

Universidad Nacional de Ingeniería

Facultad de Ciencias



TESIS

**ALGORITMO DE PUNTO INTERIOR Y DIRECCIONES FACTIBLES
FDIPA EN PROBLEMAS DE PROGRAMACIÓN LINEAL DE GRAN
ESCALA**

Para obtener el título profesional de licenciado en Matemática.

Elaborado por

Miguel Cutipa Coaquira

 [0000-0002-5097-4677](https://orcid.org/0000-0002-5097-4677)

Asesor

Mg. Jesús Cernades Gómez

 [0000-0003-3671-4358](https://orcid.org/0000-0003-3671-4358)

LIMA – PERÚ

2025

Citar/How to cite	Cutipa Coaquira [1]
Referencia/Reference	[1] M. Cutipa Coaquira, “ <i>Algoritmo de punto interior y direcciones factibles FDIPA en problemas de programación de gran escala</i> ” [Tesis de pregrado]. Lima (Perú): Universidad Nacional de Ingeniería, 2025.
Estilo/Style: IEEE	

Citar/How to cite	(Cutipa, 2025)
Referencia/Reference	Cutipa, M. (2025). <i>Algoritmo de punto interior y direcciones factibles FDIPA en problemas de programación de gran escala</i> . [Tesis de pregrado, Universidad Nacional de Ingeniería]. Repositorio Institucional UNI.
Estilo/Style: APA (7ma ed.)	

Dedicatoria

*A mis padres, por el amor y confianza que me dieron, el ejemplo de esfuerzo y
perseverancia que aprendí de ellos.*

Agradecimientos

A mi esposa Gregoria, mis padres, Emeterio Cutipa e Isidora Coaquira, y a mis hermanos Martha, Jorge, Carlos, Norma y Edith, por la compañía y apoyo recibido durante todos estos años de estudio que he compartido.

A mi asesor y amigo Jesús Cernades por su consejo y ayuda en la elaboración del presente trabajo, que durante largas conversaciones y discusiones surgió la curiosidad de estudiar este problema. También, agradecer a mi amigo Daniel Gutierrez, por las veces que compartimos ideas.

A mis entrañables amigos de la universidad, Angel, José, Diana, Rubén, Juan y mucho más, por su compañía y alegría que hicieron este camino mucho mas ligero y agradable. Además, de compartir siempre nuestras experiencias que ayudaron mucho a seguir mejorando. A mis profesores William, Irla, Fidel e otros por sus enseñanzas compartidas en cada hora de clase y la exigencia de siempre para ser mejores profesionales.

Resumen

La presente tesis estudia el algoritmo de punto interior y dirección factible FDIPA de programación no lineal adaptado a problemas de programación lineal de gran escala con restricciones de desigualdad. El desarrollo inicia con la formulación del problema lineal a restricciones de desigualdad, seguido de métodos presolver y técnicas de reacondicionamiento que analizan la infactibilidad y mal condicionamiento, el problema es sometido a las condiciones Karush Kuhn Tucker (KKT) que por el método de Newton determina una dirección de descenso a partir de un punto interior inicial, la factibilidad de la siguiente iteración es determinada por el tamaño de paso óptimo que se obtiene analizando las filas de la matriz de restricciones con la dirección de descenso encontrada, obteniendo una sucesión de puntos interiores convergiendo a una solución del problema. Además, se estudian las hipótesis y condiciones de convergencia del problema. El nuevo algoritmo ha sido nombrado algoritmo FDIPA-LP. Los problemas test de gran escala son tomados de la librería NETLIB, los resultados obtenidos son muy buenos teniendo comparaciones de hasta orden 10^{-10} , también es aplicado en resolución de problemas de transporte que comparado con el método de la esquina noroeste y la aplicación de software WINQSB, ha mostrado resultados similares a los encontrados. El algoritmo FDIPA-LP es sencillo de implementar, eficiente y eficaz, siendo una alternativa al método simplex y otros métodos de punto interior que puede ser aplicado en casos de gran complejidad y otras áreas afines.

Palabras Claves: programación lineal de gran escala, método de punto interior, direcciones factibles, primal-dual, problema de transporte.

Abstract

This paper studies the FDIPA feasible direction and interior point algorithm for nonlinear programming adapted to large-scale linear programming problems with inequality constraints. The development starts with the formulation of the linear problem with inequality constraints, followed by presolver methods and reconditioning techniques that analyse infeasibility and illconditioning, the problem is subjected to the Karush Kuhn Tucker (KKT) conditions that by Newton's method determines a direction of descent from an initial interior point, the feasibility of the next iteration is determined by the optimal step size which is obtained by succession of interior points converging to a solution of the problem. In addition, the hypotheses and convergence conditions are studied. The new algorithm has been named the FDIPA-LP algorithm. The large scale test problems are taken from the NETLIB library, the results obtained are very good with comparisons of order 10^{-10} , it is also applied to transport problems which compared to the northwest corner method and the application of WINQSB software, has shown similar results to those found. The FDIPA-LP algorithm is simple to implement, efficient and effective, being an alternative to the simplex method and other interior point methods that can be applied in cases of high complexity and other related areas.

Keywords — large scale linear programming, interior point method, feasible directions, primal-dual, transport problem.

Tabla de Contenido

	Pág.
Resumen.....	v
Abstract.....	vi
Introducción.....	xiii
I. Parte introductoria de la tesis.....	1
A. Motivación.....	1
B. Descripción del problema de investigación.....	2
C. Objetivos del estudio.....	3
1) Objetivo general.....	3
2) Objetivos específicos.....	3
D. Hipótesis.....	3
E. Antecedentes.....	4
II. Marco teórico y preliminares.....	5
A. Notaciones básicas.....	5
B. Conceptos básicos de optimización.....	7
C. Dualidad de problemas de programación lineal.....	12
D. Estrategias de solución.....	18
1) Método de Newton.....	19
2) Método de Quase-Newton.....	21
E. Métodos de búsqueda inexacta.....	22
1) Condición de Armijo.....	23
III. Algoritmo FDIPA.....	25
A. Estudio del algoritmo FDIPA.....	25
B. Análisis de convergencia.....	33
C. Ejemplos de aplicación	45

IV. Algoritmo FDIPA-LP.....	50
A. Estudio del algoritmo FDIPA-LP.....	50
B. Análisis de convergencia.....	57
V. Implementación numérica y resultados.....	66
A. Problema primal-dual.....	66
B. Resolución de sistemas lineales internos en FDIPA-LP.....	67
C. Aplicación en problemas de programación lineal de gran escala.....	73
D. Aplicación en problemas de transporte.....	75
VI. Conclusiones y discusiones.....	90
A. Conclusiones	90
B. Discusiones y recomendaciones.....	91
VII. Referencias bibliográficas.....	92

Lista de Tablas

	Pág.
Tabla I: Clasificación de problemas de optimización según Ω	8
Tabla II: Relación entre problema primal y dual.....	16
Tabla III: Aplicación de métodos presolver.....	72
Tabla IV: Tipos de técnicas de reescalamiento.....	72
Tabla V: Comparación del método Dual-simplex con el algoritmo FDIPA-LP.....	74
Tabla VI: Coste unitario de transporte del Ejemplo IV, [28].....	78
Tabla VII: Resultados del algoritmo FDIPA-LP del Ejemplo IV.....	83
Tabla VIII: Costo unitario de transporte del Ejemplo V, [28].....	85
Tabla IX: Resultados del algoritmo FDIPA-LP del Ejemplo V.....	89

Lista de Figuras

	Pág.
Figura I. Búsqueda de Armijo.....	24
Figura II. Deflexión de d_α en el punto x^k del Problema (I).....	29
Figura III. Región factible y solución optima $\bar{x} = (2, 1)^T$	35
Figura IV. Iteraciones del Ejemplo I por el algoritmo FDIPA	47
Figura V. Iteraciones del Ejemplo II por el algoritmo FDIPA	48
Figura VI. Iteraciones del Ejemplo III por el algoritmo FDIPA	49
Figura VII. Solución optima del Ejemplo IV, Fuente [28].....	84
Figura VIII. Solución optima del Ejemplo IV, por WINQSB. Fuente [28].....	84
Figura IX. Solución optima del Ejemplo V, por WINQSB. Fuente [28].....	87

Lista de notaciones

Aquí se muestra una lista de las notaciones usadas a lo largo de la tesis.

\mathbb{R}^n --- Espacio euclideo real de dimensión n .

$x \geq 0$ --- Elemento $x \in \mathbb{R}^n$ de componentes no negativas.

\mathbb{R}_+^n --- Subconjunto de \mathbb{R}^n de elementos de componentes no negativas.

$\dot{\Omega}$ --- Interior del subconjunto $\Omega \subset \mathbb{R}^n$.

$\bar{\Omega}$ --- Cerradura del subconjunto $\Omega \subset \mathbb{R}^n$.

$\sqrt{\langle v, w \rangle} = v^T w$ --- Producto interno de los vectores v y w .

$\|v\|_2$ --- Norma euclidea del vector v .

$x^k \rightarrow x^*$ --- Convergencia de sucesión de puntos $\{x^k\}_{k \in \mathbb{N}}$ a x^* .

$\mathbb{R}^{m \times n}$ --- Conjunto de matrices de componentes reales de m filas y n columnas.

$\text{diag}\{x_1, \dots, x_n\}$ --- Matriz diagonal de diagonal principal x_1, \dots, x_n .

$x < y$ --- Las componentes $x_i < y_i$ para todo $i \in \{1, 2, \dots, n\}$.

$|A|$ o $\det(A)$ --- Determinante de una matriz cuadrada A .

$A \geq 0$ --- Matriz cuadrada semidefinida positiva.

$A > 0$ --- Matriz cuadrada definida positiva.

$\text{grad } f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right)$ --- Gradiente de la función f en x .

$\nabla f(x)$ --- Tranpuesta del gradiente de f en x .

$\nabla^2 f(x)$ --- Matriz Hessiana de la función f en x .

$Jg(x)$ --- Matriz jacobiana de g en x , donde $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$.

$G(x) = \text{diag}\{g_1(x), \dots, g_m(x)\}$ --- Matriz diagonal de componentes la función $g(x) = (g_1(x), \dots, g_m(x))$.

$L(x, \lambda, \mu)$ --- Lagrangiano del problema $\min f(x)$ s.a. $\Omega = \{x \in \mathbb{R}^n: g(x) \leq 0 \text{ y } h(x) = 0\}$.

$H(x, \lambda) = \nabla_{xx}^2 L(x, \lambda)$ --- Matriz hessiana del Lagrangiano de $f(x)$ con respecto a la x .

$d(x)$ --- Campo uniforme de direcciones factibles en x para un problema de optimización.

$\Omega_a = \{x \in \Omega: f(x) \leq a\}$ ---- Conjunto de nivel de f en a .

$\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$ --- Matriz diagonal de la variable $\lambda = (\lambda_1, \dots, \lambda_m)$.

Introducción

El trabajo de la tesis, muestra el estudio del algoritmo FDIPA (Algoritmo de punto interior de direcciones factibles), en la resolución de problemas de programación lineal de gran escala de la forma:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.a.} \quad & Ax \leq b \\ & x \text{ irrestricto.} \end{aligned} \tag{1}$$

Inicialmente el Capítulo I, detalla la motivación, el planteamiento del problema, el objetivo general y específicos que se espera obtener, antecedentes e hipótesis del trabajo. El Capítulo II, presenta las diferentes notaciones, conceptos básicos, teoremas y métodos utilizados en el trabajo, así como la teoría básica de dualidad en problemas de programación lineal. El Capítulo III estudia el algoritmo FDIPA, detallando las hipótesis, el funcionamiento del algoritmo, el análisis de la convergencia y ejemplos de la aplicación. El Capítulo IV desarrolla el algoritmo FDIPA en problemas de programación lineal, llamado algoritmo FDIPA-LP, este caso disminuye las hipótesis del algoritmo FDIPA, el algoritmo FDIPA-LP presenta una forma más simple que el algoritmo FDIPA. El Capítulo V, muestra cómo se desarrolla la implementación numérica del algoritmo FDIPA-LP en el caso de Problemas de Gran Escala, la dificultad de resolver sistemas lineales de gran tamaño exige aplicar métodos que no generen mal condicionamiento, en este caso se aplica el método de Gradiente Conjugado Precondicionado Split (GCPS), se presentan los resultados de comparar el método Dual-simplex y el algoritmo FDIPA-LP en 30 problemas Test de Gran escala, tomados de la librería NETLIB. Además, se muestran resultados de aplicar el algoritmo en algunos ejemplos de Problema de Transporte. El Capítulo VI, muestra las conclusiones y discusiones. El Capítulo VII presenta todas las referencias del trabajo.

I Parte introductoria de la tesis

A. Motivación

La minimización o maximización de una función objetivo en muchos campos de la ciencia y la ingeniería es un desafío que surge del uso de modelos matemáticos que están sujetos a un conjunto de restricciones de igualdad o desigualdad. La resolución de estos modelos son problemas que están sujetos principalmente a la forma de sus restricciones: la linealidad y no linealidad, cantidad de restricciones, la diferenciabilidad; y el tipo de variables discretas o continuas.

Las aplicaciones de los problemas de programación lineal son diversas como: En la industria manufacturera para optimizar el uso de recursos (materiales, mano de obra y maquinaria); el sector energético, en la planificación y operación de sistemas de generación eléctrica; la planificación del transporte público optimizando rutas, horarios y asignación de vehículos; diseño y optimización de redes de telecomunicaciones, en la planificación de la capacidad de red y la asignación de recursos, garantizando una cobertura eficiente y de alta calidad [1], [2].

Los principales métodos de resolución de problemas de programación lineales son: El método Simplex y de punto interior. El método simplex es un algoritmo que busca la solución optima en la frontera de la región factible generada por las restricciones (los vértices del polítopo) [3]. Los métodos de punto interior son otra opción para determinar la solución optima, como el método de punto interior de Karmarkar [4] que inicia con un punto interior estricto y por medio de proyecciones sobre una esfera inscrita en la región factible genera una sucesión de puntos interiores que converge a la solución optima. En problemas de programación lineal de gran escala, los métodos de punto interior suelen superar al método Simplex debido a su complejidad polinomial y su capacidad para manejar un gran número de variables y restricciones de

manera eficiente [5]. El trabajo estudia un algoritmo de punto interior aplicado en problemas de programación lineal de gran escala, mostrando resultados muy eficientes en ejemplos test.

B. Descripción del problema de investigación

Los problemas de programación lineal como hemos mencionado tiene una cantidad variada de aplicaciones en la actualidad, muchos casos son de problemas de gran complejidad por la enorme cantidad de información que se debe manejar. La aplicación del método Simplex no es adecuado por que su velocidad de resolución de tipo exponencial al número de variables y restricciones [15], mientras que el método de punto interior es una alternativa mas adecuada pues su velocidad es de tipo polinomial [4]. Los problemas de programación lineal de gran escala, tienen una cantidad grande de variables y restricciones, dificultando su resolución por el método Simplex. Los métodos de punto interior han demostrado tener muy buenos resultados para este caso. Los problemas de gran escala pueden presentar dificultades como: restricciones redundantes o de infactibilidad, mal condicionamiento y resolución de sistemas lineales. Son necesarios aplicar métodos presolver's y técnica de escalamiento para mejorar las condiciones del problema [16]. Los sistemas lineales que resuelve el algoritmo son solucionados por el método de gradiente conjugado pre condicionado split [17], juntamente con el método preconditionado de Cholesky incompleto, (como la función ICHOL de Matlab). La programación de estos algoritmos son implementadas en Matlab, por ser un software de uso practico en algebra lineal. El algoritmo FDIPA-LP sugiere ser un algoritmo alternativo para resolver problemas de programación lineal de gran escala. El estudio tambien se aplica en problemas de transporte como un método alternativo a los ya conocidos en la literatura, alcanzando aproximadamente el valor óptimo del problema.

C. Objetivos del estudio

1). Objetivo general

El objetivo general del trabajo es: "Determinar la implementación del algoritmo punto interior de direcciones factibles FDIPA en problemas de programación lineal de gran escala". El objetivo se alcanzará desarrollando los siguientes objetivos específicos:

2). Objetivos específicos

- Determinar la formulación del problema de programación lineal con restricciones de igualdad a su forma dual para la implementación del algoritmo de punto interior FDIPA.
- Adaptar el algoritmo de punto interior FDIPA a problemas de programación lineal con restricciones de desigualdad para la implementación del algoritmo FDIPA.
- Implementar el algoritmo de punto interior FDIPA-LP para resolver problemas de programación lineal de gran escala.

D. Hipótesis

La implementación del algoritmo de punto interior FDIPA resuelve problemas de programación lineal de gran escala.

- La formulación del problema de programación lineal a su forma dual ayuda al algoritmo de punto interior FDIPA a resolver problemas de programación lineal de gran escala.
- La adaptación del algoritmo de punto interior FDIPA a problemas de programación lineal con restricciones de desigualdad resuelve problemas de programación lineal de gran escala.

- La implementación del algoritmo de punto interior FDIPA-LP resuelve problemas de programación lineal de gran escala.

E. Antecedentes

A partir del algoritmo de Karmarkar [4], se plantean otros algoritmos de puntos interiores que varían el tipo de transformación y las direcciones de búsqueda factible. Algunos ejemplos son el método proyectivo estudiado por Gonzaga [6], o el método de escala afín estudiado por Barnes [7] o el método de "path following" por Gonzaga [8]. Para un mejor estudio de estos métodos se puede revisar el libro de Wright [5] que trata sobre los métodos de puntos interiores.

El algoritmo de FDIPA es un método de punto interior en problemas de programación no lineal (PNL), formulado por Herskovits [9], basado en la generación de una sucesión de puntos interiores factibles que convergen para una solución factible (satisfaciendo las condiciones de Karush Kuhn Tucker), la ventaja del método es su sencillez de ser aplicado, la robustez y eficiencia para alcanzar una solución. Este método ha sido satisfactoriamente aplicado en problemas de ingeniería, dinámica de fluidos, electromagnetismo, [10], [11], [12].

El algoritmo FDIPA tiene una primera adaptación en problemas de programación lineal en Tits [13], encontrando del primer sistema de FDIPA una dirección de descenso y también hace uso del Hessiano exacto con un paso adecuado para garantizar la factibilidad. Vitorio [14] estudia la aplicación del algoritmo FDIPA en problemas de programación lineal, haciendo uso de los dos sistemas de FDIPA y simplifica el análisis de la condición de Armijo, solo quedando un análisis del sistema de restricciones. El algoritmo queda nombrado como FDIPA-LP. El estudio del algoritmo FDIPA y FDIPA-LP son presentados a lo largo del trabajo.

II Marco teórico y preliminares

El capítulo presenta las diferentes notaciones, conceptos básicos, teoremas y métodos que son utilizadas para el estudio de los algoritmos FDIPA y FDIPA-LP. Además, la teoría de dualidad básica en problemas de programación lineal.

A. Notaciones básicas

La sección presenta las notaciones de topología en \mathbb{R}^n , cálculo diferencial, álgebra lineal y optimización empleadas en todo del desarrollo de la tesis.

El espacio euclideo real de dimensión n es representado por \mathbb{R}^n . Escribimos $x \geq 0$ a los elementos $x \in \mathbb{R}^n$ de componentes no negativas o $x \in \mathbb{R}_+^n$. La notación v^T representa la transpuesta de un vector $v \in \mathbb{R}^n$. Dado un subconjunto $\Omega \subset \mathbb{R}^n$ se denota por $\mathring{\Omega}$ el interior del conjunto y $\overline{\Omega}$ su cerradura. Dados los vectores $v, w \in \mathbb{R}^n$, $v^T w = \langle v, w \rangle$ representa el producto interno entre v e w . La norma de un vector v es definida por $\|v\| = \sqrt{\langle v, v \rangle}$, el caso particular es la norma euclidea denotada por $\|v\|_2$. Se denota $x^k \rightarrow x^*$ a una sucesión de elementos de $\{x^k\}_{k \in \mathbb{N}} \subset \mathbb{R}^n$ que converge a un punto x^* .

Se denota $\mathbb{R}^{m \times n}$ al conjunto de matrices $A = [a_{ij}]$ de componentes reales conformada de m filas y n columnas. Se escribe $\text{diag}\{x_1, \dots, x_n\}$ la matriz diagonal de orden n con diagonal principal (x_1, \dots, x_n) .

$$\text{diag}\{x_1, \dots, x_n\} = \begin{pmatrix} x_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & x_n \end{pmatrix}.$$

$|A|$ ó $\det(A)$ es el determinante de la matriz A de orden n .

Es usual escribir $A \geq 0$, si A es semidefinida positiva y $A > 0$ cuando A es definida

positiva, donde A es una matriz cuadrada de orden n .

Definición I. Dada la función $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciable en $x \in \overset{\circ}{\Omega}$, es definido el gradiente de f en x por: $\text{grad}f(x) = (\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n})$. También se denota por $\nabla f(x)$, la transpuesta del gradiente de f en x . Si la función f es dos veces diferenciable en x se denota por $\nabla^2 f(x)$ o $H(x)$ la matriz Hessiana de la función en x . Se define respectivamente cada uno de la siguiente forma:

$$\nabla f(x) = \text{grad}f(x)^T = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix} \quad \text{y} \quad \nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{pmatrix}.$$

Definición II. Decimos $f \in C^k(\Omega)$ para indicar que las derivadas parciales de f hasta orden k son continuas en $\overset{\circ}{\Omega}$.

Definición III. Si $g : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ es diferenciable en $x \in \overset{\circ}{\Omega}$, la notación $Jg(x) \in \mathbb{R}^{n \times m}$ indica la matriz jacobiana de g en x , suponiendo que $g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T$, se tiene que:

$$\nabla g(x) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1}(x) & \cdots & \frac{\partial g_m}{\partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial x_n}(x) & \cdots & \frac{\partial g_m}{\partial x_n}(x) \end{pmatrix} = Jg(x)^T = (\nabla g_1(x), \dots, \nabla g_m(x)).$$

Definición IV. La función $G(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$ denota una matriz diagonal asociada a una función $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, tal que $G_{ii}(x) = g_i(x)$ para todo $i \in \{1, 2, \dots, m\}$ y $G_{ij}(x) = 0$, para todo $i \neq j$.

j .

$$G(x) = \begin{pmatrix} g_1(x) & 0 & 0 & \dots & 0 \\ 0 & g_2(x) & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & g_m(x) \end{pmatrix}.$$

B. Conceptos básicos de optimización

El tipo de problema que estudia la optimización puede ser considerado como un problema de la forma:

$$\text{Minimizar } f(x) \text{ sujeto a } x \in \Omega \subset \mathbb{R}^n. \quad (2)$$

La función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es llamada función objetivo y el conjunto Ω es llamado conjunto factible, Ω es definido por un conjunto de igualdades y/o desigualdades. Los puntos de Ω son llamados puntos factibles del problema (2). También, se puede buscar, maximizar la función objetivo, pero es suficiente analizar la minimización pues minimizar $-f(x)$ es igual a maximizar $f(x)$ o viceversa. Las soluciones $x^* \in \Omega$ son llamadas minimizadores si sus respectivos valores $f(x^*)$ son mínimos del problema (2). Debemos distinguir entre un minimizador local y global.

Definición V. Decimos que un punto x^* es un minimizador global del problema (2), si

$$f(x^*) \leq f(x), \quad \forall x \in \Omega.$$

Definición VI. Decimos que un punto x^* es un minimizador local del problema (2), si existe una vecindad V de x^* tal que

$$f(x^*) \leq f(x), \quad \forall x \in \Omega \cap V.$$

Definición VII. Decimos que el valor $\bar{v} \in [-\infty, +\infty[$ definido por

$$\bar{v} = \inf_{x \in \Omega} f(x)$$

es el valor óptimo del problema (2).

El problema (2) puede admitir varios minimizadores globales, mas el valor óptimo del problema es siempre el mismo (único). El problema (2) se puede clasificar según la forma del conjunto factible, el Cuadro I resume los diferentes problemas de optimización, según el conjunto factible Ω que presente.

TABLA I
Clasificación de problemas de optimización según Ω .

Ω	Problema
\mathbb{R}^n	minimización sin restricciones
$\{x \in \mathbb{R}^n : l \leq x \leq u\}$	minimización en intervalos cerrados o cajas
$\{x \in \mathbb{R}^n : Ax = b, A \in \mathbb{R}^{m \times n}\}$	minimización con restricciones lineales de igualdad
$\{x \in \mathbb{R}^n : Ax = b, Cx \leq d, A, C \in \mathbb{R}^{m \times n}\}$	minimización con restricciones lineales
$\{x \in \mathbb{R}^n : h(x) = 0, h : \mathbb{R}^n \rightarrow \mathbb{R}^m\}$	minimización con restricciones de igualdad
$\{x \in \mathbb{R}^n : h(x) = 0, h : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ y } g(x) \leq 0, g : \mathbb{R}^n \rightarrow \mathbb{R}^p\}$	problema general de programación no lineal

La notación $l \leq x \leq u$ indica las desigualdades de cada componente $l_i \leq x_i \leq u_i$ para todo $i = 1, \dots, n$.

El algoritmo FDIPA generalmente se aplica para resolver problemas de optimización no lineal, denominados problemas de programación no lineal. El problema de programación no lineal mas aplicado es el problema con restricciones de desigualdad, [9]. El objetivo del estudio, es: "Determinar la implementación del algoritmo punto interior FDIPA en problemas de programación lineal de gran escala". planteado en [14].

A continuación se define de forma particular el tipo de problema que pueden ser resuelto

por cada algoritmo.

Problema I. *Problema de programación no lineal con restricciones de desigualdad,*

$$\begin{cases} \text{mín } f(x) \\ \text{s.a. } x \in \Omega, \end{cases} \quad (3)$$

donde $\Omega = \{x \in \mathbb{R}^n : g(x) \leq 0\}$. Las funciones $f : \mathbb{R}^n \rightarrow \mathbb{R}$ y $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ deben ser al menos de clase C^1 . Si $\Omega = \mathbb{R}^n$, el Problema I define, un problema de optimización sin restricciones. El caso de las desigualdades sean estrictas queda denotado por $\overset{\circ}{\Omega} = \{x \in \mathbb{R}^n : g(x) < 0\}$.

El Lagrangiano asociado a un problema de programación no lineal I con variable dual $\lambda \in \mathbb{R}^m$ se define por:

$$L(x, \lambda) = f(x) + g(x)^T \lambda = f(x) + \sum_{i=1}^m \lambda_i g_i(x),$$

con matriz Hessiana $H(x, \lambda) = \nabla_{xx}^2 L(x, \lambda) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 g_i(x)$. El caso general del Lagrangiano para un Problema general de programación no lineal (I), con las restricciones $\Omega = \{x \in \mathbb{R}^n : g(x) \leq 0 \text{ y } h(x) = 0\}$, ver I, es dado por la función $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$,

$$L(x, \lambda, \mu) = f(x) + g(x)^T \lambda + h(x)^T \mu, \quad \text{donde } \lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^p.$$

El algoritmo FDIPA-LP determina el valor óptimo de un problema de programación lineal, definido a continuación.

Problema II. *Problema de programación lineal con restricciones lineales de igualdad (forma estándar).*

$$\begin{cases} \text{mín } f(x) \\ \text{s.a. } x \in \Omega \end{cases} \quad (4)$$

donde $\Omega = \{x \in \mathbb{R}^n : Ax = b \text{ y } x \geq 0\}$, $A \in \mathbb{R}^{m \times n}$. La función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es definida por

$$f(x) = c^T x, c \in \mathbb{R}^n.$$

El Problema (II), también tiene asociado su función Lagrangiana, que es utilizada para determinar el dual de este Problema. En la siguiente sección se estudiará, como es calculado el Problema dual asociado a un Problema de programación lineal.

Definición VIII. Se denota al vector $d \in \mathbb{R}^n$ como dirección de descenso para una función $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ en el punto $x \in \mathbb{R}^n$ si existe un $\delta > 0$ tal que $\phi(x + td) < \phi(x)$, para todo $t \in]0, \delta[$.

Definición IX. Decimos que $d \in \mathbb{R}^n$ es una dirección de descenso para una función $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ de clase \mathcal{C}^1 en $x \in \mathbb{R}^n$, si $d^T \nabla \phi(x) < 0$.

Definición X. El vector $d \in \mathbb{R}$ define una dirección factible en $x \in \Omega$, si para algún $\theta > 0$, tenemos que $x + td \in \Omega$ para todo $t \in]0, \theta[$.

Definición XI. Si $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ es una función $\mathcal{C}^1(\Omega)$, un vector $d \in \mathbb{R}^n$ tal que $d^T \nabla \phi_i(x) < 0$, para todo $i \in \{1, \dots, m\}$ y $\phi_i(x) = 0$, entonces d es una dirección factible en x .

Definición XII. Se define $d(x)$ el conjunto de vectores x en Ω , como el campo uniforme de direcciones factibles del problema (I) si para algún $\tau > 0$ tal que $x + td(x) \in \Omega$ para todo $t \in [0, \tau]$ y para todo $x \in \Omega$, [18].

Las condiciones de optimalidad de primer orden de Karush Kuhn Tucker, es entendido como condiciones necesarias para posibles puntos de mínimo local en un problema general de optimización (I). El criterio se resume en el siguiente Teorema.

Teorema I (Condiciones de optimalidad de primer orden de Karush-Kuhn-Tucker (KKT)). Sea

$f : \mathbb{R}^n \rightarrow \mathbb{R}, g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ y $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$. Considere el problema (P) dado abajo:

$$\begin{aligned} &\text{Minimizar} && f(x) \\ &\text{s.a.} && g(x) \leq 0, \\ &&& h(x) = 0. \end{aligned}$$

Sea \bar{x} una solución factible, e $I = \{i : g_i(\bar{x}) = 0\}$ el conjunto de índices de las restricciones activas en \bar{x} . Suponga que f y g_i para $i \in I$ son diferenciables en \bar{x} , que cada g_i para $i \in I$ es continua en \bar{x} , y que cada h_i para $i = 1, \dots, p$ es continuamente diferenciable en una vecindad de \bar{x} . Además, suponga que $\nabla g_i(\bar{x})$ para $i \in I$ y $\nabla h_i(\bar{x})$ para $i = 1, \dots, p$ son linealmente independientes. Si \bar{x} resuelve el problema (P) localmente, existen escalares únicos λ_i para $i \in I$ y μ tales que

$$\nabla f(\bar{x}) + \sum_{i \in I} \lambda_i \nabla g_i(\bar{x}) + \sum_{i=1}^p \mu_i \nabla h_i(\bar{x}) = 0$$

$$\lambda_i \geq 0 \quad \text{para } i \in I.$$

Además de las suposiciones de arriba, si cada g_i para $i \notin I$ es también diferenciable en \bar{x} , las condiciones KKT pueden ser escritos en la siguiente forma equivalente:

$$\nabla f(\bar{x}) + \nabla g(\bar{x}) \lambda + \nabla h(\bar{x}) \mu = 0$$

$$g(\bar{x}) \lambda = 0$$

$$g(\bar{x}) \leq 0$$

$$\lambda \geq 0.$$

De este Teorema I, se tiene la siguiente definición usada en el algoritmo FDIPA, para el siguiente Capítulo.

Definición XIII. Condiciones de optimalidad de primer orden de Karush Kuhn Tucker (KKT) del

problema (I) son definidas por el sistema:

$$\nabla f(x) + \nabla g(x)\lambda = 0, \quad (5)$$

$$G(x)\lambda = 0, \quad (6)$$

$$g(x) \leq 0, \quad (7)$$

$$\lambda \geq 0. \quad (8)$$

Como se puede notar solo han sido consideradas las restricciones de desigualdad, debido que el estudio realizado del algoritmo de FDIPA es aplicado solo a este caso. El caso donde el Problema I tienen las restricciones $h(x) = 0$, se cambia la función objetivo $f(x)$ por la función auxiliar $\phi_c(x) = f(x) + \sum_{i=1}^p c_i |h(x)|$ y se toma las restricciones de ≤ 0 , para aplicar el algoritmo FDIPA.

Definición XIV. Un punto (x^*, λ^*) que satisface las ecuaciones (5) y (6) es llamado par estacionario del problema (I) y es llamado par KKT al punto que satisface todas las ecuaciones del sistema (5) a (8).

El algoritmo FDIPA-LP es aplicado en un problema de programación lineal en la forma del Problema (II). Para un mejor entendimiento de este tipo de problema, la siguiente sección hace una revisión sobre algunos conceptos. Además, se estudia el problema primal y dual, así como de sus principales resultados.

C. Dualidad de problemas de programación lineal

El concepto de dualidad en problemas de Programación lineal consiste en asociar al problema inicial (llamado primal) a otro problema, que es llamado dual, que bajo ciertas condiciones es equivalente al valor optimo del problema inicial y que en muchas veces es más fácil de resolverlo.

Para el caso de programación lineal los resultados que se obtienen respecto al primal-dual son muy interesantes ya que en ella la brecha de dualidad se hace cero.

El problema de programación lineal tiene dos formas usuales: La forma canónica y estándar, ambas formas son equivalentes y las dualidades de ellas se explican a continuación.

Dual de la forma canónica

Sea el problema de programación lineal en su forma canónica:

$$\begin{aligned} \text{mín} \quad & c^T x \\ \text{s.a.} \quad & Ax \geq b, \\ & x \geq 0 \end{aligned}$$

donde $x, c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{n \times m}$. Entonces el Dual del problema lineal es dado por:

$$\begin{aligned} \text{máx} \quad & b^T \lambda. \\ \text{s.a.} \quad & A^T \lambda \leq c, \\ & \lambda \geq 0, \end{aligned}$$

donde $\lambda \in \mathbb{R}^m$.

Dual de la forma estándar

Ahora veamos la dualidad para el problema de programación lineal cuando el primal tiene su forma estándar:

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

Entonces el Dual del problema lineal es dado por:

$$\begin{aligned} \text{máx} \quad & b^T \lambda \\ \text{s.a.} \quad & A^T \lambda \leq c \\ & \lambda \text{ es irrestricto.} \end{aligned}$$

Ambos resultados son equivalentes, la demostración es practica, por ejemplo supongamos que aceptamos la forma estándar como definición del problema lineal, y luego se obtiene la misma forma dual de la forma canónica. Agregamos variables de holgura x_s a la forma canónica para que tenga la forma estándar y obtener su forma dual, se demuestra que el dual tiene la forma canónica. Veamos,

$$\begin{array}{ll} \text{mín} & c^T x \\ \text{s.a.} & Ax - Ix_s = b \\ & x, x_s \geq 0 \end{array} \quad \Rightarrow \quad \begin{array}{ll} \text{máx} & b^T \lambda \\ \text{s.a.} & A^T \lambda \leq c \\ & -I\lambda \leq 0 \end{array}$$

Como $-I\lambda \leq 0$ implica $\lambda \geq 0$, se obtiene que el dual es el mismo que el dual de la forma canónica.

Dual de la forma mixta

Los problemas primales pueden tener formas muy complejas, combinando diferentes tipos de restricciones, como restricciones de desigualdad \geq , \leq e igualdad $=$ a la vez; y además las variables pueden ser ≥ 0 , ≤ 0 o irrestrictas. Pero, aun así pueden ser convertidas a la forma canónica. El siguiente ejemplo muestra el proceso que se debe seguir para convertirla.

Sea el problema.

$$\begin{aligned}
& \text{mín} && c_1^T x_1 + c_2^T x_2 + c_3^T x_3. \\
& \text{s.a.} && A_{11}^T x_1 + A_{12}^T x_2 + A_{13}^T x_3 \geq b_1, \\
& && A_{21}^T x_1 + A_{22}^T x_2 + A_{23}^T x_3 \leq b_2, \\
& && A_{31}^T x_1 + A_{32}^T x_2 + A_{33}^T x_3 = b_3, \\
& && x_1 \geq 0, \quad x_2 \leq 0, \quad x_3 \text{ irrestricto}.
\end{aligned}$$

Donde: $c_i, x_i \in \mathbb{R}^{n_i}$, $b_j \in \mathbb{R}^{m_j}$ y $A_{ij} \in \mathbb{R}^{n_i \times m_j}$, tal que $n_i, m_j \in \mathbb{N}$ e $i, j \in \{1, 2, 3\}$.

Transformando a su forma canónica, multiplicando por (-1) el segundo sistema, escribiendo la restricción de igualdad por dos desigualdades y sustituyendo $x_2 = -x'_2$, $x_3 = x'_3 - x''_3$.

$$\begin{aligned}
& \text{mín} && c_1^T x_1 + c_2^T x'_2 + c_3^T x'_3 - c_3^T x''_3 \\
& \text{s.a.} && A_{11}^T x_1 - A_{12}^T x'_2 + A_{13}^T x'_3 - A_{13}^T x''_3 \geq b_1 \\
& && -A_{21}^T x_1 + A_{22}^T x'_2 - A_{23}^T x'_3 + A_{23}^T x''_3 \geq -b_2 \\
& && A_{31}^T x_1 - A_{32}^T x'_2 + A_{33}^T x'_3 - A_{33}^T x''_3 \geq b_3 \\
& && -A_{31}^T x_1 + A_{32}^T x'_2 - A_{33}^T x'_3 + A_{33}^T x''_3 \geq -b_3 \\
& && x_1 \geq 0, \quad x'_2 \geq 0, \quad x'_3 \geq 0, \quad x''_3 \geq 0
\end{aligned}$$

Denotando $\lambda_1 \geq 0$, $\lambda'_2 \geq 0$, λ'_3 y λ''_3 , las variables duales respectivas de las cuatro restricciones del problema primal, se obtiene el problema dual,

$$\begin{aligned}
& \text{máx} && b_1^T \lambda_1 + b_2^T \lambda'_2 + b_3^T \lambda'_3 - b_3^T \lambda''_3 \\
& \text{s.a.} && A_{11}^T \lambda_1 - A_{21}^T \lambda'_2 + A_{31}^T \lambda'_3 - A_{31}^T \lambda''_3 \leq c_1 \\
& && -A_{12}^T \lambda_1 + A_{22}^T \lambda'_2 - A_{32}^T \lambda'_3 + A_{32}^T \lambda''_3 \leq -c_2 \\
& && A_{13}^T \lambda_1 - A_{23}^T \lambda'_2 + A_{33}^T \lambda'_3 - A_{33}^T \lambda''_3 \leq c_3 \\
& && -A_{13}^T \lambda_1 + A_{23}^T \lambda'_2 - A_{33}^T \lambda'_3 + A_{33}^T \lambda''_3 \leq -c_3 \\
& && \lambda_1 \geq 0, \quad \lambda'_2 \geq 0, \quad \lambda'_3 \geq 0, \quad \lambda''_3 \leq 0
\end{aligned}$$

Usando $\lambda_2 = -\lambda'_2$ y $\lambda_3 = \lambda'_3 - \lambda''_3$, el problema dual queda de la siguiente forma:

$$\begin{aligned}
 &\text{máx} && b_1^T \lambda_1 + b_2^T \lambda_2 + b_3^T \lambda_3 \\
 &s.a. && A_{11}^T \lambda_1 - A_{21}^T \lambda_2 + A_{31}^T \lambda_3 \leq c_1 \\
 &&& A_{12}^T \lambda_1 + A_{22}^T \lambda_2 + A_{32}^T \lambda_3 \geq c_2 \\
 &&& A_{13}^T \lambda_1 + A_{23}^T \lambda_2 + A_{33}^T \lambda_3 = c_3 \\
 &&& \lambda_1 \geq 0, \quad \lambda_2 \leq 0, \quad \lambda_3 \text{ irrestricto}
 \end{aligned}$$

Este último resultado es muy útil, porque está representando de forma directa el dual de un problema primal sin necesidad de hacer cambios o agregar variables auxiliares. La siguiente Tabla, resume todas las relaciones entre el problema primal y dual.

TABLA II

Relación entre el problema primal y dual.

	Minimización		Maximización	
	Problema		Problema	
Variables	≥ 0	\longleftrightarrow	\leq	Restricciones
	≤ 0	\longleftrightarrow	\geq	
	irrestricto	\longleftrightarrow	$=$	
Restricciones	\geq	\longleftrightarrow	≥ 0	Variables
	\leq	\longleftrightarrow	≤ 0	
	$=$	\longleftrightarrow	irrestricto	

Otros resultados importantes para mencionar son: el dual del dual de un problema de primal es el mismo problema primal; el valor de la función objetivo del primal es el mismo del dual o viceversa y además se puede establecer una relación de factibilidad entre las funciones objetivos y sus restricciones.

Lema I. *El dual del dual de un problema primal es el mismo problema primal.*

$$\begin{aligned} \text{mín} \quad & c^T x \\ \text{s.a.} \quad & Ax \leq b \\ & x \geq 0. \end{aligned}$$

Demostración. Considerando el dual del problema primal, en su forma canónica.

$$\begin{aligned} \text{máx} \quad & b^T \lambda \\ \text{s.a.} \quad & A^T \lambda \leq c \\ & \lambda \leq 0. \end{aligned}$$

Transformando el dual como un problema primal,

$$\begin{aligned} \text{mín} \quad & (-b)^T \lambda \\ \text{s.a.} \quad & (-A^T) \lambda \geq -c \\ & \lambda \leq 0. \end{aligned}$$

Determinando el dual del problema

$$\begin{aligned} \text{máx} \quad & (-c^T) y \\ \text{s.a.} \quad & (-A^T)^T y \geq (-b) \\ & y \geq 0. \end{aligned}$$

Así el problema queda en la forma del problema primal inicial.

$$\begin{aligned} \text{mín} \quad & c^T y \\ \text{s.a.} \quad & Ay \leq b \\ & y \geq 0 \end{aligned}$$

□

Teorema II. *Un punto x^* es solución del problema primal si y solo si existe un punto λ^* que es solución factible para el problema dual respectivo. Además, se cumple: $c^T x^* = b^T \lambda^*$, es decir, los valores de las funciones objetivos de los dos problemas son iguales.*

La demostración del Teorema se puede revisar en [19]. El siguiente teorema es un resultado mas completo acerca de la relación entre el problema primal y dual.

Teorema III (Dualidad en programación lineal). *El problema primal y dual, tienen las siguientes tres posibilidades:*

- a) Ambos problemas tienen el conjunto factible vacío.*
- b) La función objetivo de uno de los problemas es ilimitada (inferiormente para el primal), superiormente para el dual) en sus respectivos conjunto factibles no vacíos.*
- c) La función objetivo de uno de los problemas es limitada (inferiormente para el primal y superiormente para el dual) en su respectivo conjunto factible no vacío.*

La demostración del Teorema se puede revisar en [19].

Los algoritmos de FDIPA y FDIPA-LP utilizan el método de Newton, porque resuelve dos sistemas de ecuaciones para encontrar direcciones de descenso factibles. La siguiente sección estudia como es la aplicación de este método.

D. Estrategias de solución

Los criterios de optimalidad de primer orden involucran resolver sobre un sistema de ecuaciones no lineales en las variables del problema y multiplicadores de Lagrange, de esta forma resolver problemas de optimización involucran conocer métodos para determinar las soluciones de sistemas no lineales. En la literatura encontraremos muchos métodos, pero fijaremos nuestro estudio en el método de clásico de Newton o el método Quasi-Newton. Recordemos que

estos métodos en caso de convergencia tienen una solución local y dependen de la posición del punto inicial.

1). Método de Newton

El método de Newton genera una sucesión de puntos $\{x^k\}_{k \in \mathbb{N}}$ que aproxima la solución del sistema no lineal $\bar{x} \in \mathbb{R}^n$ del problema,

$$\Phi(\bar{x}) = 0 \quad (9)$$

donde $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es una función diferenciable. El método de Newton plantea la aproximación de la solución del sistema (9) por un proceso iterativo de linealización del sistema, obteniéndose lo siguiente,

$$\Phi(x^k) + \Phi'(x^k)(x^{k+1} - x^k) = 0, \quad (10)$$

que se denomina ecuación de iteración del método de Newton. Suponiendo que $\Phi'(x^k)$ sea no singular, la ecuación (10) queda escrita de forma iterativa como,

$$x^{k+1} = x^k - (\Phi'(x^k))^{-1} \Phi(x^k), \quad (11)$$

también se puede plantear de la siguiente forma:

$$\begin{aligned} \Phi'(x^k)s^k &= -\Phi(x^k) \\ x^{k+1} &= x^k + s^k, \end{aligned} \quad (12)$$

donde es claro que: $s^k = x^{k+1} - x^k$.

Algoritmo del Método de Newton

Escoger $x^0 \in \mathbb{R}^n$, $\epsilon > 0$ y tomar $k = 0$.

1. Calcular x^{k+1} , como en (11) si es conocido la inversa de $\Phi'(x^k)$ o como en (12) resolviendo

un sistema de ecuaciones.

2. Tomar $k = k + 1$ y retornar a 1, hasta que $\|x^{k+1} - x^k\| < \epsilon$.

Considerando que $\{x^k\}_{k \in \mathbb{N}}$ sea convergente, se puede esperar que la convergencia sea rápida. En efecto, si $x^k \rightarrow \bar{x}$, con $\Phi'(x^k)$ no singular. Por la fórmula de Taylor de primer orden, tenemos:

$$\begin{aligned} 0 &= \Phi(\bar{x}) = \Phi(x^k) + \Phi'(x^k)(\bar{x} - x^k) + o(\|x^k - \bar{x}\|) \\ 0 &= (\Phi'(x^k))^{-1} \left(\Phi(x^k) + \Phi'(x^k)(\bar{x} - x^k) + o(\|x^k - \bar{x}\|) \right) \\ 0 &= \bar{x} - x^k + (\Phi'(x^k))^{-1} \Phi(x^k) + (\Phi'(x^k))^{-1} (o(\|x^k - \bar{x}\|)) \\ 0 &= \bar{x} - (x^k - (\Phi'(x^k))^{-1} \Phi(x^k)) + o(\|x^k - \bar{x}\|) \\ 0 &= \bar{x} - x^{k+1} + o(\|x^k - \bar{x}\|), \end{aligned}$$

lo que demuestra que la convergencia es de orden superlineal.

$$x^{k+1} - \bar{x} = o(\|x^k - \bar{x}\|).$$

El siguiente teorema analiza las condiciones para la convergencia local del método.

Teorema IV (Convergencia local del Método de Newton). *Sea $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ diferenciable en un vecindad del punto $\bar{x} \in \mathbb{R}^n$, con derivada continua en este punto. Sea \bar{x} una solución de la ecuación (9), tal que $\det \Phi'(x^k) \neq 0$. Entonces para cualquier punto inicial $x^0 \in \mathbb{R}^n$ suficientemente próximo a \bar{x} , el algoritmo de Newton genera una sucesión $\{x^k\}_{k \in \mathbb{N}}$ convergente a \bar{x} . La tasa de convergencia es superlineal y si la derivada de Φ es lipschitz continua en una vecindad de \bar{x} , entonces la tasa de convergencia es cuadrática.*

Demostración: La demostración ver [20].

□

2). Método de Quase-Newton

Esta clase de método es una performance del método de Newton que busca ser computacionalmente más económico. Los métodos Quase-Newton son aquellos algoritmos que funcionan para sistemas no lineales, que tienen el siguiente formato:

$$\begin{aligned} B_k s^k &= -\Phi(x^k) \\ x^{k+1} &= x^k + s^k \end{aligned} \tag{13}$$

donde: s^k es una expresión que depende de la iteración de orden k . Así el método de Newton también se puede considerar un método Quase-Newton, aquí $s^k = -\Phi'^{-1}(x^k)\Phi(x^k)$. Estos métodos son prácticos, la inversa B_k^{-1} pueda ser mas fácilmente obtenida, al generar un proceso de factorizaciones que evite el cálculo de la inversa.

En caso de un problema de optimización (I) pero sin restricciones $\Omega = \mathbb{R}^n$, el metodo Quase-Newton construye aproximaciones de la Hessiana de la función objetivo a medida que avancen las iteraciones. A continuación, se muestra como se construye las aproximaciones.

El proceso iterativo del algoritmo para el sistema $f(x) = 0$ toma como direcciones de descenso, el vector,

$$d^k = -H_k \nabla f(x_k), \tag{14}$$

donde $H_k \in \mathbb{R}^{n \times n}$ es simétrica definida positiva. Esta expresión tiene sentido, porque se puede observar que al realizar una aproximación cuadrática de f por el método de Newton entorno de una vecindad de x^k , se obtiene

$$m_k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T B_k d,$$

donde $B_k \in \mathbb{R}^{n \times n}$ es una matriz simétrica cualquiera en vez de $\nabla^2 f(x^k)$. Si B_k fuera definida

positiva, el minimizador del modelo cuadrático es dado por

$$-B_k^{-1}\nabla f(x^k).$$

Escogiendo $B_k = H_k^{-1}$ se obtiene la ecuación (14). De esta forma se tiene el algoritmo para este caso,

Algoritmo del Método de Quase-Newton

Escoger $x^0 \in \mathbb{R}^n$, H_0 simétrica definida positiva

$k = 0$.

Repetir mientras $\nabla f(x^k) \neq 0$

1. Defina $d^k = -H_k \nabla f(x^k)$
2. Obtener $t_k > 0$ que minimiza $f(x^k + td^k)$ en $[0, \infty[$
3. Hacer $x^{k+1} = x^k + t_k d^k$
4. Determinar H_{k+1} simétrica definida positiva

E. Métodos de búsqueda inexacta

Los métodos de búsqueda inexacta no buscan minimizar la función en cada iteración, es decir no buscan un α tal que:

$$\min f(\bar{x} + \alpha \bar{d}), \text{ sujeta a } \alpha \geq 0$$

para una dirección de descenso dada, sino que tratan de llegar en un primer instante donde la función decrece en esa dirección, evitando un costo computacional y solo toman en cuenta el tamaño de paso α . A continuación se muestra uno de los métodos de búsqueda inexacta.

1). Condición de Armijo

La búsqueda por la condición de Armijo, consiste que para un punto $\bar{x} \in \mathbb{R}^n$ y una dirección de descenso $d \in \mathbb{R}^n$ y un parámetro $\sigma \in]0, 1[$, encontrar un $\bar{\alpha}$ tal que

$$f(\bar{x} + \bar{\alpha}d) \leq f(\bar{x}) + \sigma \bar{\alpha} \nabla f^T(\bar{x})d \quad (15)$$

Esta reducción lo realiza de forma proporcional al tamaño de paso, el siguiente teorema justifica la afirmación.

Teorema V. *Considere una función diferenciable $f : \mathbb{R}^n \rightarrow \mathbb{R}$, un punto $\bar{x} \in \mathbb{R}^n$, una dirección de descenso $d \in \mathbb{R}^n$ y $\sigma \in]0, 1[$. Entonces existe un θ tal que*

$$f(\bar{x} + \alpha d) \leq f(\bar{x}) + \sigma \alpha \nabla f^T(\bar{x})d, \text{ para todo } \alpha \in [0, \theta].$$

Demostración: La prueba se puede ver en [21]

□

Algoritmo de la Búsqueda de Armijo

Dado $x^0 \in \mathbb{R}^n$, $d^0 \in \mathbb{R}^n$ una dirección de descenso, $\sigma, \theta \in]0, 1[$

Haciendo $\alpha^0 = 1$

Repetir mientras $f(x^k + \alpha^k d^k) > f(x^k) + \sigma \alpha^k \nabla f^T(x^k) d^k$

- Hacer $\alpha^{k+1} = \sigma \alpha^k$ (reducir el tamaño de paso)

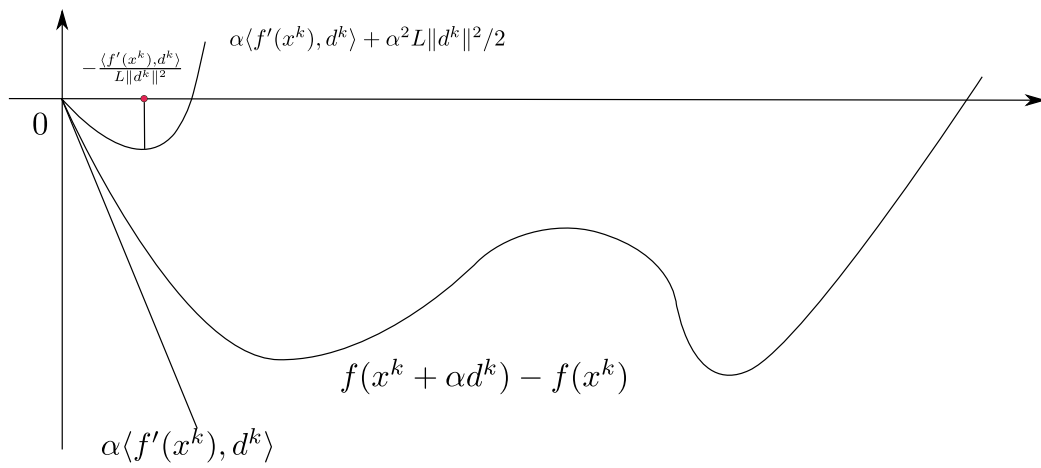


Figura I: Busqueda de Armijo.

III Algoritmo FDIPA

El capítulo trata sobre el estudio del algoritmo FDIPA (Feasible Direction Interior Point Algorithms), un algoritmo de punto interior direcciones factibles para problemas de programación no lineal desarrollado por Herskowitz [9], que consiste en determinar direcciones de descenso factibles que disminuya la función objetivo alcanzando el valor óptimo. Además, se explica las hipótesis que debe satisfacer la función objetivo y las restricciones para obtener resultados óptimos en la resolución de los problemas.

A. Estudio del algoritmo FDIPA

La aplicación del algoritmo FDIPA en problemas de programación no lineal de la forma (I) esta sujeto a un conjunto de restricciones $\Omega = \{x \in \mathbb{R}^n : g(x) \leq 0\}$. El algoritmo FDIPA es aplicado bajo hipótesis, sobre las funciones objetivo y restricciones f y g respectivamente.

Hipótesis I. Existe $a \in \mathbb{R}$ tal que el conjunto $\Omega_a = \{x \in \Omega : f(x) \leq a\}$ es compacto.

Hipótesis II. Para cada $x \in \overset{\circ}{\Omega}_a$ se satisface que $f(x) < a$.

Hipótesis III. Las funciones f y g son funciones continuamente diferenciable ($f, g \in C^1$) en Ω_a y sus derivadas parciales satisfacen la condición de Lipschitz.

Hipótesis IV. La función g satisface condiciones de regularidad. Para todo $x \in \Omega_a$ y cada $i \in \{1, \dots, m\}$ tal que $g_i(x) = 0$, los vectores $\nabla g_i(x)$ son linealmente independiente.

La Hipotesis IV son las condiciones necesarias para garantizar que se puede determinar una dirección de descenso factible por el algoritmo FDIPA, al resolver un sistema de ecuaciones que depende del $\nabla g_i(x)$.

El algoritmo se caracteriza por generar una sucesión de puntos interiores $\{x^k\}_{k \in \mathbb{N}}$ a partir de un punto inicial (punto interior) x^0 que va descendiendo el valor de la función objetivo por una dirección de descenso d y factible en cada punto $x^k \in \Omega$. Luego, se aplica una búsqueda lineal inexacta para garantizar que el nuevo punto interior factible y la función objetivo disminuya, en un tiempo óptimo. La sucesión generada converge a un punto x^* que satisface las condiciones de optimalidad de KKT, esta convergencia está garantizada por el requerimiento de factibilidad de la dirección d , que debe definir un campo uniforme de direcciones factibles. De otra manera, la longitud de paso tiende a cero, y convergiendo a puntos que no son KKT. A continuación se explica el proceso que sigue el Algoritmo.

De las condiciones de optimalidad KKT, se resuelve el sistema de ecuaciones (5) y (6) en (x, λ) por algún método iterativo de punto fijo (Newton, Quase Newton, u otro), con el objetivo de encontrar mas adelante una dirección de descenso.

Sea el sistema definido por la función $F(x, \lambda) = 0$, donde $F : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{n+m}$ es diferenciable y considere que $0 \in \mathbb{R}^{n+m}$.

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) + \nabla g(x) \cdot \lambda \\ G(x) \cdot \lambda \end{bmatrix}.$$

Aplicando el método iterativo de Newton desde el punto inicial (x, λ) y su nueva estimación (x^k, λ^k) , el sistema queda determinado como:

$$F(x, \lambda) + JF(x, \lambda) \begin{pmatrix} x^k - x \\ \lambda^k - \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow JF(x, \lambda) \begin{pmatrix} x^k - x \\ \lambda^k - \lambda \end{pmatrix} = -F(x, \lambda),$$

donde $JF(x, \lambda)$ es la matriz jacobiana de $F(x, \lambda)$. Reemplazando la forma de la función $F(x, \lambda)$ y la matriz jacobiana $JF(x, \lambda)$, queda el siguiente sistema de forma explícita:

$$\begin{bmatrix}
\frac{\partial^2 f(x)}{\partial^2 x_1} + \sum_{i=1}^m \lambda_i \frac{\partial^2 g_i(x)}{\partial^2 x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_1} + \sum_{i=1}^m \lambda_i \frac{\partial^2 g_i(x)}{\partial x_n \partial x_1} & \frac{\partial g_1(x)}{\partial x_1} & \cdots & \frac{\partial g_m(x)}{\partial x_1} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f(x)}{\partial x_1 \partial x_n} + \sum_{i=1}^m \lambda_i \frac{\partial^2 g_i(x)}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f(x)}{\partial^2 x_n} + \sum_{i=1}^m \lambda_i \frac{\partial^2 g_i(x)}{\partial^2 x_n} & \frac{\partial g_1(x)}{\partial x_n} & \cdots & \frac{\partial g_m(x)}{\partial x_n} \\
\lambda_1 \frac{\partial g_1(x)}{\partial x_1} & \cdots & \lambda_1 \frac{\partial g_1(x)}{\partial x_n} & g_1(x) & 0 & 0 \\
\vdots & \ddots & \vdots & 0 & \ddots & 0 \\
\lambda_m \frac{\partial g_m(x)}{\partial x_1} & \cdots & \lambda_m \frac{\partial g_m(x)}{\partial x_n} & 0 & 0 & g_m(x)
\end{bmatrix}
\begin{bmatrix}
x_1^k - x_1 \\
\vdots \\
x_n^k - x_n \\
\lambda_1^k - \lambda_1 \\
\vdots \\
\lambda_m^k - \lambda_m
\end{bmatrix}
= -
\begin{bmatrix}
\frac{\partial f}{\partial x_1}(x) + \sum_{i=1}^m \lambda_i \frac{\partial g_i(x)}{\partial x_1} \\
\vdots \\
\frac{\partial f}{\partial x_n}(x) + \sum_{i=1}^m \lambda_i \frac{\partial g_i(x)}{\partial x_n} \\
\lambda_1 g_1(x) \\
\vdots \\
\lambda_m g_m(x)
\end{bmatrix}.$$

De esta manera se puede reescribir la ecuación de iteración como:

$$\begin{bmatrix}
B & \nabla g(x) \\
\Lambda \nabla g(x)^T & G(x)
\end{bmatrix}
\begin{bmatrix}
x^k - x \\
\lambda^k - \lambda
\end{bmatrix}
= -
\begin{bmatrix}
\nabla f(x) + \nabla g(x)\lambda \\
G(x)\lambda
\end{bmatrix}, \quad (16)$$

donde:

$$B = \begin{bmatrix}
\frac{\partial^2 f(x)}{\partial^2 x_1} + \sum_{i=1}^m \lambda_i \frac{\partial^2 g_i(x)}{\partial^2 x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_1} + \sum_{i=1}^m \lambda_i \frac{\partial^2 g_i(x)}{\partial x_n \partial x_1} \\
\vdots & \ddots & \vdots \\
\frac{\partial^2 f(x)}{\partial x_1 \partial x_n} + \sum_{i=1}^m \lambda_i \frac{\partial^2 g_i(x)}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f(x)}{\partial^2 x_n} + \sum_{i=1}^m \lambda_i \frac{\partial^2 g_i(x)}{\partial^2 x_n}
\end{bmatrix} \quad \text{y} \quad \Lambda = \begin{bmatrix}
\lambda_1 & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & \lambda_m
\end{bmatrix}.$$

La matriz B es definida positiva y permite generar otra estimación de la solución del sistema. Si se aplica el método Quase-Newton la matriz puede ser reemplazada por $B = I$, una

matriz identidad desde el inicio del algoritmo, u otra matriz definida positiva.

Lo que sigue en el algoritmo es generar un nuevo punto interior que disminuya la función objetivo en cada nueva iteración. Con la finalidad de cumplir lo dicho, se define la dirección d_α^k en el espacio primal, como:

$$d_\alpha^k = x_\alpha^k - x^k.$$

El sistema (16) queda como:

$$\begin{bmatrix} B^k & \nabla g(x^k) \\ \Lambda \nabla g(x^k)^T & G(x^k) \end{bmatrix} \begin{bmatrix} d_\alpha^k \\ \lambda_\alpha^k - \lambda^k \end{bmatrix} = - \begin{bmatrix} \nabla f(x^k) + \nabla g(x^k) \lambda^k \\ G(x^k) \lambda^k \end{bmatrix}$$

desarrollado en forma separada, el sistema tiene la forma de dos sistemas de ecuaciones,

$$B^k d_\alpha^k + \nabla g(x^k) \lambda_\alpha^k = -\nabla f(x^k) \quad (17)$$

$$\Lambda^k \nabla g(x^k)^T d_\alpha^k + G(x^k) \lambda_\alpha^k = 0. \quad (18)$$

La solución del sistema (17)-(18), es una dirección d_α^k en el espacio primal y una nueva estimación de λ^k (λ_α^k). Si (x^k, λ_α^k) es un punto estacionario, d_α^k es cero (B^k es definida positiva). También, se demuestra que d_α^k que es una dirección de descenso para f , esto no es muy útil como dirección de búsqueda, desde que no se garantiza que produzca un campo de direcciones uniformemente factibles. Esto se debe porque d_α^k del sistema (18) tiende a ser una dirección tangente en el conjunto factible y puede darse el caso que $g_i(x^k) = 0$ y $\lambda_i^k \neq 0$ para algún $i \in \{1, 2, \dots, m\}$. Analice el desarrollo del sistema (18)

$$\lambda_{\alpha,i}^k \nabla g_i(x^k)^T d_{\alpha,i}^k + g_i(x^k) \lambda_{\alpha,i}^k = 0, \quad i = 1, \dots, m, \quad (19)$$

cuyas componentes son:

$$\begin{bmatrix} \lambda_{\alpha,1}^k \frac{\partial g_1(x^k)}{\partial x_1} & \dots & \lambda_{\alpha,1}^k \frac{\partial g_1(x^k)}{\partial x_n} & g_1(x^k) & \dots & 0 \\ \vdots & \ddots & \vdots & & \ddots & \\ \lambda_{\alpha,i}^k \frac{\partial g_i(x^k)}{\partial x_1} & \dots & \lambda_{\alpha,i}^k \frac{\partial g_i(x^k)}{\partial x_n} & \vdots & g_i(x^k) & \vdots \\ \vdots & \ddots & \vdots & & \ddots & \\ \lambda_{\alpha,m}^k \frac{\partial g_m(x^k)}{\partial x_1} & \dots & \lambda_{\alpha,m}^k \frac{\partial g_m(x^k)}{\partial x_n} & 0 & \dots & g_m(x^k) \end{bmatrix} \begin{bmatrix} d_{\alpha,1}^k \\ \vdots \\ d_{\alpha,n}^k \\ \lambda_{\alpha,1}^k \\ \vdots \\ \lambda_{\alpha,m}^k \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (20)$$

esto implica que en la fila i

$$\left[\lambda_{\alpha,i}^k \frac{\partial g_i(x^k)}{\partial x_1} \quad \dots \quad \lambda_{\alpha,i}^k \frac{\partial g_i(x^k)}{\partial x_n} \right] \begin{bmatrix} d_{\alpha,1}^k \\ \vdots \\ d_{\alpha,n}^k \end{bmatrix} = 0 \Rightarrow \nabla g_i(x^k)^T d_{\alpha}^k = 0,$$

Es decir que d_{α}^k en la frontera del conjunto factible tiende a ser una dirección tangente, lo cual no garantiza que la nueva estimativa de (x^k, λ^k) sea un punto interior. Esto se puede observar en la Figura II

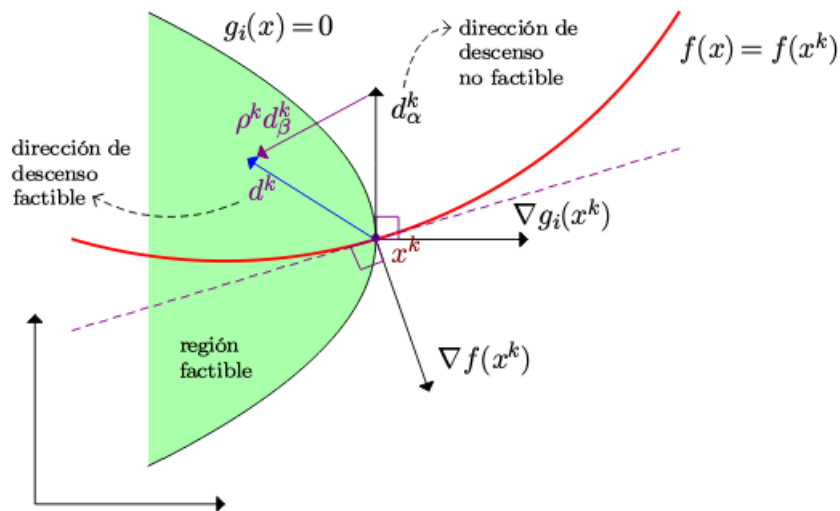


Figura II: Deflexión de d_{α} en el punto x^k del Problema (I).

Para salvar este efecto de d_α^k , agregamos un factor de desviación al sistema (18), que define un nuevo sistema en nuevas variables d^k y $\bar{\lambda}^k$:

$$B^k d^k + \nabla g(x^k) \bar{\lambda}^k = -\nabla f(x^k) \quad (21)$$

$$\Lambda \nabla g(x^k)^T d^k + G(x^k) \bar{\lambda}^k = -\rho^k \Lambda^k \omega^k \quad (22)$$

El sistema (22) se origina de agregar al sistema (18) el factor ρ y $\omega \in \mathbb{R}^m$ de signos positivos y $\bar{\lambda}^k$ la nueva estimación de λ^k . El sistema (22) queda desarrollado como,

$$\lambda_i^k \nabla g_i^T(x^k) d^k + g_i(x^k) \bar{\lambda}_i^k = -\rho^k \lambda_i^k \omega_i^k, \quad i = 1, \dots, m$$

para restricciones activas se reduce a $\nabla g_i(x^k)^T d^k = -\rho^k \omega_i^k$, por tanto d^k es una dirección factible.

El signo negativo de (22) produce el efecto de deflexión a d_α^k , en la i -ésima restricción proporcional a $\rho^k \omega_i^k$. Finalmente, como la deflexión de d_α^k es proporcional a ρ^k y como d_α^k es una dirección de descenso, se puede establecer una cota superior en ρ para garantizar que d^k sea también una dirección de descenso. Desde que $d_\alpha^k{}^T \nabla f(x^k) < 0$, obtenemos las cotas por imposición de

$$d^k{}^T \nabla f(x^k) \leq \xi d_\alpha^k{}^T \nabla f(x^k), \quad \xi \in]0, 1[\quad (23)$$

lo cual implica que $d^k{}^T \nabla f(x^k) < 0$.

La condición (23) implica que d^k es un cono circular recto con eje $\nabla f(x^k)$. En general la razón de descenso de f a lo largo de d^k será mas pequeña que a lo largo de d_α^k . Este precio se paga para obtener una buena dirección de descenso factible.

Note que por la unicidad de solución de d^k , puede ser obtenido resolviendo el sistema

auxiliar para hallar un vector d_β^k ,

$$B^k d_\beta^k + \nabla g(x^k) \lambda_\beta^k = 0, \quad (24)$$

$$\Lambda^k \nabla g^T(x^k) d_\beta^k + G(x^k) \lambda_\beta^k = -\Lambda^k \omega^k, \quad (25)$$

y luego se calcula d^k , como:

$$d^k = d_\alpha^k + \rho^k d_\beta^k$$

De esta forma, es fácil establecer las cotas de ρ^k para garantizar que (23) se satisfaga. Como un requerimiento para $d^k(x)$ sea un campo de direcciones uniformemente factible, ρ^k debe ser acotado.

Para determinar el nuevo punto primal, se aplica una búsqueda lineal inexacta, cumpliendo el requerimiento de factibilidad y decrecimiento de la función objetivo. Otras reglas de actualización pueden ser adoptadas para definir un nuevo valor positivo λ^k . Todas estas ideas forman la base del método iterativo que se estudia. Este método se resume en el algoritmo 1 que se llamará algoritmo FDIPA .

La búsqueda lineal en el paso 2 está basado en la regla de Armijo. Además (34) asegura un decrecimiento significativo de la función, las condiciones (35) y (36) fuerzan a que el nuevo punto primal esté en el interior. También (36) evita la saturación de las restricciones asociadas a las variables negativas, ya que será necesario para demostrar la convergencia a los puntos de Karush-Kuhn-Tucker.

Para la actualización, se asume algunas condiciones sobre λ^k, B^k, ω^k . Las diferentes formas de como actualizar los valores de λ^k, B^k y ω^k . dan como resultado, diferentes formas del algoritmo FDIPA. Los valores de λ^k, B^k y ω^k están sujetos a las siguientes hipótesis.

Hipótesis V. *Existen números positivos $\lambda^I, \lambda^S, \bar{g}$ tal que: $0 \leq \lambda_i^k \leq \lambda^S, i = 1, \dots, m$, y $\lambda^I \leq \lambda_i^k$, para cualquier i tal que $-\bar{g} \leq g_i(x^k)$.*

Algoritmo 1: FDIPA

Entrada: Considere los parámetros $\xi \in]0, 1[, \eta \in]0, 1[, \varphi > 0$ y $\nu \in]0, 1[, \lambda^0 \in \mathbb{R}_+^m, B^0 \in \mathbb{R}^{n \times n}$ simétrica definida positiva y un punto inicial $x^0 \in \mathring{\Omega}_\alpha \subset \mathbb{R}_+^n$.

- **Paso 1: Calculando la dirección de descenso d^k y $\bar{\lambda}^k$.**

(a) Calcular $(d_\alpha^k, \lambda_\alpha^k)$ al resolver:

$$B^k d_\alpha^k + \nabla g(x^k) \lambda_\alpha^k = -\nabla f(x^k) \quad (26)$$

$$\Lambda^k \nabla g(x^k)^T d_\alpha^k + G(x^k) \lambda_\alpha^k = 0 \quad (27)$$

Si $d_\alpha^k = 0$ entonces **FIN**.

(b) Calcular $(d_\beta^k, \lambda_\beta^k)$ al resolver:

$$B^k d_\beta^k + \nabla g(x^k) \lambda_\beta^k = 0 \quad (28)$$

$$\Lambda^k \nabla g(x^k)^T d_\beta^k + G(x^k) \lambda_\beta^k = -\Lambda^k \omega^k \quad (29)$$

(c) Si $d_\beta^k{}^T \nabla f(x^k) > 0$ definir

$$\rho^k = \min \{ \varphi \|d_\alpha^k\|_2^2, (\xi - 1) d_\alpha^k{}^T \nabla f(x^k) / d_\beta^k{}^T \nabla f(x^k) \} \quad (30)$$

Caso contrario, definir

$$\rho^k = \varphi \|d_\alpha^k\|_2^2 \quad (31)$$

(d) Definir

$$d^k = d_\alpha^k + \rho^k d_\beta^k \quad (32)$$

$$\bar{\lambda}^k = \lambda_\alpha^k + \rho^k \lambda_\beta^k. \quad (33)$$

- **Paso 2: Búsqueda lineal.** Calcule t^k siendo el primer número de la sucesión $\{1, \nu, \nu^2, \nu^3, \dots\}$ que verifique

$$f(x^k + t^k d^k) \leq f(x^k) + t^k \eta \nabla f(x^k)^T d^k \quad (34)$$

$$g_i(x^k + t^k d^k) < 0, \text{ siempre que } \bar{\lambda}_i \geq 0, \quad (35)$$

$$g_i(x^k + t^k d^k) \leq g_i(x^k), \text{ siempre que } \bar{\lambda}_i < 0. \quad (36)$$

- **Paso 3: Actualización.** Definir $x^{k+1} = x^k + t^k d^k$ y defina nuevos valores para $\omega^{k+1} > 0$, $\lambda^{k+1} \in \mathbb{R}_+^m$ y B^{k+1} simétrica definida positiva. Hacer $k = k + 1$ y retorne al **Paso 1**.
-

Hipótesis VI. *Existen números positivos σ_1 y σ_2 tal que $\sigma_1 \|d\|_2^2 \leq d^T B d \leq \sigma_2 \|d\|_2^2; \forall d \in R^n$*

Hipótesis VII. *Existen números positivos ω_1 y ω_2 tal que $\omega_1 \leq \omega \leq \omega_2$.*

B. Análisis de convergencia

Esta sección estudia las condiciones bajo las cuales la sucesión de puntos generada (x^k, λ^k) por el algoritmo FDIPA para el problema (I), converga a un punto de Karush Khun Tucker. Se presenta el análisis teórico de la demostración que cualquier sucesión generada por el Algoritmo 1, converge a puntos KKT del problema, para cualquier actualización de λ , B , ω . Primero se explica las condiciones que garantizan que nunca falle el algoritmo. Seguido se estudia las condiciones de parada del algoritmo después de un número finito de iteraciones y finalmente el estudio de la convergencia a las condiciones de KKT.

Lema II. *Dado cualquier $x \in \Omega_a$, B una matriz definida positiva, y $\lambda > 0$ tal que $\lambda_i > 0$ cuando $g_i(x) = 0$ entonces la matriz*

$$M(x, \lambda, B) = \begin{bmatrix} B & \nabla g(x) \\ \Lambda \nabla g^T(x) & G(x) \end{bmatrix}$$

es no singular.

La demostración del Lema se puede revisar en [22].

La consecuencia directa del Lema II, es que los sistemas en d_α , λ_α , d_β y λ_β para el algoritmo FDIPA tienen solución única. Además, desde que x pertenecen a un conjunto compacto, λ y B están acotados, entonces se tiene que M esta acotado desde cero y d_α , λ_α , d_β , λ_β están acotados superiormente.

Si se considera el caso especial $d_\alpha = 0$ en el Paso 1, esto indica que las iteraciones se

detienen. Además, el sistema se reduce a:

$$\nabla g(x)\lambda_\alpha = -\nabla f(x)$$

$$G(x)\lambda_\alpha = 0$$

Por lo tanto se tiene que x es un punto KKT. De esta manera de aquí en adelante, el caso donde el algoritmo FDIPA nunca se detiene es estudiado suponiendo ($d_\alpha \neq 0$). El siguiente ejemplo muestra como la condición de KKT es satisfecha en la solución óptima. Sea el Ejemplo:

$$\text{mín } f(x) = (x_1 - 3)^2 + (x_2 - 2)^2,$$

sujeto a:

$$x_1^2 + x_2^2 - 5 \leq 0,$$

$$x_1 + 2x_2 - 4 \leq 0,$$

$$-x_1 \leq 0,$$

$$-x_2 \leq 0.$$

La región factible del problema es mostrada en la Figura III, la solución óptima se da en el punto $\bar{x} = (2, 1)^T$ con las restricciones activas $I = \{1, 2\}$. Los gradientes tienen los valores:

$$\nabla f(\bar{x}) = (-2, -2)^T; \quad \nabla g_1(\bar{x}) = (4, 2)^T; \quad \nabla g_2(\bar{x}) = (1, 2)^T.$$

Entonces la solución óptima satisface que: $-\nabla f(\bar{x}) = \frac{1}{3}\nabla g_1(\bar{x}) + \frac{2}{3}\nabla g_2(\bar{x})$.

Lema III. El vector d_α , definido en el Paso 1 del algoritmo FDIPA satisface la desigualdad

$$d_\alpha^T \nabla f(x) \leq -d_\alpha^T B d_\alpha$$

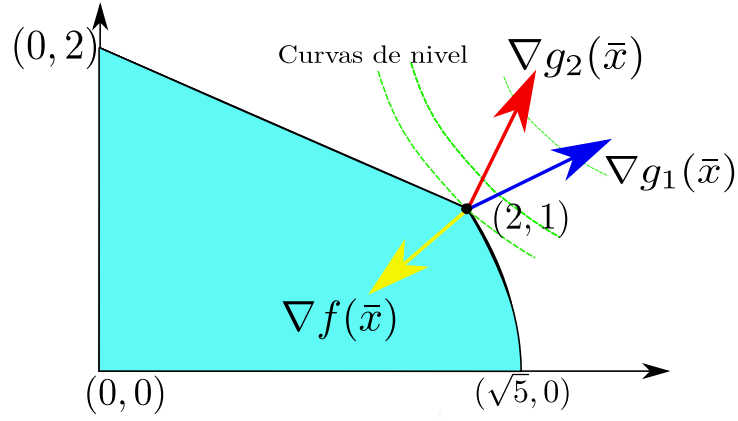


Figura III: Región factible y solución óptima $\bar{x} = (2, 1)^T$.

Demostración: Realizando el producto d_α al primer sistema del Paso 1 en ambos lados,

$$d_\alpha^T \cdot (Bd_\alpha + \nabla g(x)\lambda_\alpha = -\nabla f(x))$$

$$d_\alpha^T \nabla f(x) = -d_\alpha^T B d_\alpha - d_\alpha^T \nabla g(x) \lambda_\alpha \quad (37)$$

Del segundo sistema en el Paso 1, podemos obtener una equivalencia para $\nabla g(x)$, veamos:

$$(\Lambda \nabla g^T(x) d_\alpha) + G(x) \lambda_\alpha = 0)^T,$$

$$d_\alpha^T \nabla g(x) \Lambda^T + \lambda_\alpha^T G^T(x) = 0,$$

como Λ y $G(x)$ son simétricas, entonces el último sistema queda,

$$d_\alpha^T \nabla g(x) = -\lambda_\alpha^T G(x) \Lambda^{-1} \quad (38)$$

Reemplazando la Ecuación (38) en la Ecuación (37)

$$d_\alpha^T \nabla f(x) = -d_\alpha^T B d_\alpha + \lambda_\alpha^T G(x) \Lambda^{-1} \lambda_\alpha$$

Por las restricciones de optimalidad la matriz $G(x) \Lambda^{-1}$ es definida negativa. Así, podemos con-

cluir que,

$$d_\alpha^T \nabla f(x) \leq -d_\alpha^T B d_\alpha$$

□

Una consecuencia del Lema III, es que d_α es una dirección de descenso para f . La regla de actualización de ρ , planteado en el Paso 1 del algoritmo FDIPA, garantiza que,

$$\rho \leq \varphi \|d_\alpha\|_2^2$$

De otro lado por la hipótesis VI se tiene que:

$$-d_\alpha^T \nabla f(x) \geq \sigma_1 \|d_\alpha\|_2^2.$$

Luego de la definición de ρ en la igualdad (31),

$$\rho = \min \left\{ \varphi \|d_\alpha\|_2^2, \frac{(\xi - 1) d_\alpha^T \nabla f(x)}{d_\beta^T \nabla f(x)} \right\}$$

entonces de la desigualdad $-d_\alpha^T \nabla f(x) \geq \sigma_1 \|d_\alpha\|_2^2$, se tiene que,

$$\frac{-d_\alpha^T \nabla f(x)}{d_\beta^T \nabla f(x)} \geq \frac{\sigma_1 \|d_\alpha\|_2^2}{d_\beta^T \nabla f(x)}, \quad \text{si } d_\beta^T \nabla f(x) > 0$$

El parámetro ξ , tiene una variación $0 < \xi < 1$, entonces,

$$(1 - \xi) \frac{(-d_\alpha^T \nabla f(x))}{d_\beta^T \nabla f(x)} \geq \frac{(1 - \xi) \sigma_1 \|d_\alpha\|_2^2}{d_\beta^T \nabla f(x)}$$

$$\rho > \min \left\{ \varphi \|d_\alpha\|_2^2; \frac{(1 - \xi) \sigma_1 \|d_\alpha\|_2^2}{d_\beta^T \nabla f(x)} \right\}$$

$$\rho > \min \left\{ \varphi; \frac{(1-\xi)\sigma_1}{d_\beta^T \nabla f(x)} \right\} \|d_\alpha\|_2^2$$

Desde d_β esta acotado y $\rho = \varphi \|d_\alpha\|_2^2$ cuando $d_\beta^T \nabla f(x) < 0$, deducimos que existe $\varphi_0 > 0$ tal que

$$\rho \geq \varphi_0 \|d_\alpha\|_2^2 \quad (39)$$

Debido que d_α esta acotada, se tiene de las ecuaciones (38) y (39) que ρ es positivo y acotada por arriba.

Consideremos (38) y $d = d_\alpha + \rho d_\beta$ juntos, estos demuestran que existe $\delta > 1$ tal que la desigualdad

$$\begin{aligned} \|d\|_2 &= \|d_\alpha + \rho d_\beta\|_2 \\ &\leq \|d_\alpha\|_2 + \rho \|d_\beta\|_2 \leq \|d_\alpha\|_2 + \varphi \|d_\alpha\|_2^2 \|d_\beta\|_2 \\ &\leq \|d_\alpha\|_2 + \varphi \|d_\alpha\|_2^2 C \\ &\leq (1 + \varphi \|d_\alpha\|_2 C) \|d_\alpha\|_2 \end{aligned}$$

de esto último, denotamos a $\delta = (1 + \varphi \|d_\alpha\|_2 C) > 1$ quedando verificado que,

$$\|d\|_2 \leq \delta \|d_\alpha\|_2 \quad (40)$$

para d_α y d .

Lema IV. *La dirección de búsqueda d satisface*

$$d^T \nabla f(x) \leq \xi d_\alpha^T \nabla f(x), \text{ para algún } \xi > 1.$$

Demostración:

$$d = d_\alpha + \rho d_\beta$$

$$d^T \nabla f(x) = d_\alpha^T \nabla f(x) + \rho d_\beta^T \nabla f(x)$$

$$\text{Como: } \rho = \min \{ \varphi \|d_\alpha\|_2^2; (\xi - 1) d_\alpha^T \nabla f(x) / d_\beta^T \nabla f(x) \}, \text{ si } d_\beta^T \nabla f(x) > 0$$

de lo contrario ($d_\beta^T \nabla f(x) \leq 0$) y $\rho = \varphi \|d_\alpha\|_2^2$

Luego, se tiene dos casos:

Si $d_\beta^T \nabla f(x) > 0$ entonces,

$$\begin{aligned} d^T \nabla f(x) &\leq d_\alpha^T \nabla f(x) + \frac{(\xi - 1) d_\alpha^T \nabla f(x)}{d_\beta^T \nabla f(x)} \cdot d_\beta^T \nabla f(x) \\ d^T \nabla f(x) &\leq \xi d_\alpha^T \nabla f(x) \quad (\text{si } d_\beta^T \nabla f(x) > 0) \end{aligned}$$

Si $d_\beta^T \nabla f(x) \leq 0$ entonces,

$$\begin{aligned} d^T \nabla f(x) &\leq d_\alpha^T \nabla f(x) + \varphi \|d_\alpha\|_2^2 d_\beta^T \nabla f(x) \\ d^T \nabla f(x) &\leq d_\alpha^T \nabla f(x) + \underbrace{\varphi \|d_\alpha\|_2^2 d_\beta^T \nabla f(x)}_{\leq 0} \\ d^T \nabla f(x) &\leq d_\alpha^T \nabla f(x) \\ d^T \nabla f(x) &\leq \xi d_\alpha^T \nabla f(x) \end{aligned}$$

Por tanto: $d^T \nabla f(x) \leq \xi d_\alpha^T \nabla f(x)$

□

De este resultado, si d_α es una dirección de descenso de f , entonces d también es una dirección de descenso.

También se puede notar que $d^T \nabla f(x)$ no puede ser cero, porque si $d^T \nabla f(x) = 0$ implica $d_\alpha^T \nabla f(x) = 0$ y luego del Lema II,

$$\begin{aligned} d_\alpha^T \nabla f(x) &\leq -d_\alpha^T B d_\alpha, \\ d_\alpha^T B d_\alpha &\leq -d_\alpha^T \nabla f(x) = 0. \end{aligned}$$

Entonces: $d_\alpha^T B d_\alpha = 0$, como B es definida positiva implica que $d_\alpha = 0$.

Por tanto el algoritmo debe parar después de calcular d_α .

Proposición I. Sea $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ una función continuamente diferenciable, tal que $\nabla \phi$ satisface

la condición de Lipschitz en $\Gamma \subset \mathbb{R}^n$, es decir existe una constante $k > 0$ tal que

$$\|\nabla\phi(y) - \nabla\phi(x)\|_2 \leq \|y - x\|_2, \forall x, y \in \Gamma$$

Entonces, para $[x, y] \subset \Gamma$, se cumple que:

$$\phi(y) \leq \phi(x) + (y - x)^T \nabla\phi(x) + k\|y - x\|_2^2$$

Demostración: Aplicando el Teorema de valor medio a la función $\phi(x + t(y - x)) : [0, 1] \rightarrow \mathbb{R}$,

entonces existe $\xi \in]0, 1[$ tal que:

$$\phi(1) - \phi(0) = \phi'(\xi), \xi \in]0, 1[.$$

$$\Rightarrow \phi(y) - \phi(x) = (y - x)^T \nabla\phi(x + \xi(y - x)), \text{ para algún } \xi \in]0, 1[$$

Entonces:

$$\phi(y) = \phi(x) + (y - x)^T \nabla\phi(x) - (y - x)^T \nabla\phi(x)(x + \xi(y - x))$$

$$\phi(y) = \phi(x) + (y - x)^T \nabla\phi(x) + (y - x)^T (\nabla\phi(x + \xi(y - x)) - \nabla\phi(x))$$

$$\phi(y) \leq \phi(x) + (y - x)^T \nabla\phi(x) + \|y - x\|_2 \|\nabla\phi(x + \xi(y - x)) - \nabla\phi(x)\|_2$$

$$\phi(y) \leq \phi(x) + (y - x)^T \nabla\phi(x) + \|y - x\|_2 (k\xi\|y - x\|_2)$$

$$\phi(y) \leq \phi(x) + (y - x)^T \nabla\phi(x) + k\|y - x\|_2^2$$

Por tanto:

$$\phi(y) \leq \phi(x) + (y - x)^T \nabla\phi(x) + k\|y - x\|_2^2$$

□

Lema V. Existe $\tau > 0$ tal que cualquier $x \in \Omega_a$ y vector d calculado en el Paso 1 del algoritmo FDIPA, además las condiciones del Paso 2; (34), (35) y (36) son verificadas para todo $t \in [0, \tau]$.

Demostración: Veamos que se satisface la condición $g_i(x + td) < 0$, que se aplica cuando

$\bar{\lambda}_i \geq 0$. De la proposición I existe $k_i > 0$ tal que,

$$g_i(x + td) \leq g_i(x) + td^T \nabla g_i(x) + t^2 k_i \|d\|_2^2$$

para cualquier x y $x + td \in \Omega_a$, tal que $[x, x + td] \subset \Omega_a$. Por tanto, si

$$g_i(x) + td^T \nabla g_i(x) + t^2 k_i \|d\|_2^2 \leq 0$$

entonces $g_i(x + td) < 0$. De la ecuación (22):

$$\Lambda \nabla g^T(x) d + G(x) \bar{\lambda} = -\rho \Lambda \omega,$$

se tiene que:

$$\lambda_i \nabla g_i^T(x) d + g_i(x) \bar{\lambda}_i = -\rho \lambda_i \omega_i$$

$$\nabla g_i^T(x) d = -\rho \omega_i - g_i(x) \frac{\bar{\lambda}_i}{\lambda_i}$$

la desigualdad es equivalente a:

$$g_i(x) + t(-\rho \omega_i - g_i(x) \frac{\bar{\lambda}_i}{\lambda_i}) + t^2 k_i \|d\|_2^2 < 0$$

$$g_i(x)(1 - t \frac{\bar{\lambda}_i}{\lambda_i}) - t \rho \omega_i + t^2 k_i \|d\|_2^2 < 0$$

desde que $\lambda_i > 0$

Esta desigualdad es satisfecha si

$$1 - t \frac{\bar{\lambda}_i}{\lambda_i} > 0; \quad \rho \omega_i - t k_i \|d\|_2^2 > 0$$

De acuerdo a estas dos desigualdades cualquier t debe verificar que:

$$t < \frac{\lambda_i}{\bar{\lambda}_i} \quad ; \quad t < \frac{\rho\omega_i}{(k_i\|d\|_2^2)}$$

entonces ambas desigualdades son verdaderas para todo t tal que:

$$t < \min \left\{ \frac{\lambda_i}{\bar{\lambda}_i}; \frac{\rho\omega_i}{k_i\|d\|_2^2} \right\}$$

Como λ_i es acotada, en el caso donde $g_i(x) \geq -\bar{g}$. Por las igualdades (39) y (40) y la Hipótesis

V se deduce que existe un $\tau_i > 0$ tal que se satisface para cualquier $t \in [0, \tau_i]$. En efecto:

De la hipótesis V: $0 \leq \lambda_i \leq \lambda^S$ y $\bar{\lambda}_i \geq \lambda^I$ para cada $g_i(x) \geq -\bar{g}$, entonces

$$0 \leq \frac{\lambda_i}{\bar{\lambda}_i} \leq \frac{\lambda^S}{\lambda^I}.$$

De (39) y (40):

$$\|d\|_2^2 \leq \delta^2 \|d_\alpha\|_2^2 \leq \frac{\delta^2 \rho}{\varphi}, \delta > 1, \varphi_0 > 0$$

$$\Rightarrow \frac{\rho\omega_i}{k_i\|d\|_2^2} \geq \frac{\varphi_0\omega_i}{\delta^2 k_i}$$

$$\Rightarrow t \in [0, \underbrace{\min \left\{ \frac{\lambda^S}{\lambda^I}, \frac{\varphi_0\omega_i}{\delta^2 k_i} \right\}}_{\tau_i}]$$

Desde que g es continuamente diferenciable y d es acotado, cuando $g_i(x) < -\bar{g}$ entonces un

$\tau_i > 0$ tal que $g_i(x + td) < 0, \forall t \in [0, \tau_i]$.

De igual procedemos cuando $\lambda_i < 0$, puede ser probado que existe $\tau_i > 0$ tal que $g_i(x + td) \leq g_i(x)$ es satisfecho para cualquier $t \in [0, \tau_i]$.

De nuevo por la Proposición I, existe $k_f > 0$ tal que,

$$f(x + td) \leq f(x) + td^T \nabla f(x) + t^2 \|d\|_2^2 k_f$$

para cualquier $[x, x + td] \subset \Omega_a$. Por tanto la condición $f(x + td) \leq f(x) + t\eta \nabla d^T f(x)$ es satisfecha para cualquier t tal que,

$$f(x) + td^T \nabla f(x) + t^2 \|d\|_2^2 k_f \leq f(x) + t\eta d^T \nabla f(x)$$

o de forma equivalente que se satisfaga que

$$t(1 - \eta) d^T \nabla f(x) \leq -t^2 \|d\|_2^2 k_f$$

$$t \leq \frac{(\eta - 1) d^T \nabla f(x)}{\|d\|_2^2 k_f}$$

Por el Lema III se tiene que,

$$\frac{(\eta - 1) d^T \nabla f(x)}{\|d\|_2^2 k_f} \leq \frac{(\eta - 1) \xi d_\alpha^T \nabla f(x)}{k_f \|d\|_2^2}$$

Del Lema II y la hipótesis VI

$$\leq \frac{(\eta - 1) \xi (-d_\alpha^T B d_\alpha)}{k_f \|d\|_2^2} \leq \frac{(\eta - 1) \xi (-\sigma_1 \|d_\alpha\|_2^2)}{k_f \|d\|_2^2}$$

Usando que existe $\delta > 1$, tal que: $\|d\|_2^2 \leq \delta \|d_\alpha\|_2^2$

$$\frac{(\eta - 1) \xi (\sigma_1 (-\|d_\alpha\|_2^2))}{k_f \|d\|_2^2} \leq \frac{(1 - \eta) \xi \sigma_1}{k_f \delta^2}$$

Así, $t \leq \frac{(1 - \eta) \xi \sigma_1}{k_f \delta^2}$. Por tanto, existe $\tau > 0$ que satisface las condiciones del Lema. □

Para el caso que alguna de las restricciones son funciones lineales, no es necesario que se aplique una deflexión relativa a la función, por ejemplo si $g_q(x)$ es lineal se puede demostrar que

$$g_q(x + td) = g_q(x) + td^T \nabla g_q(x),$$

y ya no es necesario la existencia de k_f . Así, la condición de desigualdad es satisfecha para cualquier $t < \lambda_q/\bar{\lambda}_q$, para todo $\lambda_q \geq 0$. En el caso $\bar{\lambda}_q < 0$, la condición también es satisfecha para todo $t > 0$. Cualquier sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el algoritmo FDIPA está contenida en Ω_a . Desde que Ω_a es compacto, la sucesión tiene un punto de acumulación en Ω_a . El siguiente Lema demostrará que los puntos de acumulación son puntos estacionarios del problema y luego que estos mismos son puntos KKT.

Lema VI. *Todo punto de acumulación x^* de la sucesión $\{x^k\}_{k \in K}$ generada por el algoritmo FDIPA es un punto estacionario del problema. Mas aún $(x^*, \lambda_\alpha(x^*))$ es un par estacionario.*

Demostración: Considere la sucesión $\{x^k\}_{k \in K}$ donde $K \subset \mathbb{N}$, converge a x^* . Desde que λ, B, ω, ρ son acotadas, existe $K_1 \subset K$ tal que $\{x^k, \lambda^k, B^k, \omega^k, \rho^k\}_{k \in K_1}$ convergen para $x^*, \lambda^*, B^*, \omega^*, \rho^*$. Considerando que d depende continuamente en $x, \lambda, B, \omega, \rho$, deducimos que $\{d^k\}_{k \in K_1} \rightarrow d^*$, con

$$d^* = d(x^*, \lambda^*, B^*, \omega^*, \rho^*).$$

Considerando ahora $K_2 \subset K_1$ tal que $\{t^k\}_{k \in K_2} \rightarrow t^*$. Del Lema IV se tiene que $t^* > 0$. Aplicando la condición de la primera desigualdad del Paso 2 del algoritmo FDIPA, escribimos,

$$f(x^{k+1}) \leq f(x^k) + \eta t^k d^{kT} \nabla f(x^k)$$

donde se asume que $k \in K_2$, y $k+1$ no necesariamente pertenece a K_2 . Llamando $fol(k)$ los elementos que pertenecen a K_2 . Luego, desde que $fol(k) \geq k+1$.

$$f(x^{fol(k)}) \leq f(x^k) + \eta t^k d^{kT} \nabla f(x^k),$$

y tomando límite ambos lados para $k \rightarrow \infty$, tenemos

$$f(x^*) \leq f(x^*) + \eta t^* d^{*T} \nabla f(x^*),$$

y así, $d^{*T} \nabla f(x^*) \geq 0$. Considerando el resultado del Lema IV en el límite para $k \rightarrow \infty$, tenemos $d^{*T} \nabla f(x^*) = 0$. También se sigue del Lema IV que $d_\alpha^{*T} \nabla f(x^*) = 0$ y del Lema III que $d_\alpha^* = 0$. Por tanto (x^*, λ_α^*) es un par estacionario. \square

Lo que sigue es probar la convergencia a puntos KKT. Para esto se debe suponer una hipótesis adicional.

Hipótesis VIII. *El punto estacionario del Problema I son puntos aislados ó ellos constituyen conjunto compactos aislados con las mismas restricciones activas.*

Teorema VI. *Cualquier punto de acumulación x^* de cualquier sucesión generada por el algoritmo FDIPA es un punto de Karush Kuhn Tucker del problema.*

Demostración: Como x^* es un punto estacionario del problema, solo es necesario probar que los multiplicadores de Lagrange $\lambda_\alpha(x^*)$ son no negativos. Tenemos que $d_\alpha = 0$ en un punto estacionario. Luego, de la igualdad (18), los $(\lambda_\alpha)_i(x^*) = 0$ en el caso donde $g_i(x^*) < 0$. Considere ahora que una restricción g_h tal que $g_h(x^*) = 0$. Como el método tiene factibilidad estricta,

$$g_h(x^k) < 0, \quad \forall k \in \mathbb{N}.$$

Luego, definimos la sucesión $\{x^k\}_{k \in K}$, $K \subset \mathbb{N}$, convergiendo a x^* , tal que

$$g_h(x^k) > g_h(x^{k-1}), \quad \forall k \in K$$

Notar que $k - 1$ puede no pertenecer a K . Se deduce de la desigualdad $g_i(x + td) \leq g_i(x)$, si $\bar{\lambda}_i < 0$, que para estos términos de la sucesión, $\bar{\lambda}_h^{k-1} \geq 0$.

Ahora, procediendo por contradicción y suponiendo que $\lambda_{\alpha h}(x^*) < 0$. De las desigualdades $\rho \leq \varphi \|d_\alpha\|_2^2 \wedge \rho \geq \varphi_0 \|d_\alpha\|_2^2$ se deduce que $\bar{\lambda}_h(x^*) < 0$. Luego, de la Hipótesis VIII existe una bola abierta,

$$\Gamma(\gamma_h) = \{x : \|x - x^*\|_2 < \gamma_h\}$$

tal que:

$$\bar{\lambda}_h(x) < 0, \forall x \in \mathring{\Omega}_a \cap \Gamma(\gamma_h),$$

y para todo λ, B, ω, ρ generado por el algoritmo.

Considere ahora la sucesión $\{x^{k-1}\}_{k \in K}$. Como $\bar{\lambda}^{k-1} \geq 0$, para cualquier $k \in K$, tenemos que $x^{k-1} \notin \Gamma(\gamma_h)$. Por tanto, $\{x^{k-1}\}_{k \in K}$ esta en el conjunto compacto $\mathring{\Omega} - \Gamma(\gamma_h)$ y tiene puntos de acumulación. En consecuencia del Lema VI, estos puntos son puntos estacionarios del problema. Sea \bar{x} uno de estos puntos, y $\{x^{k-1}\}_{k \in \bar{K}}, \bar{K} \subset K$, una sucesión convergente para \bar{x} . Tenemos que $d(\bar{x}) = 0$. Entonces, para I suficientemente grande,

$$\|d^{k-1}\|_2 \leq \gamma_h, \quad \forall k > I, k \in \bar{K},$$

y en consecuencia de la búsqueda lineal

$$\|x^k - x^{k-1}\|_2 \leq \|d^{k-1}\|_2 < \gamma_h.$$

De este último resultado se tiene una contradicción con la parte $\bar{x} \notin \Gamma(\gamma_h)$. □

El Teorema VI explica que la convergencia de cualquier sucesión por el algoritmo FDI-PA, es condición suficiente para concluir que el punto de convergencia es una solución optima del problema de programación no lineal y que satisface todas las condiciones de Karush Kuhn Tucker. La siguiente sección muestra la aplicación del algoritmo FDIPA en varios ejemplos, quedando demostrado con sus resultados la eficacia del algoritmo en problemas de programación no lineal.

C. Ejemplos de aplicación

La sección muestra la aplicación del algoritmo FDIPA en algunos ejemplos de problemas de programación no lineal. Mostrando los resultados de las iteraciones con figuras que permiten

notar como los resultados del algoritmo van convergiendo.

La implementación se ha realizado en el software Matlab, utilizando los siguientes valores de los parámetros, en todos los ejemplos.

$$\xi = 0,8; \eta = 0,4; \varphi = 1; \nu = 0,8; x_s = 2; \lambda = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, B = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}, \omega = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Ejemplo I. Determinar el mínimo de la función, $f(x, y) = x^2 + 3y^2$, sujeto a:

$$1 - 2x - y \leq 0,$$

$$1 - 2y - x \leq 0.$$

Para las iteraciones del algoritmo FDIPA se ha tomado el punto interior inicial, $x_0 = (1, 1)$.

Los resultados son mostrados en el siguiente cuadro. La solución aproximada del problema es

$(0,42861047, 0,28569476)$, mientras que la solución real es $(\frac{3}{7}, \frac{2}{7})$, entonces

$$E = \|(3/7, 2/7) - (0,428576, 0,285712)\|_2 \approx 5,11 \times 10^{-6},$$

Se observa que la aproximación a la solución real es del orden 10^{-6} .

iteración	paso	x	$f(x)$
1	0,32978	$\begin{bmatrix} 1,429707 & 0,550306 \end{bmatrix}$	2,952574
2	1,00000	$\begin{bmatrix} 1,141401 & 0,243209 \end{bmatrix}$	1,480249
3	0,57426	$\begin{bmatrix} 0,903835 & 0,399927 \end{bmatrix}$	1,296744
4	1,00000	$\begin{bmatrix} 0,814645 & 0,189307 \end{bmatrix}$	0,771158
5	1,00000	$\begin{bmatrix} 0,593158 & 0,318587 \end{bmatrix}$	0,656329
\vdots	\vdots	\vdots	\vdots
45	1,00000	$\begin{bmatrix} 0,428580 & 0,285710 \end{bmatrix}$	0,428571
46	1,00000	$\begin{bmatrix} 0,428578 & 0,285711 \end{bmatrix}$	0,428571
47	1,00000	$\begin{bmatrix} 0,428577 & 0,285712 \end{bmatrix}$	0,428571
48	1,00000	$\begin{bmatrix} 0,428576 & 0,285712 \end{bmatrix}$	0,428571
49	1,00000	$\begin{bmatrix} 0,428576 & 0,285712 \end{bmatrix}$	0,428571

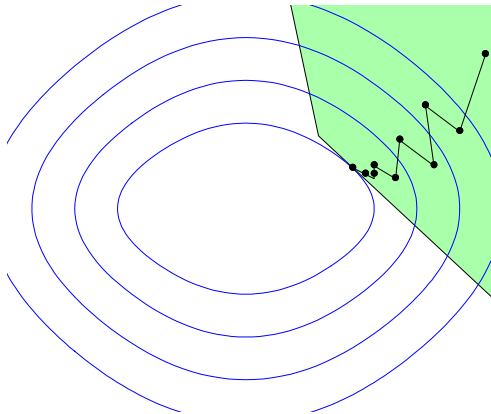


Figura IV: Iteraciones del Ejemplo I por el algoritmo FDIPA.

Ejemplo II. Determinar el mínimo de la función, $f(x, y) = x^2 + \frac{y}{2}$, sujeto a:

$$x^2 + y^2 - 2 \leq 0,$$

$$-x^2 + y \leq 0.$$

Para las iteraciones del algoritmo FDIPA se ha tomado el punto interior inicial, $x_0 = (1, 0,5)$. Los resultados son mostrados en el siguiente cuadro.

iteración	paso	x	$f(x)$
1	1,00000	$\begin{bmatrix} 0,846978 & 0,279269 \end{bmatrix}$	0,857006
2	1,00000	$\begin{bmatrix} 0,556961 & -0,397462 \end{bmatrix}$	0,111474
3	0,75780	$\begin{bmatrix} 0,085301 & -1,320341 \end{bmatrix}$	-0,652894
4	0,43517	$\begin{bmatrix} 0,000244 & -1,324314 \end{bmatrix}$	-0,662157
5	1,00000	$\begin{bmatrix} -0,000337 & -1,340707 \end{bmatrix}$	-0,670353
\vdots	\vdots	\vdots	\vdots
140	1,00000	$\begin{bmatrix} 0,000001 & -1,414214 \end{bmatrix}$	-0,707107
141	0,75780	$\begin{bmatrix} -0,000001 & -1,414214 \end{bmatrix}$	-0,707107
142	1,00000	$\begin{bmatrix} 0,000001 & -1,414214 \end{bmatrix}$	-0,707107
143	0,57426	$\begin{bmatrix} -0,000000 & -1,414214 \end{bmatrix}$	-0,707107
144	0,57426	$\begin{bmatrix} -0,000000 & -1,414214 \end{bmatrix}$	-0,707107

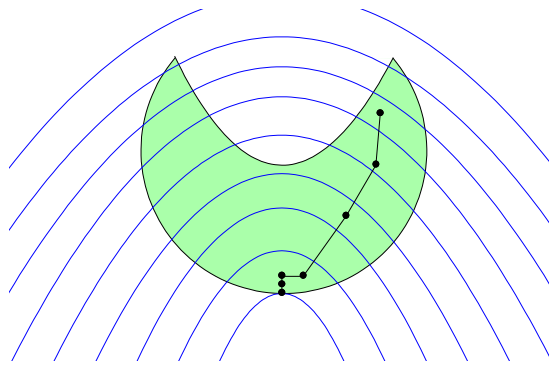


Figura V: Iteraciones del Ejemplo II por el algoritmo FDIPA.

La solución aproximada del problema es $(0,0000000, -1,41415796)$, mientras que la solución real es $(0, -\sqrt{2})$, entonces

$$E = \left\| \begin{pmatrix} 0, -\sqrt{2} \end{pmatrix} - \begin{pmatrix} -0,000000, -1,414214 \end{pmatrix} \right\|_2 \approx 4,38 \times 10^{-7},$$

Se observa que la aproximación a la solución real es del orden 10^{-7} .

Ejemplo III. Determinar el mínimo de la función,

$$f(x, y) = x^2 + (y + 21)^2,$$

sujeto a:

$$\begin{aligned} y - 20 \frac{\text{sen} x}{x} + 20 &\leq 0, \\ -y - 20 \frac{\text{sen}(x - 3)}{x} - 22 &\leq 0, \\ 0,5 - x &\leq 0. \end{aligned}$$

Para las iteraciones del algoritmo FDIPA se ha tomado el punto interior inicial, $x_0 = (15, -20)$.

Se realizó 56 iteraciones con un tamaño de paso que terminó a 0,33 aproximándose al punto $(2,80, -20,62)$. Las iteraciones del algoritmo se pueden ver en la siguiente Figura.

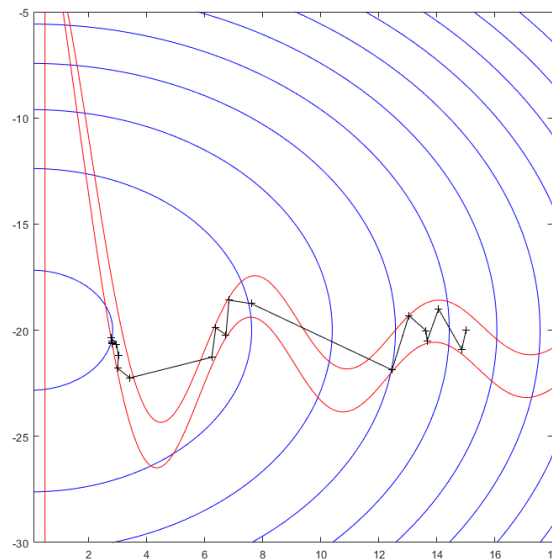


Figura VI: Iteraciones del Ejemplo III por el algoritmo FDIPA.

IV Algoritmo FDIPA-LP

El presente capítulo estudia el algoritmo FDIPA-LP, este algoritmo es una adaptación del algoritmo FDIPA [9], aplicado para problemas de programación lineal. El desarrollo del algoritmo fue realizado por Celis en [14]. El algoritmo es un método de puntos interiores de aplicación práctica en este tipo de problemas, siendo así una forma alternativa del método simplex. El capítulo muestra primero, el estudio de la adaptación del algoritmo FDIPA-LP, segundo el planteamiento del algoritmo y finalmente el estudio del análisis de convergencia. El siguiente capítulo mostrará la aplicación del algoritmo en algunos ejemplos test.

A. Estudio del algoritmo FDIPA-LP

Sea el problema de programación lineal con restricciones de desigualdad menor o igual " \leq ",

$$\begin{aligned} \text{mín} \quad & c^T x \\ \text{s.a.} \quad & Ax \leq b, \end{aligned} \tag{41}$$

donde $A \in \mathbb{R}^{m \times n}$, es la matriz de restricciones, $b \in \mathbb{R}^m$ vector independiente, $c \in \mathbb{R}^n$ es el vector de costos; $x \in \mathbb{R}^n$ es el vector de variables, con $m \geq n \geq 1$ y $\text{rango}(A) = n$.

El análisis del problema (41), es de forma semejante al problema no lineal. Para mostrar esto, primero veamos las definiciones de las funciones que se usan. Sea $I = \{1, 2, \dots, m\}$, para

cada $i \in I$, a_i^T denota la i -ésima fila de A , b_i la i -ésima componente de b . Entonces,

$$Ax - b = \begin{bmatrix} a_1^T x - b_1 \\ \vdots \\ a_i^T x - b_i \\ \vdots \\ a_m^T x - b_m \end{bmatrix}$$

De esta forma tenemos las funciones de restricciones, como las funciones $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ dadas por

$$g_i(x) = a_i^T x - b_i,$$

entonces $g(x) = Ax - b$, donde $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$. La función objetivo $f(x) = c^T x$. Además, queda claro que ambas funciones son continuamente diferenciable \mathcal{C}^∞ . Así, el Problema (41) queda denotado como un problema de programación no lineal.

$$\begin{aligned} \text{mín} \quad & f(x) = c^T x \\ \text{s.a.} \quad & x \in \Omega, \end{aligned} \tag{42}$$

donde el conjunto factible es definido por,

$$\Omega = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i \in I\}$$

cuyo interior es,

$$\overset{\circ}{\Omega} = \{x \in \mathbb{R}^n : g_i(x) < 0, i \in I\} \tag{43}$$

Se puede definir el lagrangiano asociado al Problema (41) como:

$$L(x, \lambda) = f(x) + \lambda^T g(x) = f(x) + \sum_{i=1}^m \lambda_i g_i(x),$$

$$L(x, \lambda) = c^T x + \lambda^T (Ax - b). \quad (44)$$

El gradiente del lagrangiano queda de la forma:

$$\nabla_x L(x, \lambda) = c + A^T \lambda \quad (45)$$

Como se puede observar, la matriz hessiana respectiva queda como $H(x, \lambda) = 0$ una matriz nula.

El Problema de programación lineal (41) debe cumplir las condiciones de optimalidad de primer orden de Karush Kuhn Tucker (5)-(8), de esta manera las condiciones que debe satisfacer el Problema, son las siguientes:

$$c + A^T \lambda = 0, \quad (46)$$

$$G(x) \lambda = 0, \quad (47)$$

$$(Ax - b) \leq 0, \quad (48)$$

$$\lambda \geq 0. \quad (49)$$

donde: $G(x) = \text{diag} (a_1^T x - c_1, \dots, a_m^T x - c_m)$.

Definición XV. El punto $x \in \Omega$ es llamado punto estacionario del problema (41) si existe un $\lambda \in \mathbb{R}^m$ que satisface (46) y (47). Si $\lambda \geq 0$ entonces x es un punto de Karush Kuhn Tucker (KKT).

Siguiendo las ideas básicas del algoritmo FDIPA, se plantea a continuación lo siguiente:

La región factible $\Omega_p = \{x \in \Omega : f(x) \leq p\}$ donde $p \in \mathbb{R}$ tiene interior $\overset{\circ}{\Omega}_p = \{x \in \Omega : f(x) < p\}$. Las hipótesis que se deben suponer son pocas, porque la diferenciabilidad de las funciones ya esta garantizada por ser funciones lineales. Veamos,

Hipótesis IX. Existe un número real p tal que Ω_p es compacto y el interior no vacío.

Hipótesis X. Para todo $x \in \Omega$ los vectores $\nabla g_i(x) = a_i$ para todo $i \in I$, tal que $g_i(x) = 0$, son linealmente independientes.

Lo que ahora sigue es aplicar el algoritmo de Newton al sistema formado por los sistemas (46) y (47), para esto se forma la función $F : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{n+m}$, donde

$$F(x, \lambda) = \begin{bmatrix} A^T \lambda + c \\ G(x) \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

cuyo gradiente queda de la forma:

$$\nabla F(x, \lambda) = \begin{bmatrix} 0 & A^T \\ \Lambda A & G(x) \end{bmatrix}$$

donde: $\Lambda = \text{diag} (\lambda_1, \dots, \lambda_m)$. Entonces, por el algoritmo de Newton se tiene el sistema:

$$F(x^\alpha, \lambda^\alpha) = F(x) + \nabla F(x, \lambda) \begin{pmatrix} x_\alpha - x \\ \lambda_\alpha - \lambda \end{pmatrix}.$$

Luego reemplazando $F(x, \lambda)$ y el gradiente $\nabla F(x, \lambda)$ el sistema queda de la forma,

$$\begin{bmatrix} 0 & A^T \\ \Lambda A & G(x) \end{bmatrix} \begin{pmatrix} x_\alpha - x \\ \lambda_\alpha - \lambda \end{pmatrix} = - \begin{bmatrix} b + A^T \\ G(x) \lambda \end{bmatrix},$$

donde $(x, \lambda) \in \mathring{\Omega} \times \mathbb{R}_+^m$ es el punto inicial ó punto actual y $(x_\alpha, \lambda_\alpha)$ es el nuevo punto estimado.

Denotando a la matriz

$$W(x, \lambda) = \begin{bmatrix} 0 & A^T \\ \Lambda A & G(x) \end{bmatrix}$$

Se define la dirección de descenso como $d_\alpha = x_\alpha - x$. Así, el sistema queda como:

$$\begin{bmatrix} 0 & A^T \\ \Lambda A & G(x) \end{bmatrix} \begin{bmatrix} d_\alpha \\ \lambda_\alpha \end{bmatrix} = \begin{bmatrix} -b \\ 0 \end{bmatrix}. \quad (50)$$

Perturbando el lado derecho del sistema (50) genera una nueva dirección factible hacia el $\mathring{\Omega}$,

$$\begin{bmatrix} 0 & A^T \\ \Lambda A & G(x) \end{bmatrix} \begin{bmatrix} d \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} -b \\ -\rho\lambda \end{bmatrix}, \quad (51)$$

donde $\rho > 0$ es un número positivo apropiado que garantiza que d es una dirección de descenso, y $\bar{\lambda}$ es la nueva estimación de λ . Otra manera de obtener el par $(d, \bar{\lambda})$, es resolver el sistema (50) y el siguiente sistema,

$$\begin{bmatrix} 0 & A^T \\ \Lambda A & G(x) \end{bmatrix} \begin{bmatrix} d_\beta \\ \bar{\lambda}_\beta \end{bmatrix} = \begin{bmatrix} 0 \\ -\lambda \end{bmatrix}, \quad (52)$$

donde,

$$d = d_\alpha + \rho d_\beta, \quad (53)$$

$$\lambda = \lambda_\alpha + \rho \lambda_\beta. \quad (54)$$

una vez calculado una dirección de descenso factible d , se puede determinar la siguiente iteración.

Muchas veces el tamaño de paso de la dirección d no permite que la nueva estimación sea un punto interior de la región factible y no disminuya la función objetivo. Para evitar esta dificultad se mejora la búsqueda de la dirección d reduciendo el tamaño de paso, con un factor $t \in]0, 1[$, tal que $g(x + td) < 0$ y $f(x + td) < f(x)$. Así esta nueva estimación $x + td$ asegura que los puntos de la sucesión generada pertenezcan al $\mathring{\Omega}$. Recordemos que $f(x) = c^T x$, evaluando

f en el punto $x + td$,

$$f(x + td) = c^T(x + td) = f(x) + tc^T d. \quad (55)$$

Si $c^T d < 0$, se obtiene que,

$$f(x + td) < f(x), \text{ para todo } t \in \mathbb{R}. \quad (56)$$

El resultado (56) se interpreta que la reducción de la función objetivo no depende del factor $t \in]0, 1[$, sólo de la condición $c^T d < 0$.

Lo que sigue estudiaremos bajo que condiciones se satisface la nueva estimación $x + td$ pertenesca a la región factible. Evaluando cada componente g_i , $i \in I$, en el punto $x + td$.

$$g_i(x + td) = [g(x + td)]_i = [A(x + td) - b]_i,$$

entonces:

$$g_i(x + td) = g_i(x) + ta_i^T d. \quad (57)$$

Analizando todas la posibilidades que pueda tomar el valor $a_i^T d$:

- Si $a_i^T d < 0$ entonces $g_i(x + td) < g_i(x)$, para todo $t \in \mathbb{R}_+$.
- Si $a_i^T d = 0$ entonces $g_i(x + td) = g_i(x)$, para todo $t \in \mathbb{R}$.
- Si $a_i^T d > 0$, para garantizar la factibilidad entonces $g_i(x + td) \leq 0$, ($a_i^T(x + td) - b \leq 0$),

$$t \leq \frac{-a_i^T x + b}{a_i^T d} = \frac{-g_i(x)}{a_i^T d} \quad (58)$$

Por tanto el valor adecuado del parámetro t es,

$$t = \min_{i \in I} \left\{ \frac{-g_i(x)}{a_i^T d} : a_i^T d > 0 \right\} \quad (59)$$

Luego de este análisis, podemos observar que a diferencia del algoritmo FDIPA, la condición de Armijo ya no se utiliza, es reemplazada por la igualdad (59). Quedando el siguiente algoritmo modificado, llamado algoritmo FDIPA-LP, planteado por Celis en [14].

Algoritmo 2: FDIPA-LP

Entrada: Considere los parámetros $\xi \in]0, 1[$, $\varphi > 0$, $\mu > 0$, $\gamma > 0$, λ^s .

Datos iniciales: $x^0 \in \mathring{\Omega}$ y $\lambda^0 \in \mathbb{R}_+^m$

■ **Paso 1: Calculando las direcciones de descenso factible d^k y $\bar{\lambda}^k$.**

(a) Calcular $(d_\alpha^k, \lambda_\alpha^k)$ al resolver:

$$A^T \lambda_\alpha^k = -b, \quad (60)$$

$$\Lambda^k A d_\alpha^k + G(x^k) \lambda_\alpha^k = 0. \quad (61)$$

Si $d_\alpha^k = 0$ entonces **FIN**.

(b) Calcular $(d_\beta^k, \lambda_\beta^k)$ al resolver:

$$A^T \lambda_\beta^k = 0, \quad (62)$$

$$\Lambda^k A d_\beta^k + G(x^k) \lambda_\beta^k = -\lambda^k. \quad (63)$$

(c) Calcular ρ^k . Si $d_\beta^{kT} b > 0$ entonces ρ^k es definido por,

$$\rho^k = \min \left\{ \varphi \|d_\alpha^k\|_2^2, (\xi - 1) d_\alpha^{kT} b / d_\beta^{kT} b \right\}, \quad (64)$$

$$\text{Caso contrario, definir,} \quad \rho^k = \varphi \|d_\alpha^k\|_2^2. \quad (65)$$

(d) Definir la dirección de búsqueda d^k ,

$$d^k = d_\alpha^k + \rho^k d_\beta^k, \quad \bar{\lambda}^k = \lambda_\alpha^k + \rho^k \lambda_\beta^k. \quad (66)$$

■ **Paso 2: Búsqueda lineal.** Si $a_i^T d^k > 0$; $i \in \{1, 2, \dots, m\}$, se calcula

$$\bar{t} = \min_{i \in I} \left\{ \frac{-g_i(x^k)}{a_i^T d^k} \right\}, \quad \text{de otra manera: } \bar{t} = \infty. \quad (67)$$

$$\text{Fijar } t = \min \{ \max \{ \gamma \bar{t}, \bar{t} - \|d^k\|_2 \}, 1 \}. \quad (68)$$

■ **Paso 3: Actualización.**

(a) Tomar $x^k = x^k + t d^k$.

(b) Calcular $\lambda_i = \min \{ \max \{ \lambda_{\alpha_i}^k, \mu \|d_\alpha^k\|_2^2 \}, \lambda^s \}$, para todo $i \in I$. Ir al paso 1.

El algoritmo esta diseñado para que la sucesión $\{x^k\}$ pertenezca dentro de la región factible. El Paso 1(b) obtenemos el valor de la deflexión que garantiza la dirección de descenso de d^k . El Paso 2 calcula el tamaño de paso que garantiza la factibilidad y con la expresión (68) obtenemos la factibilidad estricta. El Paso 3(b), usamos la misma regla del algoritmo FDIPA para actualizar el multiplicador de Lagrange λ^k .

B. Análisis de convergencia

La sección trata del estudio sobre las condiciones de cuando la sucesión generada por el algoritmo FDIPA-LP (x^k, λ_α^k) es convergente a un punto (x^*, λ_α^*) , a partir de un punto interior inicial $x^0 \in \mathring{\Omega}$. Además con las Hipótesis IX y X se demuestra que x^* es un punto KKT.

Para comenzar, analizamos la matriz $W(x, \lambda)$,

$$W(x, \lambda) = \begin{bmatrix} 0 & A^T \\ \Lambda A & G(x) \end{bmatrix}, \quad (69)$$

donde $x \in \Omega$ y $\lambda \in \mathbb{R}_+^n$. La matriz W es no singular, la demostración se puede ver en [13]. De esta forma se determinan de forma única a $d_\alpha, \lambda_\alpha, d_\beta, \lambda_\beta$ en el Paso 1. Además, esto garantiza que no se produzcan ambigüedades y el algoritmo quede bien definido.

El caso $d_\alpha^k = 0$ en el Paso 1, nos muestra que el par (x^k, λ_α^k) satisfacen el sistema (60) y (61). Como $G(x_k) < 0$ entonces $\lambda_\alpha^k = 0$, luego $A^T x_\lambda^k = 0$ entonces $b = 0$ lo cual es una contradicción si $b \neq 0$. Por tanto cuando $b \neq 0$, el algoritmo produce una sucesión infinita de λ_α^k . Por esto, de aquí en adelante se supondrá que $d_\alpha^k \neq 0$ en cada iteración.

Las iteraciones del algoritmo FDIPA-LP generan puntos $x^k \in \mathring{\Omega}$ y $\Lambda^k > 0$. Veamos, el siguiente Lema que demuestra que d^k es una dirección de descenso.

Lema VII. Sea $\xi \in (0, 1]$, si $d \neq 0$ y $d_\alpha \neq 0$, luego

$$d^T b \leq \xi d_\alpha^T b < 0 \quad (70)$$

Demostración: Del Paso 1 a) multiplicando a la ecuación (60) por d_α^T

$$d_\alpha^T A^T \lambda_\alpha = -d_\alpha^T b, \quad (71)$$

desde que $x^* > 0$, multiplicando a la ecuación (61) por $\lambda_\alpha^T \Lambda^{-1}$,

$$\lambda_\alpha^T \Lambda^{-1} (\Lambda A d_\alpha + G(x) \lambda_\alpha = 0),$$

$$\lambda_\alpha^T A d_\alpha + \lambda_\alpha^T \Lambda^{-1} G(x) \lambda_\alpha = 0, \quad (72)$$

de (71) $\lambda_\alpha^T A d_\alpha = -d_\alpha^T b$ reemplazando en (72),

$$d_\alpha^T b = \lambda_\alpha^T \Lambda^{-1} G(x) \lambda_\alpha \quad (73)$$

claramente $\Lambda^{-1} G(x)$ es una matriz definida negativa. Por tanto, la ecuación (73) para cualquier $\lambda_\alpha \neq 0$, es,

$$d_\alpha^T b = \lambda_\alpha^T \Lambda^{-1} G(x) \lambda_\alpha < 0. \quad (74)$$

También de la ecuación $d = d_\alpha + \rho d_\beta$ multiplicando por b , tenemos la igualdad,

$$d^T b = d_\alpha^T b + \rho d_\beta^T b. \quad (75)$$

Por el algoritmo FDIPA-LP el Paso 1 define como obtener el valor de ρ siendo no negativo en cualquier caso.

- Si $d_\beta^T b \leq 0$ de la ecuación (75) y (74) se tiene que:

$$d^T b = d_\alpha^T b + \rho d_\beta^T b$$

$$\Rightarrow d^T b \leq d_\alpha^T b < 0 \quad (76)$$

- Si $d_\beta^T b > 0$ del Paso 1 el valor de ρ permite la desigualdad $\rho \leq (\xi - 1) \frac{d_\alpha^T b}{d_\beta^T b}$,

$$\Rightarrow \rho d_\beta^T b + d_\alpha^T b \leq \xi d_\alpha^T b$$

$$d^T b \leq \xi d_\alpha^T b \quad (77)$$

Por tanto de las desigualdades (74), (76) y (77) tenemos:

$$d^T b \leq \xi d_\alpha^T b \leq 0 \quad (78)$$

□

Como consecuencia directa del Lema VII, f es monótona decreciente, d_α y d son direcciones de descenso de f . En efecto por construcción para $t \in]0, 1]$ y del Lema VII entonces se cumple que: $t d^T b < 0$. Al evaluar f en $x + td$, se tiene que,

$$f(x + td) = f(x) + t d^T b$$

entonces, se demuestra que $f(x + td) < f(x)$.

De otro lado, con la selección de ρ en el Paso 1 del algoritmo FDIPA-LP, garantizamos que

$$0 < \rho \leq \varphi \|d_\alpha\|_2^2 \quad (79)$$

Es decir a medida que $d_\alpha \rightarrow 0$ entonces $\rho \rightarrow 0$. El siguiente Lema demostrará la factibilidad de d .

Lema VIII. *Existe $\tau > 0$ tal que para cualquier Ω_p y d calculado en el Paso 1 del algoritmo 2 FDIPA-LP, tenemos que $g(x + td) \leq 0$ para todo $t \in [0, \tau]$.*

Demostración: La longitud de paso t esta definida por la fórmula $t = \min\{\max\{\gamma \bar{t}, \bar{t} - \|d\|\}, 1\}$.

Como todas las restricciones son lineales, deben satisfacer la condición de búsqueda lineal, que las siguientes desigualdades deben ser verdaderas,

$$g_i(x + td) = g_i(x) + t a_i^T d \leq 0, \quad (80)$$

para todo $i \in I$.

Si $a_i^T \leq 0$, la desigualdad es satisfecha para cualquier $t > 0$. De otro lado, si $a_i^T d > 0$, se probará mas adelante que sólo es suficiente probar que,

$$g_i(x) \left(1 - t \frac{\bar{\lambda}_i}{\lambda_i}\right) - t\rho \leq 0. \quad (81)$$

Sabemos que se satisface $\rho t > 0$ y $g(x) < 0$. Entonces la desigualdad (81) es satisfecha si:

$$t \frac{\bar{\lambda}_i}{\lambda_i} \leq 1 \quad (82)$$

Desde que λ es acotado por la definición en el paso 3, y como $\lambda_\alpha, \lambda_\beta$ y ρ estan acotados superiormente, también $\bar{\lambda}$ es acotado superiormente. Entonces, existe τ tal que $\frac{\lambda_i}{\bar{\lambda}_i} > \tau$.

Por tanto, para todo $i \in I$ y para todo $t \in [0, \tau]$, tenemos $g_i(x + td) \leq 0$. □

Considerando que $f(x^0) = p$, para un valor inicial $x^0 \in \mathring{\Omega}$ y la función esta decreciendo, tenemos que cualquier sucesión $\{x^k\}$ generada por el algoritmo esta contenida en Ω_p , entonces desde la hipótesis IX, la sucesión $\{x^k\}_{k \in \mathbb{N}}$ tienen un punto de acumulación en Ω_p . El siguiente Lema garantiza que el punto de acumulación de la sucesión $\{x^k\}_{k \in \mathbb{N}}$ generada por el algoritmo FDIPA-LP converge a un punto estacionario.

Lema IX. *Todo punto de acumulación de la sucesión $\{x^k\}_{k \in \mathbb{N}}$, generada por el algoritmo FDIPA-LP, es un punto estacionario del problema. Además de eso, (x^*, λ_α^*) es un par estacionario, donde λ_α^* es el único vector multiplicador asociado con x^* .*

Consideremos la subsucesión $\{x^k\}_{k \in K}$, donde $K \subset \mathbb{N}$, convergiendo a x^* . Por construcción de $\lambda_i^k \in [0, \lambda^s]$ y de la cota en (79) se tiene que $\{\lambda^k\}_{k \in \mathbb{N}}$ y $\{\rho^k\}_{k \in \mathbb{N}}$ estan acotados, entonces existe una subsucesión convergente, de la forma $\{(x_k, \lambda^k, \rho^k)\}_{k \in K_1}$ que converga a (x^*, λ^*, ρ^*) .

Desde que d^k depende continuamente de x^k, λ^k y ρ^k , obtenemos que $\{d^k\}_{k \in K_1} \rightarrow d^*$,

con $d^* = d(x^*, \lambda^*, \rho^*)$.

Desde que $t^k \in]0, 1]$ para todo k , se sigue del Lema VII. que existe \hat{t} tal que

$$f(x^{k+1}) \leq f(x^k) + \hat{t}d^{kT}b,$$

para todo $k \in K_1$, donde asumimos que $k + 1$ no necesariamente pertenece a K_1 .

$$f(x^{k'}) < f(x^k) + \hat{t}d^{kT}b \text{ para todo } k' > k + 1,$$

y tomando límite en ambos lados $k \rightarrow \infty$,

$$f(x^*) \leq f(x^*) + \hat{t}d^{*T}b,$$

de lo que sigue

$$0 \leq d^{*T}b. \quad (83)$$

Considerando ahora el resultado del Lema (VII) en el límite para $k \rightarrow \infty$, tenemos

$$d^{*T}b = 0.$$

Además, se tiene que $d_\alpha^* = 0$. Tomando los límites a los sistemas (60) y (61), tenemos

$$A^T \lambda^* = -b \quad (84)$$

$$G(x^*) \lambda_\alpha^* = 0 \quad (85)$$

Por tanto, x^* es estacionario, con multiplicador de Lagrange λ_α^* . De la ecuación (84) y la hipótesis X tenemos la unicidad de λ^* .

Lema X. Suponga que la sucesión $\{x^k\}_{k \in \mathbb{N}}$ es acotada. Sea L el conjunto de los puntos límites

de $\{x^k\}_{k \in \mathbb{N}}$, entonces L es un conjunto conexo y compacto con la mismas restricciones activas.

Demostración: La prueba es analoga a la del Lema 3.5 y el Lema 3.6 en [13]. El Lema 3.5 prueba que L es conexo y compacto y el Lema 3.6 prueba que L posee un único λ^* asociado con este conjunto. \square

El lema X aparece como una hipótesis en el algoritmo FDIPA [9] esto ayuda para probar la convergencia de la suceción de puntos generados por el algoritmo a un punto KKT.

Lema XI. La dirección d calculada por el algoritmo FDIPA-LP satisface que,

$$d^T a_i < 0, \text{ para cada } \bar{\lambda}_i < 0, i \in \{1, 2, \dots, m\}. \quad (86)$$

Demostración: Desarrollando las ecuaciones (61) y (63):

$$\Lambda A d_\alpha + G(x) \lambda_\alpha = 0,$$

$$\Lambda A d_\beta + G(x) \lambda_\beta = -\lambda.$$

Tenemos el productos de las matrices y vectores de la siguiente forma:

$$\begin{bmatrix} \lambda_1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & \lambda_i & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & \lambda_m \end{bmatrix} \begin{bmatrix} a_1^T \\ \vdots \\ a_i^T \\ \vdots \\ a_m^T \end{bmatrix} \begin{bmatrix} d_{\alpha_1} \\ \vdots \\ d_{\alpha_i} \\ \vdots \\ d_{\alpha_n} \end{bmatrix} + \begin{bmatrix} g_1(x) & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & g_i(x) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & g_m(x) \end{bmatrix} \begin{bmatrix} \lambda_{\alpha_1} \\ \vdots \\ \lambda_{\alpha_i} \\ \vdots \\ \lambda_{\alpha_m} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \lambda_1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & \lambda_i & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & \lambda_m \end{bmatrix} \begin{bmatrix} a_1^T \\ \vdots \\ a_i^T \\ \vdots \\ a_m^T \end{bmatrix} + \begin{bmatrix} d_{\beta_1} \\ \vdots \\ d_{\beta_i} \\ \vdots \\ d_{\beta_n} \end{bmatrix} + \begin{bmatrix} g_1(x) & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & g_i(x) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & g_m(x) \end{bmatrix} \begin{bmatrix} \lambda_{\beta_1} \\ \vdots \\ \lambda_{\beta_i} \\ \vdots \\ \lambda_{\beta_m} \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_i \\ \vdots \\ \lambda_m \end{bmatrix}$$

entonces para todo $i \in I$,

$$\lambda_i a_i^T d_\alpha + \lambda_{\alpha i} g_i(x) = 0,$$

$$\lambda_i a_i^T d_\beta + \lambda_{\beta i} g_i(x) = -\lambda_i,$$

$$a_i^T d_\alpha = -\frac{\lambda_{\alpha i}}{\lambda_i} g_i(x), \quad (87)$$

$$a_i^T d_\beta = -\frac{\lambda_{\beta i}}{\lambda_i} g_i(x) - 1, \quad (88)$$

Aplicando la multiplicación por a_i^T a la ecuación $d = d_\alpha + \rho d_\beta$

$$a_i^T d = a_i^T d_\alpha + \rho a_i^T d_\beta. \quad (89)$$

Sustituyendo las ecuaciones (87) y (88) en la ecuación (89) se obtiene

$$\begin{aligned} a_i^T d &= -\frac{\lambda_{\alpha i}}{\lambda_i} g_i(x) - \rho \frac{\lambda_{\beta i}}{\lambda_i} g_i(x) - \rho \\ &= -\frac{g_i(x)}{\lambda_i} \left(\lambda_{\alpha i} + \rho \lambda_{\beta i} \right) - \rho \\ &= -\frac{g_i(x)}{\lambda_i} \bar{\lambda}_i - \rho, \end{aligned} \quad (90)$$

En la ecuación (90), si $\bar{\lambda}_i < 0$ y de las hipótesis $\lambda_i > 0, \rho > 0$, entonces:

$$a_i^T d < 0 \quad (91)$$

□

Las funciones de restricciones asociada con los multiplicadores negativos son funciones decrecientes ($g_i(x+td) < g_i(x)$). Esto garantiza que los puntos estacionarios que no son puntos KKT pueden ser evitados.

El siguiente Teorema demuestra que el algoritmo converge a una solución del problema.

Teorema VII. *Cualquier punto de acumulación x^* de cualquier sucesión generada por el algoritmo FDIPA-LP, $\{x^k\}_{k \in \mathbb{N}}$ es un punto Karush-Kuhn-Tucker del problema.*

Demostración: Del Lema IX todo punto x^* es un punto estacionario del problema, solamente queda probar que los multiplicadores de Lagrange λ^* sean no negativos. Se conoce que en un punto estacionario $d_\alpha^* = 0$ y $d^* = 0$ en un punto estacionario. Luego en la ecuación (61) se tiene que:

$$G(x^*) \cdot \lambda_\alpha^* = 0,$$

cuando $\lambda_{\alpha i}^* = 0$, en el caso $g_i(x^*) < 0$.

Considerando una restricción, $g_j(x^*) = 0$, esto tiene algunas consecuencias. Como el método es estrictamente factible,

$$g_j(x^k) < 0, \forall k \in \mathbb{N}$$

Entonces se define la sucesión $\{x^k\}_{k \in K}$, $K \subset \mathbb{N}$ que converge para x^* tal que,

$$g_j(x^k) > g_j(x^{k-1}), \text{ para todo } k \in K.$$

Notando que $k-1$ puede no pertenecer a K y del Lema XI, obtendríamos que $\bar{\lambda}_j^{k-1} \geq 0$. Veamos

la demostración, procediendo por contradicción, y supongamos $\lambda_{\alpha j}^* < 0$, entonces se obtiene de la igualdad $0 \leq \rho \leq \varphi \|d_\alpha^*\|$ que ρ^*

$$\lambda^* = \lambda_\alpha^* + \rho^* \lambda_\beta^*$$

obtenemos que $\bar{\lambda}_j^* < 0$. Entonces del Lema X, existe una bola abierta $B(\gamma_j) = \{x : \|x - x^*\| < \gamma_j\}$ tal que $\bar{\lambda}_j < 0$ para todo $x \in \mathring{\Omega} \cap B(\gamma_j)$ y cualquier λ y ρ generando por el algoritmo FDIPA-LP.

Consideremos ahora la sucesión $\{x^{k-1}\}_{k \in K}$. Como $\bar{\lambda}_j^{k-1} \geq 0$, para cualquier $k \in K$, se tiene que $x^{k-1} \notin B(\gamma_j)$. Entonces $\{x^{k-1}\}_{k \in K}$ esta perteneciendo al conjunto posee un punto de acumulación. En consecuencia del Lema IX este punto es un punto estacionario del problema.

Sea \bar{x} es uno de estos puntos y $\{x^{k-1}\}_{k \in \bar{K}}$, $\bar{K} \subset K$, la sucesión convergente a \bar{x} , por tanto se tiene que $\|d^{k-1}\| \rightarrow 0$. Entonces para I suficientemente grande $\|d^{k-1}\| < \gamma_j$ para todo $k > I$ y en consecuencia de la búsqueda lineal

$$\|x^k - x^{k-1}\| = |t^{k-1}| \|d^{k-1}\|$$

$$\|x^* - \bar{x}\| < \gamma_j$$

Este resultado es una contradicción con el hecho que $\bar{x} \notin B(\gamma_j)$, probando el Teorema. \square

V Implementación numérica y resultados

El presente capítulo muestra dos aplicaciones del algoritmo FDIPA-LP, primero en problemas de programación lineal de gran escala y segundo en problemas de transporte. Los problemas de gran escala tienen algunas dificultades como restricciones redundantes, mal condicionamiento y no ser acotadas. Por esto la primera y segunda sección explica la metodología que se va seguir a los problemas de gran escala, para conseguir el resultado de alcanzar su valor objetivo óptimo usando la forma dual del problema. La última sección muestra los resultados de la aplicación en problemas de transportes siguiendo la misma metodología.

A. Problema primal-dual

La resolución del primal de un problema de programación lineal en muchos casos es más práctico resolver en su forma dual, porque en la forma dual puede disminuir el número de variables, resultando en un análisis menos complicado. Además, se sabe que el valor óptimo de ambos problemas es el mismo. Los problemas que vamos a resolver son ejemplos de Ensayo, dados en la biblioteca NETLIB [23]. Los problemas mostrados en esta biblioteca son problemas de gran escala (tienen una cantidad grande de variables y restricciones). Los resultados encontrados son similares a los mostrados en la biblioteca NETLIB, demostrando que el algoritmo FDIPA-LP es una alternativa muy eficiente para resolver problemas de gran escala.

Para explicar el proceso, primero vamos a presentar la formulación dual del problema

primal que vamos estudiar. El problema primal debe estar en la forma estandar:

$$\begin{aligned} \text{mín} \quad & c^T x, \\ \text{s.a} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{92}$$

donde: $A \in \mathbb{R}^{m \times n}$, $c, x \in \mathbb{R}^n$ y $b \in \mathbb{R}^m$. El rango de A es n .

La formulación del problema dual, fue presentado en un caso mas general en la sección C. Nuestro caso de aplicación es como el Problema (92). Aplicando la formulación dual tenemos que el problema dual queda como:

$$\begin{aligned} \text{máx} \quad & b^T \lambda, \\ \text{s.a} \quad & A^T \lambda \leq c, \\ & \lambda \text{ irrestricto.} \end{aligned}$$

Como se observa, el problema dual tiene restricciones de desigualdad del mismo tipo (\leq), esta forma es el tipo de problema que puede ser resuelto por el algoritmo FDIPA-LP, pero antes primero vamos hacer un previo tratamiento al problema primal. Primero se aplicará técnicas de Presolver para reducir las restricciones redundantes y el número de variables. Segundo, una técnica de escalamiento y hacer un reacondicionamiento del problema, porque al reducir las restricciones y el número de variables hay un peligro de mal acondicionamiento en las restricciones, que pueden ocasionar que el problema no obtenga una buena aproximación de la solución óptima.

B. Resolución de sistemas lineales internos en FDIPA-LP

El algoritmo FDIPA-LP es similar al FDIPA, inicialmente se deben resolver dos sistemas lineales. Para la solución de estos sistemas se aplican métodos iterativos, que se caracterizan

por aproximar la solución, generando una sucesión de puntos hasta que los puntos esten muy próximos. En muchos casos, estos métodos son mejores que otros métodos exactos, tienen un costo computacional mucho menor y obtienen una buena aproximación de la solución real.

La aproximación de la solución de los sistemas serán obtenidos aplicando el método de gradiente conjugado preconditionado split (GCPS). En la siguiente sección se desarrolla el algoritmo del método.

Método del Gradiente Conjugado Precondicionado Split (GCPS)

Los métodos directos tienen la desventaja de generar una cantidad grande de iteraciones, por eso tienen un costo computacional muy elevado. Sus operaciones generan muchos resultados inexactos, siendo mas inestable. la obtención de la solución exacta pueden ser incorrecta. Los métodos iterativos son métodos que en general se aproximan a la solución exacta y suelen ser mas practico. El método de gradiente conjugado es un método iterativo, no es muy robusto en sistemas de gran tamaño, pero con un preconditionamiento funcionan muy bien a comparación de otros métodos, como eliminación gaussiana y otros métodos iterativos. Cuando la matriz ha sido preconditionada se tienen buenos resultados en aproximadamente \sqrt{n} iteraciones [24].

El método de Gradiente conjugado fue planteado por Hestenes y Stiefel [25] para resolver sistemas de ecuaciones lineales del tipo.

$$Ax = b, \tag{93}$$

donde $A \in \mathbb{R}^{n \times n}$ es una matriz simétrica definida positiva, $x \in \mathbb{R}^n$ es el vector de incognitas del problema y $b \in \mathbb{R}^n$ el término independiente. El método GCPS es un algoritmo para la solución numérica de sistemas lineales, tal que las matrices son simétricas definidas y positivas. El algoritmo emplea el uso de una matriz preconditionada L , cuando esta empleando matrices

pobremente condicionadas, para garantizar la convergencia del método [17].

El algoritmo resuelve un sistema $Qx = B$, para ingresar la matriz Q esta se descompone aplicando la descomposición Cholesky Incompleta. La descomposición factoriza $Q = LL^T$ donde L es una matriz L triangular inferior, luego se transforma el sistema en un nuevo sistema de matrices,

$$L^{-1}QL^{-T}u = L^{-1}b, \quad x = L^{-T}u.$$

Algoritmo 3: Algoritmo del Método Gradiente Conjugado Precondicionado Split GCPS

Entrada: Dados $Q \in \mathbb{R}^{n \times n}$ simétrica definida positiva, $b \in \mathbb{R}^n$ y $x_0 \in \mathbb{R}^n$.

■ **Paso 1:** Calcular

$$\begin{aligned} r_0 &= b - Qx_0, \\ \hat{r}_0 &= L^{-1}r_0, \\ p_0 &= L^{-T}r_0. \end{aligned}$$

■ **Paso 2:** Determinación de la iteraciones.

Desde $k = 0, 1, \dots$, hasta converger, hacer

$$\begin{aligned} \alpha_k &= \frac{\hat{r}_k^T \hat{r}_k}{(Qp_k)^T p_k}, \\ x_{k+1} &= x_k + \alpha_k p_k, \\ \hat{r}_{k+1} &= \hat{r}_k + \alpha_k L^{-1}Qp_k, \\ \beta_k &= \frac{\hat{r}_{k+1}^T \hat{r}_{k+1}}{\hat{r}_k^T \hat{r}_k}, \\ p_{k+1} &= L^{-T} \hat{r}_{k+1} - \beta_k p_k. \end{aligned}$$

Fin.

El criterio de convergencia usado aquí es $\frac{\|\hat{r}_j\|}{\hat{r}_0} < \epsilon$. Además, se calcula la matriz L , usando la función ICHOL de Matlab.

Resolución de los sistema lineales internos

El sistema lineal interno originado por la aplicación del algoritmo FDIPA-LP puede ser resueltos por varios métodos numéricos, el cuál depende de la forma y característica que tienen los sistemas. Las matrices de los sistemas lineales para la aplicación del algoritmo FDIPA-LP son manipuladas antes de aplicar el método iterativo. Del sistema (61) despejamos el multiplicador de Lagrange λ_α , obteniendo $\lambda_\alpha = -G^{-1}(x)\Lambda Ad_\alpha$. Luego reemplazamos en la ecuación (60)

tenemos el sistema,

$$-A^T G^{-1}(x) \Lambda Ad_\alpha = -b, \quad (94)$$

$$\lambda_\alpha = -G^{-1}(x) \Lambda Ad_\alpha. \quad (95)$$

De forma análoga, en las ecuaciones (62) y (63) tenemos,

$$-A^T G^{-1}(x) \Lambda Ad_\beta = -A^T G^{-1}(x) \lambda, \quad (96)$$

$$\lambda_\beta = -G^{-1}(x) (\lambda + \Lambda Ad_\beta). \quad (97)$$

En los dos últimos sistemas se denota como: $Q = -A^T G^{-1}(x) \Lambda A$, se puede demostrar que la matriz es simétrica y definida positiva para todo $x \in \mathring{\Omega}_p$. Quedando el nuevo sistema de la forma:

$$Qd_\alpha = -b, \quad (98)$$

$$\lambda_\alpha = -G^{-1}(x) \Lambda Ad_\alpha, \quad (99)$$

y,

$$Qd_\beta = -A^T G^{-1}(x) \lambda, \quad (100)$$

$$\lambda_\beta = -G^{-1}(x) (\lambda + \Lambda Ad_\beta). \quad (101)$$

Por lo tanto, se puede aplicar el método GCPs para resolver los nuevos sistemas (98) y (100), y luego de forma inmediata se reemplaza las soluciones en los siguientes sistemas (99) y (101) respectivamente, para encontrar los parámetros λ_α y λ_β .

Métodos presolver en problema de PL

Los métodos presolver's son métodos aplicados en problemas de PL, porque puede ocurrir que algunas restricciones sean redundantes como: filas o columnas nulas, ecuaciones o inecuaciones repetidas. La aplicación de los métodos presolver's reducen el tamaño del sistema y el tiempo de ejecución, además pueden detectar si un problema es no acotado o infactible. Los métodos presolvers que se aplican están detallados en [16]. Apesar de reducir el número de variables y restricciones del problema de PL, la solución óptima verifica el mismo valor de la función objetivo original. El Tabla III muestra ejemplos de [14], como cambian las dimensiones para algunos problemas Test en [23], antes y después de aplicar el presolver. Los resultados demuestran que no todos los problemas se reducen sus dimensiones, como los problemas: Afiro, Degen2, Degen3, Israel, Qap8, y Qap12.

Al aplicar un presolver puede ocurrir que el nuevo sistema quede mal condicionado. Por este motivo se recomienda aplicar una técnica de reescalamiento, que disminuya el condicionamiento del sistema pero sin alterar la solución del problema. En [16] se pueden encontrar 11 tipos de técnicas, de las cuales en [14] se aplicaron 6 tipos, mostrados la Tabla IV. Para mas detalles de las técnicas de reescalamiento ver [16].

Determinación del punto interior inicial

El algoritmo FDIPA-LP es un algoritmo de punto interior, entonces se necesita de un punto interior inicial x^0 factible para iniciar el algoritmo. Este punto inicial se puede determina resolviendo el siguiente problema de PL.

$$\begin{aligned} \text{mín} \quad & z, \\ \text{s.a} \quad & Ax - c \leq 1^T z, \\ & x \geq 0, \end{aligned} \tag{102}$$

TABLA III
Aplicación de métodos presolver.

Nombre	Antes		Después	
	Nº variables	Nº restricciones	Nº variables	Nº restricciones
Adlittle	138	56	137	55
Afiro	51	27	51	27
Agg	615	488	478	390
Agg2	758	516	755	514
Agg3	758	516	755	514
Blend	114	74	111	71
Degen2	757	444	757	444
Degen3	2604	1503	2604	1503
Israel	316	174	316	174
Qap8	1632	912	1632	912
Qap12	8856	3192	8856	3192
Sc205	317	205	315	203
Sc50A	78	50	77	49
Sc50B	78	50	76	48
Scagr7	185	129	183	127
Scagr25	671	471	669	469
Scorpion	466	388	412	346
Sctap1	660	300	644	284
Sctap2	2500	1090	2443	1033
Sctap3	3340	1480	3268	1408
Share1b	253	117	248	112
Stocfor1	165	117	161	113

TABLA IV
Tipos de técnicas de reescalamiento.

Número	Técnica de reescalamiento
1	Media aritmética
2	Buchet para el caso $p = 1$
3	Buchet para el caso $p = 2$
4	Buchet para el caso $p = \infty$
5	Entropy
6	Equilibración

el punto factible es encontrado cuando $z < 0$.

C. Aplicación en problema de programación lineal de gran escala

La sección muestra la experimentación del algoritmo FDIPA-LP en problemas de programación lineal de gran escala, estos resultados ya fueron mostrados en [14]. Los problemas de gran escala tienen un número grande de variables y restricciones, cientos de variables y restricciones. Los ejemplos son elegidos de la librería NETLIB [23], que posee problemas Test de gran escala.

Los problemas Test de NETLIB, son problemas de programación lineal en formato estándar, mientras que el algoritmo FDIPA-LP resuelve problemas con restricciones de desigualdad menor o igual. Por este motivo, para la aplicación del algoritmo, se planteó formular el problema Test a su forma dual. Pues, se sabe el valor objetivo del problema primal es igual al valor objetivo de su problema dual [26], [16] y además la forma del problema dual queda con restricciones de desigualdad, que se necesita para aplicar el algoritmo. La experimentación ha sido realizada en MATLAB. Los cálculos fueron realizados en una computadora con microprocesador Intel(R) Core(TM) i7-7700HQ de frecuencia 2.80GHz×2.80GHz con una RAM de 8GB en el sistema operativo Windows 11 Home.

El procedimiento seguido para la aplicación del algoritmo FDIPA-LP a los problemas test de gran escala es mostrado en los siguientes pasos:

1. Revisar la factibilidad y redundancias del problema Test por métodos presolver.
2. Reacondicionar el problema aplicando una técnica de reescalamiento.
3. Resolver el problema (102) asociado al problema test, para determinar el punto interior inicial estrictamente factible aplicando el algoritmo FDIPA-LP.
4. Aplicar el algoritmo FDIPA-LP con el punto interior inicial hallado.

TABLA V

Comparación del método Dual-simplex con el algoritmo FDIPA-LP.

Problema	Valor objetivo (DUAL-SIMPLEX)	Valor objetivo (FDIPA-LP)	Error	Técnica de escalamiento
Adlittle	2.2549496316E+05	2.2549496316E+05	2.1269952413E-06	3
Afiro	-4.6475314286E+02	-4.6475314329E+02	4.3685201945E-07	6
Agg	-3.5991767287E+07	-3.5991767287E+07	9.0152025223E-07	2
Agg2	-2.0239252356E+07	-2.0239440369E+07	1.8801280320E+02	1
Agg3	1.0312115935E+07	1.0312115935E+07	0.0000000000E+00	1
Beaconfd	3.3592485807E+04	3.3592485807E+04	2.0008883439E-09	6
Blend	-3.0812149846E+01	-3.0812151576E+01	1.7299507000E-06	2
Degen2	-1.4351780000E+03	-1.4351780000E+03	0.0000000000E+00	1
Degen3	-9.8729400000E+02	-9.8729400000E+02	2.3703705665E-10	1
Ffff800	5.5567956482E+05	5.5567956482E+05	7.2701368481E-07	2
Israel	-8.9664482186E+05	-8.9664482186E+05	1.0477378964E-09	6
Qap8	2.0350000000E+02	2.0350000000E+02	9.9475983006E-13	4
Qap12	5.2289435056E+02	5.2289413517E+02	2.1538924796E-04	1
Sc105	-5.2202061212E+01	-5.2202061212E+01	8.6599320070E-07	1
Sc205	-5.2202061212E+01	-5.2202067078E+01	8.6599330018E-07	1
Sc50A	-6.4575077059E+01	-6.4575077059E+01	6.9633188104E-13	1
Sc50B	-7.0000000000E+01	-7.0000000000E+01	1.0013962992E-09	1
Scagr25	-1.4753433061E+07	-1.4753433061E+07	1.4994293451E-06	1
Scorpion	1.8781248227E+03	1.8781248227E+03	1.1160190070E-05	1
Sctap1	1.4122500000E+03	1.4122500000E+03	3.6490064304E-08	1
Sctap2	1.7248071429E+03	1.7248071429E+03	2.5859890229E-08	6
Sctap3	1.4240000000E+03	1.4240000000E+03	7.8900029621E-08	1
Share1b	-7.6589318579E+04	-7.6589318579E+04	3.9290171117E-10	1
Share2b	-4.1573224074E+02	-4.1573224074E+02	2.2299673219E-10	6
Ship04l	1.7933245380E+06	1.7933245380E+06	3.7599820644E-06	4
Ship04s	1.7987147004E+06	1.7987147004E+06	0.0000000000E+00	6
Stocfor1	-4.1131976219E+04	-4.1131976220E+04	1.2559758034E-07	6
Stocfor2	-3.9024408538E+04	-3.9024408538E+04	1.0186340660E-10	4
Bored3	1.3730803942E+03	1.3730803942E+03	8.4899056674E-09	6
Fit1d	-9.1463780924E+03	-9.1463780920E+03	4.2092869990E-07	6

Los resultados del algoritmo en 30 problemas Test son mostrados en el Tabla V. La tabla presenta en la primera columna el problema Test, la segunda columna el valor objetivo del problema por el método Dual-Simplex, la tercera columna el valor objetivo determinado por el

algoritmo FDIPA-LP, la cuarta columna el error de la comparación del método Dual-simplex y el algoritmo, y la última columna la técnica de escalamiento aplicada.

Se puede notar que los resultados del valor objetivo por el algoritmo FDIPA-LP son similares a los resultados del valor objetivo de cada problema por el método Dual-simplex. Demostrando que el algoritmo FDIPA-LP funciona de forma muy eficiente para encontrar el valor objetivo del problema Test de gran escala hasta orden de 10^{-10} , 10^{-13} y algunas situaciones de coincidencia como Agg3, Degen2 y Ship04s.

D. Aplicación en problemas de transporte

El problema de transporte fue presentado por FL Hitchcock (1941) [27] en su estudio "The distribution of a product from several sources to numerous localities". Un problema de transporte se define como la entrega de un producto desde un conjunto de centros de suministro, llamados **orígenes**, a otro conjunto de centros de recepción, llamados **destinos**, con el objetivo de minimizar los costos asociados con la distribución. Por ejemplo el costo total de transportar x_{ij} unidades a un costo de c_{ij} por unidad del origen i al destino j esta dado por:

$$C_T = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

El planteamiento de estos problemas resultan en modelos de programación lineal, cuya complejidad depende del número de orígenes y destinos. Además, de las condiciones de oferta y demanda del producto, que definen parte de las restricciones del problema. La formulación del problema de transporte supone que se tiene almacenados cantidades s_1, s_2, \dots, s_m unidades en los orígenes y la demanda de los destinos sean d_1, d_2, \dots, d_n unidades

El problema de transporte debe satisfacer tanto la oferta como la demanda, por esta

razón el problema queda formulado de la siguiente manera:

$$\begin{aligned}
 &\text{minimizar } C_T = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\
 &\text{sujeto a} \\
 &\quad \sum_{j=1}^n x_{ij} \leq s_i \quad i=1,2,\dots,m \quad (\text{Oferta}) \\
 &\quad \sum_{i=1}^m x_{ij} \leq d_j \quad j=1,2,\dots,n \quad (\text{Demanda}) \\
 &\quad x_{ij} \geq 0 \quad \text{para todo } i \text{ y } j.
 \end{aligned}$$

Donde los términos y variables son:

- C_T : costo total de la distribución.
- x_{ij} : cantidad de unidades distribuidas del origen i al destino j .
- c_{ij} : costo unitario de transportar una unidad desde el origen i al destino j .
- s_i : cantidad de unidades que suministra el origen i .
- d_j : cantidad de unidades que recibe el destino j .

De la formulación del problema de transporte, se interpreta que la Oferta esta restringido por la disponibilidad de los **orígenes** y la Demanda esta restringido por el requerimiento de los **destinos**. Entonces puede ocurrir algunos de los siguientes casos:

1. Lo demandado en total por los **destinos** sea mayor que la disponibilidad total que distribuye los **orígenes**, es decir: $\sum_{i=1}^m s_i < \sum_{j=1}^n d_j$. Resulta en un problema infactible, pues no se puede cubrir la demanda.
2. Lo demandado en total por los **destinos** es igual que la disponibilidad total que distribuye los **orígenes**, $\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$. Resulta en un problema factible y las restricciones del problema son igualdades.

3. Lo demandado en total por los **destinos** es menor que la disponibilidad total que distribuye los **orígenes**, $\sum_{i=1}^m s_i > \sum_{j=1}^n d_j$. Resulta en un problema con factibilidad, pues se puede cubrir la demanda. Además, se puede crear un destino ficticio con demanda $s_{m+1} = \sum_{i=1}^m s_i - \sum_{j=1}^n d_j$ y un costo unitario de envío $c_{m+1} = 0$, para no variar el costo total de transporte.

De estos casos, queda claro que el problema de transporte queda como un problema de programación lineal,

$$\begin{aligned} &\text{minimizar } C_T = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ &\text{sujeto a} \\ &\quad \sum_{j=1}^n x_{ij} \leq s_i \quad i=1,2,\dots,m \\ &\quad \sum_{i=1}^m x_{ij} \leq d_j \quad j=1,2,\dots,n \\ &\quad x_{ij} \geq 0 \quad \text{para todo } i \text{ y } j, \\ &\quad \sum_{i=1}^m s_i = \sum_{j=1}^n d_j = A \end{aligned}$$

Los problemas de transporte en sumo presentan las siguientes características:

1. Como mínimo tiene una solución factible.
2. Puede tener una cantidad finita de soluciones óptimas, pues las restricciones forman un conjunto factible acotado.
3. El número de variables es $m \times n$ y $m + n$ restricciones. Existen $m + n - 1$ restricciones lineales independientes y una redundante, de la igualdad $\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$. Esto implica, que una solución factible depende a lo mas de $m + n - 1$ variables estrictamente positivas.
4. Si lo suministrado y demandado (s_i y d_j) son cantidades enteros positivos, entonces las cantidades repartidas x_{ij} también deben ser valores enteros. Así, la solución básica factible (también la óptima), tienen valores enteros, por la estructura que presenta la matriz de

restricciones.

De esta última parte podemos decir que el problema de transporte tiene la forma de un problema de programación lineal con restricciones de igualdad y variables no negativas. Esta forma tiene su problema dual como un problema con restricciones de desigualdad ≤ 0 , lo cual se ajusta a los tipos de problema que pueden ser resueltos por el Algoritmo de FDIPA-LP. Para mostrar que algoritmo FDIPA-LP aproxima al valor óptimo del problema, se ha tomado dos ejemplos resueltos en [28], los valores óptimos obtenidos por el Algoritmo son iguales o aproximadamente iguales a los encontrados aplicando el software WINQSB 2.0, y el Método de la Esquina Noroeste y Modi. Sea el primer ejemplo.

Ejemplo IV. (Ejemplo 1 [28]) La empresa Mantecosa S.L, dedicada a la fabricación y distribución de mantecados, cuenta con dos fábricas una ubicada en Sevilla y la otra en Almería, y tres centros de distribución: Granada, Huelva y Córdoba. La fábrica de Sevilla tiene disponibles 8000 kilos de mantecados, mientras Almería tiene unas existencias de 6000 kilos de mantecados. Se estima que la demanda de cada centro de distribución es de 3000 kilos para Granada, 4000 kilos para Huelva y 7000 kilos para Córdoba. Por otro lado, los costes de transporte unitarios asociados desde cada fábrica hasta cada centro de distribución se detallan a continuación:

TABLA VI
Coste unitario de transporte del Ejemplo IV, [28].

	Granada	Huelva	Cordova
Sevilla	15	11	8
Almería	7	14	12

Para la formulación del problema usamos la variable x_{ij} para indicar la cantidad distribuida del origen i al destino j , donde $i \in \{1, 2\}$ y $j \in \{1, 2, 3\}$.

De la Tabla VI se tiene los costos unitarios c_{ij} por distribuir del origen i al destino j ,

quedando el costo total por transporte, como la función:

$$C_T = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = 15x_{11} + 11x_{12} + 8x_{13} + 7x_{21} + 14x_{22} + 12x_{23}$$

Se puede comprobar que la oferta total (s_i) y la demanda total (d_j) son iguales,

$$\sum_{i=1}^m s_i = 8000 + 6000 = 14000, \quad \sum_{j=1}^n d_j = 3000 + 4000 + 7000 = 14000.$$

$$\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$$

La distribución de una fábrica a un centro de distribución debe satisfacer la condición de la oferta:

$$\sum_{j=1}^3 x_{1j} = x_{11} + x_{12} + x_{13} = 8000, \quad \sum_{j=1}^3 x_{2j} = x_{21} + x_{22} + x_{23} = 6000$$

De igual forma, las condiciones de la demanda de cada centro de distribución se debe satisfacer:

$$\sum_{i=1}^2 x_{i1} = x_{11} + x_{21} = 3000,$$

$$\sum_{i=1}^2 x_{i2} = x_{12} + x_{22} = 4000,$$

$$\sum_{i=1}^2 x_{i3} = x_{13} + x_{23} = 7000.$$

Así, el problema queda formulado como en [28],

Minimizar

$$\begin{aligned}
 C_T &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = 15x_{11} + 11x_{12} + 8x_{13} + 7x_{21} + 14x_{22} + 12x_{23} \\
 \text{sujeto a: } &\sum_{j=1}^3 x_{1j} = x_{11} + x_{12} + x_{13} = 8000 \\
 &\sum_{j=1}^3 x_{2j} = x_{21} + x_{22} + x_{23} = 6000 \\
 &\sum_{i=1}^2 x_{i1} = x_{11} + x_{21} = 3000 \\
 &\sum_{i=1}^2 x_{i2} = x_{12} + x_{22} = 4000 \\
 &\sum_{i=1}^2 x_{i3} = x_{13} + x_{23} = 7000 \\
 &x_{ij} \geq 0, \forall i \in \{1, 2\}, j \in \{1, 2, 3\}.
 \end{aligned} \tag{103}$$

Se puede observar que el problema de transporte (103) queda transformado en un problema de programación lineal en forma estandar, la solución óptima del problema es demostrada en [28], $x_{11} = 0$, $x_{12} = 1000$, $x_{13} = 7000$, $x_{21} = 3000$, $x_{22} = 3000$, $x_{23} = 3000$ y el valor óptimo es: $Z = 130000$.

Aplicación del algoritmo FDIPA-LP

El problema (103) como presenta la forma estándar entonces su dual correspondiente tiene la forma de un problema de programación lineal con restricciones ≤ 0 y variables irrestrictas. Para formular el dual vamos a denotar las variables $x_{ij} = y_k$, $\forall i \in \{1, 2\}$, $j \in \{1, 2, 3\}$, $k \in \{1, \dots, 6\}$. Reformulando el problema (103) con las nuevas variable, tenemos la siguiente forma:

$$\begin{aligned}
& \text{mín} \quad Z = 15y_1 + 11y_2 + 8y_3 + 7y_4 + 14y_5 + 12y_6 \\
& \text{s.a.} \quad y_1 + y_2 + y_3 = 8000, \\
& \quad y_4 + y_5 + y_6 = 6000, \\
& \quad y_1 + y_4 = 3000, \\
& \quad y_2 + y_5 = 4000, \\
& \quad y_3 + y_6 = 7000, \\
& \quad y_k \geq 0 \quad k \in \{1, \dots, 6\}.
\end{aligned} \tag{104}$$

Entonces las componentes del problema primal son:

$$c = \begin{pmatrix} 15 \\ 11 \\ 8 \\ 7 \\ 14 \\ 12 \end{pmatrix}; \quad b = \begin{pmatrix} 8000 \\ 6000 \\ 3000 \\ 4000 \\ 7000 \end{pmatrix}; \quad A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Luego el problema dual tiene la forma:

$$\text{máx} \quad b^T \lambda \quad \text{s.a.} \quad A^T \lambda \leq c; \lambda \text{ irrestricto}$$

$$\begin{aligned}
& \text{máx} \quad 8000\lambda_1 + 6000\lambda_2 + 3000\lambda_3 + 4000\lambda_4 + 7000\lambda_5 \\
& \text{s.a.} \quad \lambda_1 + \lambda_3 \leq 15, \\
& \quad \lambda_1 + \lambda_4 \leq 11, \\
& \quad \lambda_1 + \lambda_5 \leq 8, \\
& \quad \lambda_2 + \lambda_3 \leq 7, \\
& \quad \lambda_2 + \lambda_4 \leq 14, \\
& \quad \lambda_2 + \lambda_5 \leq 12, \\
& \quad \lambda_k \text{ irrestricto.}
\end{aligned} \tag{105}$$

Antes de aplicar el algoritmo FDIPA-LP cambiamos el problema (105) a un problema de minimización.

$$\begin{aligned}
& \text{mín} \quad -(8000\lambda_1 + 6000\lambda_2 + 3000\lambda_3 + 4000\lambda_4 + 7000\lambda_5) \\
& \text{s.a.} \quad \lambda_1 + \lambda_3 \leq 15, \\
& \quad \lambda_1 + \lambda_4 \leq 11, \\
& \quad \lambda_1 + \lambda_5 \leq 8, \\
& \quad \lambda_2 + \lambda_3 \leq 7, \\
& \quad \lambda_2 + \lambda_4 \leq 14, \\
& \quad \lambda_2 + \lambda_5 \leq 12, \\
& \quad \lambda_k \text{ irrestricto.}
\end{aligned} \tag{106}$$

Así, el problema dual (106) asociado al problema de transporte (103) queda listo para aplicar el algoritmo FDIPA-LP, los datos de entrada para el algoritmo son:

- La matriz de restricciones A^T .
- El vector de costos o precios de la función objetivo $-b$.
- El vector de recursos o disponibilidad c .

Los parámetros usados en el algoritmo son:

$$\xi = 0,7; \varphi = 1,5; \mu = 1,2; \gamma = 0,9995; xs = 2; Tol = 0,000001.$$

Los resultados del algoritmo son mostrados en la Tabla VII El valor optimo encontrado por el algoritmo es $-12999,999997$, entonces el problema dual (105) tiene como valor optimo $12999,999997$. Así, podemos notar que la aproximación es casi perfecta a la función objetivo del problema de transporte (103).

TABLA VII
Resultados del algoritmo FDIPA-LP del Ejemplo IV.

Iteración	Paso	Valor
1	0.00024	2182752010.373181
2	0.00053	1280542046.557217
3	0.00058	762760018.049394
4	0.00092	304063347.812918
5	0.00070	156011479.064167
6	0.00096	57473996.000484
7	0.00062	32087393.540134
8	0.00105	9651245.021503
⋮	⋮	⋮
48	0.00036	-129999.999884
49	0.00034	-129999.999904
50	0.00074	-129999.999939
51	0.00076	-129999.999962
52	0.00124	-129999.999982
53	0.00029	-129999.999987
54	0.00203	-129999.999997
55	0.00203	-129999.999997

El resultado de la Tabla VII, puede ser comparado con los resultados obtenidos aplicando el método de la Esquina Noroeste y el método Modi. El desarrollo de estos métodos no es el objetivo del trabajo puede revisar [28], para mas detalles. La Figura VII, muestra la solución optima del Ejemplo IV resuelto con los métodos mencionados, el valor de la minimización de la

función objetivo es: $Z = 1000 \times 11 + 7000 \times 8 + 3000 \times 7 + 3000 \times 14 = 130000$.

Origen	Destino						Oferta
	GRANADA		HUELVA		CÓRDOBA		
SEVILLA	15		11		8		8000
	0		1000		7000		
ALMERÍA	7		14		12		6000
	3000		3000		0		
Demanda	3000		4000		7000		

Figura VII: Solución óptima del Ejemplo IV, Fuente [28].

La comprobación de este resultado, también se realizó por el software WINQSB ver la Figura VIII. Se observa que el valor de la función objetivo es $Z = 130000$.

Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)
X1	0	15,0000	0	11,0000	at bound	4,0000	M
X2	1.000,0000	11,0000	11.000,0000	0	basic	10,0000	14,0000
X3	7.000,0000	8,0000	56.000,0000	0	basic	-M	9,0000
X4	3.000,0000	7,0000	21.000,0000	0	basic	-M	18,0000
X5	3.000,0000	14,0000	42.000,0000	0	basic	11,0000	15,0000
X6	0	12,0000	0	1,0000	at bound	11,0000	M
Objective	Function	(Min.) =	130.000,0000				
Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS
C1	8.000,0000	=	8.000,0000	0	-3,0000	8.000,0000	11.000,0000
C2	6.000,0000	=	6.000,0000	0	0	6.000,0000	M
C3	3.000,0000	=	3.000,0000	0	7,0000	0	3.000,0000
C4	4.000,0000	=	4.000,0000	0	14,0000	1.000,0000	4.000,0000
C5	7.000,0000	=	7.000,0000	0	11,0000	4.000,0000	7.000,0000

Figura VIII: Solución óptima del Ejemplo IV, por WINQSB. Fuente [28].

De esta forma se demuestra que la solución hallado por los métodos tradicionales tienen un error muy bajo al encontrado por el algoritmo FDIPA-LP,

$$Error = |130000 - 129999,999997| = 3 \times 10^{-6}.$$

El siguiente ejemplo es un caso aplicado en la industria agrícola presentado en [28].

Ejemplo V. (Ejemplo 2 [28]) La sociedad agrícola Antonio Martínez Legal e Hijos S.L posee varias fincas de naranjas en la zona de Lora del Rio y sus cercanías. En la actualidad se compone de cuatro fincas: Los Silos, Puerto Chico, Alcolea y El Membrillo. Distribuye a siete fábricas sus productos: Guadex, Toñifruit, Serviagro, Matias Colom, Expalmex, Particular y Extrafru. Elaboraron una tabla de costos unitarios de transporte de la producción según los datos recogidos en los años 2015 y 2016, ver el Tabla VIII .

TABLA VIII
Coste unitario de transporte del Ejemplo V, [28].

Destino	Origen				Demanda
	LOS SILOS	PUERTO CHICO	ALCOLEA	MEMBRILLO	
GUARDEX	79.40	70.37	124.85	102.15	32
TOÑIFRUIT	1225.50	1240.98	1194.54	1161.00	12
MATIAS COLOM	835.50	851.40	923.64	903.00	8
EXPALMEX	27.24	36.32	63.56	47.67	1
SERVIAGRO	497.94	516.00	570.18	541.80	39
PARTICULAR	11.35	22.70	34.05	45.40	5
EXAFRUT	980.40	1001.40	1070.70	1039.70	2
Oferta	6	14	13	66	

El problema de transporte asociado a este caso queda formulado como en [28],

Minimizar:

$$C_T = \sum_{i=1}^4 \sum_{j=1}^7 c_{ij}x_{ij} = 79,40x_{11} + 1225,50x_{12} + 835,50x_{13} + 27,24x_{14} + 497,94x_{15} + 11,35x_{16} + 980,40x_{17} + 70,37x_{21} + 1240,98x_{22} + 851,40x_{23} + 36,32x_{24} + 516,00x_{25} + 22,70x_{26} + 1001,40x_{27} + 124,85x_{31} + 1194,54x_{32} + 923,64x_{33} + 63,56x_{34} + 570,18x_{35} + 34,05x_{36} + 1070,70x_{37} + 102,15x_{41} + 1161,00x_{42} + 903,00x_{43} + 47,67x_{44} + 541,80x_{45} + 45,40x_{46} + 1039,74x_{47}$$

sujeto a

$$\sum_{j=1}^7 x_{1j} = x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} = 6$$

$$\sum_{j=1}^7 x_{2j} = x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} = 14$$

$$\sum_{j=1}^7 x_{3j} = x_{31} + x_{32} + x_{33} + x_{34} + x_{35} + x_{36} + x_{37} = 13$$

$$\sum_{j=1}^7 x_{4j} = x_{41} + x_{42} + x_{43} + x_{44} + x_{45} + x_{46} + x_{47} = 66$$

$$\sum_{i=1}^4 x_{i1} = x_{11} + x_{21} + x_{31} + x_{41} = 32$$

$$\sum_{i=1}^4 x_{i2} = x_{12} + x_{22} + x_{32} + x_{42} = 12$$

$$\sum_{i=1}^4 x_{i3} = x_{13} + x_{23} + x_{33} + x_{43} = 8$$

$$\sum_{i=1}^4 x_{i4} = x_{14} + x_{24} + x_{34} + x_{44} = 1$$

$$\sum_{i=1}^4 x_{i5} = x_{15} + x_{25} + x_{35} + x_{45} = 39$$

$$\sum_{i=1}^4 x_{i6} = x_{16} + x_{26} + x_{36} + x_{46} = 5$$

$$\sum_{i=1}^4 x_{i7} = x_{17} + x_{27} + x_{37} + x_{47} = 2$$

$$x_{ij} \geq 0 \quad \forall i \text{ y } j.$$

(107)

La solución de este problema fue hallado usando el programa **WINQSB**. El software

resuelve diferentes problemas de investigación de operaciones, veamos la solución encontrada:

- La finca Los Silos ha de transportar a Matías Colom 4 camiones y a Exafrut 2 camiones.
- Puerto chico ha de distribuir a Guadex 10 camiones y a Matías Colom 4 camiones.
- Alcolea deberá llevar a Guadex 7 camiones, a Expalmex 1 camión y a Particular 5 camiones.
- El Membrillo transportará a Guadex 15 camiones, a Toñifruit 12 camiones y a Serviagro 39 camiones.

El costo total optimo de transporte por la empresa según [28] es de 47114,31 euros ver la Figura

IX. La aplicación del algoritmo FDIPA-LP es mostrado a continuación.

From	To	Shipment	Unit Cost	Total Cost	Reduced Cost
Los Silos	Matias Colom	4	835,50	3342	0
Los Silos	Exafrut	2	980,40	1.960,80	0
Puerto Chico	Guadex	10	70,37	703,70	0
Puerto Chico	Matias Colom	4	851,40	3.405,60	0
Alcolea	Guadex	7	124,85	873,95	0
Alcolea	Expalmex	1	63,56	63,56	0
Alcolea	Particular	5	34,05	170,25	0
Membrillo	Guadex	15	102,15	1.532,25	0
Membrillo	Toñifruit	12	1161	13932	0,00
Membrillo	Seviagro	39	541,80	21.130,20	0
Total	Objective	Function	Value =	47.114,31	

Figura IX: Solución optima del Ejemplo V, por WINQSB. Fuente [28].

Aplicación del algoritmo FDIPA-LP

De forma analoga al Ejemplo VI, antes de aplicar el algoritmo FDIPA-LP, vamos primero escribir el problema (107) en su forma primal. Las variables x_{ij} son denotadas por $y_k = x_{ij}$, $\forall i \in \{1, \dots, 4\}$, $j \in \{1, \dots, 7\}$, $k \in \{1, \dots, 28\}$, $(k = 7(i - 1) + j)$.

El problema (107) tiene la forma primal:

$$\text{mín } Z = c^T y$$

$$\text{s.a. } Ay = b$$

$$y_k \geq 0, \forall k \in 1, \dots, 28.$$

Donde:

$$c = (79,40, 1225,50, 835,50, 27,24, 497,94, 11,35, 980,40, 70,37, 1240,98, 851,40, 36,32, \dots \\ 516,00, 22,70, 1001,40, 124,85, 1194,54, 923,64, 63,56, 570,18, 34,05, 1070,70, 102,15, 1161,00, \dots \\ 903,00, 47,67, 541,80, 45,40, 1039,74)^T,$$

$$b = (6, 14, 13, 66, 32, 12, 8, 1, 39, 5, 2)^T,$$

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Luego el problema dual tiene la forma:

$$\text{máx } b^T \lambda \quad \text{s.a. } A^T \lambda \leq c; \lambda \text{ irrestricto.}$$

TABLA IX
Resultados del algoritmo FDIPA-LP del Ejemplo V

Iteración	Paso	Valor
1	0.01770	10083113.154133
2	0.08296	7514577.838465
3	0.06702	6488121.393880
4	0.08465	5759923.738179
5	0.10670	5193371.408166
6	0.29066	3949083.717781
7	0.10869	3596244.224665
⋮	⋮	⋮
280	0.05097	-47115.509968
281	0.15224	-47115.509973
282	0.14327	-47115.509977
283	0.11857	-47115.509979
284	0.00005	-47115.509979
285	0.00002	-47115.509979
286	0.03562	-47115.509980
287	-0.00000	-47115.509980

El problema de programación lineal asociado es de la forma:

$$\begin{aligned}
 &\text{mín} \quad -b^T y, \\
 &\text{s.a.} \quad A^T y \leq c, \\
 &\quad y_k \text{ irrestricto.}
 \end{aligned}$$

La aplicación del algoritmo FDIPA-LP en este ejemplo se realiza usando los mismos parámetros del Ejemplo VIII. Se observa los resultados en el Tabla IX

El valor encontrado de la función objetivo es $-47115,509980$, entonces el valor positivo de la función objetivo del problema dual (107) es $47115,509980$. De esta manera la aproximación del valor optimo del problema de transporte (105) es muy buena comparado con el valor mostrado en [28] encontrado por el software WINQSB $47114,31$ euros. El error se puede observar que es relativamente pequeño $Error = |47115,509980 - 47114,31| = 1,19998$.

VI Conclusiones y discusiones

A. Conclusiones

El algoritmo de punto interior FDIPA ha sido aplicado de forma satisfactoria para la resolución de problemas de programación lineal de gran escala. La adaptación a problemas de programación lineal se ha realizado siguiendo los mismos pasos del algoritmo FDIPA, cambiando el PASO 2 de la búsqueda lineal, por una relación de filas de la matriz de restricciones con la dirección de descenso. El nuevo algoritmo se ha nombrado FDIPA-LP, la aplicación no es compleja, la convergencia puede ser lenta pero eficiente en aproximarse a una solución factible.

La aplicación del Teorema de Dualidad en problemas de programación lineal garantiza que la formulación dual del problema de programación lineal de gran escala obtenga el mismo valor óptimo del problema primal. De esta manera la formulación dual es utilizada para la solución del problema. Además, la forma dual en restricciones de desigualdad es necesaria para la aplicación del algoritmo FDIPA.

La adaptación del algoritmo de punto interior FDIPA a problemas de programación lineal de gran escala es logrado, debido que el problema lineal es transformado a su forma dual con restricciones de desigualdades menor o igual a cero. Los resultados son mostrados en 30 problemas test de la librería Netlib obteniendo aproximaciones hasta el orden de 10^{-10} y 10^{-13} .

La implementación del algoritmo de punto interior FDIPA-LP resuelve problemas de programación lineal de gran escala, se ha verificado su implementación en problemas test mostrados en la tabla V. Antes de la aplicación del algoritmo se realizan métodos de presolver y técnicas de reacondicionamiento a la matriz de restricciones. La implementación del algoritmo resuelve dos sistemas de ecuaciones lineales del PASO 1 del algoritmo, haciendo uso del método de

Descomposición de Choleski Incompleto.

La aplicación también se realizó en problemas de transporte por la forma que tiene el problema primal (forma estándar). La aplicación se realizó en dos casos: El ejemplo IV comprueba que la solución es eficiente, comparado con los valores óptimos hallados por el método de la esquina Noroeste y el software WINQSB. El segundo ejemplo V más complejo también obtuvo una buena aproximación comparada con el resultado hallado por el software WINQSB.

B. Discusiones y recomendaciones

El algoritmo de punto interior FDIPA-LP muestra resultados eficientes, pero no se ha centrado en la obtención del valor de la variable, quedando esto como una futura investigación o aplicación.

El estudio del algoritmo FDIPA-LP, nos llevan a cuestionar si es posible adaptar la aplicación del algoritmo a otros problemas de contexto real como se hizo al problema de Transporte.

El algoritmo FDIPA-LP se adaptó en problema de forma estándar, para obtener el valor objetivo, quedando la interrogante ¿El algoritmo FDIPA-LP puede resolver problemas de programación lineal en forma mixta?.

La implementación del algoritmo FDIPA-LP se ha realizado con los mismos parámetros del algoritmo FDIPA estos valores deben ser estudiados para determinar si existen otros valores que optimicen el rendimiento del algoritmo.

VII Referencias bibliográficas

- [1] J. M. Sallan, O. Lordan y V. Fernandez, *Modeling and solving linear programming with R*. OmniaScience, 2015.
- [2] Y. B. Colina, "Aplicaciones de programación lineal, entera y mixta," *Ingeniería Industrial. Actualidad y Nuevas Tendencias*, vol. 2, n.º 7, págs. 85-104, 2011.
- [3] G. B. Dantzig, "Maximization of a linear function of variables subject to linear inequalities," *Activity analysis of production and allocation*, vol. 13, págs. 339-347, 1951.
- [4] N. Karmarkar, "A New Polynomial-Time Algorithm for Linear Programming-II," *Combinatorica*, vol. 4, págs. 373-395, dic. de 1984. DOI: 10.1007/BF02579150.
- [5] S. J. Wright, *Primal-dual interior-point methods*. SIAM, 1997.
- [6] C. C. Gonzaga, "Conical projection algorithms for linear programming," *Mathematical Programming*, vol. 43, n.º 1, págs. 151-173, 1989.
- [7] E. R. Barnes, "A variation on Karmarkars algorithm for solving linear programming problems," *Mathematical programming*, vol. 36, págs. 174-182, 1986.
- [8] C. C. Gonzaga, "Path-following methods for linear programming," *SIAM review*, vol. 34, n.º 2, págs. 167-224, 1992.
- [9] J. Herskovits, "Feasible direction interior-point technique for nonlinear optimization," *Journal of optimization theory and applications*, vol. 99, n.º 1, págs. 121-146, 1998.
- [10] J. Herskovits, G. Dias, G. Santos y M. Soares, "Shape structural optimization with an interior point nonlinear programming algorithm," *Structural and multidisciplinary Optimization*, vol. 20, n.º 2, págs. 107-115, 2000.
- [11] J. Herskovits y S. R. Mazorche, "A feasible directions algorithm for nonlinear complementarity problems and applications in mechanics," *Structural and*

Multidisciplinary Optimization, vol. 37, n.º 5, págs. 435-446, 2009.

- [12] J. Herskovits, P. Mappa, E. Goulart y C. M. Soares, "Mathematical programming models and algorithms for engineering design optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, n.º 30-33, págs. 3244-3268, 2005.
- [13] A. L. Tits y J. L. Zhou, "A Simple, Quadratically Convergent Interior Point Algorithm for Linear Programming and Convex Quadratic Programming," en *Large Scale Optimization: State of the Art*, W. W. Hager, D. W. Hearn y P. M. Pardalos, eds. Boston, MA: Springer US, 1994, págs. 411-427, ISBN: 978-1-4613-3632-7. DOI: 10.1007/978-1-4613-3632-7_20. dirección: https://doi.org/10.1007/978-1-4613-3632-7_20.
- [14] A. Victorio Celis y J. Herskovits, "Linear programming with a feasible direction interior point technique for smooth optimization," *RAIRO - Operations Research*, vol. 56, sep. de 2022. DOI: 10.1051/ro/2022165.
- [15] V. Klee y G. J. Minty, "How good is the simplex algorithm," *Inequalities*, vol. 3, n.º 3, págs. 159-175, 1972.
- [16] N. Ploshkas, N. Samaras et al., *Linear programming using MATLAB*. Springer, 2017, vol. 127.
- [17] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.
- [18] J. Herskovits, "A two-stage feasible directions algorithm for nonlinear constrained optimization," *Mathematical Programming*, vol. 36, n.º 1, págs. 19-38, 1986.
- [19] A. Iov y M. Solodov, *Otimização, volume 1: condições de otimalidade, elementos de análise convexa e de dualidade*. IMPA, 2005.
- [20] A. Izmailov y M. Solodov, *Otimização, volume 2: métodos computacionais*. IMPA, 2007.

- [21] A. A. Ribeiro y E. W. Karas, "Otimização Continua: Aspectos teóricos e computacionais," *São Paulo: Cengage Learning*, pág. 9, 2013.
- [22] E. R. Panier, A. L. Tits y J. N. Herskovits, "A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization," *SIAM Journal on Control and Optimization*, vol. 26, n.º 4, págs. 788-811, 1988.
- [23] D. M. Gay, "Electronic mail distribution of linear programming test problems," *Mathematical Programming Society COAL Newsletter*, vol. 13, págs. 10-12, 1985.
- [24] R. Burden, J. Faires y A. Burden, *Análisis Numérico* (10ma ed.) USA. 2017.
- [25] M. R. Hestenes, E. Stiefel et al., "Methods of conjugate gradients for solving linear systems," *Journal of research of the National Bureau of Standards*, vol. 49, n.º 6, págs. 409-436, 1952.
- [26] M. Bazaraa, J. Jarvis y H. Sherali, *Linear Programming and Network Flows*. Wiley, 2009, ISBN: 9780470462720. dirección: https://books.google.com.pe/books?id=2DKKHvV_xVwC.
- [27] F. L. Hitchcock, "The distribution of a product from several sources to numerous localities," *Journal of mathematics and physics*, vol. 20, n.º 1-4, págs. 224-230, 1941.
- [28] I. M. Ramos Pérez, "Aplicaciones de la Programación Lineal: El Problema de Transporte," 2016.