.	SOL. INF.	SOL. INF.

Resuelto por el Método de Escala Afín Dual.

UDU ^t	CHOLESKY	QR	MIN. CUAD.
*****	*****	SOL. INF.	SOL. INF.

Como podemos observar el método de escala afín primal en todos los casos detecta infactibilidad, en cambio en la versión dual sólo con factorización QR y mínimos cuadrados; con UDU^t y Cholesky se detecta mal condicionamiento y el IMSL detiene el programa. En [Col] se llega a un valor erróneo sin detectar infactibilidad. En [ABR] se detecta la infactibilidad y en [HBA], al igual que para el problema anterior, únicamente con mínimos cuadrados detecta que el problema es infactible.

PROBLEMA 5.

Resuelto por el Método de Escala Afín Primal.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	-1.99977	-1.99977	-1.99977	-1.99977
ITER.	5	5	5	5
SEG.	(0.113)	(0.109)	(0.109)	(0.113)

Resuelto por el Método de Escala Afín Dual.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	-2.01144		-2.00000	-2.00000
ITER.	6	*****	6	6
SEG.	(0.551)		(0.109)	(0.059)

Para este problema tanto el método de escala afín primal como el dual alcanzan la solución óptima en 5 y 6 iteraciones respectivamente en menos de un segundo. Sólo para la versión dual con factorización de Cholesky el programa se detiene por mal condicionamiento. En [Col] el método para con $z^* = -1.1327$ en 2 iteraciones en menos de un segundo. En [ABR] se utilizan 12 iteraciones y 7 segundos. En [HBA] en todas las factorizaciones que instrumentaron llegan a la solución óptima, siendo sus mejores resultados en 4 iteraciones y 3 segundos, para el método de barrera reportan 53 iteraciones y 5 segundos.

PROBLEMA 6.

Resuelto por el Método de Escala Afín Primal (para $N=3$).

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	3.56646	3.56645	3.56648	3.56646
ITER.	4	4	4	4
SEG.	(0.051)	(0.109)	(0.059)	(0.063)

Resuelto por el Método de Escala Afín Dual (para $N=3$).

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	3.43038	3.43038	3.43038	3.43038
ITER.	4	4	4	4
SEG.	(0.106)	(0.160)	(0.059)	(0.051)

Resuelto por el Método de Escala Afin Primal (para N=5).

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	6.2642	6.26422	6.26421	6.2642
ITER.	4	4	4	4
SEG.	(0.109)	(0.160)	(0.109)	(0.109)

Resuelto por el Método de Escala Afin Dual (para N=5).

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	6.39410	6.39409	6.39408	6.39409
ITER.	6	6	6	6
SEG.	(0.492)	(0.556)	(0.160)	(0.109)

En este problema para $N = 3$ y $N = 5$ ambos métodos llegan a la solución óptima en todos los casos. En la versión dual se modificaron los parámetros $\beta = 0.994$, $c_3 = 10^{-6}$ (es cota de error para h_c) y $\theta = 3$. Para $N = 3$, las 2 versiones utilizan 4 iteraciones y se llega a la solución en menos de un segundo. Para $N = 5$ la versión primal ocupa 4 iteraciones y la dual 6, las dos versiones obtienen la solución en menos de un segundo. En [ABR] para $N = 3$ se reportan 11 iteraciones y 2 segundos y para $N = 5$, 11 iteraciones y 4 segundos. En [HBA] con los 4 métodos de factorización que utilizaron y el método de barrera llegan a la solución óptima, obteniendo su mejor resultado con QR en 3 iteraciones y 3 segundos para $N = 3$; y 5 iteraciones y 4 segundos para $N = 5$. Con el método de barrera reporten 52 iteraciones y 19 segundos para $N = 3$ y, 54 iteraciones y 54 segundos para $N = 5$.

PROBLEMAS PRUEBA, GRUPO 2.

PROBLEMA 7.

Resuelto por el Método de Escala Afín Primal.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	-1.24985	-1.24986	-1.24985	-1.24986
ITER.	6	6	6	6
SEG.	(0.109)	(0.113)	(0.172)	(0.051)

Resuelto por el Método de Escala Afín Dual.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	-1.33332	-1.33332	-1.33332	-1.33332
ITER.	4	4	4	4
SEG.	(0.219)	(0.281)	(0.059)	(0.059)

Para este problema el método de escala afín primal alcanza la solución óptima en 6 iteraciones en menos de un segundo en cualesquiera de los casos. En cambio el método de escala afín dual no llega a la solución óptima y se queda en una aproximación (aún con cota de holguras complementarias en cero), por lo que podemos concluir que se ve afectado por la degeneración primal.

PROBLEMA 8

Resuelto por el Método de Escala Afín Primal.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*				10745
ITER.	6
SEG.				(0.434)

Resuelto por el Método de Escala Afín Dual.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	10787	11229.59		10745.45
ITER.	9	9	9
SEG.	(0.879)	(1.211)		(0.547)

Para el problema 8, el método de escala afín primal alcanza la solución en 6 iteraciones y menos de un segundo, sólo para el caso de mínimos cuadrados y con la modificación de los parámetros $M = 10^5$ y con cota sobre la variable artificial de 10^{-3} ; para los casos UDU^t y QR detecta no acotación y para Cholesky, IMSL detiene el programa por mal condicionamiento. En cambio en el método de escala afín dual, la solución óptima se alcanza en 9 iteraciones en menos de un segundo para UDU^t, Cholesky y mínimos cuadrados (con parámetros modificados $\beta = 0.994$, $\epsilon_3 = 10^{-4}$ y $\theta = 3$) aunque la estimación primal en UDU^t y Cholesky no es muy aceptable.

PROBLEMA 9.

Resuelto por el Método de Escala Afín Primal.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*				2220.02900
ITER.	24
SEG.				(3.30)

Resuelto por el Método de Escala Afín Dual.

	UDU ^t	CHOLESKY	QR	MIN. CUAD.
Z*	2219.91400	2220.00000	2220.01100	2220.08800
ITER.	7	7	7	7
SEG.	(0.930)	(1.211)	(0.871)	(0.820)

Para este problema la versión primal sólo alcanza la solución óptima con mínimos cuadrados utilizando 24 iteraciones y 3.3 segundos, los parámetros que se modificaron en este caso son $M = 10^9$, $\beta = 0.99$ y $\epsilon_3 = 10^{-5}$. El método de escala afín dual, como puede observarse tuvo un mejor comportamiento alcanzando en todos los casos la solución óptima en 7 iteraciones y sólo con Cholesky en más de un segundo, los parámetros que se modificaron son $\beta = 0.994$, $\epsilon_3 = 10^{-5}$ y $\theta = 3$.

En general, nuestros resultados con respecto a los problemas del Grupo 1 fueron buenos, pues como puede verificarse, para los problemas 1,2,5, y 6 utilizamos menos iteraciones con respecto a [Col] y [ABR]. Los resultados reportados por [HBA] son mejores en una iteración para el problema 5 y para el problema 6 en el caso $N = 3$, aunque para $N = 5$ nosotros mejoramos en una iteración. Con respecto a [ABR] y [HBA] dado

que instrumentaron el método de escala proyectiva de Karmarkar, consideramos válido sólo comparar en iteraciones y no en tiempo, considerando el tipo de computadora que utilizaron. Con respecto a [Col] en los problemas evaluados podemos decir que tanto en número de iteraciones y en tiempo nuestros resultados fueron superiores.

Para los problemas 3 y 4 (Grupo 1) el método de escala afín primal en todos los casos detecta la no acotación y la infactibilidad, el mismo resultado es reportado en [Col] y [ABR]. El método de escala afín dual detecta la no acotación para todos los casos, pero no la infactibilidad. En [HBA], en el cual se efectuó también una instrumentación con distintas alternativas de factorización matricial, sólo mínimos cuadrados detecta la no acotación y la infactibilidad. Podemos concluir que los métodos de escala afín no se ven muy afectados por el tipo de factorización en este problema.

Con respecto a los problemas del Grupo 2, el método de escala afín primal proporciona mejores resultados que el método de escala afín dual para el problema 7. Para el problema 8, la versión primal obtiene la solución sólo para el caso de mínimos cuadrados en un menor número de iteraciones que el método de escala afín dual, aunque este último, alcanza la solución con otros dos tipos de factorizaciones más. Para el problema 9, el método de escala afín dual muestra superioridad con respecto al método de escala afín primal.

Aunque nuestro propósito no es realizar comparaciones con el método simplex, siempre es interesante el efectuarlas aún en el número de iteraciones (dado lo explicado en la sección V.2). Podemos observar que los

resultados proporcionados por el método simplex, son mejores que los proporcionados por el método de escala afín primal en los problemas 1,2,5, 6 (con $N=3$) y 9. Con respecto al método de escala afín dual, el simplex es superior en los problemas 1,5,6 y 8. Debe notarse que esta superioridad es por una o dos iteraciones. En cambio el método de escala afín primal es superior al método simplex utilizado, en los problemas 4,6 (con $N=5$), 7 y 8. El método de escala afín dual es superior al simplex en los problemas 2,4 y 9. Un aspecto importante a considerar es que con el método de escala afín primal se pudieron resolver problemas que el simplex que utilizamos no pudo resolver, además para los problemas 2 y 9 el método de escala afín dual supera en un 50% de iteraciones al método simplex con el que contamos.

CONCLUSIONES

Como hemos podido observar a través de este trabajo los métodos de puntos interiores muestran una nueva filosofía para resolver problemas de programación lineal distinta a la utilizada por el método simplex. El surgimiento de estos nuevos algoritmos ha propiciado, inclusive que las instrumentaciones comerciales del método simplex hayan sido mejoradas para poder competir con las instrumentaciones de los algoritmos de puntos interiores. Por lo que la investigación en estas áreas presentan mucha actividad, de ahí nuestro interés por abordar este tipo de algoritmos y de proporcionar una base para aquellos que quieran continuar con este tipo de trabajo.

A nuestro parecer, por el momento no es importante saber qué método es el mejor sino qué tipo de problemas pueden resolverse con cada método. De acuerdo a nuestras observaciones, el método de escala afín primal tiene un mejor comportamiento con problemas degenerados y el método de escala afín dual con problemas esparcos. En cuanto al tipo de factorizaciones de acuerdo a los resultados, consideramos que el método de mínimos cuadrados es el más adecuado para instrumentaciones robustas.

Por otro lado quisiéramos mencionar que para quienes están interesados en realizar instrumentaciones más eficientes, de acuerdo a nuestra experimentación, se debe poner mayor atención a la obtención de la solución interior inicial, a la selección de tamaño de paso y a la detección de no acotación. Es importante hacer notar también que el éxito o fracaso de los métodos de puntos interiores depende de las técnicas numéricas con las

CONCLUSIONES

que se instrumente, por lo que se pueden manejar adicionalmente técnicas de esparsidad, modificaciones de rango uno, métodos de gradiente preconditionado y todas aquellas técnicas que surjan en el álgebra lineal numérica, especialmente para la solución de sistemas de ecuaciones lineales.

Por último es importante hacer notar que estos métodos pueden extenderse a la solución de problemas de programación no lineal, programación con múltiples objetivos o programación entera en donde la solución no necesariamente se encuentra en un vértice del poliedro de soluciones.

APÉNDICE A

CONCEPTOS DE ALGEBRA LINEAL

En este apéndice se proporcionan las definiciones y teoremas del Algebra Lineal que se consideran necesarias, así como la notación a seguir, en el presente trabajo. Los conceptos expuestos aquí pueden consultarse más ampliamente en [Bat], [Baz], [Cal], [Fan], [Gro].

Vectores.

Definición A1. Un vector n -dimensional $x \in \mathbb{R}^n$ es un conjunto ordenado, vertical, de n escalares llamados sus componentes (x_i). Este se escribe como

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

En particular nos interesan los siguientes vectores especiales en \mathbb{R}^n :

- i) el vector cero cuyas n componentes son cero : $0 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.
- ii) el vector "suma" cuyas n componentes son uno : $e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$.

Un vector x en su forma horizontal o transpuesta se denota por x^t . Este se escribe como $x^t = (x_1 \ x_2 \ \dots \ x_n)$.

Dos vectores de la misma dimensión se pueden sumar, y la suma se efectúa por componentes. De la misma manera, la operación de multiplicar un vector x por un escalar α se realiza por componentes.

El producto interno entre dos vectores arbitrarios x y v en \mathbb{R}^n es una operación que les asocia el escalar dado por: $x^t v = \sum_{i=1}^n x_i v_i$. Si $x^t v = 0$, se dice que los vectores x y v son ortogonales y se escribe $x \perp v$. En base a esta operación, se define la norma euclidiana de un vector $x \in \mathbb{R}^n$ arbitrario como el número $\|x\| = \sqrt{x^t x}$ con $x^t x = \sum_{i=1}^n x_i^2 \geq 0$, así como los siguientes conjuntos especiales:

- i) La bola abierta de centro $x^0 \in \mathbb{R}^n$ y radio $r > 0$ tal que:

$$B(x^0, r) = \{x \in \mathbb{R}^n \mid \|x - x^0\| < r\}.$$

- ii) La bola cerrada de centro $x^0 \in \mathbb{R}^n$ y radio $r > 0$ tal que:

$$B_c(x^0, r) = \{x \in \mathbb{R}^n \mid \|x - x^0\| \leq r\}.$$

- iii) Un conjunto abierto $C \subset \mathbb{R}^n$ donde para toda $x \in C$ existe $r > 0$ tal que $B(x, r) \subset C$.

- iv) Un conjunto cerrado $D \subset \mathbb{R}^n$ tal que:

$$\mathbb{R}^n - D = \{x \in \mathbb{R}^n \mid x \notin D\} \text{ es un conjunto abierto.}$$

Definición A2. Un punto $x \in \mathbb{R}^n$ es un punto de acumulación (o punto límite) de un subconjunto $A \subset \mathbb{R}^n$, si para toda $r > 0$ se tiene que $B(x, r) \cap A$ es no vacía y distinta de x .

Teorema A1. Un conjunto es cerrado si y sólo si contiene a todos sus puntos de acumulación.

Definición A3. Un espacio vectorial real V es un conjunto de vectores, junto con dos operaciones, llamadas *suma* y *multiplicación escalar* que satisfacen los diez axiomas que se enumeran a continuación.

1. Si $x \in V$ e $y \in V$, entonces $x+y \in V$ (V es cerrado bajo la suma).
2. Para todo x, y, z en V , $(x+y)+z = x+(y+z)$ (asociatividad en la suma).
3. Existe un vector $0 \in V$ tal que para todo $x \in V$, $x+0 = 0+x = x$ (0 se conoce como neutro aditivo.)
4. Si $x \in V$, existe un vector $-x$ en V tal que $x+(-x) = 0$ ($-x$ se conoce como el inverso aditivo de x).
5. Si x e y están en V , entonces $x + y = y + x$ (conmutatividad en la suma de vectores).
6. Si $x \in V$, entonces $\alpha x \in V$, donde α es un escalar (V es cerrado bajo la multiplicación por escalar).
7. Si x e $y \in V$, y α es un escalar, entonces $\alpha(x+y) = \alpha x + \alpha y$ (ley distributiva para la multiplicación por un escalar).
8. Si $x \in V$ y si α y $\beta \in \mathbb{R}$, entonces $(\alpha+\beta)x = \alpha x + \beta x$ (ley distributiva para la suma de escalares).
9. Si $x \in V$ y si α y β son escalares, entonces $\alpha(\beta x) = \alpha\beta x$. (ley asociativa de la multiplicación escalar).
10. Para todo vector $x \in V$, $1x = x$ (al escalar 1 se le conoce como neutro multiplicativo).

En particular \mathbb{R}^n es un espacio vectorial, ya que éste satisface los axiomas anteriores si tomamos a \mathbb{R} como el conjunto de escalares.

Definición A4. Una combinación lineal de los vectores $x_1, x_2, \dots, x_m \in \mathbb{R}^n$ es un vector $w = \sum_{i=1}^m a_i x_i$, donde a_1, a_2, \dots, a_m son escalares.

Se dice que los vectores x_1, x_2, \dots, x_m son linealmente independientes, cuando $\sum_{i=1}^m a_i x_i = 0$ implica que $a_i = 0$ para $i = 1, \dots, m$. De lo contrario se dice que los vectores son linealmente dependientes.

Definición A5. Se dice que los vectores v_1, v_2, \dots, v_n , del espacio vectorial V generan a V si cualquier vector en V puede ser escrito como una combinación lineal de ellos. Esto es, para todo $v \in V$ existen escalares $\alpha_1, \alpha_2, \dots, \alpha_n$ tales que $v = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n$.

Definición A6. Un conjunto de vectores $\{v_1, v_2, \dots, v_n\}$ constituye una base para V si:

- i. $\{v_1, v_2, \dots, v_n\}$ es linealmente independiente.
- ii. $\{v_1, v_2, \dots, v_n\}$ genera a V .

En particular cualquier conjunto de n vectores linealmente independientes en \mathbb{R}^n es una base en \mathbb{R}^n .

Definición A7. Un conjunto de vectores $\{p_1, p_2, \dots, p_k\}$ no cero es ortogonal si $p_i^T p_j = 0$ para $i \neq j$.

Definición A8. Si cada vector en un conjunto ortogonal cumple además con $p_i^T p_i = 1$ para todo $1 \leq i \leq k$, entonces el conjunto de vectores es ortonormal.

Matrices.

Definición A9. Una matriz $A = (a_{ij})$ de dimensión $m \times n$ es un arreglo rectangular de mn escalares ordenados en m renglones y n columnas. Su componente a_{ij} es el número que aparece en el i -ésimo renglón y la j -ésima columna de A .

Al i -ésimo renglón de A lo denotaremos por A_i y a la j -ésima columna de A por A^j .

La suma de dos matrices de la misma dimensión se define componente por componente; es decir, si A y B son matrices $m \times n$, entonces $C = A+B$ se define tomando $c_{ij} = a_{ij} + b_{ij}$ para $i=1, \dots, m$ y $j=1, \dots, n$. Si A es una matriz de $m \times n$ y k un escalar, entonces el producto de un escalar por una matriz (kA) es una matriz de $m \times n$ cuyo elemento ij es ka_{ij} . Por otro lado si A es una matriz $m \times n$ y B es una matriz $n \times p$, entonces el producto AB se define como la matriz $C_{m \times p}$ con $c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$ para $i=1, \dots, m$ y $j=1, \dots, p$. En otras palabras, el elemento ij de C se determina como el producto interno del renglón A_i por la columna B^j .

Quando una matriz es de dimensión $n \times n$ se dice que es cuadrada. Algunas matrices cuadradas especiales son:

- i) la matriz cero cuyos n^2 elementos son cero,
- ii) la matriz identidad $I = (i_{jk})$ donde $i_{jk} = \begin{cases} 1 & \text{si } j=k \\ 0 & \text{si } j \neq k, \end{cases}$
- iii) la matriz transpuesta $A^t = (a_{ji}^t)$ de $A = (a_{ij})$ dada por $a_{ij}^t = a_{ji}$.

- iv) la matriz simétrica $A = (a_{ij})$ tal que $A = A^t$,
- v) la matriz positiva definida $A = (a_{ij})$, donde A es una matriz cuadrada, simétrica y tal que $\forall x \in \mathbb{R}^n, x \neq 0 : x^t A x > 0$. Si $x^t A x \geq 0$ la matriz A se considera una matriz positiva semi-definida.
- vi) la matriz triangular superior $U = (u_{ij})$ donde $u_{ij} = 0$ si $i > j$.
- vii) la matriz triangular inferior $L = (l_{ij})$ donde $l_{ij} = 0$ si $i < j$.
- viii) la matriz diagonal $D = (d_{ij})$ donde $d_{ij} = 0$ si $i \neq j$. Este tipo de matrices se denotan por $D = [d_{11}, d_{22}, \dots, d_{nn}]$.

El rango de una matriz es el número máximo de renglones (o columnas) de ésta que son linealmente independientes.

Si A es una matriz de dimensión $m \times n$ y es de rango m , es decir que todos sus renglones son linealmente independientes, se dice que A es de rango completo por renglones. Por otro lado, si A es de rango n , es decir que todas sus columnas son linealmente independientes, se dice que A es de rango completo por columnas.

Teorema A2. Sea A una matriz de $m \times n$ de rango q . Entonces, la matriz simétrica AA^t de $m \times m$ tiene también rango q .

Si B es una matriz $p \times q$, con rango r , y A es una matriz $q \times q$ no singular, entonces la matriz BA es también de rango r . Así mismo, el rango de AB^t es r .

Si A e I son matrices cuadradas $n \times n$ entonces se satisface $AI = IA = A$. Esto es, I conmuta con cualquier matriz $n \times n$ y la deja intacta después de la multiplicación por la izquierda o por la derecha.

Definición A10. Sea $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$. Entonces el determinante de una matriz cuadrada de 2×2 está dado por

$$\det(A) = |A| = a_{11}a_{22} - a_{21}a_{12}.$$

Definición A11. Sea A una matriz de $n \times n$ y sea M_{ij} la matriz de $(n-1) \times (n-1)$ que se obtiene de A , al eliminar el i -ésimo renglón y la j -ésima columna de A . Entonces M_{ij} es llamado el ij -ésimo menor de A .

Definición A12. Sea A una matriz de $n \times n$. El ij -ésimo cofactor de A , denotado por A_{ij} , está dado por

$$A_{ij} = (-1)^{i+j} |M_{ij}|$$

Definición A13. Sea A una matriz de $n \times n$. Entonces el determinante de A , escrito $\det(A)$ o $|A|$ está dado por:

$$\det(A) = |A| = \sum_{i=1}^n a_{ii} A_{ii}$$

Así pues a cada matriz cuadrada $n \times n$ se le puede asociar un número real, el cual se llama el determinante de la matriz.

Se dice que una matriz A de $n \times n$ es no singular si $\det(A) \neq 0$. En caso contrario, es decir si $\det(A) = 0$ se dice que A es singular.

Algunas propiedades de los determinantes son:

- $\det(A) = \det(A^t)$.
- Sea B la matriz que se obtiene de A al intercambiar dos renglones (o columnas). Entonces $\det(B) = -\det(A)$.
- Sea B la matriz que se obtiene de A al sumar a un renglón (o a una columna) algún otro renglón (o columna) multiplicado por una constante. Entonces $\det(B) = \det(A)$.
- Sea B la matriz que se obtiene de A al multiplicar un renglón (o columna) por un escalar k . Entonces $\det(B) = k \cdot \det(A)$.
- Sean A y B matrices de $n \times n$. Entonces $\det(AB) = \det(A) \cdot \det(B)$.
- $\det(A) \neq 0$ si y sólo si, las columnas (y renglones) de A son linealmente independientes.
- El determinante de una matriz triangular es el producto de los elementos en la diagonal.

Definición A14. Sean A y B dos matrices $n \times n$, tales que $AB = BA = I$. Entonces se dice que B es la matriz inversa de A , y se escribe $B = A^{-1}$.

Si A tiene inversa entonces decimos que A es invertible y su inversa es única. Puede probarse que A^{-1} existe si y sólo si se da una de las siguientes condiciones: el rango de A es n o el determinante de A es distinto de cero.

Además se tienen los siguiente teoremas cuya demostración puede ser consultada en cualquier libro de algebra lineal.

Teorema A3. Sean A y B matrices invertibles de $n \times n$. Entonces AB es invertible y

$$(AB)^{-1} = B^{-1}A^{-1}.$$

Teorema A4. Supongamos que A es una matriz de $n \times m$ y B es una matriz de $m \times p$. Entonces:

- i. $(A^t)^t = A$,
- ii. $(AB)^t = B^tA^t$,
- iii. Si A y B son de $n \times m$, entonces $(A+B)^t = A^t + B^t$,
- iv. Si A es no singular, entonces A^t también es no singular y $(A^t)^{-1} = (A^{-1})^t$.

Definición A15. Una norma matricial en el conjunto de todas las matrices reales de $n \times n$ es una función de valores reales, $\| \cdot \|$, definida en este conjunto que satisface, para todas las matrices A y B de $n \times n$ y todo número real α :

- i. $\| A \| \geq 0$,
- ii. $\| A \| = 0$ si y sólo si A es 0 ,
- iii. $\| \alpha A \| = |\alpha| \| A \|$,
- iv. $\| A + B \| \leq \| A \| + \| B \|$,
- v. $\| AB \| \leq \| A \| \| B \|$.

Definición A16. Si $\| \cdot \|$ es cualquier norma vectorial en \mathbb{R}^n , entonces

$$\|A\| = \max_{\|x\|=1} \|Ax\|$$

define una norma matricial en el conjunto de las matrices reales de $n \times n$, la cual se llama la norma natural.

Si $A = (a_{ij})$ es una matriz de $n \times n$, entonces

$$\|A\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|.$$

Definición A17 El número de condición $K(A)$ de la matriz no singular A relativo a la norma $\| \cdot \|$ se define como

$$K(A) = \|A\| \|A^{-1}\|.$$

Una matriz es una matriz bien condicionada si $K(A)$ está cerca de uno y está mal condicionada cuando $K(A)$ sea significativamente mayor que uno.

Definición A18. Una matriz de permutación P de $n \times n$ es, una "identidad revuelta", es decir, sus columnas (renglones) son las columnas (renglones) de I_n en cualquier orden.

Una matriz de permutación es ortogonal, con todos los elementos cero excepto por exactamente un "uno" en cada renglón y columna.

La aplicación de una matriz de permutación P a otra matriz cualquiera A intercambia todos los renglones o columnas de A , dependiendo de si A

es premultiplicada o posmultiplicada por P . Cuando P premultiplica a A , los renglones de PA son los renglones de A intercambiados, de acuerdo al orden de los renglones de P . Cuando P multiplica al lado derecho de A , las columnas de AP son las columnas de A intercambiadas, de acuerdo al orden de las columnas de P .

Definición A19 Una matriz de $n \times n$ es una matriz ortogonal si sus columnas son ortonormales.

Hiperplanos y Semiespacios.

Definición A20. Un hiperplano en \mathbb{R}^n es un conjunto de la forma $H = \{x \in \mathbb{R}^n \mid a^t x = \beta\}$ donde β es un escalar y $a \neq 0$ es un vector en \mathbb{R}^n llamado normal al hiperplano H .

Un hiperplano H divide a \mathbb{R}^n en dos semiespacios H^+ y H^- , dados por $H^+ = \{x \in \mathbb{R}^n \mid a^t x \geq \beta\}$ y $H^- = \{x \in \mathbb{R}^n \mid a^t x \leq \beta\}$. Además se cumple que $H^+ \cup H^- = \mathbb{R}^n$ y $H^+ \cap H^- = H$.

Nota : Sin pérdida de generalidad, todo semiespacio se puede representar por una desigualdad del tipo $a^t x \leq \beta$. En efecto, este hecho se cumple trivialmente para H^- , mientras que $H^+ = \{x \in \mathbb{R}^n \mid (-a)^t x \leq (-\beta)\} = \hat{H}^-$ relacionado al hiperplano $\hat{H} = \{x \in \mathbb{R}^n \mid (-a)^t x = (-\beta)\}$ que es idéntico a H pero cuyo vector normal formalmente es $-a$.

Cabe hacer notar que los hiperplanos y los semiespacios son ejemplos de conjuntos convexos.

Ortante Positivo

Otro ejemplo de conjunto convexo lo constituye el primer ortante u ortante positivo, que se define como el conjunto de todos los vectores en \mathbb{R}^n con componentes no negativas y se denota como

$$\mathbb{R}_+^n = \{x \in \mathbb{R}^n \mid x_i \geq 0, i=1, \dots, n\}$$

Simplejo

Un simplejo n-dimensional contenido en \mathbb{R}^{n+1} se denota y define como:

$$\Delta^n = \{x \in \mathbb{R}^{n+1} \mid \sum_{i=1}^{n+1} x_i = 1 \text{ y } x_i \geq 0 \text{ donde } i=1, \dots, n+1\}$$

Así $\Delta^0 \subset \mathbb{R}^1$ es el punto $+1$ sobre la recta de los reales; $\Delta^1 \subset \mathbb{R}^2$ es el segmento de línea entre los puntos $(0,1)$ y $(1,0)$ en el plano; $\Delta^2 \subset \mathbb{R}^3$ es el área triangular formada por $(0,0,1)$, $(0,1,0)$ y $(1,0,0)$; y $\Delta^3 \subset \mathbb{R}^4$, llega a ser la pirámide con vértices en $(0,0,0,1)$, $(0,0,1,0)$, $(0,1,0,0)$ y $(1,0,0,0)$. Observe que en \mathbb{R}^{n+1} , Δ^n tiene exactamente $n+1$ vértices, $C(n+1,2)$ ejes, $C(n+1,n)$ facetas y su centro está en $\frac{e}{n+1}$. La distancia del centro del simplejo a uno de sus vértices (ver fig A.1), digamos $(1,0, \dots, 0)$, nos proporciona el radio de la bola más pequeña circunscrita a Δ^n ; es decir

$$R = \frac{\sqrt{n}}{\sqrt{n+1}}$$

Similarmente, la distancia del centro del simplejo al centro de una de sus facetas (ver fig A.1) nos proporciona el radio de la bola más grande inscrita en Δ^n . Por ejemplo el centro de la faceta $x_{n+1} = 0$ es $(1/n, \dots, 1/n, 0)$. Por tanto el radio es

$$r = \frac{1}{\sqrt{n(n+1)}}$$

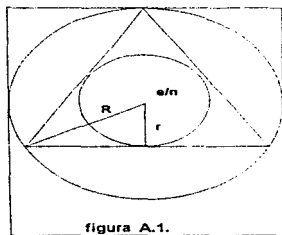


figura A.1.

APÉNDICE B.

CONCEPTOS DE PROGRAMACIÓN NO LINEAL

En este apéndice se proporcionan los conceptos de Programación no Lineal que se consideran necesarios, así como la notación a seguir, en el presente trabajo. Los conceptos expuestos aquí pueden consultarse más ampliamente en [Pra], [Tah].

El Problema de Programación No Lineal.

Sea $f(x)$ una función continua, que representa a la función objetivo; sean $h_1(x), h_2(x), \dots, h_m(x)$ funciones continuas que representan a las restricciones de igualdad, $g_{m+1}(x), g_{m+2}(x), \dots, g_p(x)$, funciones continuas que representan a las restricciones de desigualdad y $x \in \mathbb{R}^n$ un vector columna. El problema de programación no lineal se define como:

Opt $f(x)$

s. a

$$h_j(x) = 0, \quad j=1, \dots, m$$

$$g_j(x) \geq 0, \quad j=m+1, \dots, p$$

$$x \in \mathbb{R}^n.$$

La palabra optimización (Opt) puede significar maximización o minimización.

Los problemas de programación no lineal pueden ser:

- ◆ *restringidos* : cuando se tienen restricciones (lineales o no lineales),
- ◆ *no-restringidos* : cuando no se tienen restricciones y sólo se optimiza la función objetivo, que desde luego, no es lineal,
- ◆ *continuos* : cuando las variables y funciones son continuas,
- ◆ *discretos* : cuando alguna de las variables y/o funciones es discreta,
- ◆ *diferenciables* : cuando todas las funciones del problema son doblemente diferenciables (se puede derivar la matriz Hessiana),
- ◆ con restricciones de igualdad y/o desigualdad,
- ◆ convexos, cuadráticos, separables,
- ◆ con una *sola* variable independiente o con *varias* variables independientes.

En específico en este trabajo nos vamos a interesar en el problema de programación no lineal, no restringido, continuo, diferenciable y con varias variables.

Problemas de Optimización no Restringidos.

El problema a resolver es

$$\begin{aligned} & \text{Min } f(x) \\ & \text{s. a} \\ & \quad x \in \mathbb{R}^n. \end{aligned}$$

donde $f(x)$ es una función real. Como $\text{Max } f(x) = -\text{Min } -f(x)$, sólo se tomarán en cuenta los procesos de minimización.

En los problemas no lineales generalmente existen mínimos locales y globales, a diferencia de los problemas lineales en los que sólo se tienen mínimos globales. Se dice que un vector $x^* \in \mathbb{R}^n$ es un mínimo relativo o local de $f(x^*)$, si existe una $\epsilon > 0$ tal que $f(x^*) \leq f(x)$ para toda x tal que $\|x - x^*\| < \epsilon$. Si $f(x^*) \leq f(x)$ para toda $x \in \mathbb{R}^n$, entonces el vector x^* es un mínimo absoluto o global.

Contorno de una Función.

Se llama un contorno de $f(x)$ al conjunto de puntos para los cuales $f(x)$ es constante.

Gradiente de una Función.

Si $f(x)$ es continua y diferenciable, se define como el gradiente de $f(x)$, escrito como $\nabla f(x)$, al siguiente vector de funciones:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix} \quad (\text{B.1})$$

Como el gradiente puede evaluarse para un valor determinado, del vector x , se denotará por $\nabla f(x^k)$, al valor del gradiente cuando el vector x toma el valor x^k .

Para un hiperplano de la forma $\{x \in \mathbb{R}^n \mid f(x) = a_1x_1 + a_2x_2 + \dots + a_nx_n\}$, el gradiente de $f(x)$ obtenido como en (B.1) es $\nabla f(x) = a = (a_1, a_2, \dots, a_n)^t$, donde a , de acuerdo a la definición A.8, es la normal al hiperplano $f(x) = a^t x$.

Hessiano de una Función.

Si $f(x)$ es continua y diferenciable se define como la matriz Hessiana o simplemente el Hessiano de $f(x)$, escrito como $H(x)$ a la siguiente matriz cuadrada:

$$H(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

El Hessiano evaluado en el punto $x^k = (x_1^k, x_2^k, \dots, x_n^k)$ se escribe $H(x^k)$.

Condiciones Necesarias para Obtener el Mínimo Local de una Función.

En general los métodos de programación no lineal encuentran un mínimo local o relativo, por lo que es importante conocer las condiciones que debe cumplir dicho punto.

Las condiciones necesarias y suficientes que debe satisfacer el punto mínimo local de una función $f(x)$ se encuentran consideradas en el siguiente teorema, cuya prueba puede consultarse en [Pra].

Teorema B.1. Sea $f(x)$ una función continua y diferenciable en \mathbb{R}^n . Una condición necesaria para que x^* sea un mínimo local de $f(x)$ es que $\nabla f(x^*) = 0$ y que $H(x^*)$ sea positiva semi-definida (ver definición en el apéndice A). Una condición suficiente para que x^* sea un mínimo local de $f(x)$ es que $\nabla f(x^*) = 0$ y que $H(x^*)$ sea positiva definida (ver definición en el apéndice A).

El resolver un problema no restringido por medio de los métodos clásicos del cálculo presenta grandes dificultades al querer satisfacer las condiciones del teorema B.1. Sobre todo al calcular $\nabla f(x) = 0$ pues se generan n ecuaciones no lineales para resolver, lo que generalmente es muy difícil o imposible. Por lo que se requieren otras técnicas de solución como lo son los métodos de gradiente.

Filosofía de los Métodos de Gradiente.

Los métodos de gradiente para optimizar funciones no restringidas, de varias variables, tienen como común denominador el uso del gradiente. Existen una gran variedad de estos métodos (los cuales pueden ser consultados en cualquier libro de programación no lineal), pero en este trabajo sólo estamos interesados en el método de ascenso o descenso acelerado.

La filosofía de los métodos de gradiente está basada en tres aspectos fundamentales para la búsqueda de un óptimo local, los cuales se enumeran a continuación:

- i) un punto conocido de partida,
- ii) una dirección de búsqueda, y
- iii) una longitud a recorrer sobre dicha dirección.

La idea es pues que a partir de un punto dado nos traslademos hacia otro punto mejor que el anterior; identificando para ello una dirección que nos indique hacia donde movernos y que distancia recorrer sobre esta dirección para obtener dicho punto. Este proceso (el cual ocupará distintos puntos de partida, direcciones y longitudes a recorrer sobre dichas direcciones) debe realizarse sucesivamente hasta que no nos sea posible desplazarnos a otro punto mejor que el actual.

Específicamente, una vez que se tiene el punto conocido de partida, los métodos de gradiente determinan la mejor dirección (utilizando el gradiente de la función evaluado en el punto de partida) y la mejor longitud de recorrido en esa dirección, para poder alcanzar el máximo o mínimo local, en el menor tiempo posible (menor número de iteraciones). Debe observarse que el gradiente es siempre perpendicular a la tangente del contorno de la función.

Ahora procederemos a explicar el método de ascenso o descenso acelerado.

Método de Ascenso o Descenso Acelerado.

Sea $f(x)$ una función diferenciable en \mathbb{R}^n . Dado un punto inicial de partida x^0 , la dirección de descenso acelerado en este punto está dado por $-\nabla f(x^0)$ (se usa el descenso acelerado para el cálculo de mínimos locales; en el caso de máximos, la dirección de ascenso acelerado está dada por $\nabla f(x^0)$), y la longitud de recorrido por un parámetro, $r_k > 0$ (llamado tamaño de paso óptimo), tal que el punto generado en la iteración k , donde $k = 0, 1, 2, \dots$, es:

$$x^{k+1} = x^k - r_k \nabla f(x^k). \quad (B.2)$$

La expresión (B.2) es conocida como la ecuación de descenso acelerado (para maximización se conoce como la ecuación de ascenso acelerado). El parámetro r_k se determina de modo que x^{k+1} resulta en una mayor mejora en f . En otras palabras si una función $h(r)$ se define de manera que

$$h(r) = f(x^k - r \nabla f(x^k)).$$

r_k toma el valor de r que maximiza a $h(r)$, ya que $h(r)$ es una función de una sola variable, pues en la iteración k se conocen a x^k y a $\nabla f(x^k)$, para $k=0,1,2,\dots$

El vector $x^k - r_k \nabla f(x^k)$ es tangente al contorno $h(r)$ y como $\nabla f(x^{k+1})$ es perpendicular a este contorno, los movimientos sucesivos son en ángulos rectos de 90° .

El algoritmo puede ser detenido por medio de alguno de los siguiente criterios. El primero de ellos consiste en parar cuando $\nabla f(x^*) = 0$ en el punto x^* , pues se sabe que esta debe ser una condición necesaria (más no suficiente) que debe cumplir un punto máximo o mínimo. En el segundo criterio se debe comparar el valor de la función en dos iteraciones sucesivas y parar el algoritmo cuando esa diferencia es menor a una tolerancia arbitraria, es decir, cuando $|f(x^{k+1}) - f(x^k)| < \epsilon$, donde $\epsilon > 0$.

Este método encontrara un mínimo (o máximo) local, generalmente el que está cercano al punto de inicialización x^0 . Para encontrar todos los mínimos (o máximos) locales hay que repetir el método con diferentes puntos de inicialización x^0 .

La desventaja de este método es que su convergencia al óptimo local buscado es muy lenta, debido precisamente a los movimientos perpendiculares. Pero si la función en cuestión tiene contornos circulares (como sucede en los PPL transformados por el algoritmo de Karmarkar para la optimización sobre una bola), el óptimo local se encuentra en una sola iteración.

APÉNDICE C

MANUAL DEL USUARIO

En este apéndice se proporcionan los detalles para ejecutar las instrumentaciones computacionales.

La instrumentación computacional del método de escala afín primal y del método de escala afín dual reciben como entrada un PPL de la siguiente forma:

$$\text{Opt } z = c^T x$$

s. a

$$A_1 x \leq b_1$$

$$A_2 x \geq b_2$$

$$A_3 x = b_3$$

$$x \geq 0$$

donde Opt estará definido como minimizar o maximizar.

A continuación se describen los datos de entrada para que la ejecución se realice correctamente. Los puntos en este manual representan las distintas pantallas de las instrumentaciones.

C.1 MÉTODO DE ESCALA AFÍN PRIMAL

1. En la primera pantalla se despliega la presentación del método.
Presione enter para continuar.

2. Se captura el número de variables y restricciones que tiene el PPL.

EL NÚMERO DE VARIABLES QUE TIENE EL PPL ES? : N
EL NÚMERO DE RESTRICCIONES QUE TIENE EL PPL ES? : M

donde:

N : es un número entero positivo, $2 < N < 25$.

M : es un número entero positivo, $1 < M < 25$.

3. Se captura las características del PPL a resolver.

LA FUNCIÓN OBJETIVO DEL PPL SE TIENE QUE:

MINIMIZAR (1)

MAXIMIZAR (2)

SU SELECCIÓN ES? : _

La selección debe ser un número entero positivo 1 ó 2.

4. Se captura el tipo de restricción para cada una de las restricciones

MENOR O IGUAL (L)

MAYOR O IGUAL (G)

IGUAL A (E)

EL TIPO DE LA RESTRICCIÓN I ES : _

El tipo de restricción debe ser un caracter L, G ó E para cada una de las I restricciones.

5. Se capturan los coeficientes de la función objetivo.

EL COEFICIENTE $X(J)$ DE LA FUNCIÓN OBJETIVO ES : C(J)

C(J) puede ser un número entero o real.

6. Se captura la estructura de la matriz de restricciones.

LA MATRIZ DE RESTRICCIONES SE CAPTURA COMO:

MATRIZ ESPARSA (1)

MATRIZ NO ESPARSA (2)

SU SELECCIÓN ES: _

La selección debe ser un número entero positivo 1 ó 2. Si su selección es 1:

6.1 Se captura la matriz esparsa de restricciones:

EN LA RESTRICCIÓN I
QUE VARIABLE TIENE COEFICIENTE DISTINTO DE CERO: J
EL COEFICIENTE DE LA VARIABLE X(I) EN LA RESTRICCIÓN J
ES: A(I,J)

La respuesta a la variable que tienen coeficiente distinto de cero deberá ser un entero positivo entre 1 y N. En el coeficiente se puede introducir un número entero o real.

Cuando se haya terminado de introducir los coeficientes de cada restricción se tecleará un 0 (cero) en:

QUE VARIABLE TIENE COEFICIENTE DISTINTO DE CERO: 0

para que continúe con la siguiente restricción, o bien para indicar que ha finalizado la captura de la matriz.

Si su selección es 2:

6.2 Se captura la matriz de restricciones sin esparcidad.

EL COEFICIENTE DE LA VARIABLE X(J) EN LA RESTRICCIÓN I
ES: A(I,J)

Se capturan todos los elementos de la matriz, los valores deben ser enteros o reales.

7. Se capturan los términos independientes.

EL TÉRMINO INDEPENDIENTE DE LA RESTRICCIÓN I ES: B(I)

Los términos a introducir pueden ser enteros o reales.

8. Se despliega el PPL proporcionado y, se hace la siguiente pregunta:

DESEA EFECTUAR ALGUNA CORRECCIÓN AL PPL (S/N)?:_

Debe ingresar un caracter S ó N. Si su respuesta es S:

8.1 Se despliega una pantalla para la corrección de los datos de entrada del PPL proporcionado

TIPO DE PROBLEMA (MAXIMIZAR Ó MINIMIZAR)	(1)
COEFICIENTE EN LA FUNCIÓN OBJETIVO	(2)
COEFICIENTE EN LA MATRIZ DE RESTRICCIONES	(3)
COEFICIENTE DEL VECTOR DEL LADO DERECHO	(4)
TIPO DE RESTRICCIÓN (L:<=, G:>=, E=)	(5)
SALIR	(6)

SU SELECCIÓN ES:_

La selección debe ser un número entero positivo 1, 2, 3, 4, 5 ó 6.

8.1.1 Si su selección es 1, se despliega

"CORRECCIÓN DEL TIPO DE PROBLEMA"

TIPO DE PROBLEMA ACTUAL:

INGRESE (1) PARA MINIMIZACIÓN, (2) PARA MAXIMIZACIÓN: _

En el tipo de problema actual se muestra el que ingresó inicialmente (minimización ó maximización). Debe introducir un número entero positivo 1 ó 2.

Enseguida aparece:

DESEA EFECTUAR OTRA CORRECCIÓN (S/N)?: _

Debe ingresar un caracter S ó N. Si la respuesta es S, regresa a la pantalla para la corrección de los datos de entrada del PPL proporcionado. Si la respuesta es N, se despliega el PPL proporcionado corregido. Esta pregunta aparece cuando ha terminado la operación de corrección en cada una de las opciones.

8.1.2 Si su selección es 2, se despliega

"CORRECCIÓN DE COEFICIENTE EN LA F.O."

INGRESE EL NÚMERO DE LA VARIABLE DE DECISIÓN: I

EL COEFICIENTE ACTUAL DE LA VARIABLE DE DECISIÓN ES: C(I)

INGRESE EL NUEVO COEFICIENTE: C(I)

En el número de la variable de decisión deberá introducir un número entero positivo entre 1 y N. En el coeficiente actual, se muestra el que ingresó inicialmente. En ingrese el nuevo coeficiente, debe proporcionar un número entero ó real

8.1.3 Si su selección es 3, se despliega

"CORRECCIÓN DE COEFICIENTE EN LA MATRIZ DE RESTRICCIONES"
INGRESE EL NÚMERO DE LA RESTRICCIÓN: I
INGRESE EL NÚMERO DE LA VARIABLE DE DECISIÓN: J
EL COEFICIENTE ACTUAL DE ESTA VARIABLE ES: A(I,J)
INGRESE EL NUEVO COEFICIENTE: A(I,J)

El número de la restricción y la variable de decisión deben ser números enteros positivos entre 1 y M y, entre 1 y N respectivamente. Se muestra el coeficiente actual que ingresó inicialmente. Posteriormente el coeficiente a introducir debe ser un número entero ó real.

8.1.4 Si su selección es 4, se despliega

"CORRECCIÓN DE TÉRMINO INDEPENDIENTE"
INGRESE EL NÚMERO DE LA RESTRICCIÓN: I
EL TÉRMINO INDEPENDIENTE ACTUAL ES: B(I)
INGRESE EL NUEVO TÉRMINO INDEPENDIENTE: B(I)

El número de la restricción debe ser un número entero positivo entre 1 y M. Se muestra el término que ingresó inicialmente. El término independiente a ingresar puede ser un número entero o real.

8.1.5 Si su selección es 5, se despliega

"CORRECCIÓN DEL TIPO DE RESTRICCIÓN"
INGRESE EL NÚMERO DE LA RESTRICCIÓN: I
EL TIPO ACTUAL DE LA RESTRICCIÓN ES: (L,G,M)
INGRESE EL NUEVO TIPO DE RESTRICCIÓN (L<=,G>=,E=):_

El número de la restricción debe ser un entero positivo entre 1 y M. Se muestra el tipo de restricción que introdujo inicialmente. En el tipo de restricción deberá ingresar un caracter L, E ó G.

8.1.6 Si su selección es 6, se despliega el PPL proporcionado con las correcciones y la pregunta:

DESEA EFECTUAR ALGUNA CORRECCIÓN AL PPL (S/N)?:_

Si su respuesta es N, continúe con el punto 9. En caso contrario regresará al punto 8.

9. Se despliega una pantalla para la captura de los parámetros de ejecución del método de escala afín primal.

PARA EJECUTAR EL PROGRAMA, USTED DESEA:

PROPORCIONAR LOS VALORES DE LOS PARÁMETROS (1)
EL PROGRAMA SE EJECUTE CON LOS VALORES QUE SE TIENEN ASIGNADOS (2)

SU SELECCIÓN ES:_

La selección debe ser un número entero positivo 1 ó 2. Si su selección es 1, se capturan los parámetros de ejecución del método de escala afín primal

EL VALOR DE LA M-GRANDE: MG
EL FACTOR DE TAMAÑO DE PASO ES: R
LA COTA SOBRE LA FUNCIÓN OBJETIVO ES: COTA
LA COTA PARA LA FACTIBILIDAD PRIMAL ES: E1
LA COTA PARA LA FACTIBILIDAD DUAL ES: E2
LA COTA PARA HOLGURAS COMPLEMENTARIAS ES: E3
LA COTA SOBRE LA VARIABLE ARTIFICIAL ES: E4
LA COTA SOBRE EL VECTOR DIRECCIÓN ES: E5

donde

MG: es un número real, $MG > 0$.
R: es un número real, $0 < R \leq 1$
COTA: es un número real, $COTA > 0$
E1: es un número real, $0 \leq E1 < 1$
E2: es un número real, $0 \leq E2 < 1$
E3: es un número real, $0 \leq E3 < 1$
E4: es un número real, $0 \leq E4 < 1$
E5: es un número real, $0 \leq E5 < 1$

NOTA: Si se desea $MG=10^4$ debe introducirse 10E5, si se desea escribir en algunas de las cotas por ejemplo 10^{-3} debe teclearse 10E-4

Enseguida aparece la pregunta:

REQUIERE VOLVER A CAPTURAR LOS PARÁMETROS (S/N)?_

Debe ingresar un caracter S ó N. Si la respuesta es S, se regresa a la captura de los parámetros de ejecución del método de escala afn primal. En caso contrario, se despliega el PPL de la M-grande asociado al PPL proporcionado.

Si la selección es 2, se despliega el PPL de la M-Grande asociado al PPL proporcionado.

10. Se despliegan los métodos de factorización matricial que pueden ser utilizados en la obtención del vector dirección

FACTORIZACIÓN UDU ^t	(1)
FACTORIZACIÓN DE CHOLESKY	(2)
FACTORIZACIÓN QR	(3)
MINIMOS CUADRADOS	(4)

EL MÉTODO SELECCIONADO ES?:_

La selección debe ser un número entero positivo entre 1 y 4. Todos los métodos funcionan de la misma forma, por lo cual bastará mencionar el manejo en forma general.

Enseguida aparece la pregunta:

DESEA VER CADA SOLUCIÓN INTERIOR OBTENIDA EN CADA ITERACIÓN DEL MÉTODO (S/N)?:_

Debe ingresar un caracter S ó N. Si su respuesta es S, se despliega la solución interior obtenida en cada iteración, el error relativo de factibilidad primal, y si la solución es factible o no para la cota de error dada.

El método se detiene cuando se satisface alguna de las condiciones de paro. Si la variable artificial es distinta de cero, se detecta infactibilidad; si el

vector dirección es menor que cero ó el valor absoluto de la f.o. es mayor que una cota dada (a partir de que la variable artificial es cero), se detecta que el PPL es no acotado; si la norma 2 del vector dirección es menor que una cota dada, se despliega f.o. constante. En caso de que ninguna de estas condiciones se cumpla y el método se haya detenido, se habrá alcanzado la solución óptima, entonces se mostrará el número de iteraciones, el valor de las variables, los costos reducidos a cada variable y el vector estimado dual.

11. Se despliega la siguiente pregunta:

DESEA RESOLVER NUEVAMENTE EL PPL PROPORCIONADO
PERO CON OTRO MÉTODO DE FACTORIZACIÓN (S/N)? :_

Debe ingresar un caracter S ó N. Si su respuesta es S regresa al punto

10. En caso contrario se despliega la siguiente pregunta:

DESEA RESOLVER NUEVAMENTE EL PPL PROPORCIONADO
PERO CON DISTINTOS PARÁMETROS (S/N)? :_

Debe ingresar un caracter S ó N. Si su respuesta es S regresa al punto 9.

En caso contrario se despliega la siguiente pregunta:

DESEA RESOLVER OTRO PROBLEMA DE PROGRAMACIÓN
LINEAL CON ESTA INSTRUMENTACIÓN COMPUTACIONAL
(S/N)?:_

Debe ingresar un caracter S ó N. Si su respuesta es S regresa al punto 2.

En caso contrario, se da por terminada la ejecución de la instrumentación computacional para el método de escala afín primal.

C.2 MÉTODO DE ESCALA AFÍN DUAL.

Los datos de entrada requeridos en este método son los mismos que para el método de escala afín primal, esto es, se deben seguir los puntos del 1 al 8 de la versión primal. Por lo que el punto siguiente es:

9. Se despliega una pantalla para la captura de los parámetros de ejecución del método de escala afín dual.

PARA EJECUTAR EL PROGRAMA, USTED DESEA:

PROPORCIONAR LOS VALORES DE LOS PARÁMETROS	(1)
EL PROGRAMA SE EJECUTE CON LOS VALORES QUE SE TIENEN ASIGNADOS	(2)

SU SELECCIÓN ES: _

La selección debe ser un número entero 1 ó 2. Si su selección es (1), se capturan los parámetros de ejecución del método de escala afín dual.

EL VALOR DE LA M-GRANDE: MG
EL FACTOR DE TAMAÑO DE PASO ES: R
LA COTA SOBRE LA FUNCIÓN OBJETIVO ES: COTA
LA COTA PARA LA FACTIBILIDAD PRIMAL ES: E1
LA COTA PARA LA FACTIBILIDAD DUAL ES: E2
LA COTA PARA HOLGURAS COMPLEMENTARIAS ES: E3
LA COTA SOBRE EL VECTOR DIRECCIÓN ES: E4
EL PARÁMETRO DE CÁLCULO DE LA VAR. ARTIFICIAL ES: 0

donde

MG: es un número real, $MG > 0$.
R: es un número real, $0 < R \leq 1$
COTA: es un número real, $COTA > 0$
E1: es un número real, $0 \leq E1 < 1$
E2: es un número real, $0 \leq E2 < 1$
E3: es un número real, $0 \leq E3 < 1$
E4: es un número real, $0 \leq E4 < 1$
 θ : es un número real, $\theta > 0$.

NOTA: Si se desea $MG=10^4$ debe introducirse 10E5, si se desea escribir en algunas de las cotas por ejemplo 10^{-3} debe teclearse 10E-4

Enseguida aparece la pregunta:

REQUIERE VOLVER A CAPTURAR LOS PARÁMETROS (S/N)?:_

Debe ingresar un caracter S ó N. Si la respuesta es S, se regresa a la captura de los parámetros de ejecución del método de escala afín dual. En caso contrario, se despliega el PPL de la M-grande dual asociado al PPL proporcionado.

Si la selección es (2), se despliega el PPL de la M-Grande dual asociado al PPL proporcionado.

10. Se despliegan los métodos de factorización matricial que pueden ser utilizados en la obtención del vector dirección

FACTORIZACIÓN UDU ⁴	(1)
FACTORIZACIÓN DE CHOLESKY	(2)
FACTORIZACIÓN QR	(3)
MINIMOS CUADRADOS	(4)

EL MÉTODO SELECCIONADO ES?:_

La selección debe ser un número entero positivo entre 1 y 4. Todos los métodos funcionan de la misma forma, por lo que se supondrá que alguno de ellos fue seleccionado, sin importar cual, y se continuará con los pasos siguientes.

Enseguida aparece la pregunta:

DESEA VER CADA SOLUCIÓN INTERIOR OBTENIDA EN CADA ITERACIÓN DEL MÉTODO (S/N)?:_

Debe ingresar un caracter S ó N. Si su respuesta es S, se despliega el valor de las variables no restringidas; el valor de las variables restringidas, el error relativo de factibilidad dual, y si la solución es factible o no para la cota de error dada, el valor de la función objetivo del PPL dual.

El método se detiene cuando se satisface alguna de las condiciones de paro. Si la variable artificial es distinta de cero, se detecta que el PPL es no acotado; si el vector dirección es mayor o igual que una cota dada (a partir de que la variable artificial es cero), se detecta que el PPL es infactible; si la

norma 2 del vector dirección es menor que una cota dada se detecta f.o. constante. En caso de que ninguna de estas condiciones se cumpla y el método se haya detenido, se habrá alcanzado la solución óptima, entonces se desplegará el número de iteraciones, la solución óptima dual; el valor óptimo de la función objetivo (dual); el vector estimado primal y el valor de la función objetivo primal correspondiente a dicha estimación.

11. Se despliega la siguiente pregunta:

DESEA RESOLVER NUEVAMENTE EL PPL PROPORCIONADO
PERO CON OTRO MÉTODO DE FACTORIZACIÓN (S/N)? :_

Debe ingresar un caracter S ó N. Si su respuesta es S regresa al punto
10. En caso contrario se despliega la siguiente pregunta:

DESEA RESOLVER NUEVAMENTE EL PPL PROPORCIONADO
PERO CON DISTINTOS PARÁMETROS (S/N)? :_

Debe ingresar un caracter S ó N. Si su respuesta es S regresa al punto 9.
En caso contrario se despliega la siguiente pregunta:

DESEA RESOLVER OTRO PROBLEMA DE PROGRAMACIÓN
LINEAL CON ESTA INSTRUMENTACIÓN COMPUTACIONAL
(S/N)?:_

Debe ingresar un caracter S ó N. Si su respuesta es S regresa al punto 2.
En caso contrario, se da por terminada la ejecución de la instrumentación
computacional para el método de escala afín dual.

NOTA: En la captura de coeficientes de la función objetivo, de la matriz de restricciones ó del término independiente debe tenerse cuidado de no proporcionar un valor caracter porque el programa se interrumpe.

APÉNDICE D

CÓDIGO CORRESPONDIENTE AL MÉTODO DE ESCALA AFÍN PRIMAL

```

C      INSTRUMENTACION COMPUTACIONAL DEL ALGORITMO DE ESCALA AFIN
C      PRIMAL UTILIZANDO IMSL
      INCLUDE 'GRAPH.FIT'
      INCLUDE 'GRAPH.FIT.F'
      INTEGER MNRA,MNCA
      PARAMETER (MNRA=25,MNCA=51)
      INTEGER I,FLAG1,FLAG2,MXMN,OP
      INTEGER*4 K
      INTEGER M,N,NV11
      CHARACTER TRST(MNRA)
      REAL A(MNRA,MNCA),AT(MNCA,MNRA),R(MNRA),C(MNCA),X(MNCA),TIME0,TF
      REAL TIME1,LAMDA(MNRA),V(MNCA),DW(MNCA),MG,R,E1,E2,E3,E4,E5,PCX
      REAL COTA,TIEMPO1,TIEMPO2,TIEMPO3,T1,T2,TIME2,TIME3,TIME4,TIME5
      LOGICAL ASK1,ASK2,ASK3,OP1
      REAL SECNDS
      ASK1=.TRUE.
      CALL TTULO
      DO WHILE (ASK1)
        CALL LECTAM(M,N)
        CALL INICIA(M,N,A,TRST)
        CALL CHARACTER(M,NV11,MXMN,TRST)
        CALL LECTURAM(N,A,B,C)
        OP1=.TRUE.
        DO WHILE (OP1)
          CALL IMPRESO1(M,N,A,B,C,MXMN,TRST,OP1,1)
          IF (OP1) CALL CORREGIR(M,N,A,B,C,MXMN,TRST)
        END DO
        TIME0=SECNDS()
        CALL FORMIST(M,N,NV11,MXMN,A,C,TRST)
        TIME1=SECNDS()
        TIEMPO1=TIME1-TIME0
        ASK2=.TRUE.
        DO WHILE (ASK2)
          CALL PARAMET(MG,R,COTA,E1,E2,E3,E4,E5)
          TIME2=SECNDS()
          CALL OSHEIM(N,A,B,C,AT,MG)
          TIME3=SECNDS()
          TIEMPO2=TIME3-TIME2
          CALL IMPRESO1(M,N+1,A,B,C,MXMN,TRST,OP1,2)
          ASK2=.TRUE.
          DO WHILE (ASK3)
            CALL SELMET(OP,OP1,N,N)
            T2=0
            TIME4=SECNDS()
            FLAG1=0
            FLAG2=0
            K=0
            DO WHILE (FLAG1.EQ.0)
              IF (OP1) CALL IMPRESO2(N,N,K,FF,E1)
              IF (OP.NE.1) THEN
                CALL CVED(M,N,A,AT,C,LAMDA,N,OP,T1)
                CALL CVCR(M,N,C,AT,LAMDA,V)
              ELSE
                CALL VEDVCRDW(M,N,AT,C,LAMDA,V,DW,T1)
              ENDIF
              T2=T1+T2
              CALL PARCOM(N,A,B,C,V,LAMDA,FLAG1,PCX,E1,E2,E3,E4,FF)
              IF (FLAG1.EQ.0) THEN
                CALL DMNAOC(N,N,V,DW,E4,E5,FLAG1,FLAG2,OP,PCX,COTA)
            
```

```

IF (FLAG1.EQ.0) THEN
  CALL I.PC8M(N,DW,X,R,E5)
  K=K+1
ENDIF
ENDIF
END DO
TIMES=SECNDS1)
TIEMPO3=(TIME2-TIME4)+TIEMPO2+TIEMPO1
IF (OP1) CALL IMPRESO2(N,X,K,FP,E1)
CALL IMPRESO3(M,N,X,LAMDA,PC,K,K,FLAG2,E4,TIEMPO3,OP,MAXMN,V,
OP1,T2)
CALL CONTINUA(ASK1,ASK2,ASK3,FP)
END DO
END DO
END DO
CALL AUTORES
STOP
END
SUBROUTINE TITULO
  CALL CLEARSCREEN($G$CLEARSCREEN)
  PRINT(9,'127X.A /// 40X.A /// 27X.A,F')
  * 'INSTRUMENTACION COMPUTACIONAL:DEB.'
  * 'METODO DE ESCALA AFIN PRIMAL'
  PRINT(9,'127X.A (,)'PRESIONE ENTER PARA CONTINUAR'
  READ(*,*)
  RETURN
END
SUBROUTINE LECTAM(M1,N1)
  INTEGER M1,N1
  REAL M2,N2
  CHARACTER*60 F1
  CALL CLEARSCREEN($G$CLEARSCREEN)
  M2=0
  N2=0
  F1='0/ 14X.A / 14X.A // '
  PRINT F1,CAPTURA DEL NUMERO DE VARIABLES Y RESTRICCIONES QUE
  * 'TIENE EL PROBLEMA DE PROGRAMACION LINEAL A SOLUCIONAR'
  DO WHILE ((N2.GT.25).OR.(N2.LT.2))
    PRINT '(M1)/16X.A,F'
    * ' EL NUMERO DE VARIABLES QUE TIENE EL PPL ES ?'
    READ*,N2
    N1=INT(N2)
    IF (N1.NE.N2) N2=0
  END DO
  * '
  DO WHILE ((M2.GT.25).OR.(M2.LT.1))
    PRINT '(M1)/16X.A,F'
    * ' EL NUMERO DE RESTRICCIONES QUE TIENE EL PPL ES ?'
    READ*,M2
    M1=INT(M2)
    IF (M1.NE.M2) M2=0
  END DO
  RETURN
END
SUBROUTINE INICIA(M1,N1,A1,TREST)
  INTEGER M1,N1
  CHARACTER TREST(M1)
  REAL A1(M1,N1)
  INTEGER I,J
  DO I=1,N1
    DO J=1,N1
      A1(I,J)=0
    END DO
  END DO
  DO I=1,M1
    TREST(I)=0
  END DO
END DO

```

```

RETURN
END
SUBROUTINE CARACTER(MI,NVI,LMXMN,TREST)
INTEGER*2 MMAXN
INTEGER M1,NV1
CHARACTER*TRIST(M1)
INTEGER I
CHARACTER*2 ASK
ASK=0
DO WHILE ((ASK.NE.'1').AND.(ASK.NE.'2'))
  CALL CLEARSCREEN($GCLARS$CREEN)
  PRINT '/ 16%.AF,'
  * 'CAPTURA DE LAS CARACTERISTICAS DEL PPL.A RESOLVER'
  PRINT('/ // 19%.AF,LA FUNCION OBJETIVO DEL PPL SE TIENE QUE?'
  PRINT('/ // 29%.AF,MINIMIZAR -> (1)'
  PRINT('/ // 29%.AF,MAXIMIZAR -> (2)'
  PRINT('/ // 29%.A.V.'SU SELECCION ES ?:'
  READ(A)ASK
END DO
IF (ASK.EQ.'1') MMAXN=1
IF (ASK.EQ.'2') MMAXN=-1
PRINT '/ 16%.AF,'
CALL CLEARSCREEN($GCLARS$CREEN)
* 'CAPTURA DE LAS CARACTERISTICAS DEL PPL.A
RESOLVER' NV1=0
PRINT('/ // 21%.A / 21%.AF,'
* 'PARA CADA UNA DE LAS RESTRICCIONES DEL PPL,'
* 'INDICAR QUE TIPO DE RESTRICCION ES?'
PRINT('/ // 29%.AF, MENOR O IGUAL A -> (1)'
PRINT('/ 29%.AF, MAYOR O IGUAL A -> (2)'
PRINT('/ 29%.A // PPL.A -> (1)' DO I=1,M1
  DO WHILE ((TREST(I,NE.'C').AND.(TREST(I,NE.'L').AND.
    (TREST(I,NE.'E').AND.(TREST(I,NE.'g').AND.
    (TREST(I,NE.'I').AND.(TREST(I,NE.'g'))
    PRINT('/ 25%.A,1,12,1%.A.V,'
  * 'EL TIPO DE LA RESTRICCION ES:'
  READ(A)TREST(I)
  END DO
  IF ((TREST(I).EQ.'1').OR.(TREST(I).EQ.'2').OR.
    (TREST(I).EQ.'3').OR.(TREST(I).EQ.'4')) NV1=NV1+1
END DO
RETURN
END
SUBROUTINE LECTURA(M,N,A,B,C)
INTEGER M,N
REAL A(M,N),B(M),C(N)
INTEGER I,J
LOGICAL ASK
CHARACTER*2 R1
CALL CLEARSCREEN($GCLARS$CREEN)
PRINT('/ // 16%.A // /'
* 'CAPTURA DE LOS COEFICIENTES DE LA FUNCION OBJETIVO'
DO J=1,N
  IF (J.LE.10) THEN
    PRINT('/ 5%.A,11,1%.A.V,'
    * 'EL COEFICIENTE DE LA VARIABLE N,J EN LA F.O. ES:'
  ELSE
    PRINT('/ 5%.A,12,1%.A.V,'
    * 'EL COEFICIENTE DE LA VARIABLE N,J EN LA F.O. ES:'
  ENDIF
  READ*(C)(J)
END DO
R1=0
DO WHILE ((R1.NE.'1').AND.(R1.NE.'2'))
  CALL CLEARSCREEN($GCLARS$CREEN)
  PRINT('/ // 14%.AF,'

```

```

+ 'FORMA DE CAPTURA DE LA MATRIZ DE RESTRICCIONES'
PRINT// 17%A//
+ 'LA MATRIZ DE RESTRICCIONES SE CAPTURARA COMO ?'
PRINT// 20%A// 'MATRIZ ESPARSA -> (1)'
PRINT// 20%A// 'MATRIZ NO ESPARSA -> (2)'
PRINT// 12%A \V,5U S/ELECCION ES: '
READ(A7,R1)
END DO
IF (R1.EQ.'1') THEN
DO I=1,M
ASK=.TRUE.
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT// 18%A // '
'CAPTURA DE LA MATRIZ ESPARSA DE RESTRICCIONES'
DO WHILE (ASK)
I=1
PRINT(10%,A,I2) 'EN LA RESTRICCION 'I'
PRINT(10%,A \V,
+ 'QUE VARIABLE TIENE COEFICIENTE DISTINTO DE CERO: '
DO WHILE ((L.T.0)OR.(L.GT.N))
READ(J)
END DO
IF (L.GT.0) THEN
IF (L.T.10) THEN
PRINT(10%,A,A,11,A,I2,A \V, 'EL COEFICIENTE DE LA,
+ 'VARIABLE N.I.' EN LA RESTRICCION 'I,' ES: '
ELSE
PRINT(10%,A,A,I2,A,I2,A \V, 'EL COEFICIENTE DE LA,
+ 'VARIABLE N.J.' EN LA RESTRICCION 'I,' ES: '
END IF
READ(A(L))
PRINT*,''
ELSE
ASK=.FALSE.
END IF
END DO
ELSE
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT// 16%A // '
'CAPTURA DE LA MATRIZ NO ESPARSA DE RESTRICCIONES'
DO I=1,M
DO J=1,N
IF (L.T.10) THEN
PRINT(11%,A,A,11,A,I2,A \V, 'EL COEFICIENTE DE LA, +VARIABLE N.I.'
EN LA RESTRICCION 'I,' ES: '
ELSE
PRINT(11%,A,A,I2,A,I2,A \V, 'EL COEFICIENTE DE LA, +
VARIABLE N.J.' EN LA RESTRICCION 'I,' ES: '
END IF
READ(A(I))
PRINT*,''
END DO
ENDIF
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT// 21%A // '
'CAPTURA DE LOS TERMINOS INDEPENDIENTES'
DO I=1,M
PRINT(10%,A,I2,A \V,
+ 'EL TERMINO INDEPENDIENTE DE LA RESTRICCION 'I,' ES: '
READ*(R(I))
END DO
RETURN
END

```

```

SUBROUTINE IMPRESO1(M1,N1,A1,B1,C1,MXMIN,TREST,OP1,OP2)
  INTEGER*2 MXMIN
  INTEGER M1,N1,OP2
  REAL A1(M1,N1),B1(M1),C1(N1)
  CHARACTER TREST(M1)
  LOGICAL OP1
  INTEGER I,J,K
  REAL E11
  PARAMETER (E11=10E-4)
  CHARACTER*2 R
  CALL CLEARSCREEN(1)GCLEARSCREEN)
  IF (OP2.EQ.1) THEN
    PRINT/ 15X,A //
  * 'EL PROBLEMA DE PROGRAMACION LINEAL PROPORCIONADO ES '
  ELSE
    PRINT/ 14X,A //
  * 'EL PPL DE LA M-GRANDE ASOCIADO AL PPL PROPORCIONADO ES '
  ENDF
  IF ((MXMIN.EQ.-1).AND.(OP2.EQ.1)) THEN
    IF (N1.GT.10) THEN
      PRINT/ 4X,A //,'MAX Z = '
    ELSE
      PRINT/ 10X,A //,'MAX Z = '
    ENDF
  ELSE
    IF (N1.GT.10) THEN
      PRINT/ 4X,A //,'MIN Z = '
    ELSE
      PRINT/ 10X,A //,'MIN Z = '
    ENDF
  ENDF
  K=0
  DO I=1,N1
    IF (ABS(C1(I)).GT.E11) THEN
      K=K+1
      IF ((C1(I).GT.0).AND.(I.NE.1).AND.(K.NE.0)) THEN PRINT(A //,' '
      ELSE
        IF (C1(I).LT.0) PRINT(A //,' '
      ENDF
      IF (I.LT.10) THEN
        IF (ABS(C1(I)).LT.1) THEN
          PRINT(A,F2.1,A1 //,'0',ABS(C1(I)),N1) ELSE
          IF (ABS(C1(I)).LT.10)
            * PRINT(F8.1,A1 //,'ABS(C1(I)),N1)
          ENDF
          IF ((ABS(C1(I)).GE.10).AND.(ABS(C1(I)).LT.100))
            * PRINT(F4.1,A1 //,'ABS(C1(I)),N1)
          IF (ABS(C1(I)).GE.100) .AND. (ABS(C1(I)).LT.1000) +
            * PRINT(F5.1,A1 //,'ABS(C1(I)),N1)
          IF ((ABS(C1(I)).GE.1000).AND.(ABS(C1(I)).LT.10000)) +
            * PRINT(F6.1,A1 //,'ABS(C1(I)),N1)
          IF (ABS(C1(I)).GE.10000).AND.(ABS(C1(I)).LT.10E5) +
            * PRINT(F8.1,A1 //,'ABS(C1(I)),N1)
          IF (ABS(C1(I)).GE.10E5)
            * PRINT(F13.1,A1 //,'ABS(C1(I)),N1)
        ELSE
          IF (ABS(C1(I)).LT.1) THEN
            PRINT(A,F2.1,A1 //,'0',ABS(C1(I)),N1) ELSE
            IF (ABS(C1(I)).LT.10)
              * PRINT(F8.1,A1 //,'ABS(C1(I)),N1)
            ENDF
            IF ((ABS(C1(I)).GE.10).AND.(ABS(C1(I)).LT.100))
              * PRINT(F4.1,A1 //,'ABS(C1(I)),N1)
            IF (ABS(C1(I)).GE.100).AND.(ABS(C1(I)).LT.1000) +
              * PRINT(F5.1,A1 //,'ABS(C1(I)),N1)

```

```

      IF ((ABS(C(I)).GE.1000).AND.(ABS(C(I)).LT.10000)) *
        PRINT('6.1.A.12 \',ABS(C(I)),'N')
      IF ((ABS(C(I)).GE.1000).AND.(ABS(C(I)).LT.10E5)) *
        PRINT('8.1.A.12 \',ABS(C(I)),'N')
      IF (ABS(C(I)).GE.10E5)
        PRINT('13.1.A.12 \',ABS(C(I)),'N')
      *
    ENDF
  END DO
  PRINT('// 12\A //Y.S. A'
  DO I=1,M1
    K=0
    IF (I.LT.10) THEN
      IF (N1.GT.24) THEN
        PRINT('4X,A,1,A,1X \',R,I,' '
      ELSE
        PRINT('10X,A,1,A,1X \',R,I,'
      : ' ENDF
    ELSE
      IF (N1.GT.24) THEN
        PRINT('4X,A,2,A,1X \',R,I,'
      ELSE
        PRINT('10X,A,2,A,1X \',R,I,'
      ENDF
    ENDF
  ENDIF
  DO J=1,N1
    IF (ABS(A(I,J)).GT.E1) THEN
      K=K+1
      IF (A(I,J).GT.0) THEN
        IF ((I.NE.1).AND.(K.NE.1)) PRINT('A \', ' *
      ELSE
        PRINT('A \', '
      ENDF
      IF (I.LT.10) THEN
        IF (ABS(A(I,J)).LT.1) THEN
          PRINT('A,F2.1,A,1 \',0,ABS(A(I,J)),'N')
        ELSE
          IF (ABS(A(I,J)).LT.10) PRINT('3.1,A,1 \',
            ABS(A(I,J)),'N')
        *
      ENDF
      IF (ABS(A(I,J)).GE.10).AND.(ABS(A(I,J)).LT.100)
        PRINT('4.1,A,1 \',ABS(A(I,J)),'N')
      *
      IF ((ABS(A(I,J)).GE.100).AND.(ABS(A(I,J)).LT.1000))
        PRINT('5.1,A,1 \',ABS(A(I,J)),'N')
      *
      IF (ABS(A(I,J)).GE.1000).AND.(ABS(A(I,J)).LT.10000)
        PRINT('6.1,A,1 \',ABS(A(I,J)),'N')
      *
      IF ((ABS(A(I,J)).GE.10000).AND.(ABS(A(I,J)).LT.10E5))
        PRINT('8.1,A,1 \',ABS(A(I,J)),'N')
      *
      ELSE
        IF (ABS(A(I,J)).LT.1) THEN
          PRINT('A,F2.1,A,12 \',0,ABS(A(I,J)),'N')
        ELSE
          IF (ABS(A(I,J)).LT.10) PRINT('3.1,A,12 \',
            ABS(A(I,J)),'N')
        *
      ENDF
      IF (ABS(A(I,J)).GE.10).AND.(ABS(A(I,J)).LT.100)
        PRINT('4.1,A,12 \',ABS(A(I,J)),'N')
      *
      IF ((ABS(A(I,J)).GE.100).AND.(ABS(A(I,J)).LT.1000))
        PRINT('5.1,A,12 \',ABS(A(I,J)),'N')
      *
      IF (ABS(A(I,J)).GE.1000).AND.(ABS(A(I,J)).LT.10000)
        PRINT('6.1,A,12 \',ABS(A(I,J)),'N')
      *
      IF ((ABS(A(I,J)).GE.10000).AND.(ABS(A(I,J)).LT.10E5))
        PRINT('8.1,A,12 \',ABS(A(I,J)),'N')
      *
    ENDF
  ENDF
END DO

```

```

IF (OP2.EQ.1) THEN
  SELECT CASE (TRIST(1))
  REJECT CASE (TRIST(1))
  CASE (L)
    PRINT(A \Y) <= '
  CASE (O)
    PRINT(A \Y) >= '
  CASE (F)
    PRINT(A \Y) <= '
  CASE (g)
    PRINT(A \Y) >= '
  CASE DEFAULT
    PRINT(A \Y) = '
  END SELECT
ELSE
  PRINT(A \Y) = '
ENDIF
IF (B1(1),LT.0) PRINT(A \Y)
IF (ABS(B1(1)),LT.1) THEN PRINT(A,F2.1,F,0,ABS(B1(1)))
ELSE
  IF (ABS(B1(1)),LT.10) PRINT(F3.1,F,ABS(B1(1))) ENDIF
  IF ((ABS(B1(1)),GE.10).AND.(ABS(B1(1)),LT.100))
    PRINT(F4.1,F,ABS(B1(1)))
  IF ((ABS(B1(1)),GE.100).AND.(ABS(B1(1)),LT.1000))
    PRINT(F5.1,F,ABS(B1(1)))
  IF ((ABS(B1(1)),GE.1000).AND.(ABS(B1(1)),LT.10000))
    PRINT(F6.1,F,ABS(B1(1)))
  IF ((ABS(B1(1)),GE.10000).AND.(ABS(B1(1)),LT.100000))
    PRINT(F8.1,F,ABS(B1(1)))
END DO
IF (OP2.EQ.1) THEN
  R=0
  DO WHILE ((R.NE.'T').AND.(R.NE.'L').AND.(R.NE.'P').AND. + (R.NE.'F'))
    IF ((M1.GE.15).OR.(N1.GE.25)) THEN
      PRINT(/ 16% A \Y,
+      'DESEA EFECTUAR ALGUNA CORRECCION AL PPL (T/F) ? '
      ELSE
        PRINT(/ / / 16% A \Y,
+      'DESEA EFECTUAR ALGUNA CORRECCION AL PPL (T/F) ? '
      ENDF
      READ(A,F,R)
    END DO
    IF ((R.EQ.'L').OR.(R.EQ.'F')) OP1 = TRUE
    IF ((R.EQ.'T').OR.(R.EQ.'P')) OP1 = FALSE
  ELSE
    IF ((M1.GE.15).OR.(N1.GE.25)) THEN
      PRINT(/ 27% A \Y,PRISIONE ENTER PARA CONTINUAR' ELSE
      PRINT(/ / / 27% A \Y,PRISIONE ENTER PARA CONTINUAR' ENDF
      READ(*,*)
    ENDF
    RETURN
  END
  SUBROUTINE CORREGIR(M,N,A,B,C,MM,NM,TRIST)
  INTEGER*2 MM,N
  REAL A(M,N),B(N),C(N)
  CHARACTER TRIST(M)
  REAL I1,I1
  INTEGER IJ
  LOGICAL OPCION
  CHARACTER*2 R,R1
  CHARACTER R2
  OPCION = TRUE
  DO WHILE (OPCION)
    R=0
    DO WHILE ((R.NE.'1').AND.(R.NE.'2').AND.(R.NE.'3').AND.
+      (R.NE.'4').AND.(R.NE.'5').AND.(R.NE.'6'))
      CALL CLEARSCREEN($GCLEARSCREEN)

```



```

PRINT// 12X,AJ,
* 'CORRECCION DE LOS DATOS DE ENTRADA DEL PPL PROPORCIONADO'
PRINT// 16X,AJ,
+ 'TIPO DE PROBLEMA (MAXIMIZAR/MINIMIZAR) -->(1)'
PRINT(16X,AJ,
+ 'COEFICIENTE EN LA FUNCION OBJETIVO -->(2)'
PRINT(16X,AJ,
+ 'COEFICIENTE DE LA MATRIZ DE RESTRICCIONES -->(3)'
PRINT(16X,AJ,
+ 'COEFICIENTE DEL VECTOR DEL LADO DERECHO -->(4)'
PRINT(16X,AJ,
+ 'TIPO DE RESTRICCION (L<=, G=>, E=) -->(5)'
PRINT(16X,AJ,
+ 'SALIR -->(6)'
PRINT(X//1,22X,A \), 'SU SELECCION ES : '
READ(AJ,R
END DO
R2=R
SELECT CASE(R2)
CASE (1)
R1=0
PRINT// 24X,AJ, 'CORRECCION DEL TIPO DE PROBLEMA'
PRINT// 12X,A \), 'TIPO DE PROBLEMA ACTUAL : '
IF (MNMN.EQ.1)
PRINT(AJ, 'MINIMIZAR' IF
(MNMN.EQ.-1)
PRINT(AJ, 'MAXIMIZAR' DO
WHILE
((R1.NE.1) .AND. (R1.NE.2))
PRINT(12X,A \),
+ 'INGRESE (1) PARA MINIMIZACION, (2) PARA MAXIMIZACION : '
READ(AJ,R1
END DO
IF (R1.EQ.1) MNMN=1
IF (R1.EQ.2) MNMN=-1
CASE (2)
I1=0
PRINT// 21X,AJ, 'CORRECCION DE COEFICIENTE EN LA F.O.' DO WHILE
((I1.LE.0) .OR. (I1.GT.N))
PRINT// 18X,A \),
+ 'INGRESE EL NUMERO DE LA VARIABLE DE DECISION : '
READ(I1)
I=INT(I1)
IF (I.NE.I1) I1=0
END DO
PRINT(15X,AJ, I2,
+ 'EL COEFICIENTE ACTUAL DE ESTA VARIABLE ES : ', C(I)
PRINT(15X,A \), 'INGRESE EL NUEVO COEFICIENTE : ' READ*(C(I)
CASE (3)
I1=0
PRINT// 12X,AJ, 'CORRECCION DE COEFICIENTE EN LA',
+ 'MATRIZ DE RESTRICCIONES'
DO WHILE ((I1.LE.0) .OR. (I1.GT.M))
PRINT// 15X,A \), 'INGRESE EL NUMERO DE LA RESTRICCION : ' READ*(I1)
I=INT(I1)
IF (I.NE.I1) I1=0
END DO
I1=0
DO WHILE ((I1.LE.0) .OR. (I1.GT.N))
PRINT(15X,A \),
+ 'INGRESE EL NUMERO DE LA VARIABLE DE DECISION : '
READ*(I1)
I=INT(I1)
IF (I.NE.I1) I1=0
END DO
PRINT(15X,AJ, I2,

```

```

+      'EL COEFICIENTE ACTUAL DE ESTA VARIABLE ES :',A(I))
      PRINT(19,A \,,'INGRESE EL NUEVO COEFICIENTE :')
      READ*,A(I)
      CASE ('#')
        I1=0
        PRINT(// 21\,A \,,'CORRECCION DE TERMINO INDEPENDIENTE')
        DO WHILE ((I1.LE.0).OR.(I1.GT.M))
          PRINT(// 19\,A \,,'INGRESE EL NUMERO DE LA RESTRICCION :') READ*,I1
          I=INT(I1)
          IF (I.LE.I1) I1=0
          END DO
          PRINT(19,A \,,'F7.2')
+      'EL TERMINO INDEPENDIENTE ACTUAL ES :',B(I)
      PRINT(19,A \,,'INGRESE EL NUEVO TERMINO INDEPENDIENTE :') READ*,B(I)
      CASE ('#')
        I1=0
        PRINT(// 22\,A \,,'CORRECCION DEL TIPO DE RESTRICCION')
        DO WHILE ((I1.LE.0).OR.(I1.GT.M))
          PRINT(// 12\,A \,,'INGRESE EL NUMERO DE LA RESTRICCION :') READ*,I1
          I=INT(I1)
          IF (I.LE.I1) I1=0
          END DO
          PRINT(12\,A \,,'A')
+      'EL TIPO ACTUAL DE LA RESTRICCION ES :',TREST(I)
          TREST(I)=0
          DO WHILE ((TREST(I).NE.'C').AND.(TREST(I).NE.'L').AND.
+      (TREST(I).NE.'F').AND.(TREST(I).NE.'g').AND.
+      (TREST(I).NE.'I').AND.(TREST(I).NE.'j'))
            PRINT(12\,A \,,'')
+      'INGRESE EL NUEVO TIPO DE RESTRICCION (L=<=, G=>=, E=) :'
            READ(A \,TREST(I)
            END DO
            END SELECT
            IF (R.NE.'g') THEN
              R1=0
              DO WHILE ((R1.NE.'I').AND.(R1.NE.'F').AND.(R1.NE.'j').AND.
+      (R1.NE.'g'))
                PRINT(// 19\,A \,,'DISEA EFECTUAR OTRA CORRECCION (T/F) ?') READ(A \,R1
                END DO
                IF ((R1.EQ.'I').OR.(R1.EQ.'F')).OPCION=.TRUE.
                IF ((R1.EQ.'I').OR.(R1.EQ.'F')).OPCION=.FALSE.
                ELSE
                  OPCION=.FALSE.
                ENDIF
              END DO
            RETURN
          END
          SUBROUTINE FORMHST(M1,N1,NV1,M,N,N,A1,C1,TREST)
            INTEGER*2 M,N,N1
            INTEGER M1,N1,NV1
            CHARACTER TREST(M1)
            REAL A1(M1,N1+NV1),C1(N1+NV1)
            INTEGER I,J
            DO I=1,M1
              DO J=N1+1,N1+NV1
                IF (LQ I-1,N1) THEN
                  SELECT CASE (TREST(I))
                    CASE ('L')
                      A1(I,J)=1
                    CASE ('G')
                      A1(I,J)=-1
                    CASE ('F')
                      A1(I,J)=1
                    CASE ('g')
                      A1(I,J)=-1
                    CASE DEFAULT

```

```

                                A1(I,J)=0
END SELECT
      ELSE
A1(I,J)=0
ENDIF
      END DO
      END DO
      IF (MAXN.EQ.-1) CALL SSCAL(N1,-1.0,C1,1)
      DO I=N1+1,N1+NV11
        C1(I)=0
      END DO
      N1=N1+NV11
      RETURN
      END
      SUBROUTINE PARAMET(MG,R,COTA,E1,E2,E3,E4,E5)
      REAL MG,R,COTA,E1,E2,E3,E4,E5
      LOGICAL DEC1
      CHARACTER*2 RI,R2
      RI='0'
      DO WHILE ((RI.NE.'1').AND.(RI.NE.'2'))
        CALL CLEARSCREEN($GCLEARSCREEN)
        PRINT(/ 20X'A / 20X'A //,
+ 'CAPTURA DE LOS PARAMETROS DE EJECUCION DEL,
+ ' METODO DE ESCALA AFIN PRIMAL'
        PRINT(/ 21X'A /// 19X'A /// 18X'A / 16X'A //,
+ 'PARA EJECUTAR EL PROGRAMA, USTED DESEA ',
+ ' PROPORCIONAR LOS VALORES DE LOS PARAMETROS --> (1)',
+ ' EL PROGRAMA SE EJECUTE CON LOS VALORES QUE,
+ ' SE TIENEN ASIGNADOS          --> (2) PRINT('0',32X'A \/, 'SU SELECCION ES : '
        READ(A),RI
      END DO
      IF (RI.EQ.'1') THEN
        DEC1=.TRUE.
        DO WHILE (DEC1)
          CALL CLEARSCREEN($GCLEARSCREEN)
          PRINT(/ 19X'A / 19X'A //,
+ 'CAPTURA DE LOS PARAMETROS DE EJECUCION DEL,
+ ' METODO DE ESCALA AFIN PRIMAL'
          MG=0
          DO WHILE (MG.LE.0)
            PRINT(/ 18X'A \/, 'EL VALOR DE LA M-GRANDE ES : '
            READ(A),MG
          END DO
          R=0
          DO WHILE ((R.LE.0).OR.(R.GT.1))
            PRINT(/ 18X'A \/, 'EL FACTOR DEL TAMAÑO DE PASO ES : ' READ(A),R
          END DO
          COTA=0
          DO WHILE (COTA.LE.0)
            PRINT(/ 18X'A \/, 'LA COTA SOBRE LA FUNCION OBJETIVO ES : ' READ(A),COTA
          END DO
          E1=1
          DO WHILE ((E1.LT.0).OR.(E1.GE.1))
            PRINT(/ 18X'A \/, 'LA COTA PARA LA FACTIBILIDAD PRIMAL ES : '
            READ(A),E1
          END DO
          E2=1
          DO WHILE ((E2.LT.0).OR.(E2.GE.1))
            PRINT(/ 18X'A \/, 'LA COTA PARA LA FACTIBILIDAD DUAL ES : ' READ(A),E2
          END DO
          E3=1
          DO WHILE ((E3.LT.0).OR.(E3.GE.1))
            PRINT(/ 18X'A \/,
+ 'LA COTA PARA HOLGURAS COMPLEMENTARIAS ES : '

```

```

      READ*,E3
      END DO
      E4=1
      DO WHILE ((E4.LT.0).OR.(E4.GE.1))
        PRINT// '18X,A',F,
        * 'LA COTA SOBRE LA VARIABLE ARTIFICIAL ES:'
      READ*,E4
      END DO
      E5=1
      DO WHILE ((E5.LT.0).OR.(E5.GE.1))
        PRINT// '18X,A',F,'LA COTA SOBRE EL VECTOR DIRECCION ES:' READ*,E5
      END DO
      R2=0
      DO WHILE ((R2.NE.'1').AND.(R2.NE.'0').AND.(R2.NE.'F').AND.
        * (R2.NE.'?'))
        PRINT// '15X,A',F,
        * 'REQUIERE VOLVER A CAPTURAR LOS PARAMETROS (T/F)?:'
      READ*(A,F,R2)
      END DO
      IF ((R2.EQ.'1').OR.(R2.EQ.'0')) DEC1=.TRUE.
      IF ((R2.EQ.'F').OR.(R2.EQ.'?')) DEC1=.FALSE.
      END DO
    ELSE
      MG=10E3
      R=0.9999
      COTA=10E5
      E1=10E-4
      E2=10E-4
      E3=10E-4
      E4=10E-8
      E5=10E-4
    ENDIF
    RETURN
  END
  SUBROUTINE CSIFRM1,N1,A1,B1,C1,A11,MG1)
    INTEGER M1,N1
    REAL A1(M1,N1+1),A11(N1+1,M1),B1(M1),C1(N1+1),MG1
    INTEGER L,M2,N2
    PARAMETER (M2=25,N2=51)
    REAL E(N2),TEM(M2)
    CH(N1+1)=MG1
    CALL SSSET(N1,L,O,N1)
    CALL MURFV(M1,N1,A1,M1,N1,E1,M1,TEM)
    DO I=1,M1
      A11(N1+1)=B1(I)/TEM(I)
    END DO
    CALL TRNRRM1,N1+1,A1,M1,N1+1,M1,AT1,N1+1)
    RETURN
  END
  SUBROUTINE SELMET(OP,OP1,N1,X1)
    INTEGER N1
    REAL X1(N1+1)
    LOGICAL OP1
    INTEGER*2 OP
    CHARACTER*2 R
    CALL SSSET(N1+1,L,O,N1)
    R=0
    DO WHILE ((R.NE.'1').AND.(R.NE.'2').AND.(R.NE.'3').AND.(R.NE.'4')) CALL
      CLEARSCREEN//CALL CLEARSCREEN
      PRINT// '17X,A',F,
      * 'MÉTODOS DE FACTORIZACION MATRICIAL QUE PUEDEN SER,
      * 'UTILIZADOS EN LA OBTENCION DEL VECTOR DIRECCION:'
      PRINT// '25X,A',FACTORIZACION U*DU,      -> (1) PRINT//
      25X,A',FACTORIZACION DE CHOLSKY -> (2) PRINT//
      25X,A',FACTORIZACION QR,      -> (3) PRINT//
      25X,A',MINIMOS CUADRADOS,      -> (4) PRINT//0,1),20X,A',F,

```

```

* 'EL METODO SELECCIONADO (1,2,3 o 4) ES ? ' READ(A7,R
END DO
IF (R.EQ.'1') OP=1
IF (R.EQ.'2') OP=2
IF (R.EQ.'3') OP=3
IF (R.EQ.'4') OP=4
DO WHILE (R.NE.'1').AND.(R.NE.'2').AND.(R.NE.'3').AND.(R.NE.'4')
PRINT// '16X, A / 16X, A \ /
* 'DISEÑAR CADA SOLUCION INTERIOR OBTENIDA EN CADA'.
* 'ITERACION DEL METODO (I,F) ? '
  READ(A7,R
  END DO
  IF ((R.EQ.'1')OR.(R.EQ.'2')) OP1=.TRUE.
  IF ((R.EQ.'3')OR.(R.EQ.'4')) OP1=.FALSE.
  RETURN
  END
  SUBROUTINE CVED(M1,N1,AL,AT1,C1,L,N1,OP,T1)
  INTEGER M1,N1
  INTEGER 2 OF
  REAL A1(M1,N1+1),AT1(N1+1,M1),C1(N1+1),L,M1,N1+1,T1
  INTEGER LL,KBASIS,M2,N2
  PARAMETER (M2=25,N2=51)
  REAL AD2(N2,N2),AD2AT(M2,M2),AD2c(M2),RES(N2),N2,SECNDS,T11,T12,C
  DO J=1,N1+1
    N2=N1(I)*X1(I)
    DO I=1,M1
      AD2(I,J)=A1(I,J)*N2
    END DO
  END DO
  CALL MURRR(M1,N1+1,AD2,M2,N1+1,M1,AT1,N1+1,M1,M1,AD2AT,M2)
  CALL MURRV(M1,N1+1,AD2,M2,N1+1,C1,L,M1,AD2c)
  SELECT CASE (OP)
    CASE (1)
      T11=SECNDS()
      CALL LLSFM1,AD2AT,M2,AD2c,L
      T12=SECNDS()
    CASE (2)
      T11=SECNDS()
      CALL LSADR(M1,AD2AT,M2,AD2c,L)
      T12=SECNDS()
    CASE (3)
      T11=SECNDS()
      CALL LSQRR(N1,M1,AD2AT,M2,AD2c,0,L,RES,KBASIS)
      T12=SECNDS()
  END SELECT
  T1=T12-T11
  RETURN
  END
  SUBROUTINE CVCR(M1,N1,C1,AT1,LV1)
  INTEGER M1,N1
  REAL C1(N1+1),AT1(N1+1,M1),L,M1),V1(N1+1)
  INTEGER LN2
  PARAMETER (N2=51)
  REAL AT1(LN2)
  CALL MURRV(N1+1,M1,AT1,N1+1,M1,L,N1+1,AT1)
  DO I=1,N1+1
    V1(I)=C1(I)-AT1(I)
  END DO
  RETURN
  END
  SUBROUTINE VEDVCRDW(M1,N1,AT1,C1,X1,LV1,DW1,T1)
  INTEGER M1,N1
  REAL AT1(N1+1,M1),C1(N1+1),X1(N1+1),L,M1),V1(N1+1),DW1(N1+1),T1
  INTEGER LL,KBASIS,M2,N2
  PARAMETER (M2=25,N2=51)
  REAL DAT(N2,M2),Dc(N2),SECNDS,T11,T12

```

```

CALL SHPROD(N1+1,X1,J,C1,I,DG,1)
DO I=1,N1+1
  DO J=1,M1
    DAT(I)=X1(I)*AT(I,J)
  END DO
END DO
T11=SECDNS0)
CALL LSQRR(N1+1,M1,DAT,N2,DG,0,L,DW1,KIIASIS)
T12=SECDNS0)
T1=T12-T11
DO I=1,N1+1
  V1(I)=DW1(I)/X1(I)
END DO
RETURN
END
SUBROUTINE PARQ(N1,N1A1,N1B1,C1,V1,L,FLAG1,PCX,E1,E2,E3,E4,FP)
INTEGER M1,N1
INTEGER*2 FLAG1
REAL A1(M1,N1+1),X1(N1+1),B1(M1),V1(N1+1),L(M1),C1(N1+1) REAL
E1,E2,E3,E4,FP,PCX
INTEGER M2,N2,FLAG,LIK
PARAMETER (N2=25,N2=51)
REAL AX(M2),TEM1(N2),TEM2(N2),NAX,NB,NT1,NT2,NT3,FD,HC,PBL
FLAG=0
I=ISMIN(N1+1,X1,1)
IF (X1(I).GE.0) THEN
  CALL MURRV(M1,N1+1,A1,M1,N1+1,X1,L,M1,AX)
  CALL SASPY(M1,-1.0,B1,LAN,1)
  NAX=SNRM2(M1,AX,1)
  NB=SNRM2(M1,B1,1)
  FP=NAX/(NB+1)
  IF (FP.GT.E1) FLAG=1
ELSE
  FLAG=1
ENDIF
IF (FLAG.EQ.0) THEN
  I=ISMIN(N1+1,V1,1)
  IF (V1(I).LT.0) THEN
    K=0
    DO J=1,N1+1
      K=K+1
      IF (V1(J).LT.0) THEN
        TEM1(K)=V1(I)
        TEM2(K)=C1(I)
      ENDIF
    END DO
    NT1=SNRM2(K,TEM1,1)
    NT2=SNRM2(K,TEM2,1)
    FD=NT1/(NT2+1)
    IF (FD.GT.E2) FLAG=1
  ENDIF
ENDIF
IF (N1*(N1+1)*E.E4).OR.(FLAG.EQ.0) PCX=SDOT(N1+1,C1,1,X1,1) IF (FLAG.EQ.0)
THEN
  PBL=SDOT(M1,B1,1,L,1)
  NTB=SNRM2(N1+1,PCX,1)
  HC=ABS(PCX/PBL)/(NTB+1)
  IF (HC.LT.E3) FLAG=1
ENDIF
RETURN
END
SUBROUTINE DMNAOC(N1,N1V1,DW1,E4,E5,FLAG1,FLAG2,OP,PCX,COTA)
INTEGER N1
INTEGER*2 FLAG1,FLAG2,OP
REAL X1(N1+1),V1(N1+1),DW1(N1+1),E4,E5,PCX,COTA
INTEGER LI

```

```

REAL NDW
IF (OP.NE.4) THEN
  CALL SHPROD(N1+1,N1,1,V1,1,DW1,1)
ENDIF
NDW=SNRN2(N1+1,DW1,1)
IF (NDW.LE.E5) THEN
  I=SMIN(N1+1,DW1,1)
  IF (DW1(I),LT.0) THEN
    FLAG1=1
    FLAG2=2
  ELSE
    FLAG1=1
  ENDIF
ELSE
  I=SMIN(N1+1,DW1,1)
  I=SMAX(N1+1,DW1,1)
  IF ((DW1(I),GT.0).AND.(DW1(I),LT.0)) THEN
    FLAG1=1
    FLAG2=2
  ENDIF
ENDIF
IF (X1(N1+1),LE.E4) THEN
  IF (ABS(PCN),GT.COTA) THEN
    FLAG1=1
    FLAG2=2
  ENDIF
ENDIF
RETURN
END
SUBROUTINE LPOS6(N1,DW1,X1,R,E5)
INTEGER N1
REAL DW1(N1+1),X1(N1+1),E5,R
INTEGER N2,T,T1
PARAMETER(N2=51)
REAL ALFA,TEM(N2)
T=0
DO T=1,N1+1
  IF (DW1(T),GT.E5) THEN
    T=T+1
    TEM(T)=R/DW1(T)
  ENDIF
END DO
IF (T.GT.0) THEN
  T1=ISSIN(T,TEM,1)
  ALFA=TEM(T1)
ELSE
  ALFA=R
ENDIF
CALL SHPROD(N1+1,N1,1,DW1,1,TEM,1)
CALL SANFY(N1+1,ALFA,TEM,1,N1,1)
END
SUBROUTINE IMPRESO2(N,X,K,FP,E1)
INTEGER N,K
REAL X(N+1),FP,E1
INTEGER I,(2),N1,L
REAL X(2)
CALL CLEARSCREEN(5,CLEARSCREEN)
PRINT/ / 13X,A12A //Y:EN LA ITERACION *K
* LA SOLUCION INTERIOR OBTENIDA ES *
J=0
IF (MOD(N,2),EQ.0) THEN
  N1=N
ELSE
  N1=N+1
ENDIF

```

```

L=0
DO I=1,N1
  L=L+1
  I=I+1
  II(I)=I
  X2(I)=X(I)
  IF (I.EQ.2) THEN
    IF (I.LT.10) THEN PRINT(23%,A,II,A,F10.5,9%,A,II,A,F10.5),
      * 'X,(I-1)' =',X2(I-1);X,II(I)' =',X2(I)
    ELSE
      PRINT(23%,A,II,A,F10.5,8%,A,II,A,F10.5),
      * 'X,(I-1)' =',X2(I-1);X,II(I)' =',X2(I)
    ENDIF
  ENDIF
  IF (I.EQ.30).AND.(I.NE.N1) THEN
    L=0
    PRINT(// 25%,A //,'PRESIONE ENTER PARA CONTINUAR.'
    READ*,)
    CALL CLEARSCREEN(56,CLEARSCREEN)
    PRINT(// 13%,A,II,A //,'EN LA ITERACION ',K
    * ' LA SOLUCION INTERIOR OBTENIDA ES.'
  ENDIF
END DO
IF (MOD(N,2).EQ.0) THEN
  IF (N*.1.LT.10) THEN PRINT(23%,A,II,A,F10.5,'N,N+1.' =',X(N+1)
  ELSE
    PRINT(23%,A,II,A,F10.5,'N,N+1.' =',X(N+1)
  ENDIF
ENDIF
PRINT(// 7%,A,F5.4),
* 'EN ESTE PUNTO EL ERROR RELATIVO DE FACTIBILIDAD PRIMAL ES:',E6,F10.1,LE,II)
THEN
  PRINT(7%,A),
  * 'ESTA SOLUCION ES FACTIBLE PARA LA COTA DE ERROR DADA.'
ELSE
  PRINT(7%,A),
  * 'ESTA SOLUCION ES INFACHTIBLE PARA LA COTA DE ERROR DADA.'
ENDIF
PRINT(// 25%,A //,'PRESIONE ENTER PARA CONTINUAR.'
READ*,)
RETURN
END
SUBROUTINE IMPRISO(M1,N1,N1,LL,PCN,K,FLAG2,E4,THEMPC03,OP,MN,MN,V1,OP1,T)
  INTEGER*2 FLAG,FLAG2,MN,MN,OP
  INTEGER*4 K
  INTEGER M1,N1
  REAL X1(N1+1),L1(M1),V1(N1),PCN,E4,THEMPC03,T
  LOGICAL OP1
  INTEGER I1,I1(2),I1,N2
  REAL X2(I),T1
  CALL CLEARSCREEN(56,CLEARSCREEN)
  FLAG=0
  IF (OP.FQ.2) THEN
    PRINT(// 7%,A //,'EL RESULTADO OBTENIDO CON EL METODO DE '
  ELSE
    PRINT(// 10%,A //,'EL RESULTADO OBTENIDO CON EL METODO DE '
  ENDIF
  ENDIF
  SELECT CASE (OP1)
    CASE (1)
      PRINT(A //,'FACTORIZACION U*DUYES.'
    CASE (2)
      PRINT(A //,'FACTORIZACION DE CHOLESKY ES.'
    CASE (3)
      PRINT(A //,'FACTORIZACION Q*R ES.'
    CASE (4)

```



```

PRINT(A // 'MINIMOS CUADRADOS ES :
END SELECT
T1=(T/TEMPO3)*100
IF (ABS(X(N1+1))-GT.E4) THEN
PRINT('7,1,23X.A //',EL PPL PROPORCIONADO ES INFECTIBLE
ELSE
IF (FLAC2.EQ.2) THEN
PRINT('1,1,9X.A //',
* 'LA F.O. DEL PPL PROPORCIONADO ES NO ACOTADA'
ELSE
FLAC=1
PRINT('4,X.A,12.A //',
* 'LA SOLUCION OPTIMA PRIMAL OBTENIDA EN 'K,
* 'ITERACIONES ES'
j=0
IF (MOD(N1,2).EQ.0) THEN
N2=N1
ELSE
N2=N1-1
ENDIF
I1=0
DO I=1,N2
I1=I+1
I=I+1
I1(I)=I
X20=X1(I)
IF (J.EQ.2) THEN
IF (I.LT.10) THEN PRINT('23X.A,11.A,F10.5,9X.A,11.A,F10.5',
* 'X,11(I-1)' = 'X20-1')X,11(I)' = 'X20)
ELSE
PRINT('23X.A,12.A,F10.5,8X.A,12.A,F10.5',
* 'X,11(I-1)' = 'X20-1')X,11(I)' = 'X20)
ENDIF
j=j+1
ENDIF
IF ((I.EQ.30).AND.(LN2.N1)) THEN
I1=0
PRINT(' // 25X.A //',TRESIONE ENTER PARA CONTINUAR'
READ(C,'')
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT('14,X.A,12.A //',
* 'LA SOLUCION OPTIMA PRIMAL OBTENIDA EN 'K,
* 'ITERACIONES ES'
ENDIF
END DO
IF (MOD(N1,2).NE.0) THEN
IF (N1.LT.10) THEN
PRINT('23X.A,11.A,F10.5//X,N1)' = 'X(N1)
ELSE
PRINT('23X.A,12.A,F10.5//X,N1)' = 'X(N1)
ENDIF
ENDIF
IF (MXMN.EQ.-1) PCX=PCN*MXMN
IF (N1.GE.24) THEN
PRINT(' // 13X.A,F12.5',
* 'EL VALOR OPTIMO DE LA FUNCION OBJETIVO ES :',PCX
ELSE
PRINT('8//,13X.A,F12.5',
* 'EL VALOR OPTIMO DE LA FUNCION OBJETIVO ES :',PCX
ENDIF
IF (.NOT.OP1) THEN
PRINT('13X.A,F6.3,A',
* 'TIEMPO DE CPU EN QUE SE OBTUVO EL OPTIMO :',TIEMPOS,
* 'SEG',
* PRINT('13X.A,F5.1,A',
* 'EN TIEMPO QUE OCURTO EL METODO DE FACTORIZACION :',TI

```

```

+      *%
+      ENDIF
+      PRINT(2//1,25% A \Y, PRESIONE ENTER PARA CONTINUAR'
+      READ(*,*)
+      CALL CLEARSCREEN($CLEARSCREEN)
+      J=0
+      PRINT(// 17% A,11% A,8% A //,
+      'VARIABLE/VALOR,COSTOS REDUCIDOS'
+      DO I=1,N1
+         I=I+1
+         IF (I.LT.10) THEN PRINT(20% A,I,4% N,F10.5,8% F10.5Y,
+         '%,LN1(I),V1(I)
+         ELSE
+         PRINT(20% A,I,2,8% F10.5,8% F10.5Y,
+         '%,LN1(I),V1(I)
+         ENDIF
+         IF (I.EQ.18) AND (4*LNEN1) THEN
+         PRINT(2//1,25% A \Y, PRESIONE ENTER PARA CONTINUAR'
+         READ(*,*)
+         CALL CLEARSCREEN($CLEARSCREEN)
+         J=0
+         PRINT(// 17% A,11% A,8% A //,
+         'VARIABLE/VALOR,COSTOS REDUCIDOS'
+         ENDIF
+         END DO
+         PRINT(2//1,25% A \Y, PRESIONE ENTER PARA CONTINUAR'
+         READ(*,*)
+         CALL CLEARSCREEN($CLEARSCREEN)
+         J=0
+         PRINT(// 26% A //, EL VECTOR ESTIMADO DUAL ES ?
+         DO I=1,N1
+            I=I+1
+            IF (I.LT.10) THEN PRINT(29% A,I,1,1,F10.5Y,LAMDA(I) =',L(I)
+            ELSE
+            PRINT(29% A,I,2,1,F10.5Y,LAMDA(I) =',L(I) ENDIF
+            IF (I.EQ.18) THEN
+            PRINT(2//1,26% A \Y, PRESIONE ENTER PARA CONTINUAR'
+            READ(*,*)
+            CALL CLEARSCREEN($CLEARSCREEN)
+            J=0
+            PRINT(// 26% A //, EL VECTOR ESTIMADO DUAL ES ?
+            ENDIF
+            END DO
+            PRINT(2//1,26% A \Y, PRESIONE ENTER PARA CONTINUAR'
+            READ(*,*)
+            END IF
+            END IF
+            IF (FLAG.NE.1) THEN
+            PRINT(8//1,26% A \Y, PRESIONE ENTER PARA CONTINUAR'
+            READ(*,*)
+            ENDIF
+            RETURN
+            END
+            SUBROUTINE CONTINUA(ASK1,ASK2,ASK3,F)
+            REAL F,F
+            LOGICAL ASK1,ASK2,ASK3
+            CHARACTER*2 R1,R2,R3
+            CALL CLEARSCREEN($CLEARSCREEN)
+            R1='0'
+            R2='0'
+            R3='0'
+            DO WHILE ((R3.NE.'1') AND (R3.NE.'F') AND (R3.NE.'P') AND (R3.NE.'#'))
+            PRINT(// 15% A // 15% A \Y,
+            * 'DESEA RESOLVER NUEVAMENTE EL P/L PROPORCIONADO PERO,
+            * 'CON OTRO METODO DE FACTORIZACION (T/F) ?'
+            READ(A,R3)

```

```

END DO
IF ((R3.EQ.'T').OR.(R3.EQ.'F')) ASK3=.TRUE
IF ((R3.EQ.'F').OR.(R3.EQ.'?')) ASK3=.FALSE
IF (.NOT.ASK3) THEN
  DO WHILE ((R2.NE.'T').AND.(R2.NE.'F').AND.(R2.NE.'?').AND.(R2.NE.'F'))
    PRINT('/ / 15X,A / 15X,A \)
    * 'DESEA RESOLVER NUEVAMENTE EL PPL PROPORCIONADO PERO,'
    * 'CON DISTINTOS PARAMETROS (T/F) ?:'
    READ(A),R2
  END DO
IF ((R2.EQ.'T').OR.(R2.EQ.'F')) ASK2=.TRUE
IF ((R2.EQ.'F').OR.(R2.EQ.'?')) ASK2=.FALSE
IF (.NOT.ASK2) THEN
  DO WHILE ((R1.NE.'T').AND.(R1.NE.'F').AND.(R1.NE.'?').AND.(R1.NE.'F'))
    PRINT('/ / 15X,A / 15X,A \)
    * 'DESEA RESOLVER OTRO PROBLEMA DE PROGRAMACION LINEAL,'
    * 'CON ESTA INSTRUMENTACION COMPUTACIONAL (T/F) ?:'
    READ(A),R1
  END DO
IF ((R1.EQ.'T').OR.(R1.EQ.'F')) ASK1=.TRUE
IF ((R1.EQ.'F').OR.(R1.EQ.'?')) ASK1=.FALSE
ENDIF
ENDIF
FP=0
RETURN
END
SUBROUTINE AUTORES
  CALL CLEARSCREEN(9)CLEARSCREEN)
  PRINT('/,25X,A // 35X,A // // 30X,A // 39X,A // 25X,A \)
  * 'INSTRUMENTACION COMPUTACIONAL, REALIZADA POR,'
  * 'SOFIA LOPEZ AGUILERA',Y', GUSTAVO ERNESTO FLORES VAZQUEZ'
  PRINT('/,1,25X,A \) 'PRESIONE ENTER PARA CONTINUAR' READ(*,*)
  CALL CLEARSCREEN(9)CLEARSCREEN)
  RETURN
END
REAL FUNCTION SECND(S)
  INTEGER*2 HOUR, MINUTE, SECOND, HUNDREDTH
  CALL GETTIME(HOUR,MINUTE,SECOND,HUNDREDTH)
  SECND = ((DBLE(HOUR)*3600.0)+(DBLE(MINUTE)* 60.0)+
  * DBLE(SECOND)+(DBLE(HUNDREDTH)/100.0))
END

```

APÉNDICE E

CÓDIGO CORRESPONDIENTE AL MÉTODO DE ESCALA AFÍN DUAL

```

INSTRUMENTACION COMPUTACIONAL DEL ALGORITMO DE ESCALA AFIN DUAL
UTILIZANDO IBM/360
INCLUDE 'GRAPH.FP'
INCLUDE 'GRAPH.FD'
INTEGER MNRA,MNCA
PARAMETER (MINRA=26,MNCA=30)
INTEGER*2 FLAG1,FLAG2,MXMIN,OP
INTEGER*4 K
INTEGER M,M1,N,NVH
CHARACTER TREST(MNRA)
REAL A(MNRA,MNCA),LAT(MNCA,MNRA),B(MNRA),C(MNCA),X(MNCA),FD
REAL Y(MNRA),Y1(MNRA),V(MNCA),V1(MNCA),D(V(MNCA)),DY(MNRA)
REAL COTA,MG,PBY,R,TEMPO1,TEMPO2,TEMPO3,TETA,TEMP1
REAL E1,E2,E3,E4,SECNDS,TIME0,TIME1,TIME2,TIME3,TIME4,TIME5
LOGICAL ASK1,ASK2,ASK3,OP1
ASK1=.TRUE.
CALL TTULO
DO WHILE (ASK1)
  CALL LECTAM(M,N)
  CALL INICIA(M,N,ATREST)
  CALL CARACTER(M,N,V1,MXMIN,TREST)
  CALL LECTURA(M,N,A,B,C)
  OP1=.TRUE.
  DO WHILE (OP1)
    CALL IMPRESO1(M,N,A,B,C,MXMIN,TREST,OP1)
    IF (OP1) CALL CORREGIR(M,N,A,B,C,MXMIN,TREST)
  END DO
  TIME0=SECNDS()
  CALL FORMET(M,N,NVH,MXMIN,A,C,TREST)
  TIME1=SECNDS()
  TEMPO1=TIME1-TIME0
  ASK2=.TRUE.
  DO WHILE (ASK2)
    CALL PARAMET(MG,R,COTA,E1,E2,E3,E4,TETA)
    TIME2=SECNDS()
    CALL OSIFID(M,N,A,B,C,AT,V1,Y,MG,TETA)
    TIME3=SECNDS()
    TEMPO2=TIME3-TIME2
    CALL IMPRESO4(M,N,AT,B,C)
    ASK3=.TRUE.
    DO WHILE (ASK3)
      CALL SELMET(M1,M3,N,B,OP1,V1,Y,VI)
      TIME4=SECNDS()
      FLAG1=0
      FLAG2=0
      K=0
      DO WHILE (FLAG1.EQ.0)
        IF (OP1) CALL IMPRESO2(M1,N,B,V,Y,E2,FD,K)
        TEMP1=Y(M1)
        IF (OP1.NE.4) THEN
          CALL DIRSMOVS(M3,N,AT,B,DV,DY,V,OP1)
          CALL CVEP(N,DV,V,N)
        ELSE
          CALL MINCUAD(M3,N,AT,B,DV,DY,V)
          CALL CVEP(N,DV,V,N)
        ENDIF
        CALL NAFOC(M1,N,B,DV,Y,COTA,E4,FLAG1,FLAG2,PBY)
        IF (FLAG1.EQ.0) THEN
          CALL PARCMB(M1,N,AT,B,C,X,V,Y,FLAG1,PBY,E1,E2,E3,FD)
          IF (FLAG1.EQ.0) THEN

```

```

CALL LPFORM(M3,N,DV,DY,V,Y,R,E4)
IF (Y(M+1).NE.0)
CALL CHECVART(M3,M,N,AT,V,Y,FLAG1,TEMP1)
K=K+1
ENDIF
ENDIF
END DO
TIMES=SECND$0
TIEMPO3=(TIMES-TIME4)+TIEMPO2+TIEMPO1
IF (OP1) CALL IMPRESO2(M,N,B,V,Y,E2,FD,K)
CALL IMPRESOR(M,N,C,V,X,Y,FLAG2,R,OP,OP1,PBY,MXMN,TIEMPO3)
CALL CONTINUA(ASK1,ASK2,ASK3,FD)
END DO
END DO
CALL AUTORES
STOP
END
SUBROUTINE TITULO
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT(6/),27%A // 40%A // 29%A,
'INSTRUMENTACION COMPUTACIONAL',
'DEL METODO DE ESCALA AFIN DUAL',
PRINT(6/),27%A \), 'PRESIONE ENTER PARA CONTINUAR '
READ(*,*)
RETURN
END
SUBROUTINE LECTAM(M1,N1)
INTEGER M1,N1
REAL M2,N2
CHARACTER*60 F1
CALL CLEARSCREEN($GCLEARSCREEN)
M2=0
N2=0
F1=(/ 14%A / 14%A //
PRINT F1,CAPTURA DEL NUMERO DE VARIABLES Y RESTRICCIONES QUE
*TIENE EL PROBLEMA DE PROGRAMACION LINEAL A SOLUCIONAR'
DO WHILE ((N2.GT.25).OR.(N2.LT.2))
PRINT '(3/),16%A \),
* 'EL NUMERO DE VARIABLES QUE TIENE EL PPL ES 7:'
READ*,N2
N1=INT(N2)
IF (N1.NE.N2) N2=0
END DO
DO WHILE ((M2.GT.25).OR.(M2.LT.1))
PRINT '(3/),15%A \),
* 'EL NUMERO DE RESTRICCIONES QUE TIENE EL PPL ES 7:'
READ*,M2
M1=INT(M2)
IF (M1.NE.M2) M2=0
END DO
RETURN
END
SUBROUTINE INICIA(M1,N1,A1,TREST1)
INTEGER M1,N1
CHARACTER TRIST(M1)
REAL A1(M1,N1)
INTEGER I,J
DO I=1,M1
DO J=1,N1
A1(I,J)=0
END DO
END DO
DO I=1,M1
TREST1(I)=0
END DO

```

```

RETURN
END
SUBROUTINE CARACTER(M1,NV1,MXMN,TRIST)
INTEGER*2 MXMN
INTEGER M1,NV1
CHARACTER*2 TRIST(M1)
INTEGER I
CHARACTER*2 ASK
ASK='0'
DO WHILE ((ASK.NE.'1').AND.(ASK.NE.'2'))
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT / 15X,AF
* 'CAPTURA DE LAS CARACTERISTICAS DEL PPL A RESOLVER'
PRINT / 19X,AF,LA FUNCION OBJETIVO DEL PPL SE TIENE QUE ?
PRINT / 20X,AF,MINIMIZAR -> (1)
PRINT / 20X,AF,MAXIMIZAR -> (2)
PRINT / 7(1,2)X,A V,¿SU SELECCION ES ?
READ(A,F,ASK)
END DO
IF (ASK.EQ.'1') MXMN=1
IF (ASK.EQ.'2') MXMN=-1
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT / 16X,AF,CAPTURA DE LAS CARACTERISTICAS DEL
'PPL A RESOLVER'
NV1=0
PRINT / 21X,A / 21X,AF
* PARA CADA UNA DE LAS RESTRICCIONES DEL PPL.
* INDICAR QUE TIPO DE RESTRICCION ES +
PRINT / 20X,AF, MENOR O IGUAL A -> (L)
PRINT / 20X,AF, MAYOR O IGUAL A -> (G)
PRINT / 20X,A //,IGUAL A -> (E)
DO I=1,M1
DO WHILE ((TRIST(I).NE.'G').AND.(TRIST(I).NE.'L').AND.
(TRIST(I).NE.'E').AND.(TRIST(I).NE.'G').AND.(TRIST(I).NE.'L').AND.
(TRIST(I).NE.'E'))
PRINT(25X,A,1,2,1X,A //,E)
READ(A,F,TRIST(I))
END DO
IF ((TRIST(I).EQ.'L').OR.(TRIST(I).EQ.'G').OR.(TRIST(I).EQ.'E'))
OR.(TRIST(I).EQ.'G')) NV1=NV1+1
END DO
RETURN
END
SUBROUTINE LECTURA(M,N,A,B,C)
INTEGER M,N
REAL A(M,N),B(M),C(N)
INTEGER I,J
LOGICAL ASK
CHARACTER*2 RI
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT / 16X,A //
*CAPTURA DE LOS COEFICIENTES DE LA FUNCION OBJETIVO
DO I=1,N
IF (I.LT.10) THEN
PRINT(15X,A,1,1X,A //,
PRINT(15X,A,12,1X,A //,
* 'EL COEFICIENTE DE LA VARIABLE N,I,' EN LA F. O. ES '+'
ELSE
PRINT(15X,A,12,1X,A //,
* 'EL COEFICIENTE DE LA VARIABLE N,I,' EN LA F. O. ES '-'
ENDIF
READ(F,C(I))
END DO
RI='0'
DO WHILE ((RI.NE.'1').AND.(RI.NE.'2'))
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT / 14X,AF,

```

```

+ 'FORMA DE CAPTURA DE LA MATRIZ DE RESTRICCIONES'
PRINT//17X//
+ 'LA MATRIZ DE RESTRICCIONES SE CAPTURARA COMO.'
PRINT/// 29X//MTRIZ ESPARSA -> (1/
PRINT// 29X//MTRIZ NO ESPARSA -> (2/
PRINT//132X//SU SELECCION ES.'
READ(A),R1
END DO
IF (R1.EQ.'1') THEN
DO I=1,M
ASK=.TRUE
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT// 18X//
+ 'CAPTURA DE LA MATRIZ ESPARSA DE RESTRICCIONES'
DO WHILE (ASK)
J=1
PRINT(10X,A,I2//EN LA RESTRICCION ',I'
PRINT(10X,A //
+ 'QUE VARIABLE TIENE COEFICIENTE DISTINTO DE CERO.'
DO WHILE ((I.LT.0)OR.(J.GT.N))
READ*,J
END DO
IF (J.GT.0) THEN
IF (I.LT.10) THEN
PRINT(10X,A,I1,A,I2,A //,EL COEFICIENTE DE LA',
' VARIABLE N',I,' EN LA RESTRICCION ',I,' ES.'
+ ELSE
PRINT(10X,A,I2,A,I2,A //,EL COEFICIENTE DE LA',
' VARIABLE N',I,' EN LA RESTRICCION ',I,' ES.'
+ END IF
READ*,A(I)
PRINT*,''
ELSE
ASK=.FALSE
END IF
END DO
ELSE
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT// 18X//
+ 'CAPTURA DE LA MATRIZ NO ESPARSA DE RESTRICCIONES'
DO I=1,M
DO J=1,N
IF (J.LT.10) THEN
PRINT(11X,A,A,I1,A,I2,A //,EL COEFICIENTE DE LA',
' VARIABLE N',I,' EN LA RESTRICCION ',I,' ES.'
+ ELSE
PRINT(11X,A,A,I2,A,I2,A //,EL COEFICIENTE DE LA',
' VARIABLE N',I,' EN LA RESTRICCION ',I,' ES.'
+ END IF
READ*,A(I)
END DO
PRINT*,''
END DO
ENDIF
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT// 21X//
+ 'CAPTURA DE LOS TERMINOS INDEPENDIENTES'
DO I=1,M
PRINT(10X,A,I2,A //
+ 'EL TERMINO INDEPENDIENTE DE LA RESTRICCION ',I,' ES.'
READ*,R(I)
END DO
RETURN
END
SUBROUTINE IMPRESO1(M1,N1,A1,B1,C1,MXXIN,TREST,OP1)

```

```

INTEGER*2 M,MMN
INTEGER M1,N1
REAL A(M1,N1),B(M1),C1(N1)
CHARACTER TRIST(M1)
LOGICAL OP1
INTEGER L,K
REAL EH
PARAMETER (EH=10E-4)
CHARACTER*2 R
CALL CLEARSCREEN(5GCLEARSCREEN)
PRINT/ 15X,A /,
*EL PROBLEMA DE PROGRAMACION LINEAL PROPORCIONADO ES :
IF (M*MMN.EQ.1) THEN
  IF (N1.GT.10) THEN
    PRINT/ 4X,A //,MAX Z = '
  ELSE
    PRINT/ 10X,A //,MAX Z = '
  ENDIF
ELSE
  IF (N1.GT.10) THEN
    PRINT/ 4X,A //,MIN Z = '
  ELSE
    PRINT/ 10X,A //,MIN Z = '
  ENDIF
ENDIF
ENDIF
K=0
DO I=1,N1
  IF (ABS(C1(I)).GT.EH) THEN
    K=K+1
    IF ((C1(I).GT.0).AND.(LINE1).AND.(K.NE.1)) THEN PRINT(A //, '
    ELSE
      IF (C1(I).LT.0) PRINT(A //, '
    ENDIF
    IF (L.LT.10) THEN
      IF (ABS(C1(I)).LT.1) THEN
        PRINT(A,F2.1,A,1 //,ABS(C1(I)),N,1 ELSE
          IF (ABS(C1(I)).LT.10)
            PRINT(F3.1,A,1 //,ABS(C1(I)),N,1
      ENDIF
      IF (ABS(C1(I)).GE.10).AND.(ABS(C1(I)).DLT.100)
        PRINT(F4.1,A,1 //,ABS(C1(I)),N,1
      IF ((ABS(C1(I)).GE.100).AND.(ABS(C1(I)).LT.1000))
        PRINT(F5.1,A,1 //,ABS(C1(I)),N,1
      IF ((ABS(C1(I)).GE.1000).AND.(ABS(C1(I)).LT.10000))
        PRINT(F6.1,A,1 //,ABS(C1(I)),N,1
      IF ((ABS(C1(I)).GE.10000).AND.(ABS(C1(I)).LT.10E5))
        PRINT(F8.1,A,1 //,ABS(C1(I)),N,1
      IF (ABS(C1(I)).GE.10E5)
        PRINT(F11.1,A,1 //,ABS(C1(I)),N,1
    ELSE
      IF (ABS(C1(I)).LT.1) THEN
        PRINT(A,F2.1,A,12 //,ABS(C1(I)),N,1 ELSE
          IF (ABS(C1(I)).LT.10)
            PRINT(F3.1,A,12 //,ABS(C1(I)),N,1
      ENDIF
      IF (ABS(C1(I)).GE.10).AND.(ABS(C1(I)).LT.100)
        PRINT(F4.1,A,12 //,ABS(C1(I)),N,1
      IF ((ABS(C1(I)).GE.100).AND.(ABS(C1(I)).LT.1000))
        PRINT(F5.1,A,12 //,ABS(C1(I)),N,1
      IF (ABS(C1(I)).GE.1000).AND.(ABS(C1(I)).LT.10000))
        PRINT(F6.1,A,12 //,ABS(C1(I)),N,1
      IF ((ABS(C1(I)).GE.10000).AND.(ABS(C1(I)).LT.10E5))
        PRINT(F8.1,A,12 //,ABS(C1(I)),N,1
      IF (ABS(C1(I)).GE.10E5)
        PRINT(F11.1,A,12 //,ABS(C1(I)),N,1
    ENDIF
  ENDIF

```



```

ENDIF
END DO
PRINT// 12%,A //7%, A'
DO I=1,M1
  K=0
  IF (L1.T.10) THEN
    IF (N1.GT.24) THEN
      PRINT(4%,A,11,A,1%,R,U) ' '
    ELSE
      PRINT(10%,A,11,A,1%,R,U) ' '
    ENDIF
  ELSE
    IF (N1.GT.24) THEN
      PRINT(4%,A,12,A,1%,R,U) ' '
    ELSE
      PRINT(10%,A,12,A,1%,R,U) ' '
    ENDIF
  ENDIF
  DO J=1,N1
    IF (ABS(A1(I),J).GT.E1) THEN
      K=K+1
      IF (A1(I).GT.0) THEN
        IF ((J.NE.1).AND.(K.NE.1)) PRINT(A \Y) ' '
      ELSE
        PRINT(A \Y) ' '
      ENDIF
      IF (L1.T.10) THEN
        IF (ABS(A1(I),J).LT.1) THEN
          PRINT(A,F2,A,11 \Y,%ABS(A1(I),J)/N1)
        ELSE
          IF (ABS(A1(I),J).LT.10) PRINT(F3,1,A,11 \Y,
            ABS(A1(I),J)/N1)
        ENDIF
      ELSE
        IF (ABS(A1(I),J).GE.10).AND.(ABS(A1(I),J).LT.100)
          PRINT(F4,1,A,11 \Y,ABS(A1(I),J)/N1)
        IF (ABS(A1(I),J).GE.100).AND.(ABS(A1(I),J).LT.1000)
          PRINT(F5,1,A,11 \Y,ABS(A1(I),J)/N1)
        IF (ABS(A1(I),J).GE.1000).AND.(ABS(A1(I),J).LT.10000)
          PRINT(F6,1,A,11 \Y,ABS(A1(I),J)/N1)
        IF (ABS(A1(I),J).GE.10000).AND.(ABS(A1(I),J).LT.10E5)
          PRINT(F8,1,A,11 \Y,ABS(A1(I),J)/N1)
      ELSE
        IF (ABS(A1(I),J).LT.1) THEN
          PRINT(A,F2,1,A,12 \Y,%ABS(A1(I),J)/N1)
        ELSE
          IF (ABS(A1(I),J).LT.10) PRINT(F3,1,A,12 \Y,
            ABS(A1(I),J)/N1)
        ENDIF
      ELSE
        IF (ABS(A1(I),J).GE.10).AND.(ABS(A1(I),J).LT.100)
          PRINT(F4,1,A,12 \Y,ABS(A1(I),J)/N1)
        IF (ABS(A1(I),J).GE.100).AND.(ABS(A1(I),J).LT.1000)
          PRINT(F5,1,A,12 \Y,ABS(A1(I),J)/N1)
        IF (ABS(A1(I),J).GE.1000).AND.(ABS(A1(I),J).LT.10000)
          PRINT(F6,1,A,12 \Y,ABS(A1(I),J)/N1)
        IF (ABS(A1(I),J).GE.10000).AND.(ABS(A1(I),J).LT.10E5)
          PRINT(F8,1,A,12 \Y,ABS(A1(I),J)/N1)
      ENDIF
    END DO
  SELECT CASE (TRINST(I))
  CASE ('U')
    PRINT(A \Y) '<- '
  CASE ('O')
    PRINT(A \Y) '>= '
  CASE ('T')
    PRINT(A \Y) '<= '
  END SELECT

```

```

CASE (G)
  PRINT('A Y.' >= '
CASE DEFAULT
  PRINT('A Y.' = '
END SELECT
IF (B1(1).LT.0) PRINT('A Y.'
IF (ABS(B1(1)).LT.1) THEN
  PRINT('A.F2.17./O.ABS(B1(1))
ELSE
  IF (ABS(B1(1)).LT.10) PRINT(F3.17,ABS(B1(1))
ENDIF
IF ((ABS(B1(1)).GE.10).AND.(ABS(B1(1)).LT.100))
  PRINT('F4.17,ABS(B1(1))
  IF ((ABS(B1(1)).GE.100).AND.(ABS(B1(1)).LT.1000))
  PRINT('F5.17,ABS(B1(1))
  IF ((ABS(B1(1)).GE.1000).AND.(ABS(B1(1)).LT.10000))
  PRINT('F6.17,ABS(B1(1))
  IF ((ABS(B1(1)).GE.10000).AND.(ABS(B1(1)).LT.10E5))
  PRINT('F8.17,ABS(B1(1))
END DO
R=0
DO WHILE ((R.NE.'1').AND.(R.NE.'1').AND.(R.NE.'1').AND.(R.NE.'1'))
  IF ((M1.GE.15).OR.(N1.GE.25)) THEN
    PRINT(/ 16X,A,Y)
  * 'DESEA EFECTUAR ALGUNA CORRECCION AL PPL (T/F) ?:'
  ELSE
    PRINT(/// 16X,A,Y)
  * 'DESEA EFECTUAR ALGUNA CORRECCION AL PPL (T/F) ?:'
  ENDF
  READ(A,Y)
END DO
IF ((R.EQ.'1').OR.(R.EQ.'0')) OP1=.TRUE.
IF ((R.EQ.'1').OR.(R.EQ.'0')) OP1=.FALSE.
RETURN
END
SUBROUTINE CORREGIRM,N,A,B,C,MXMIN,TREST)
INTEGER*2 MXMIN
REAL A(M,N),B(M),C(N)
CHARACTER TREST(M)
INTEGER I1
REAL B1,I1
CHARACTER*2 R,R1
CHARACTER R2
LOGICAL OPCION
OPCION=.TRUE.
DO WHILE(OPCION)
  R=0
  DO WHILE ((R.NE.'1').AND.(R.NE.'1').AND.(R.NE.'1').AND.
  (R.NE.'1').AND.(R.NE.'1').AND.(R.NE.'1'))
    CALL CLEARSCREEN($GCLEARSCREEN)
  * 'CORRECCION DE LOS DATOS DE ENTRADA DEL PPL PROPORCIONADO'
  PRINT(/ 16X,A,Y)
  * 'TIPO DE PROBLEMA (MANIMIZAR/MINIMIZAR) ->(1)F'
  PRINT(16X,A,Y)
  * 'COEFICIENTE EN LA FUNCION OBJETIVO ->(2)F'
  PRINT(16X,A,Y)
  * 'COEFICIENTE DE LA MATRIZ DE RESTRICCIONES ->(3)F'
  PRINT(16X,A,Y)
  * 'COEFICIENTE DEL VECTOR DEL LADO DERECHO ->(4)F'
  PRINT(16X,A,Y)
  * 'TIPO DE RESTRICCION (L<=, G==, E=) ->(5)F'
  PRINT(16X,A,Y)
  * 'SALIR ->(6)F'
  PRINT(/,32X,A,Y,50'SELECCION ES:'
  READ(A,Y)

```

```

END DO
R2=R
SELECT CASE(R2)
  CASE (1)
    R1=0
    PRINT(// 2)X,A,Y,"CORRECCION DEL TIPO DE PROBLEMA"
    PRINT(// 12)X,A,Y,"TIPO DE PROBLEMA ACTUAL:"
    IF (MXX.NE.1) PRINT(A,Y,"MINIMIZAR"
    IF (MXX.NE.1) PRINT(A,Y,"MAXIMIZAR"
    DO WHILE ((R1.NE.1).AND.(R1.NE.2))
      PRINT(12)X,A,Y
      * INGRESE (1) PARA MINIMIZACION, (2) PARA MAXIMIZACION:
      READ(A,R1)
    END DO
    IF (R1.EQ.1) MXXN=1
    IF (R1.EQ.2) MXXN=-1
  CASE (2)
    I1=0
    PRINT(// 2)X,A,Y,"CORRECCION DE COEFICIENTE EN LA F.O."
    DO WHILE ((I1.LE.0).OR.(I1.GT.N))
      PRINT(// 15)X,A,Y
      * INGRESE EL NUMERO DE LA VARIABLE DE DECISION:
      READ*,I1
      I=INT(I1)
      IF (I.NE.I1) I1=0
    END DO
    PRINT(15)X,A,F7.2F
    * EL COEFICIENTE ACTUAL DE ESTA VARIABLE ES: (C1)
    PRINT(15)X,A,Y,"INGRESE EL NUEVO COEFICIENTE:"
    READ*,C(I)
  CASE (3)
    I1=0
    PRINT(// 12)X,A,Y,"CORRECCION DE COEFICIENTE EN LA,
    * MATRIZ DE RESTRICCIONES"
    DO WHILE ((I1.LE.0).OR.(I1.GT.M))
      PRINT(// 15)X,A,Y,"INGRESE EL NUMERO DE LA RESTRICCION:"
      READ*,I1
      I=INT(I1)
      IF (I.NE.I1) I1=0
    END DO
    I1=0
    DO WHILE ((I1.LE.0).OR.(I1.GT.N))
      PRINT(15)X,A,Y
      * INGRESE EL NUMERO DE LA VARIABLE DE DECISION:
      READ*,I1
      I=INT(I1)
      IF (I.NE.I1) I1=0
    END DO
    PRINT(15)X,A,F7.2F
    * EL COEFICIENTE ACTUAL DE ESTA VARIABLE ES: (A(I))
    PRINT(15)X,A,Y,"INGRESE EL NUEVO COEFICIENTE:"
    READ*,A(I)
  CASE (4)
    I1=0
    PRINT(// 2)X,A,Y,"CORRECCION DE TERMINO INDEPENDIENTE"
    DO WHILE ((I1.LE.0).OR.(I1.GT.M))
      PRINT(// 15)X,A,Y,"INGRESE EL NUMERO DE LA RESTRICCION:"
      READ*,I1
      I=INT(I1)
      IF (I.NE.I1) I1=0
    END DO
    PRINT(15)X,A,F7.2F
    * EL TERMINO INDEPENDIENTE ACTUAL ES: (B(I))
    PRINT(15)X,A,Y,"INGRESE EL NUEVO TERMINO INDEPENDIENTE:"
    READ*,B(I)
  CASE (5)

```

```

      II=0
      PRINT// 22X'AY, 'CORRECCION DEL TIPO DE RESTRICCION"
      DO WHILE ((I1.LE.0).OR.(I1.GT.M))
        PRINT// 12X'A \F, 'INGRESE EL NUMERO DE LA RESTRICCION : '
        READ* I1
        I=INT(I1)
        IF (I.NE.I1) II=0
        END DO
        PRINT(12X'A \F,
+      'EL TIPO ACTUAL DE LA RESTRICCION ES :', TREST(I)
+      TREST(I)=0
+      DO WHILE ((TREST(I).NE.C') .AND. (TREST(I).NE.L') .AND.
+      (TREST(I).NE.F') .AND. (TREST(I).NE.E') .AND.
+      (TREST(I).NE.P') .AND. (TREST(I).NE.G'))
+      PRINT(12X'A \F,
+      'INGRESE EL NUEVO TIPO DE RESTRICCION (L=>, G=>, E=>): '
+      READ* (A \F, TREST(I)
        END DO
      END SELECT
      IF (I.NE.'G') THEN
        R1=0
        DO WHILE ((R1.NE.'P') .AND. (R1.NE.'F') .AND. (R1.NE.'L') .AND.
+      (R1.NE.'E'))
          PRINT// 19X'A \F, 'DESEA EFECTUAR OTRA CORRECCION (T/F) ? : '
          READ* (A \F, R1
          END DO
          IF ((R1.EQ.'T') .OR. (R1.EQ.'F')) OPCION='TRUE'
          IF ((R1.EQ.'P') .OR. (R1.EQ.'E')) OPCION='FALSE'
        ELSE
          OPCION='FALSE'
        ENDIF
      END DO
      RETURN
    END
    SUBROUTINE FORMEST(M1, N1, NVH, MNMN, A1, C1, TREST)
      INTEGER*2 MNMN
      INTEGER N1, NVH
      CHARACTER TREST(M1)
      REAL A1(M1, N1+NVH), C1(N1+NVH)
      INTEGER I, J
      DO I=1, M1
        DO J=N1+1, N1+NVH
          IF (I.EQ.I-N1) THEN
            SELECT CASE (TREST(I))
              CASE (L')
                A1(I, J)=1
              CASE (G')
                A1(I, J)=1
              CASE (T')
                A1(I, J)=1
              CASE (F')
                A1(I, J)=1
              CASE DEFAULT
                A1(I, J)=0
            END SELECT
          ELSE
            A1(I, J)=0
          ENDIF
        END DO
      END DO
      IF (MIN(MN, EQ-1) CALL
      SSCAL(N1-1, 0, C1, 1) DO
      J=N1+1, N1+NVH
      C1(J)=0
      END DO
      N1=N1+NVH

```

```

RETURN
END
SUBROUTINE PARAMET(MG,R,COTA,E1,E2,E3,E4,TETA)
REAL MG,R,COTA,E1,E2,E3,E4,TETA
CHARACTER*2 R1,R2
LOGICAL DEC1
R1='0'
DO WHILE ((R1.NE.'1').AND.(R1.NE.'2'))
  CALL CLEARSCREEN($CLEARSCREEN)
  PRINT(/ 20\A / 21\A //)
  * 'CAPTURA DE LOS PARAMETROS DE EJECUCION DEL,
  * 'METODO DE ESCALA AFIN DUAL'
  PRINT(/ 21\A /// 18\A /// 16\A / 16\A //)
  * 'PARA EJECUTAR EL PROGRAMA, ¿STED DESEA?',
  * 'PROPORCIONAR LOS VALORES DE LOS PARAMETROS -> (1),
  * 'EL PROGRAMA SE EJECUTE CON LOS VALORES QUE,
  * 'SE TIENEN ASIGNADOS -> (2)
  PRINT(/ 13\A \/,SU SELECCION ES: '
  READ*(A7,R1)
END DO
IF (R1.EQ.'1') THEN
  DEC1=.TRUE.
  DO WHILE (DEC1)
    CALL CLEARSCREEN($CLEARSCREEN)
    PRINT(/ 19\A / 20\A //)
    * 'CAPTURA DE LOS PARAMETROS DE EJECUCION DEL,
    * 'METODO DE ESCALA AFIN DUAL'
    MG=0
    DO WHILE (MG.LE.0)
      PRINT(/ 18\A \/,EL VALOR DE LA M-GRANDE ES: '
      READ*MG
    END DO
    R=0
    DO WHILE ((R.LE.0).OR.(R.GT.1))
      PRINT(/ 18\A \/,EL FACTOR DEL TAMAWO DE PASO ES: '
      READ*R
    END DO
    COTA=0
    DO WHILE (COTA.LE.0)
      PRINT(/ 18\A \/,LA COTA SOBRE LA FUNCION OBJETIVO ES: '
      READ*COTA
    END DO
    E1=1
    DO WHILE ((E1.LT.0).OR.(E1.GE.1))
      PRINT(/ 18\A \/,LA COTA PARA LA FACTIBILIDAD PRIMAL ES: '
      READ*E1
    END DO
    E2=1
    DO WHILE ((E2.LT.0).OR.(E2.GE.1))
      PRINT(/ 18\A \/,LA COTA PARA LA FACTIBILIDAD DUAL ES: '
      READ*E2
    END DO
    E3=1
    DO WHILE ((E3.LT.0).OR.(E3.GE.1))
      PRINT(/ 18\A \/,
  * 'LA COTA PARA HOLGURAS COMPLEMENTARIAS ES: '
      READ*E3
    END DO
    E4=1
    DO WHILE ((E4.LT.0).OR.(E4.GE.1))
      PRINT(/ 18\A \/,LA COTA SOBRE EL VECTOR DIRECCION ES: '
      READ*E4
    END DO
    TETA=0
    DO WHILE (TETA.LE.0)
      PRINT(/ 18\A \/,

```

```

      * 'EL PARAMETRO DE CALCULO DE LA VARIABLE ARTIFICIAL ES : '
      READ*,TETA
      END DO
      R2=0
      DO WHILE ((R2.NE.'1').AND.(R2.NE.'F').AND.(R2.NE.'F').AND.(R2.NE.'F'))
      * 'REQUIERE VOLVER A CAPTURAR LOS PARAMETROS (T/F) ? : '
      READ*(A?,R2
      END DO
      IF ((R2.EQ.'1').OR.(R2.EQ.'F')) DEC1=.TRUE.
      IF ((R2.EQ.'F').OR.(R2.EQ.'F')) DEC1=.FALSE.
      END DO
      ELSE
      MG=10E3
      R=0.9999
      COTA=10E5
      E1=10E-4
      E2=10E-4
      E3=10E-4
      E4=10E-4
      TETA=2
      ENDIF
      RETURN
      END
      SUBROUTINE OSIFID(M1,N1,A1,B1,C1,AT1,V1,Y1,MG1,TETA)
      INTEGER M1,N1
      REAL A1(M1),N1,AT1(N1,M1+1),B1(M1+1),C1(N1),V1(N1),Y1(M1+1)
      REAL MG1,TETA
      INTEGER I
      REAL CMAX
      CALL TRNRRR(M1,N1,A1,M1,N1,M1,AT1,N1)
      CALL SSUT(N1,0.0,Y1,1)
      I=ISMIN(N1,C1,1)
      IF (C1(I).LE.0) THEN
      B1(M1+1)=(-MG1)
      DO I=1,N1
      IF (C1(I).LE.0) THEN
      AT1(LM1+1)=1
      ELSE
      AT1(LM1+1)=0
      END IF
      END DO
      I=ISAMAX(N1,C1,1)
      CMAX=ABS(C1(I))
      V1(M1+1)=TETA*CMAX
      DO I=1,N1
      V1(I)=C1(I)+AT1(LM1+1)*V1(M1+1)
      END DO
      ELSE
      CALL SCOPY(N1,C1,1,V1,1)
      B1(M1+1)=0
      ENDIF
      RETURN
      END
      SUBROUTINE IMPRESO4(M1,N1,AT1,B1,C1)
      INTEGER M1,N1
      REAL AT1(N1,M1+1),B1(M1+1),C1(N1)
      INTEGER I,J,K,M3
      REAL E11
      PARAMETER (E11=10E-4)
      CALL CLEARSCREEN(5GCLEARSCREEN)
      PRINT/12XA //
      * 'EL PROBLEMA DE PROGRAMACION LINEAL DUAL A RESOLVER ES '
      IF (B1(M1+1).NE.0) THEN
      M3=M1+1
      ELSE

```

```

M3=M41
ENDIF
IF (M3.GT.10) THEN
  PRINT// 2X,A //,MAX W = '
ELSE
  PRINT// 10X,A //,MAX W = '
ENDIF
K=0
DO I=1,M3
  IF (ABS(B1(I)).GT.E1) THEN
    K=K+1
    IF ((B1(I).GT.0).AND.(I.NE.1).AND.(K.NE.1)) THEN
      PRINT(A //, ' *
    ELSE
      IF (B1(I).LT.0) PRINT(A //, ' *
    ENDFI
    IF (I.LT.10) THEN
      IF (ABS(B1(I)).LT.1) THEN
        PRINT(A,F2.1,A,1 //,0,ABS(B1(I))//,Y,1
      ELSE
        IF (ABS(B1(I)).LT.10)
          PRINT(F3.1,A,1 //,Y,ABS(B1(I))//,Y,1
        ENDFI
        IF ((ABS(B1(I)).GE.10).AND.(ABS(B1(I)).LT.100))
          PRINT(F4.1,A,1 //,Y,ABS(B1(I))//,Y,1
        IF ((ABS(B1(I)).GE.100).AND.(ABS(B1(I)).LT.1000))
          PRINT(F5.1,A,1 //,Y,ABS(B1(I))//,Y,1
        IF ((ABS(B1(I)).GE.1000).AND.(ABS(B1(I)).LT.10000))
          PRINT(F6.1,A,1 //,Y,ABS(B1(I))//,Y,1
        IF ((ABS(B1(I)).GE.10000).AND.(ABS(B1(I)).LT.10E5))
          PRINT(F8.1,A,1 //,Y,ABS(B1(I))//,Y,1
        IF (ABS(B1(I)).GE.10E5)
          PRINT(F13.1,A,1 //,Y,ABS(B1(I))//,Y,1
      ELSE
        IF (ABS(B1(I)).LT.1) THEN
          PRINT(A,F2.1,A,12 //,0,ABS(B1(I))//,Y,1
        ELSE
          IF (ABS(B1(I)).LT.10)
            PRINT(F3.1,A,12 //,Y,ABS(B1(I))//,Y,1
          ENDFI
          IF (ABS(B1(I)).GE.10).AND.(ABS(B1(I)).LT.100)
            PRINT(F4.1,A,12 //,Y,ABS(B1(I))//,Y,1
          IF (ABS(B1(I)).GE.100).AND.(ABS(B1(I)).LT.1000)
            PRINT(F5.1,A,12 //,Y,ABS(B1(I))//,Y,1
          IF (ABS(B1(I)).GE.1000).AND.(ABS(B1(I)).LT.10000)
            PRINT(F6.1,A,12 //,Y,ABS(B1(I))//,Y,1
          IF (ABS(B1(I)).GE.10000).AND.(ABS(B1(I)).LT.10E5)
            PRINT(F8.1,A,12 //,Y,ABS(B1(I))//,Y,1
          IF (ABS(B1(I)).GE.10E5)
            PRINT(F13.1,A,12 //,Y,ABS(B1(I))//,Y,1
        ENDFI
      ENDFI
    ENDDO
  PRINT// 12X,A //,S. A '
  DO I=1,N1
    K=0
    IF (I.LT.10) THEN
      IF (M3.GT.24) THEN
        PRINT(4X,A,1,A,1X //,R,L, ' *
      ELSE
        PRINT(10X,A,1,A,1X //,R,L, ' *
      ENDFI
    ELSE
      IF (M3.GT.24) THEN
        PRINT(4X,A,12,A,1X //,R,L, ' *
      ELSE

```

```

      PRINT(10X,A,I2,A,I1X\Y,'S:L: ')
    ENDF
  ENDDO
DO J=1,N3
  IF (ABS(AT(I,J),GT.0E1) THEN
    K=K+1
    IF (AT(I,J),GT.0) THEN
      IF (J.NE.1)AND(K.NE.1) PRINT(A \Y: '
    ELSE
      PRINT(A \Y: '
    ENDF
    IF (L.T.10) THEN
      IF (ABS(AT(I,D),L.T.1) THEN
        PRINT(A,F2.1,A,1 \Y,*,ABS(AT(I,D)),Y)
      ELSE
        IF (ABS(AT(I,D),L.T.10) PRINT(F3.1,A,1 \Y,
        ABS(AT(I,D)),Y)
      ENDF
      IF (ABS(AT(I,J),GE.10)AND(ABS(AT(I,D),L.T.100)
      PRINT(F4.1,A,1 \Y,ABS(AT(I,D)),Y)
      IF (ABS(AT(I,J),GE.100)AND(ABS(AT(I,D),L.T.1000)
      PRINT(F5.1,A,1 \Y,ABS(AT(I,D)),Y)
      IF (ABS(AT(I,J),GE.1000)AND(ABS(AT(I,D),L.T.10000)
      PRINT(F6.1,A,1 \Y,ABS(AT(I,D)),Y)
      IF (ABS(AT(I,J),GE.10000)AND(ABS(AT(I,D),L.T.10E5)
      PRINT(F8.1,A,1 \Y,ABS(AT(I,D)),Y)
    ELSE
      IF (ABS(AT(I,J),L.T.1) THEN
        PRINT(A,F2.1,A,I2 \Y,*,ABS(AT(I,D)),Y)
      ELSE
        IF (ABS(AT(I,J),L.T.10) PRINT(F3.1,A,I2 \Y,
        ABS(AT(I,D)),Y)
      ENDF
      IF (ABS(AT(I,D),GE.10)AND(ABS(AT(I,D),L.T.100)
      PRINT(F4.1,A,I2 \Y,ABS(AT(I,D)),Y)
      IF (ABS(AT(I,D),GE.100)AND(ABS(AT(I,D),L.T.1000)
      PRINT(F5.1,A,I2 \Y,ABS(AT(I,D)),Y)
      IF (ABS(AT(I,D),GE.1000)AND(ABS(AT(I,D),L.T.10000)
      PRINT(F6.1,A,I2 \Y,ABS(AT(I,D)),Y)
      IF (ABS(AT(I,D),GE.10000)AND(ABS(AT(I,D),L.T.10E5)
      PRINT(F8.1,A,I2 \Y,ABS(AT(I,D)),Y)
    ENDF
  ENDDO
ENDDO
IF (L.T.10) THEN
  PRINT(A,I1,A \Y: '+1.0V,I: '=
ELSE
  PRINT(A,I2,A \Y: '+1.0V,I: '=
ENDF
IF (C(I),L.T.0) PRINT(A \Y: '
IF (ABS(C(I)),L.T.1) THEN PRINT(A,F2.1,Y,*,ABS(C(I))
ELSE
  IF (ABS(C(I)),L.T.10) PRINT(F3.1,Y,ABS(C(I))
ENDF
IF (ABS(C(I),GE.10)AND(ABS(C(I)),L.T.100)
PRINT(F4.1,Y,ABS(C(I))
IF (ABS(C(I),GE.100)AND(ABS(C(I)),L.T.1000)
PRINT(F5.1,Y,ABS(C(I))
IF (ABS(C(I),GE.1000)AND(ABS(C(I)),L.T.10000)
PRINT(F6.1,Y,ABS(C(I))
IF (ABS(C(I),GE.10000)AND(ABS(C(I)),L.T.10E5)
PRINT(F8.1,Y,ABS(C(I))
ENDDO
IF (N1.GE.15)OR(N3.GE.25) THEN
  PRINT(V, 27X,A \Y,PRESIONE ENTER PARA CONTINUAR '
ELSE

```



```

PRINT// ' 27X.A \), PRESIONE ENTER PARA CONTINUAR '
ENDIF
READ(*,*)
RETURN
END
SUBROUTINE SELMET(M1,M3,N1,B1,OP1,V1,Y1)
INTEGER M1,M3,N1
REAL B1(M1+1),V1(N1),Y1(N1),Y1(M1+1)
LOGICAL OP1
INTEGER*2 OP
CHARACTER*2 R
IF (B1(M1+1),NE.0) THEN
M3=M1+1
CALL SCOPY(M3,Y1,I,Y,1)
ELSE
M3=M1
CALL SCOPY(M3,Y1,I,Y,1)
Y(N1+1)=0
ENDIF
CALL SCOPY(N1,V1,1,V,1)
R='0'
DO WHILE ((R.NE.'1').AND.(R.NE.'2').AND.(R.NE.'3').AND.(R.NE.'4'))
CALL CLEARSCREEN($CLEARSCREEN)
PRINT// ' 17X.A / 17X.A)',
+ 'METODOS DE FACTORIZACION MATRICIAL QUE PUEDEN SER,'
+ 'UTILIZADOS EN LA OBTENCION DEL VECTOR DIRECCION .'
PRINT// ' 25X.A)', FACTORIZACION U'D'U' -> (1)
PRINT// ' 25X.A)', FACTORIZACION DE CHOLESKY -> (2)
PRINT// ' 25X.A)', FACTORIZACION Q'R -> (3)
PRINT// ' 25X.A)', MINIMOS CUADRADOS -> (4)
PRINT// ' 1,2,3,4 \), EL METODO SELECCIONADO (1,2,3 o 4) ES ? '
READ(A,Y,R)
END DO
IF (R.EQ.'1') OP=1
IF (R.EQ.'2') OP=2
IF (R.EQ.'3') OP=3
IF (R.EQ.'4') OP=4
DO WHILE ((R.NE.'1').AND.(R.NE.'2').AND.(R.NE.'3').AND.(R.NE.'4'))
PRINT// ' 16X.A / 16X.A \),
+ 'DESEA VER CADA SOLUCION INTERIOR OBTENIDA EN CADA',
+ 'ITERACION DEL METODO (T/F) ? '
READ(A,Y,R)
END DO
IF ((R.EQ.'T').OR.(R.EQ.'F')) OP1=.TRUE.
IF ((R.EQ.'F').OR.(R.EQ.'T')) OP1=.FALSE.
RETURN
END
SUBROUTINE DISSMOV(S,M3,N1,AT1,B1,DV,DY,V1,OP)
INTEGER M3,N1
INTEGER*2 OP
REAL AT1(N1),M3,B1(M3),V1(N1),DV(N1),DY(N1)
INTEGER I1,KRASS,M2,N2
PARAMETER (M2=20,N2=50)
REAL AV2AT(M2,M2),RES(M2),VAT(N2,M2)
DO I=1,N1
DO J=1,M3
VAT(I,J)=AT1(I)/V1(J)
END DO
END DO
CALL MN1F(N1,M3,VAT,N2,M3,AV2AT,M2)
SELECT CASE (OP)
CASE (1)
CALL LSLSF(M3,AV2AT,M2,B1,DY)
CASE (2)
CALL LSADS(M3,AV2AT,M2,B1,DY)
CASE (3)

```

```

      CALL LSQR(M3,M3,AVZAT,M2,B1,0,DY,RES,KBASIS)
END SELECT
CALL MURRV(N1,M3,AT1,N1,M3,DY,1,N1,DV)
CALL SSCAL(N1,-1.0,DV,1)
RETURN
END
SUBROUTINE CVEP(N1,DV,V1,A)
  INTEGER N1
  REAL DV(N1),V1(N1),X(N1)
  INTEGER I
  REAL T
  DO I=1,N1
    T=V1(I)/V1(I)
    X(I)=(DV(I)/T)
  END DO
  RETURN
END
SUBROUTINE MINCUAD(M3,N1,AT1,B1,DV,DY,V1)
  INTEGER M3,N1
  REAL AT1(N1,M3),B1(M3),DV(N1),DY(M3),V1(N1)
  INTEGER I,J,M2,N2
  PARAMETER (M2=26,N2=50)
  REAL QRVAT(N2,M2),VAT(N2,M2),T1(N2),T2(M2),Z(M2)
  INTEGER IPVT(M2)
  LOGICAL PIV
  DO I=1,N1
    DO J=1,M3
      VAT(I,J)=AT1(I,J)/V1(I)
    END DO
  END DO
  PIV=FALSE
  CALL ISET(M3,0,IPVT,1)
  CALL LQRRR(N1,M3,VAT,N2,PIV,IPVT,QRVAT,N2,T1,T2)
  CALL LSLRT(M3,QRVAT,N2,B1,4,Z)
  CALL LSLRT(M3,QRVAT,N2,Z,DY)
  CALL MURRV(N1,M3,AT1,N1,M3,DY,1,N1,DV)
  CALL SSCAL(N1,-1.0,DV,1)
  RETURN
END
SUBROUTINE PARC(M3,M1,N1,AT1,B1,C1,A1,V1,Y1,FLAG1,PBY,E1,E2,E3,FD)
  INTEGER M1,M3,N1
  INTEGER*2 FLAG1
  REAL AT1(N1,M3),B1(M3),C1(N1),X1(N1),V1(N1),Y1(M3)
  REAL E1,E2,E3,FD,PBY
  INTEGER N2,N2,FLAG
  PARAMETER (M2=26,N2=50)
  REAL AX(M2),T(N2),NAX,NB,NC,NT,ABY,FP,HC,PCX
  FLAG=0
  CALL MURRV(N1,M3,AT1,N1,M3,Y1,1,N1,T)
  CALL SAXPY(N1,1.0,V1,1,T,1)
  CALL SAXPY(N1,-1.0,C1,1,T,1)
  NT=#NRM2(N1,T,1)
  NC=#NRM2(N1,C1,1)
  FD=NT/(NC*1)
  IF (FD.GT.E2) FLAG=1
  IF (FLAG.EQ.0) THEN
    CALL MURRV(N1,M1,AT1,N1,N1,N2,M1,AX)
    CALL SAXPY(N1,-1.0,B1,1,AX,1)
    NAX=#NRM2(M1,AX,1)
    NB=#NRM2(N1,B1,1)
    FP=NAX/(NB*1)
    IF (FP.GT.E1) FLAG=1
  ENDIF
  IF (FLAG.EQ.0) THEN
    PCX=#SDOT(N1,C1,1,N1,1)
    ABY=ABS(PBY)
  
```

```

      HC=ABS(PCN-PBY)/(ABY+1)
      IF (HC.LT.E3) FLAG1=1
    ENDF
  RETURN
  END
  SURROUTINE NAFOC(M1,N1,B1,DV,Y1,COTA,E4,FLAG1,FLAG2,PBY) C
  INTEGER M1,N1
  INTEGER*2 FLAG1,FLAG2
  REAL B1(M1+1),DV(N1),Y1(M1+1),E4,PBY
  INTEGER I,J,I1
  REAL NDV
  LOGICAL FLAG
  I=SMIN(N1,DV,1)
  I=SMAX(N1,DV,1)
  NDV=SNRM2(N1,DV,1)
  IF (NDV.LE.E4) THEN
    IF ((DV(I).LT.0).AND.(DV(J).LT.0)) THEN
      FLAG1=1
    ELSE
      I=0
      J=0
      J1=0
      FLAG=.TRUE.
      DO WHILE (FLAG)
        I=I+1
        IF (DV(I).GE.0) THEN
          J=J+1
        ELSE
          J1=J1+1
        ENDIF
        IF (.EQ.N1) FLAG=.FALSE.
      END DO
      IF (J1.LE.I) THEN
        FLAG1=1
        FLAG2=2
      ELSE
        FLAG1=1
      ENDF
    ELSE
      IF ((DV(I).GE.0).AND.(DV(J).GE.0)) THEN
        FLAG1=1
        FLAG2=2
      ELSE
        I=0
        J=0
        J1=0
        FLAG=.TRUE.
        DO WHILE (FLAG)
          I=I+1
          IF (DV(I).GE.0) THEN
            J=J+1
          ELSE
            IF (ABS(DV(I)).LE.E4) THEN
              J1=J1+1
            ELSE
              FLAG=.FALSE.
            ENDIF
          ENDF
          IF (.EQ.N1) FLAG=.FALSE.
        END DO
        IF (J1.LE.I) THEN
          I=I+1
          IF (.EQ.N1) THEN
            FLAG1=1
            FLAG2=2
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF

```

```

        ENDIF
      ENDIF
    ENDIF
  ENDIF
  PHY=SDOT(M1+1,B1,Y1,1)
  IF ((FLAG1.NE.1).AND.(Y1(M1+1)EQ.0).AND.(ABS(PHY).GT.COTA)) THEN FLAG1=1
    FLAG2=2
  ENDIF
  RETURN
END
SUBROUTINE LP08M(N3,N1,DV,DY,V1,Y1,R,E4)
  INTEGER M3,N1
  REAL DV(N1),DY(M3),V1(N1),Y1(M3),E4,R
  INTEGER N2,I,T1
  PARAMETER (N2=50)
  REAL BETA,TEM1(N2)
  T=0
  DO T=1,N1
    IF (DV(T).LT.(E4)) THEN
      T=T+1
      TEM(T)=(R*V1(T))/(C-DV(T))
    ENDIF
  END DO
  IF (T.GT.0) THEN
    T1=BSMIN(T,TEM,1)
    BETA=TEM(T1)
  ELSE
    BETA=R
  ENDIF
  CALL SAXPY(M3,BETA,DY,1,Y1,1)
  CALL SAXPY(N1,BETA,DV,1,V1,1)
  RETURN
END
SUBROUTINE CHECVART(M3,M1,N1,AT1,V1,Y1,FLAG1,TEMP1)
  INTEGER M3,M1,N1
  INTEGER 2 FLAG1
  REAL AT1(N1,M1+1),V1(N1),Y1(M1+1),TEMP1
  INTEGER I,N2
  REAL EP,T1
  PARAMETER (EP=10E-6,N2=50)
  REAL T(N2)
  IF (Y1(M1+1).GT.EP) THEN
    T1=TEMP1-Y1(M1+1)
    IF (T1.LT.0) FLAG1=1
  ELSE
    DO I=1,N1
      T(I)=C-AT1(I,M1+1)*V1(M1+1)
    END DO
    CALL SAXPY(N1,-1.0,T,1,V1,1)
    Y1(M1+1)=0
    M3=M1
  ENDIF
  RETURN
END
SUBROUTINE IMPRISO2(M1,N1,B1,V1,Y1,E2,FD,K)
  INTEGER M1,N1,K
  REAL B1(M1+1),V1(N1),Y1(M1+1),E2,FD
  INTEGER I1(2),I,L,M2,N2
  REAL V2(2),Y2(2),PHYT
  CALL CLEARSCREEN(3)CALL CLEARSCREEN)
  PRINT// '80. AIZ.A // EN LA ITERACION 'K
  * EL VALOR DE LAS VAR. NO RESTRINGIDAS ES '
  I=0
  IF (MOD(M1+1,2).EQ.0) THEN
    M2=M1-1
  ELSE

```

```

M2=M1
ENDIF
L=0
DO I=1,M2
  L=L+1
  J=J+1
  H(I)=I
  V2(I)=V1(I)
  IF (I.EQ.2) THEN
    IF (I.LT.10) THEN PRINT(23X,A11,A,F10.5,9X,A11,A,F10.5F,
      * 'V',H(I-1)' = 'V2(I-1)',V,H(I)' = 'V2(I)
    ELSE
      PRINT(23X,A12,A,F10.5,8X,A12,A,F10.5F,
      * 'V',H(I-1)' = 'V2(I-1)',V,H(I)' = 'V2(I)
    ENDF
  I=0
ENDIF
ENDIF
IF ((I.EQ.30)AND(I.NE.M2)) THEN
  L=0
  PRINT// 25X A V, PRESIONE ENTER PARA CONTINUAR '
  READ(,"")
  CALL CLEARSCREEN($GCLEARSCREEN)
  PRINT// 9X,A12 A // EN LA ITERACION 'K,
  * ' EL VALOR DE LAS VAR. NO RESTRINGIDAS ES '
  ENDF
END DO
IF (MOD(M1+1,2).NE.0) THEN
  IF (M1+1.LT.10) THEN PRINT(23X,A11,A,F10.5F,'V',M1+1' = 'V1(M1+1)
  ELSE
    PRINT(23X,A12,A,F10.5F,'V',M1+1' = 'V1(M1+1)
  ENDF
ENDIF
PRINT// 25X A V, PRESIONE ENTER PARA CONTINUAR '
READ(,"")
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT// 11X,A12 A // EN LA ITERACION 'K,
* ' EL VALOR DE LAS VAR. RESTRINGIDAS ES '
J=0
IF (MOD(N1,2).EQ.0) THEN
  N2=N1
ELSE
  N2=N1-1
ENDIF
L=0
DO I=1,N2
  L=L+1
  J=J+1
  H(I)=I
  V2(I)=V1(I)
  IF (I.EQ.2) THEN
    IF (I.LT.10) THEN PRINT(23X,A11,A,F10.5,9X,A11,A,F10.5F,
      * 'V',H(I-1)' = 'V2(I-1)',V,H(I)' = 'V2(I)
    ELSE
      PRINT(23X,A12,A,F10.5,8X,A12,A,F10.5F,
      * 'V',H(I-1)' = 'V2(I-1)',V,H(I)' = 'V2(I)
    ENDF
  I=0
ENDIF
IF ((I.EQ.30)AND(I.NE.N2)) THEN
  L=0
  PRINT// 25X A V, PRESIONE ENTER PARA CONTINUAR '
  READ(,"")
  CALL CLEARSCREEN($GCLEARSCREEN)
  PRINT// 11X,A12 A // EN LA ITERACION 'K,
  * ' EL VALOR DE LAS VAR. RESTRINGIDAS ES '
  ENDF

```

```

END DO
IF (MOD(N1,2).NE.0) THEN
  IF (N1.LT.10) THEN PRINT(23%,A1,A,F10.5F,'V',N1) = 'V1(N1)
  ELSE
    PRINT(23%,A12,A,F10.5F,'V',N1) = 'V1(N1)
  ENDIF
ENDIF
PRINT(// 8%,A,F5.4F,
* 'EN ESTE PUNTO EL ERROR RELATIVO DE FACTIBILIDAD DUAL ES: 'FD
IF (ABS(FD).LE.F2) THEN
  PRINT(8%,A,F
    + 'ESTA SOLUCION ES FACTIBLE PARA LA COTA DE ERROR DADA.'
    PBYT=SDO1(M1+1,B1,1,Y1,1)
    PRINT(8%,A,F10.2F,
    + 'EN ESTE PUNTO, EL VALOR DE LA F.O. DEL PPL DUAL ES: PBYT
ELSE
  PRINT(8%,A,F,
    'ESTA SOLUCION ES INFACIBLE PARA LA COTA DE ERROR DADA.'
ENDIF
PRINT(// 25%,A,F,'PRESIONE ENTER PARA CONTINUAR'
READ(*,*)
RETURN
END
SUBROUTINE IMPRESO(M1,N1,C1,V1,M1,Y1,FLAG2,K,OP1,PBY,MXNMN,
*TIEMPO3)
INTEGER*2 FLAG,FLAG2,MXNMN,OP1
INTEGER*4 K
INTEGER M1,N1
REAL C1(N1),M1(N1),Y1(M1+1),V1(N1),PBY,TIEMPO3
LOGICAL OP1
INTEGER I,J(2),L,M2,N2
REAL X(2),Y(2)
CALL CLEARSCREEN(5,CLEARSCREEN)
FLAG=0
IF (OP.EQ.2) THEN
  PRINT(// 7%,A,F,'EL RESULTADO OBTENIDO CON EL METODO DE '
ELSE
  PRINT(// 10%,A,F,'EL RESULTADO OBTENIDO CON EL METODO DE '
ENDIF
SELECT CASE (OP1)
CASE (1)
  PRINT(A // F,FACTORIZACION U'D'U' ES '
CASE (2)
  PRINT(A // F,FACTORIZACION DE CHOLSKY ES '
CASE (3)
  PRINT(A // F,FACTORIZACION Q'R' ES '
CASE (4)
  PRINT(A // F,MINIMOS CUADRADOS ES '
END SELECT
IF (N1.EQ.10).AND.(M1.EQ.10) Y1(M1+1)=0
IF (ABS(Y1(M1+1)).NE.0) THEN
  PRINT(// 1,19%,A // F,
* 'LA F.O. DEL PPL PROPORCIONADO ES NO ACOTADA'
ELSE
  IF (FLAG2.EQ.2) THEN
    PRINT(// 1,23%,A // F,EL PPL PROPORCIONADO ES INFACIBLE
  ELSE
    FLAG=1
    PRINT(15%,A12,A // F,
    + 'LA SOLUCION OPTIMA DUAL OBTENIDA EN 'K' ITERACIONES ES '
    I=0
    IF (MOD(M1,2).EQ.0) THEN
      M2=M1
    ELSE
      M2=M1-1
    ENDIF
  ENDIF

```

```

L=0
DO I=1,M2
  L=L+1
  J=J+1
  H(I)=1
  Y2(I)=Y1(I)
  IF (I.EQ.2) THEN
    IF ((L.T.10) THEN PRINT(23X,A,I1,A,F10.5,9X,A,I1,A,F10.5F,
+   'Y',H(I-1), '=',Y2(I-1),'Y',H(I)), '=',Y2(I)
    ELSE
+   PRINT(23X,A,I2,A,F10.5,8X,A,I2,A,F10.5F,
+   'Y',H(I-1), '=',Y2(I-1),'Y',H(I)), '=',Y2(I)
    ENDIF
    J=0
  ENDF
  IF ((L.EQ.30).AND.(L.NE.M2)) THEN
    L=0
    PRINT(// 25X,A //,'PRESIONE ENTER PARA CONTINUAR'
+   READ(*,*)
    CALL CLEARSCREEN($GCL,EARSSCREEN)
    PRINT(15X,A,I2,A //,'
+   'LA SOLUCION OPTIMA DUAL OBTENIDA EN 'K,
+   ' ITERACIONES ES:'
    ENDF
  END DO
  IF (MOD(M1,2).NE.0) THEN
    IF (M1.T.10) THEN
      PRINT(23X,A,I1,A,F10.5F,'Y',M1, '=',Y1(M1) ELSE
      PRINT(23X,A,I2,A,F10.5F,'Y',M1, '=',Y1(M1) ENDF
    ENDF
    PRINT(// 25X,A //,'PRESIONE ENTER PARA CONTINUAR'
+   READ(*,*)
    CALL CLEARSCREEN($GCL,EARSSCREEN)
    PRINT(15X,A,I2,A //,'
+   'LA SOLUCION OPTIMA DUAL OBTENIDA EN 'K,
+   ' ITERACIONES ES:'
    J=0
    IF (MOD(N1,2).EQ.0) THEN
      N2=N1
    ELSE
      N2=N1-1
    ENDF
    L=0
    DO I=1,N2
      L=L+1
      J=J+1
      H(I)=1
      V2(I)=V1(I)
      IF (I.EQ.2) THEN
+       IF ((L.T.10) THEN PRINT(23X,A,I1,A,F10.5,9X,A,I1,A,F10.5F,
+       'V',H(I-1), '=',V2(I-1),'V',H(I)), '=',V2(I)
+       ELSE
+       PRINT(23X,A,I2,A,F10.5,8X,A,I2,A,F10.5F,
+       'V',H(I-1), '=',V2(I-1),'V',H(I)), '=',V2(I)
+       ENDF
+       J=0
      ENDF
      IF ((L.EQ.30).AND.(L.NE.N2)) THEN
        L=0
        PRINT(// 25X,A //,'PRESIONE ENTER PARA CONTINUAR'
+       READ(*,*)
        CALL CLEARSCREEN($GCL,EARSSCREEN)
        PRINT(15X,A,I2,A //,'
+       'LA SOLUCION OPTIMA DUAL OBTENIDA EN 'K,
+       ' ITERACIONES ES:'
        ENDF
      END DO
    ENDIF
  ENDIF

```

```

END DO
IF (MOD(N1,2) NE 0) THEN
IF (N1.LT.10) THEN
PRINT(23%,A,I2,A,F10.5,'V',N1,' = ',V1(N1))
ELSE
PRINT(23%,A,I2,A,F10.5,'V',N1,' = ',V1(N1))
ENDIF
ENDIF
ENDIF
IF (MXMIN.EQ.-1) PBY=PBV*MXMIN
IF (N1.GE.24) THEN
PRINT(// 13%,A,F12.5,'
+ 'EL VALOR OPTIMO DE LA FUNCION OBJETIVO ES :',PBY
ELSE
PRINT(5// 13%,A,F12.5,'
+ 'EL VALOR OPTIMO DE LA FUNCION OBJETIVO ES :',PBY
ENDIF
IF (.NOT.OP1) THEN
PRINT(13%,A,F6.3,A,F,
+ 'TIEMPO DE CPU EN QUE SE OBTUVO EL OPTIMO :',TIEMPOS,
+ 'SEG.'
ENDIF
PRINT(2// 12%,A ),'PRESIONE ENTER PARA CONTINUAR.'
READ(*,*)
CALL CLEARSCREEN($CLEARSCREEN)
PRINT(4%,A,I2,A,F,
+ 'EL VECTOR ESTIMADO PRIMAL OBTENIDO EN ',K,
+ ' ITERACIONES ES:'
PRINT(// 17%,A,I1%,A,8%,A //,
+ 'VARIABLE',VALOR,'COSTOS REDUCIDOS'
DO I=1,N1
I=I+1
IF (I.LT.10) THEN
PRINT(20%,A,I,9%,F10.
5,8%,F10.5,'
+ ',N1,I(0),V1(I)
ELSE
PRINT(20%,A,I2,8%,F10.5,8%,F10.5,'N1,I(0),V1(I)
ENDIF
ENDIF
IF ((I.EQ.18).AND.(I.NE.N1)) THEN
PRINT(2// 12%,A ),'PRESIONE ENTER PARA CONTINUAR.'
READ(*,*)
CALL CLEARSCREEN($CLEARSCREEN)
PRINT(14%,A,I2,A,F,
+ 'EL VECTOR ESTIMADO PRIMAL OBTENIDO EN ',K,
+ ' ITERACIONES ES:'
I=0
PRINT(// 17%,A,I1%,A,8%,A //,
+ 'VARIABLE',VALOR,'COSTOS REDUCIDOS'
ENDIF
ENDIF
END DO
PCXT=8*BT(N1,C1,I,1,1)
IF (NMIN.EQ.-1) PCXT=PCXT*MXMIN
PRINT(// 9%,A,F12.5,'
+ 'EL VALOR DE LA F.O. PRIMAL CON ESTA ESTIMACION ES :',PCXT
PRINT(2// 12%,A ),'PRESIONE ENTER PARA CONTINUAR.'
READ(*,*)
CALL CLEARSCREEN($CLEARSCREEN)
ENDIF
END IF
IF (FLAG.NE.1) THEN
PRINT(8// 12%,A ),'PRESIONE ENTER PARA CONTINUAR.'
READ(*,*)
ENDIF
RETURN
END

```



```

SUBROUTINE CONTINUA(ASK1,ASK2,ASK3,FD)
CALL CLEARSCREEN($CLEARSCREEN)
FD=0
R1=0
R2=0
R3=0
DO WHILE ((R3.NE.'T').AND.(R3.NE.'F').AND.(R3.NE.'P').AND.
+ (R3.NE.'?'))
  PRINT(// 15X.A // 15X.A.Y,
+ 'DESEA RESOLVER NUEVAMENTE EL PPL PROPORCIONADO PERO,'
+ 'CON OTRO METODO DE FACTORIZACION (T/F)?')
  READ(A7,R3)
END DO
IF ((R3.EQ.'T').OR.(R3.EQ.'F')) ASK3=.TRUE
IF ((R3.EQ.'P').OR.(R3.EQ.'?')) ASK3=.FALSE
IF (.NOT.ASK3) THEN
  DO WHILE ((R2.NE.'T').AND.(R2.NE.'F').AND.(R2.NE.'P').AND.
+ (R2.NE.'?'))
    PRINT(// 15X.A // 15X.A.Y,
+ 'DESEA RESOLVER NUEVAMENTE EL PPL PROPORCIONADO PERO,'
+ 'CON DISTINTOS PARAMETROS (T/F)?')
    READ(A7,R2)
  END DO
  IF ((R2.EQ.'T').OR.(R2.EQ.'F')) ASK2=.TRUE
  IF ((R2.EQ.'P').OR.(R2.EQ.'?')) ASK2=.FALSE
  IF (.NOT.ASK2) THEN
    DO WHILE ((R1.NE.'T').AND.(R1.NE.'F').AND.(R1.NE.'P').AND.
+ (R1.NE.'?'))
      PRINT(// 15X.A // 15X.A.Y,
+ 'DESEA RESOLVER OTRO PROBLEMA DE PROGRAMACION LINEAL,'
+ 'CON ESTA INSTRUMENTACION COMPUTACIONAL (T/F)?')
      READ(A7,R1)
    END DO
    IF ((R1.EQ.'T').OR.(R1.EQ.'F')) ASK1=.TRUE
    IF ((R1.EQ.'P').OR.(R1.EQ.'?')) ASK1=.FALSE
  ENDIF
ENDIF
RETURN
END
SUBROUTINE AUTORIS
CALL CLEARSCREEN($GCLEARSCREEN)
PRINT(6(7),26X.A // 30X.A,8(7),20X.A // 30X.A // 25X.A.Y,
+ 'INSTRUMENTACION COMPUTACIONAL REALIZADA POR:'
+ 'SOFIA LOPEZ AGUILERA Y GUSTAVO BERNIERI FLORES VAZQUEZ'
PRINT(6(7),27X.A.Y,PRIMONE.ENTER PARA CONTINUAR')
READ(*,*)
CALL CLEARSCREEN($GCLEARSCREEN)
RETURN
END
REAL FUNCTION SECND$0)
INTEGER*2 HOUR, MINUTE, SECOND, HUNDREDTH
CALL GETTIME(HOUR,MINUTE,SECOND,HUNDREDTH)
SECND$ = (DBLE(HOUR)*3600.0)+(DBLE(MINUTE)*60.0)+
+ DBLE(SECOND)+(DBLE(HUNDREDTH)/100.0)
END

```

GLOSARIO

F.O.: Función objetivo.

FCK: Forma canónica de Karmarkar.

HIPERPLANO: Es un conjunto de la forma $H = \{x \in \mathbb{R}^n \mid a^t x = \beta\}$ donde β es un escalar y $a \neq 0$ es un vector en \mathbb{R}^n .

INSTRUMENTACIÓN: Programación en computadora.

ITER: Iteraciones.

KKT: Karush-Kuhn-Tucker.

L.I.: Linealmente independiente.

MATRIZ ESPARSA: Una matriz es esparsa si la mayoría de sus elementos tienen valor cero.

MATRIZ NO ESPARSA: Una matriz no esparsa o densa es cuando la mayoría de los elementos de la matriz tienen valor distinto de cero.

MAX: Maximizar.

MIN: Minimizar.

N.A.: No acotación.

ORTANTE POSITIVO: el conjunto de todos los vectores en \mathbb{R}^n con componentes no negativas y se denota como $\mathbb{R}_+^n = \{x \in \mathbb{R}^n \mid x_i \geq 0, i=1, \dots, n\}$

POLIEDRO: Es un conjunto definido por la intersección de un conjunto finito de semiespacios cerrados.

POLIEDRO DE SOLUCIONES: Región Factible del Problema de Programación Lineal..

POLIEDRO SIMPLE: Se dice que P es un poliedro simple cuando en cada una de sus vértices se intersectan exactamente d de los hiperplanos que definen a P .

POLITOPO: Es un poliedro no vacío y acotado.

PL: Programación Lineal.

PPL NO DEGENERADO: Un PPL es no degenerado si todas las soluciones básicas factibles son no degeneradas

PPL: Problema de Programación Lineal.

PUNTO INTERIOR: Dado un conjunto $S \subset \mathbb{R}^n$ y $x \in S$, se dice que x es un punto interior de S si existe un escalar $\epsilon > 0$ tal que la bola abierta $B(x, \epsilon) \subset S$.

SEG: Segundos.

SEMIESPACIO: Un hiperplano H divide a \mathbb{R}^n en dos semiespacios H^+ y H^- , dados por $H^+ = \{x \in \mathbb{R}^n \mid a^T x \geq \beta\}$ y $H^- = \{x \in \mathbb{R}^n \mid a^T x \leq \beta\}$.

SIMPLEJO: Es un poliedro simple y acotado.

SISTEMA HOMOGÉNEO: El sistema $Ax = b$ de m ecuaciones lineales en n incógnitas, se conoce como homogéneo si todas las constantes b_i para $i = 1, \dots, m$ son cero.

SOL. INF.: Solución infactible.

SOLUCIÓN INTERIOR: Es un vector cuyos componentes son estrictamente positivas (es decir, $x > 0$).

SOLUCIÓN INTERIOR FACTIBLE: Es una solución interior que satisface las restricciones del PPL.

VALOR O PARÁMETRO POR DEFAULT: Son los valores establecidos en el programa instrumentado.

BIBLIOGRAFÍA

- [ABR] Allende, S., Bermúdez, B., Romero, I.
"Experiencias en la Instrumentación Computacional del Método
Proyectivo de Karmarkar para Computadoras Personales".
Revista Investigación Operacional, Vol. 14, 1993, pp. 58-72.
- [AKR] Adelr, I., Karmarkar, N., Resende, M. and Veiga, G.
"An Implementation of Karmarkar's algorithm for Linear
Programming".
Mathematical Programming, Vol. 44, 1989, pp. 297-335.
- [Arb1] Arbel, A.
"An Interior Multiobjective Linear Programming Algorithm".
Computers and Operations Research, Vol. 20, No. 7, 1993, pp.
723-735.
- [Arb2] Arbel, A.
"Anchoring Points and Cones of Opportunities in Interior
Multiobjective Linear Programming".
Journal of Operations Research Society, Vol. 45, No. 1, 1994, pp.
83-96.
- [ASB] Allende, S., Bermúdez, B. y Romero, I.
"Experiencias en la Instrumentación Computacional del Método
Proyectivo de Karmarkar para Computadoras Personales".
Revista Investigación Operacional, Vol. 14, No. 1, 1993, pp. 58-
72.
- [Bar] Barnes, E.
"A Variation on Karmarkar's Algorithm for Solving Linear
Programming Problems".
Mathematical Programming, Vol. 36, 1986, pp. 174-182.
- [Bat] Bartle, R.
"Introducción al Análisis Matemático"
Editorial Limusa, México, 1987.
- [Baz] Bazaraa, M., Jarvis, J. and Sherali, H.
"Linear Programming and Network Flows".
John Wiley & Sons, U.S.A., 1990.

BIBLIOGRAFÍA

- [BCJ] Barnes, E., Chopra, S. and Jensen, D.
 "A Polynomial Time Version of the Affine Scaling Algorithm".
 I.B.M. Research Division, T.J: Watson Center, Yorktown Heights,
 N.Y., 1988, 39 pages.
- [Cal] Calvillo, G.
 "Métodos de la Programación Lineal".
 V Coloquio del Departamento de Matemáticas,
 Centro de Investigación y Estudios Avanzados del I.P.N., México,
 1987.
- [Col] Colmenares, O.
 "Karmarkar's Linear Programming Algorithm Better or Worse
 Than the Classical Method?".
 Graduate School of Management, UCLA, pp. 1-29.
- [CyM] Cavalier, T. and Sosyter, A.
 "Some Computational Experience and a Modification of the
 Karmarkar's Algorithm".
 The Pennsylvania State University, ISME Working Paper, 1985,
 pp. 85-105.
- [Dan] Dantzig, G.
 "Linear Programming and Extensions".
 Princeton University Press, Princeton, New Jersey, U.S.A., 1963.
- [Dik] Dikin, I.
 "Iterative Solution of Problems of Linear and Quadratic
 Programming".
 Doklady Akademii Nauk SSSR, Vol 174, 1967, pp. 174-182.
 Soviet Math. Dokl. Vol. 8, 1967, pp. 674-675.
- [DMB] Dongarra, J., Bonch, J., Moller, C. and Stewart, W.
 "LINPAK User's Guide".
 SIAM, Philadelphia
- [DyB] Daellenbach, H. and Bell, E.
 "User's Guide to Linear Programming".
 Prentice Hall, 1970.
- [Fan] Fang, S. and Puthenpura, S.
 "Linear Optimization and Extensions: Theory and Algorithms".
 Prentice Hall, Englewood Cliffs, New Jersey, U.S.A., 1993.

BIBLIOGRAFÍA

- [FyK] Friedman, F. and Koffman, E.
"FORTRAN"
"Introducción al Lenguaje y Resolución de Problemas con Programación Estructurada".
Fondo Educativo Interamericano, México, 1986.
- [GMS] Gill, P., Murray, W. and Saunders, M.
"On Projective Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar's Projective Method".
Mathematical Programming, Vol. 36, 1986, pp. 183-209.
- [GMW] Gill, P., Murray, W. and Wright, M.
"Numerical Linear Algebra and Optimization".
Addison-Wesley Publishing Company, Vol. 1, U.S.A., 1991.
- [Gro] Grossman, S.
"Álgebra Lineal".
Grupo Editorial Iberoamérica, México., 1983
- [HBA] Hernández, C., Bermúdez, B., Allende, S.
"Comparación Computacional del Método de Karmarkar y un Método de Barrera para Problemas de Programación Lineal".
Proyecto: Métodos de Optimización No Lineal y sus Aplicaciones.
Benemérita Universidad Autónoma de Puebla, pp. 1-20.
- [Her] Hernández, C.
"Comparación Computacional del Método de Karmarkar y un Método de Barrera para Problemas de Programación Lineal".
Tesis, Universidad Autónoma de Puebla, México.
- [Hoo] Hooker, J.
"Karmarkar's Linear Programming Algorithm".
Interfaces, Vol. 16, No. 4, 1986, pp. 75-90.
- [IMSL] User's Manual.
"IMSL Math/Library".
FORTRAN Subroutines for Mathematical Applications.
MALB-USM-PERFCT-EN8912-1.1, Versión 1.1
Houston, Texas, 1989.
- [Kar] Karmarkar, N.
"A New Polynomial - Time Algorithm for Linear Programming".
Combinatorica, Vol. 4, 1984, pp. 373-395.

- [Kha] Khachiyan, L.
"A Polynomial Algorithm in Linear Programming".
Traducido en Soviet Mathematics Doklady 20, 1979, pp. 191-194.
- [KyM] Klee, V. and G. Minty.
"How Good is the Simplex Algorithm?".
Inequalities III, O. Shisha, Editor, Academic Press, New York,
1972, pp. 159-175.
- [MMT] Martínez, E., Mejía, J. y Tapia, H.
"Elementos de Programación Lineal".
Centro de Investigación y de Estudios Avanzados del I.P.N.,
México, 1980.
- [MSS] Marsten, R., Saltzman, M., Shanno, D., Pierce, G.
"Implementation of a Dual Affine Interior Point Algorithm for Linear
Programming".
ORSA Journal on Computing, Vol. 1, No. 4, 1989, pp 287-297.
- [MyM] Monma, C. and Morton, A.
"Computational Experience with a Dual Affine Variant of
Karmarkar's Method for Linear Programming".
Operations Research Letters, Vol. 6, 1987, pp 261-267.
- [MyO] Martínez, E. y Ortega, J.
"Programación Matemática".
Serie en Ciencias e Ingeniería, Vol. 5.
SEPI-ESIME, Instituto Politécnico Nacional, México 1994.
- [PQV] Ponnambalam, K., Quintana, V. and Vanelli, A.
"A Fast Algorithm for Power System Optimization Problems Using
an Interior Point Method".
IEEE Transactions on Powers Systems, Vol. 7, No. 2, 1992, pp.
892-899.
- [Pra] Prawda, J.
"Métodos y Modelos de Investigación de Operaciones".
Limusa, Vol. 1 (Modelos Determinísticos), México, 1989.
- [RyS] Rockett, A. y Stevenson, J.
"Karmarkar's Algorithm, a Method for Solving Large Linear
Programming Problems".
Byte, September 1987, pp 146-160.

BIBLIOGRAFÍA

- [Sou] Souza, P.
"El Método Elipsoidal".
Tesis, ITAM, México, 1984.
- [Tah] Taha, H.
"Investigación de Operaciones, una Introducción".
Representaciones y Servicios de Ingeniería, México, 1981
- [Van] Vanelli, A.
"Teaching Large-Scale Optimization by an Interior Point Approach".
IEEE Transactions on Education, Vol. 36, No. 1, 1993, pp. 204-209.
- [VMF] Vanderbei, R., Meketon, M. and Freedman, B.
"A Modification of Karmarkar's Linear Programming Algorithm".
Algorithmica, Vol. 1, 1986, pp. 395-407.
- [VQV] Vargas, L., Quintana, V. and Vanelli, A.
"A Tutorial Description of an Interior Point Method and its Applications to Security-Constrained Economic Dispatch".
IEEE Transactions on Powers Systems, Vol. 8, No. 3, 1993, pp. 1315-1324.