T
510
P45

UNIVERSISDAD NACIONAL DEL CALLAO

FACULTAD DE CIENCIAS NATURALES Y MATEMATICA

ESCUELA PROFESIONAL DE MATEMATICA

ALGORITMOS DE TRAYECTORIA CENTRAL PARA EL PROBLEMA DE PROGRAMACIÓN LINEAL CONTINUA

TESIS PARA OPTAR EL TITULO PROFESIONAL DE

LICENCIADO EN MATEMÁTICA

KARLA ESTHER PÉREZ COLÁN

CALLAO – PERÚ

AGOSTO – 2008

Id. Publ. 15865
Id. Ejemplar: 39105

Algoritmos de Trayectoria Central para

el Problema de Programación Lineal

Continua

Karla Esther Pérez Colán

Tesis presentada a consideración del Cuerpo Docente de la Facultad de Ciencias Naturales y Matemática de la Universidad Nacional del Callao, como parte de los requisitos para obtener el Título Profesional de Licenciado en Matemática.

Aprobada por:

Mg. Roel Vidal Guzmán (Presidente)

Lic. Gabriel Rodríguez Varillas (Miembro)

Lic. Marco Antonio Rubio Gallarday (Miembro)

Mg. Carlos Enrique Vargas Trujillo (Asesor)

CALLAO - PERÚ

Febrero - 2007

FICHA CATALOGRÁFICA

PÉREZ COLAN KARLA ESTHER

Algoritmos de Trayectoria Central para el Problema de Programación Lineal Continua. Callao (2006).

xi, 97, 29,7cm (UNAC, Licenciado en Matemática 2006).

Tesis, Universidad Nacional del Callao, Facultad de Ciencias Naturales y Matemática 1. Matemática

1. UNAC/FCNM II. Título (Serie).

Este trabajo está dedicado

para mis tres grandes amores:

Wilfredo, Ximena y Matias por

haber sacrificado valiosas horas

de tiempo juntos.

Agradecimientos

A Jehová DIOS pues sin su ayuda nada de esto se hubiera logrado.

A Carlos Vargas, mi asesor, por toda su paciencia, tiempo y dedicación mostrada hacia este trabajo de tésis.

A Carlos y Esther (mis padres) que nunca dejaron de alentarme y me dieron su apoyo en todo sentido.

A Wilfredo Bardales (mi esposo) por su amor, comprensión y apoyo incondicional.

A Magno Bardales (mi suegro) quien siempre estuvo dispuesto a ser un apoyo en todo momento para llegar a la culminación de este trabajo.

A todos aquellos profesores y amigos que en general estuvieron siempre interesados en el avance y culminación de éste trabajo como el Prof. Angel Coca Balta, Prof. Absalón Castillo, Prof. Ezequiel Fajardo, Prof Wilfredo Mendoza, quienes siempre tuvieron unas palabras de aliento y motivación durante todos los años que nos conocemos y más aun en el tiempo dedicado a ésta tésis.

RESÚMEN

Algoritmos de trayectoria central para el problema de programación lineal continua.

Karla Esther Pérez colán

Febrero - 2007

Asesor: Mg. Carlos Enrique Vargas Trujillo.

Título Obtenido: Licenciado en Matemática.

En el presente trabajo de tésis estudiaremos una familia de algoritmos de punto interior, conocido como algoritmos de trayectoria central, para el problema de programación lineal el cual en la llamada forma estándar, consiste en:

$$\begin{aligned} &\text{minimizar} \quad c^T x \\ &\text{sujeto a} \\ &\qquad Ax = b, \\ &\qquad x \geqslant 0, \end{aligned} \tag{1}$$

donde $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ y $A \in \mathbb{R}^{m \times n}$.

Empleando una metodología ampliamente utilizada para los problemas de programación no lineal llamada métodos de barrera y penalización pero especializada al caso lineal, es asi que generamos una familia de algoritmos de punto interior para el problema de programación lineal.

Presentamos tres algoritmos basados en esta metodología: el algoritmo de trayectoria central de pasos cortos, el algoritmo de trayectoria central predictor - corrector, el algoritmo de trayectoria central de pasos largos y un algoritmo alternativo llamado algoritmo de trayectoria central no factible.

Además presentamos pruebas de convergencia para cada uno de ellos. Una experiencia numérica básica es también presentada.

SUMMARY

Algorithms of central trajectory

stop the problem of linear programming

continuous.

Karla Esther Perez Colán

February - 2007

Adviser: Mg. Carlos Enrique Vargas Trujillo.

Obtained title: Lawyer in Mathematical.

In the present thesis work we will study a family of algorithms of inner point, known like algorithms central trajectory, for the problem of linear programming which in the call forms standard, it consists of:

$$\begin{aligned} &\text{minimizar} \quad c^T x \\ &\text{sujeto a} \\ &\qquad Ax = b, \\ &\qquad x \geqslant 0, \end{aligned} \tag{1}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ y $A \in \mathbb{R}^{m \times n}$.

Using a methodology widely used for the programming problems nonlinear call methods of barrier and penalty but specialized to the linear case, it is so we generated a family of algorithms of inner point for the problem of linear programming.

We presented three algorithms based on this methodology: the algorithm of central trajectory of half-steps, the deflection algorithm of central trajectory, the algorithm of central trajectory of lengthened paces and an alternative algorithm called algorithm of nonfeasible central trajectory.

In addition we presented tests of convergence for each one of them. Basic a numerical experience also is presented.

INDICE

Notaciones

En éste trabajo usaremos letras minúsculas de nuestro alfabeto para representar vectores y letras mayúsculas para representar matrices; además de las siguientes notaciones:

1.- $\mathbb{R}^n$: conjunto de vectores columna

$$x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}$$

2.- $\mathbb{R}^{m \times n}$: conjunto de matrices $m \times n$.

3.- A: matriz de orden $m \times n$.

4.- $\mathcal{F}$: conjunto de soluciones factibles del problema primal.

5.- $\mathcal{F}^0$: conjunto de puntos interiores factibles del problema primal.

6.- S: conjunto de soluciones factibles del problema dual.

7.- S^0: conjunto de puntos interiores factibles del problema dual.

8.- $f(x, \mu)$: función barrera

9.- μ : parámetro positivo; medida de dualidad definido como $\mu := \dfrac{x^T s}{n}$

10.- $\mathcal{L}$: región primal - dual factible.

11.- $\mathcal{L}^0$: conjunto interior de $\mathcal{L}$.

12.- $(\Delta x, \Delta u, \Delta s)$: dirección de Newton.

13.- $(\Delta x_\mu, \Delta u_\mu, \Delta s_\mu)$: un punto en la trayectoria central.

14.- Z : conjunto de las concretizaciones de un problema lineal.

15.- M : conjunto solución del problema lineal.

16.- $\lceil \alpha \rceil$: el menor número entero mayor o igual que α.

17.- L : es el tamaño de un problema lineal.

18.- $T(L)$: representa el tiempo de ejecución de un algoritmo.

19.- $\|.\|$, $\|.\|_2$: norma euclidiana, para $u \in \mathbb{R}^n$,

$\|u\| = (\sum_{i=1}^{n} u_i^2)^{1/2}$.

Para una matriz M, $\|M\| = \max_{\|u\|=1} \|Mu\|$.

20.- $\|.\|_\infty$: norma ∞. Para $u \in \mathbb{R}^n$, $\|u\|_\infty = \max_{i=1,...,n} |u_i|$.

(Note que $\|u\|_\infty \leq \|u\|_2 \leq \|u\|_1$, para cada vector μ).

21.- $e : (1, 1, ..., 1)^T$.

22.- $\ln(.)$: logaritmo natural $\log_e$.

23.- $Z(L)$: subconjunto de Z formado por las concretizaciones de Z de tamaño L.

24.- (KM) : problema de Klee y Minty.

25.- $(KM)_2$: problema de Klee y Minty para $n = 2$.

26.- $\mathcal{R}_\mu$: conjunto de nivel factible.

27.- $\mathcal{N}(\alpha)$: vecindad de la trayectoria central.

28.- ρ_b : residuo primal.

29.- ρ_c : residuo dual.

30.- x : vector - $\mathbb{R}^n$ de variables primales.

31.- u : vector - $\mathbb{R}^m$ de variables duales, que es el multiplicador de lagrange para las restricciones de igualdad $Ax = b$.

32.- s : vector - $\mathbb{R}^n$ de holguras duales, que es, el multiplicador de lagrange para las restricciones de frontera $x \geq 0$.

Capítulo 1

INTRODUCCION

La historia de la programación lineal comienza con el método simplex, creado por Dantzig en 1946. Este método fue el único merecedor de atención hasta 1978, cuando salió a la luz el algoritmo elipsoidal de Khachian. Este algoritmo resolvía un problema teórico importante: mostraba que el problema de programación lineal podía ser resuelto con un número de operaciones aritméticas limitadas por un polinomio cuya variable era un indicador del "tamaño" del problema (definimos tamaño de un problema en el capítulo 3). Pero rápidamente se llegó a verificar que aunque este algoritmo tuviese buenas propiedades teóricas era irremediablemente lento en problemas reales.

Por varios años, vivimos con la existencia de dos algoritmos:
1°) El algoritmo simplex que tenia una complejidad exponencial (pésima) en estudios del peor caso, pero un desempeño bueno en problemas del mundo real.
2°) El algoritmo elipsoidal que era polinomial pero ineficiente en el sentido que realizaba un número grande de iteraciones.

En 1984 Karmarkar publicó un notable algoritmo en tiempo polinomial para resolver problemas de programación lineal, dando inicio a una gran investigación en el área de la programación lineal a través del estudio de los llamados algoritmos de punto interior. Las cuales se clasifican en 3 grandes grupos: los algoritmos afines, los algoritmos de reducción de potencial (al cual pertenece el algoritmo de Karmarkar), y los algoritmos basados en

la trayectoria central.

En esta tesis, estamos particularmente interesados en el estudio de los algoritmos de trayectoria central. Estos algoritmos resuelven el siguiente problema de programación lineal

$$\begin{aligned} &\text{minimizar} && c^T x \\ &\text{sujeto a} && \\ & && Ax = b, \\ & && x \geqslant 0, \end{aligned}$$

y su dual

$$\begin{aligned} &\text{maximizar} && b^T u \\ &\text{sujeto a} && \\ & && A^T u + s = c, \\ & && s \geqslant 0, \end{aligned}$$

la idea básica en los algoritmos basados en la trayectoria central es motivada de la observación de que aquello que hace esencialmente dificil la solución del problema lineal es la presencia de las restricciones de no negatividad, i.e., las restricciones $x \geq 0$. Por esta razón, convertimos el problema de programación lineal en un problema con sólo restricciones de igualdad, usando una función llamada función de barrera

$$f(x, \mu) := c^T x - \mu \sum_{j=1}^{n} \ln x_j,$$

que reemplazará a la función objetivo original. A continuación consideramos la siguiente familia de problemas de programación no lineal (llamado problema de barrera) parametrizado por el parámetro $\mu > 0$:

$$\begin{aligned} &\text{minimizar} \quad f(x,\mu) \\ &\text{sujeto a} \\ &\qquad\qquad Ax = b, \end{aligned}$$

siendo este problema de barrera un problema convexo (capitulo 2 - teorema 2.1), las condiciones de Karush - Kunh - Tucker (KKT)

$$\begin{aligned} A^T u + s &= c \\ Ax &= b \\ XSe &= \mu e \end{aligned}$$

donde: X =diag$(x_1, x_2, ..., x_n)$; S =diag$(s_1, s_2, ..., s_n)$ y $e = (1, 1, ..., 1)^T$. Son condiciones necesarias y suficientes de optimalidad. Probamos (capitulo 2 - teorema 2.1), que para cada $\mu > 0$ los problemas de barrera tienen una solución óptima la cual denotaremos por $x(\mu)$.

Cuando μ varia, los minimizadores $x(\mu)$ forman una curva continua llamada trayectoria central.

La cual es ilustrada en la siguiente figura:

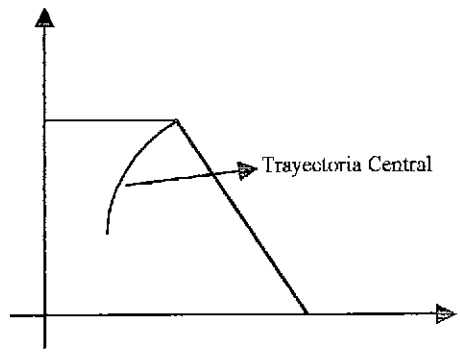

Figura 1.1.

En particular, puede probarse (capitulo 2 - teorema 2.1) que el límite cuando $\mu \to 0$ de $x(\mu)$ existe y es una solución óptima x^* del problema de programación lineal.

El cálculo explícito de esta trayectoria central

$$\Gamma = \{(x, u, s) \in \mathcal{L}^0 : Xs = \frac{x^T s}{n} e\},$$

es en general para la mayoria de problemas, muy complicado, razón por la cual los algoritmos basados en la trayectoria central solo generan puntos que se encuentran próximos (en el algún sentido que precisaremos posteriormente) a dicha trayectoria.

Como veremos la diferencia central entre los diversos algoritmos derivados de la trayectoria central está en el empleo de vecindades que nos dan la aproximación a la trayectoria central.

Nuestro trabajo de tésis está estructurado de la siguiente manera:

En el capitulo 2 establecemos conceptos importantes que nos permitiran tener claras las ideas fundamentales de los algoritmos de seguimiento de trayectoria central, como la función barrera la cual nos servirá para discutir la existencia y unicidad de una solución (x_μ, u_μ, s_μ) del problema de barrera, así como otros principios importantes, además analizaremos las condiciones de Karush - Kunh - Tucker (KKT) y el método de Newton.

En el capítulo 3 hacemos un estudio de la complejidad computacional de los algoritmos, definiendo para ello el tamaño de un problema lineal. Además, haremos un estudio de la complejidad del algoritmo simplex en el peor caso y en el caso promedio.

En el capitulo 4 analizaremos los algoritmos de trayectoria central discutiendo la medida de proximidad entre la trayectoria central y puntos próximos a ella; para luego estudiar cada uno de los algoritmos de trayectoria central que consideramos en el presente trabajo de tésis.

Finalmente, en el Capitulo 5 mostramos resultados numéricos donde comprobamos

que los algoritmos de trayectoria central tienen mejor perfomance, para determinados tipos de problemas, que el algoritmo simplex, asi como también presentaremos resultados que muestran la mejor estabilidad numérica de los algoritmos de puntos interiores en relación al algoritmo simplex.

Capítulo 2

PRELIMINARES

En este capítulo hacemos un recuento de las principales herramientas matemáticas asociadas con los algoritmos de puntos interiores que estudiaremos en nuestro trabajo de tésis, en la sección 2.1 estudiamos el problema de barrera asociado a un problema de programación lineal y la curva llamada trayectoria central que se genera en la solución de estos problemas de barrera, asi como analizaremos algunos teoremas importantes que nos servirán para establecer con más claridad el fundamento de éste trabajo de tésis. En la sección 2.2 estudiamos las condiciones de Karush - Kunh - Tucker (KKT) y la especializamos para el caso de la programación lineal. En la sección 2.3 estudiamos el método de Newton para la solución de un sistema de ecuaciones no lineales.

2.1 Método de barrera en la programación lineal

Consideremos el problema de programación lineal en su forma estándar

$$\begin{aligned} &\text{minimizar} && c^T x \\ &\text{sujeto a} && \\ & && Ax = b, \\ & && x \geqslant 0, \end{aligned} \tag{2.1}$$

donde $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, y $A \in \mathbb{R}^{m \times n}$.

El conjunto de soluciones factibles del problema estándar y el conjunto de sus puntos interiores asociados al problema son dados respectivamente por:

$$\begin{aligned} \mathcal{F} &= \{x \in \mathbb{R}^n \ / \ Ax = b, x \geqslant 0\}, \text{ es acotado} \\ \mathcal{F}^0 &= \{x \in \mathcal{F} \ / x > 0\}, \text{ es no vacío.} \end{aligned}$$

El respectivo problema dual será:

$$\begin{aligned} &\text{maximizar} \quad b^T u \\ &\text{sujeto a} \\ &\qquad\qquad A^T u + s = c, \\ &\qquad\qquad s \geqslant 0, \end{aligned} \tag{2.2}$$

donde $u \in \mathbb{R}^m$ y $s \in \mathbb{R}^n$.

El conjunto de soluciones factibles del problema dual y el conjunto de puntos interiores asociado a este son dados por:

$$\begin{aligned} \mathcal{S} &= \left\{s \in \mathbb{R}^n \ / \ A^T u + s = c; \text{ para algún } u \in \mathbb{R}^m, s \geqslant 0\right\}, \\ \mathcal{S}^0 &= \{s \in \mathcal{S} \ / \ s > 0\}. \end{aligned}$$

Ahora definiremos la región primal - dual factible $\mathcal{L}$ y su interior $\mathcal{L}^0$ como

$$\mathcal{L} := \left\{(x, u, s) \ / \ Ax = b, \ A^T u + s = c, \ (x, s) \geqslant 0\right\},$$

y

$$\mathcal{L}^0 := \left\{(x, u, s) \ / \ Ax = b, \ A^T u + s = c; \ (x, s) > 0\right\},$$

podemos decir que para cada iteración, los algoritmos de trayectoria central parten de un punto $(x, u, s) \in \mathcal{L}^0$. Debido a la dificultad de trabajar con restricciones de desigualdad

$x \geq 0$ en los problemas de programación lineal, convertimos el problema lineal a un problema de restricciones de igualdad solamente, usando una función barrera que impida que alguna variable llegue a la frontera ($x_j = 0$). Esto se logra al añadir el término $-\ln x_j$ a la función objetivo.

Este término permite que el valor de la función objetivo se incremente sin límite cuando x_j tiende a 0.

Definición 2.1: La siguiente función:

$$f(x, \mu) := c^T x - \mu \sum_{j=1}^{n} \ln x_j$$

es llamada función barrera logarítmica, donde μ es un parámetro mayor que cero.

La función barrera fue utilizada por primera vez en optimización por Frisch (Ver [3]).

Ahora consideremos las siguiente definición.

Definición 2.2: La siguiente familia de problemas de programación no lineal, parametrizados por $\mu > 0$:

$$\begin{aligned} &\text{minimizar} \quad f(x, \mu) \\ &\text{sujeto a} \\ &\qquad Ax = b, \end{aligned}$$

es conocida como **problemas de barrera.**

El primer término de $f(x, \mu)$ mide el valor de la función objetivo del problema lineal estándar y el segundo término funciona como una penalización de los puntos que se aproximan a la frontera de la región factible del problema.

Respecto a este término haremos que $-\sum_{j=1}^{n} \ln x_j$ sea igual a $\varphi(x)$, luego tenemos

$$\varphi(x) = -\sum_{j=1}^{n} \ln x_j.$$

Es así que nuestro problema de barrera quedará expresado de la siguiente manera

$$\begin{aligned} &\text{minimizar} \quad c^T x + \mu\varphi(x) \\ &\text{sujeto a} \\ &\qquad Ax = b, \end{aligned} \tag{2.3}$$

Ahora mostraremos que para algún $\mu > 0$, el problema (2.3) tiene una única solución óptima sobre $\mathcal{F}^0$. Además, cuando μ tiende a cero, la curva formada por toda la minimización es continua en μ y converge a la única solución óptima x^* del problema (2.1).

Teorema 2.1: Para $\mu > 0$, el problema (2.3) tiene una única solución óptima $x(\mu) \in \mathcal{F}^0$.

Prueba:

Puesto que $\varphi(x)$ es estrictamente convexa sobre $\mathcal{F}$ entonces $c^T x + \mu\varphi(x)$ también es estrictamente convexa. Además, el dominio factible $\mathcal{F}$ es convexo y acotado. Esto garantiza que (2.3) tenga una única solución óptima $x(\mu)$ sobre $\mathcal{F}$. Por otra parte, esta solución tiene que satisfacer las siguientes condiciones llamadas condiciones de Karush - Kuhn - Tucker (KKT) (ver sección 2.2)

$$\begin{aligned} &c + \mu\nabla\varphi(x) - A^T u - s = 0, \\ &Ax = b, \\ &x^T s = 0, \\ &x \geq 0, \ s \geq 0 \end{aligned} \tag{2.4}$$

donde $u \in \mathbb{R}^m$ y $s \in \mathbb{R}^n$ son conocidos como multiplicadores de Lagrange.

Por la definición de la función $\varphi(x)$ (Ver [10]), $(\nabla\varphi(x))$, se aproxima al menos infinito si x_j llega a ser cero para algún j. Puesto que μ es positivo, si x_j es cero para algún j, $\mu(\nabla\varphi(x))$, se aproxima a menos infinito y por lo tanto (2.4) podría no tener alguna solución finita. Por lo tanto, $x(\mu)$ tiene que hallarse en $\mathcal{F}^0$ y (2.4) se reduce al siguiente sistema:

$$
\begin{aligned}
& c + \mu \nabla \varphi(x) - A^T u = 0, \\
& Ax = b, \\
& x > 0.
\end{aligned} \tag{2.5}
$$

Teorema 2.2: $\{x(\mu) : \mu > 0\}$ caracteriza un interior y una curva continua en $\mathcal{F}^0$.

Prueba:

Por el teorema 2.1, solamente nos queda por probar la parte de la continuidad. Supongamos que la curva es discontinua en $\overline{\mu} > 0$. Entonces, existe $\varepsilon > 0$ tal que no importa que tan pequeño sea $\Delta\mu$, si $\|x(\overline{\mu} + \Delta\mu) - x(\overline{\mu})\| \geq 2\varepsilon > 0$. Probaremos por contradicción que $x(\overline{\mu} + \Delta\mu)$ no es el mínimo del problema (2.3) para $\mu = \overline{\mu} + \Delta\mu$.

Como hemos mencionado anteriormente, $c^T x + \mu\varphi(x)$ es estrictamente convexa. Puesto que $x(\overline{\mu})$ es el mínimo del problema (2.3) para $\mu = \overline{\mu}$, por $x \in \mathcal{F}^0$ y $\|x(\overline{\mu}) - x\| > \varepsilon$, existe un $\tau > 0$, tal que

$$
c^T x + \overline{\mu}\varphi(x) > c^T x(\overline{\mu}) + \overline{\mu}\varphi(x(\overline{\mu})) + \tau. \tag{2.6}
$$

En particular,

$$
c^T x(\overline{\mu} + \Delta\mu) + \overline{\mu}\varphi(x(\overline{\mu} + \Delta\mu)) > c^T x(\overline{\mu}) + \overline{\mu}\varphi(x(\overline{\mu})) + \tau,
$$

no importa que tan pequeño sea $\Delta\mu$.

Supongamos que el dominio de $\varphi(x)$ sea $\mathcal{F}$. Puesto que $\mathcal{F}$ es compacto y $\varphi(x)$ es continua sobre $\mathcal{F}$ (Ver [10]), elegiremos $\Delta\mu$ suficientemente pequeño tal que

$$
0 < |\Delta\mu| \, \|\varphi(x)\|_\infty < \frac{\tau}{2}, \tag{2.7}
$$

donde $\|.\|_\infty$ es la norma ∞. Por consiguiente

$$
\begin{aligned}
& c^T x(\overline{\mu}) + (\overline{\mu} + \Delta\mu)\overline{\mu}\varphi(x(\overline{\mu})), \\
& < c^T x(\overline{\mu}) + \overline{\mu}\varphi(x(\overline{\mu})) + \frac{\tau}{2}, && \text{por (2.7)} \\
& < c^T x(\overline{\mu} + \Delta\mu) + \overline{\mu}\varphi(x(\overline{\mu} + \Delta\mu)) - \frac{\tau}{2}, && \text{por (2.6)} \\
& < c^T x(\overline{\mu} + \Delta\mu) + (\overline{\mu} + \Delta\mu)\varphi(x(\overline{\mu} + \Delta\mu)), && \text{por (2.7)}
\end{aligned}
$$

con lo cual mostramos que $x(\overline{\mu} + \Delta\mu)$ no es el mínimo del problema (2.3) para $\mu = \overline{\mu} + \Delta\mu$.■

Teorema 2.3: Para $\mu > 0$, $\{c^T x(\mu)\}$ es una sucesión monótonamente decreciente en μ.

Prueba:

Por la definición de $x(\mu_1)$ y $x(\mu_2)$, tenemos

$$c^T x(\mu_1) + \mu_1\varphi(x(\mu_1)) \leq c^T x(\mu_2) + \mu_1\varphi(x(\mu_2)), \tag{2.8}$$

y

$$c^T x(\mu_2) + \mu_2\varphi(x(\mu_2)) \leq c^T x(\mu_1) + \mu_2\varphi(x(\mu_1)), \tag{2.9}$$

multiplicando ambos lados de (2.9) por -1, tenemos

$$-c^T x(\mu_1) - \mu_2\varphi(x(\mu_1)) \leq -c^T x(\mu_2) - \mu_2\varphi(x(\mu_2)), \tag{2.10}$$

sumando (2.8) y (2.9) a la vez se tiene

$$(\mu_1 - \mu_2)\varphi(x(\mu_1)) \leq (\mu_1 - \mu_2)\varphi(x(\mu_2)),$$

si $\mu_1 > \mu_2 > 0$, entonces $\varphi(x(\mu_1)) \leq \varphi(x(\mu_2))$.

Además, por (2.9)

$$\begin{aligned} c^T x(\mu_2) + \mu_2 \varphi(x(\mu_2)) &\leq c^T x(\mu_1) + \mu_2 \varphi(x(\mu_1)) \\ &\leq c^T x(\mu_1) + \mu_2 \varphi(x(\mu_2)). \end{aligned}$$

Esto implica que $c^T x(\mu_2) \leq c^T x(\mu_1)$, si $0 < \mu_2 < \mu_1$.■

Teorema 2.4: Dada una sucesión decreciente positiva $\{\mu_k\}$ tal que $\lim_{k\to\infty} \mu_k = 0$, si $\lim_{k\to\infty} x(\mu_k) = x^*$, entonces x^* es el mínimo de (2.1). Esto indica que $x(\mu)$ es realmente continua sobre $\mu \geq 0$.

Prueba:

Si $\varphi(x)$ toma el valor de $+\infty$ sobre la frontera, haremos la prueba por contradicción. Supongamos que x^* no es el mínimo del problema (2.1), pero v^* si lo es.

En este caso, podemos encontrar $w \in \mathcal{F}^0$ que satisface $c^T v^* < c^T w < c^T x^*$, puesto que $\lim_{k\to\infty} \mu_k = 0$, podemos elegir un k suficientemente grande y hallar $\delta > 0$, tal que

$$c^T w + \mu_k \varphi(w) < c^T x^* - \delta.$$

Para $x^* \in \mathcal{F}^0$, dado que este es el limite de $\{x(\mu_k)\}$, cuando k es suficientemente grande $x(\mu_k)$ se tiene una vecindad pequeña de x^* con el cual $\varphi(x(\mu_k))$ es cerrado y se tiene finito valores. Por lo tanto, $c^T x^* - \delta < c^T x(\mu_k) + \mu_k \varphi(x(\mu_k))$, para k suficientemente grande.

Pero esto contradice el hecho que $x(\mu_k)$ es el mínimo de (2.3) cuando $\mu = \mu_k$.

Ahora, si $x^* \in \mathcal{F} \ / \ \mathcal{F}^0$ y $\varphi(x)$ toman el valor $+\infty$ en x^*, $\mu_k \varphi(x(\mu_k))$ puede que no tenga un límite. Sin embargo, tenemos que $\lim_{k\to\infty} \inf\{\mu_k \varphi(x(\mu_k))\} \geq 0$.

Por consiguiente para k suficientemente grande,

$$c^T x^* - \frac{\delta}{2} < c^T x(\mu_k) + \lim_{k\to\infty} \inf\{\mu_k \varphi(x(\mu_k))\},$$

$$< \quad c^T x(\mu_k) + \mu_k \varphi(x(\mu_k)) + \frac{\delta}{2},$$

como consecuencia,

$$c^T w + \mu_k \varphi(w) < c^T x^* - \delta < c^T x(\mu_k) + \mu_k \varphi(x(\mu_k)),$$

con lo que se tiene una contradicción. Por lo tanto, x^* debe ser la solución óptima de (2.1).■

Teorema 2.5: Para alguna sucesión positiva decreciente $\{\mu_k\}$ con $\lim_{k\to\infty} \mu_k = 0$, $\{x(\mu_k)\}$ converge. Consecuentemente, $\{x(\mu_k)\}$ converge a la solución óptima de (2.1).

Prueba:

Puesto que $\mathcal{F}$ es compacto, la sucesión $\{x(\mu_k)\}$ tiene al menos un punto límite. Para cada punto límite podemos encontrar una subsucesión la cual realmente converga. Aplicando el teorema 2.4, podemos concluir que cada punto límite es de hecho una solución óptima de (2.1). Pero como (2.1) se supone que tiene una única solución óptima, por lo tanto $\{x(\mu_k)\}$ tiene uno y sólo un punto límite y por lo tanto converge.■

A continuación ponemos un ejemplo de un problema de barrera simple.

Ejemplo 2.1: Consideremos el problema

$$\begin{aligned} &\text{minimizar} \quad x \\ &\text{sujeto a} \\ &\qquad\qquad x \geqslant 0 \end{aligned}$$

La función barrera en este caso es

$$f(x, \mu) = x - \mu \ln x,$$

calculando, encontramos que el mínimo $x(\mu)$es igual a μ.

Como μ decrece a cero (en particular se observa que $\lim_{\mu \longrightarrow 0} x(\mu)$ existe y es una solución optima x^* del problema lineal primal inicial (2.1) dado que cuando μ es muy pequeño, el término logaritmico en cualquier caso es casi insignificante) luego la solución óptima se aproxima a $x^* = 0$.

Ahora como μ toma diferentes valores, para cada valor de μ obtenemos una solución de (2.3) la cual es llamada el punto central; este se encuentra en el interior de la región factible del problema lineal primal (2.1).

Definición 2.3: El conjunto de los puntos centrales de un problema, describe una curva llamada la Trayectoria Central del problema. Ver figura 2.1.

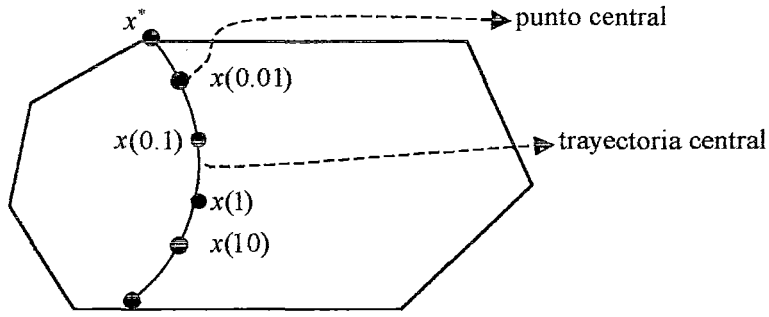

figura 2.1 : *La trayectoria central y los puntos centrales.*

Ejemplo2.2: (Cálculo de la trayectoria central)

$$\begin{aligned} \text{minimizar} \quad & x_2 \\ \text{sujeto a} \quad & \\ & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Sea el conjunto de soluciones factibles. Para hallar el camino central necesitamos

resolver el siguiente problema de optimización

$$\begin{aligned} \text{minimizar} \quad & x_2 - \mu \ln x_1 - \mu \ln x_2 - \mu \ln x_3 \\ \text{sujeto a} \quad & \\ & x_1 + x_2 + x_3 = 1 \end{aligned}$$

hacemos $x_3 = 1 - x_1 - x_2$, ahora resolvemos el siguiente problema de optimización no lineal sin restricciones

$$\text{minimizar } x_2 - \mu \ln x_1 - \mu \ln x_2 - \mu \ln(1 - x_1 - x_2)$$

Derivando e igualando a cero, encontramos que el camino central es dado por:

$$\begin{aligned} x_1(\mu) &= \frac{1 - x_2(\mu)}{2} \\ x_2(\mu) &= \frac{1 + 3\mu - \sqrt{1 + 9\mu^2 + 2\mu}}{2} \\ x_3(\mu) &= \frac{1 - x_2(\mu)}{2} \end{aligned}$$

El centro analítico[1] se obtiene haciendo que $\mu \to \infty$. Este viene a ser el punto $(1/3, 1/3, 1/3)$.

Sea Q el conjunto de soluciones óptimas del problema original, dado por:

$$Q = \{x \ / \ x = (x_1,\ 0,\ x_3),\ x_1 + x_3 = 1,\ x \geq 0\}$$

El centro analítico del poliedro Q es el punto $(1/2,\ 0,\ 1/2)$. Este punto se obtiene haciendo $\mu = 0$ encontrando así una solución óptima que es el centro analítico del conjunto de soluciones óptimas. Ver la figura 2.1.

[1] Definamos el centro analítico de un poliedro general $P = \{(x, s)/a_i^t x \leq b_i;\ i = 1, ..., m\}$ como la minimización de la función $-\sum_{i=1}^{n} \ln(b_i - a_i^t x)$; el centro analítico no se representa de forma independiente, por el contrario su representación depende del sistema de desigualdades lineales usadas para representar el poliedro P. Si aumentamos una restricción redundante, el centro analítico cambia.

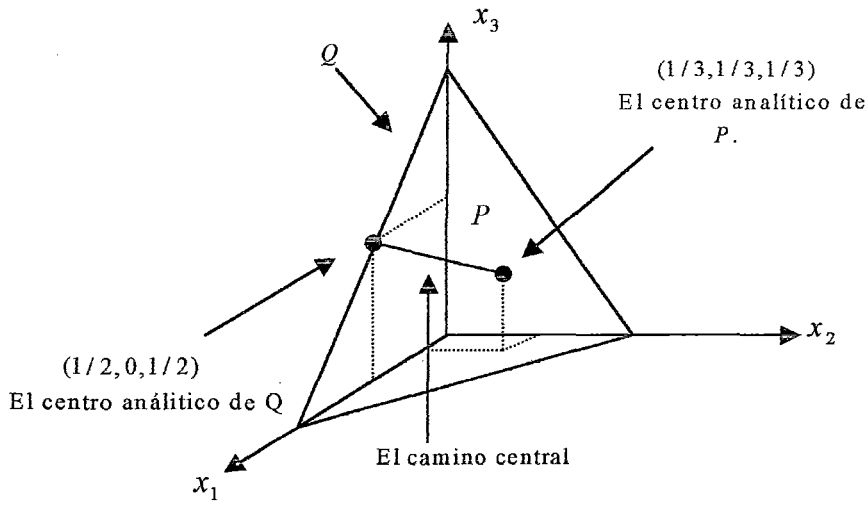

Figura 2.2: La trayectoria central y el centro analítico para el ejemplo 2.2.

La trayectoria central desempeña un papel importante en la presentación de los algoritmos de puntos interiores, de hecho en el capitulo 4 de esta tesis estudiaremos una clase de estos algoritmos que siguen esta curva en cada iteración.

Como ya se mencionó, cada punto que pertenece a la trayectoria central de un problema soluciona el problema barrera asociado, para un determinado valor de μ. A continuación estudiaremos las condiciones de optimalidad que deben de ser satisfechas para solucionar un problema general de optimización con restricciones, las cuales son conocidas como las condiciones de Karush - Kunh - Tucker (o KKT).

2.2 Condiciones de Karush - Kunh - Tucker (o KKT)

Consideremos el siguiente problema de programación no lineal con restricciones.

$$\begin{aligned} &\text{minimizar} && f(x) \\ &\text{sujeto a} && \\ & && g_1(x) = 0 \\ & && g_2(x) = 0 \\ & && \vdots \\ & && g_m(x) = 0 \end{aligned}$$

donde $x \in \mathbb{R}^n$ y $f, g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$; $i = 1, ..., m$ son funciones diferenciales.

Las condiciones de KKT para este problema son descritas a continuación:

Si $\bar{x}$ es una solución óptima del problema no lineal, entonces existen multiplicadores $y_i \in \mathbb{R}$, $i = 1, ..., m$ tal que

$$\begin{aligned} g_i(\bar{x}) &= 0;\ i = 1, ..., m \\ \nabla f(\bar{x}) &= \sum_{i=1}^{m} y_i \nabla g_i(\bar{x});\ i = 1, ..., m, \end{aligned}$$

particularizando estas condiciones para el problema barrera (2.3), definido para un determinado μ, verificamos que su solución óptima debe satisfacer el siguiente sistema de ecuaciones no lineales:

$$\begin{aligned} c_j - \mu . \frac{1}{x_j} - \sum_{i=1}^{q} u_i a_{ij} &= 0;\ j = 1, 2..., n \\ b_i - \sum_{j=1}^{q} a_{ij} x_j &= 0;\ i = 1, 2..., m. \end{aligned}$$

donde los multiplicadores $u_i \in \mathbb{R}$, $i = 1, 2..., m$.

Utilizando la notación matricial, podemos reescribir este sistema de ecuaciones, como:

$$\begin{aligned} A^T u + \mu X^{-1} e &= c, \\ Ax &= b, \end{aligned}$$

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

Definiendo ahora $s = \mu X^{-1} e$, $s \in \mathbb{R}^n$ reescribiremos el sistema como

$$\begin{aligned} A^T u + s &= c, \\ Ax &= b, \\ s &= \mu X^{-1} e, \end{aligned}$$

finalmente multiplicando la tercera ecuación por X, obtenemos las condiciones de optimalidad para el problema barrera en la forma primal - dual.

$$\begin{aligned} A^T u + s &= c, \\ Ax &= b, \\ XSe &= \mu e, \end{aligned} \tag{2.11}$$

donde:

$X = diagonal\,(x_1, x_2, ..., x_n)$; $S = diagonal\,(s_1, s_2, ..., s_n)$ y $e = (1, 1, ..., 1)^T$.

Llamaremos a (x_μ, u_μ, s_μ) la solución del sistema (2.4) para un μ especifico.

Si $\mu = 0$ las ecuaciones de (2.4) juntamente con las restricciones $x, s \geqslant 0$ son exactamente las condiciones de optimalidad de los problemas (2.1) y (2.2).

El sistema (2.11) es un sistema no lineal que tiene $2n + m$ restricciones y $2n + m$ variables. La utilización del método de Newton para resolver el sistema, da origen a la familia de métodos de puntos interiores primales - duales. Para cada iteración de estos métodos, se parte de un punto (x, u, s) tal que x es primal factible, (u, s) es dual factible y $x, s > 0$.

UNIVERSIDAD NACIONAL DEL CALLAO
BIBLIOTECA CENTRAL
UNAC

Se definió la región primal - dual factible $\mathcal{L}$ y su interior $\mathcal{L}^0$ como

$$\mathcal{L} := \left\{(x,u,s) \,/\, Ax = b,\ A^T u + s = c;\ (x,s) \geqslant 0\right\},$$

y

$$\mathcal{L}^0 := \left\{(x,u,s) \,/\, Ax = b,\ A^T u + s = c;\ (x,s) > 0\right\},$$

podemos decir que para cada iteración, los métodos primales-duales parten de un punto $(x,u,s) \in \mathcal{L}^0$.

La dirección sobre la cual se camina a partir de este punto se basa en la aplicación del método de Newton a las ecuaciones (2.11).

2.3 El Método de Newton

Dada una función $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, el método de Newton busca un cero de la función, esto es, busca un punto $\bar{\lambda} \in \mathbb{R}^n$ tal que $F(\bar{\lambda}) = 0$, esto es realizado a travez de un proceso iterativo, en el cual para cada iteración se parte de un punto dado $\lambda \in \mathbb{R}^n$ y se camina una aproximación de la dirección $\Delta\lambda$ tal que $F(\lambda + \Delta\lambda) = 0$. Esta aproximación es calculada de la siguiente manera:

$$F(\lambda + \Delta\lambda) \approx F(\lambda) + J(\lambda)\Delta\lambda,$$

donde J es el jacobiano de F. El cálculo de la dirección $\Delta\lambda$ a ser tomada para cada iteración es realizado a través de la solución del sistema lineal $J(\lambda)\Delta\lambda = -F(\lambda)$.

Ahora podemos reescribir el sistema (2.11) como $F(\lambda) = 0$ donde $\lambda := (x,u,s)$ y

$$F(\lambda) = F(x,u,s) := \begin{bmatrix} A^T u + s - c \\ Ax - b \\ XSe - \mu e \end{bmatrix},$$

2105

En este caso el jacobiano de F es dado por:

$$J(\lambda) = J(x,u,s) = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix},$$

y la dirección de Newton $\Delta\lambda := (\Delta x, \Delta u, \Delta s)$; que es calculada a partir de un punto $(x,u,s) \in \mathcal{L}^0$, es dada por la solución del siguiente sistema lineal

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e - XSe \end{bmatrix}. \tag{2.12}$$

Por lo tanto el vector $\Delta\lambda := (\Delta x, \Delta u, \Delta s)$ es llamado la dirección de Newton y el proceso de calcular esta dirección es llamado paso de Newton.

Luego sin perdida de generalidad llamaremos a

$$\Delta\lambda^k := \left(\Delta x^k, \Delta u^k, \Delta s^k\right),$$

la dirección de Newton para la k - ésima iteración, la nueva solución será:

$$\begin{aligned} x^{k+1} &= x^k + \beta_p^k \Delta x^k, \\ u^{k+1} &= u^k + \beta_D^k \Delta u^k, \\ s^{k+1} &= s^k + \beta_D^k \Delta s^k, \end{aligned}$$

donde β_p^k y β_D^k son longitudes de paso para las variables primal y dual, respectivamente. La no negatividad condición para x^{k+1} y s^{k+1} reglamentan la elección de β_p^k y β_D^k.

Una posibilidad para preservar la no negatividad es elegir

$$\beta_p^k \ := \ = \text{minimo}\left\{1, \alpha \underset{\left\{i/\ (\Delta x^k)_i < 0\right\}}{\text{minimo}} \left(-\frac{x_i^k}{(\Delta x^k)_i}\right)\right\},$$

$$\beta_D^k \ := \text{minimo}\left\{1, \alpha \underset{\{i/\,(\Delta s^k)_i<0\}}{\text{minimo}}\left(-\frac{s_i^k}{(\Delta s^k)_i}\right)\right\},$$

donde $0 < \alpha < 1$, además $(\Delta x^k)_i$ y $(\Delta s^k)_i$ es la i - ésima componente de Δx^k y Δs^k respectivamente; la razón por la que buscamos un α menor que 1 es evitar aproximarnos a la frontera del conjunto factible primal y del dual, respectivamente.

Finalmente notamos que las restricciones de igualdad del problema primal y dual se satisfacen, en efecto observemos el sistema (2.12) de donde se desprende que:

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta u + \Delta s &= 0, \end{aligned}$$

para la k - ésima iteración tendriamos

$$\begin{aligned} A\Delta x^k &= 0, \\ A^T\Delta u^k + \Delta s^k &= 0. \end{aligned}$$

Por lo tanto, la nueva solución es factible.

Ahora regresando al sistema (2.12), vamos a obtener de su tercera ecuación la siguiente expresión para Δs.

$$\Delta s = X^{-1}\left(\mu e - XSe - S\Delta x\right), \tag{2.13}$$

sustituyendo (2.13) en la primera ecuación de (2.5), tenemos

$$A^T\Delta u + X^{-1}S\Delta x = S - \mu X^{-1}e. \tag{2.14}$$

Ahora resolvemos (2.14) para Δx, obteniendo

$$\Delta x = XS^{-1}\left(S - \mu X^{-1}e - A^T\Delta u\right),$$

hacemos $S = -A^T u + c$

$$\Delta x = XS^{-1}\left(-A^T u + c - \mu X^{-1} e - A^T \Delta u\right), \tag{2.15}$$

sustituyendo (2.15) en la segunda ecuación de (2.5)

$$-AXS^{-1}A^T \Delta u = -AXS^{-1}\left(-A^T u + c - \mu X^{-1} e\right), \tag{2.16}$$

resolviendo para Δu y sustituyendo en (2.12) tenemos

$$A^T\left(\left(AXS^{-1}A^T\right)^{-1} AXS^{-1}\left(c - A^T u - \mu X^{-1} e\right)\right) + X^{-1}S\Delta x = S - \mu X^{-1} e,$$

despejando

$$X^{-1}S\Delta x = -\mu X^{-1} e + S - A^T\left(\left(AXS^{-1}A^T\right)^{-1} AXS^{-1}\left(c - A^T u - \mu X^{-1} e\right)\right), \tag{2.17}$$

definiendo: $D^2 = XS^{-1}$ y haciendo nuevamente $S = c - A^T u$, podemos reescribir (2.17) como

$$\begin{aligned} \Delta x &= \left(-D^2 - D^2 A^T \left(AD^2 A^T\right)^{-1} AD^2\right)\left(c - A^T u - \mu X^{-1} e\right), \\ &= \left(-D^2 - D^2 A^T \left(AD^2 A^T\right)^{-1} AD^2\right) c, \\ &\quad -\mu\left(-D^2 - D^2 A^T \left(AD^2 - A^T\right)^{-1} AD^2\right) X^{-1} e, \end{aligned} \tag{2.18}$$

cuando $\mu = 0$, observamos que (2.18) se reduce a

$$\Delta x = \left(-D^2 - D^2 A^T \left(AD^2 A^T\right)^{-1} AD^2\right) c,$$

donde Δx es la solución del sistema (2.12).

Como ya se mencionó, si $\mu = 0$, el sistema de ecuaciones (2.11) en conjunto con las restricciones de no negatividad para x y s representan las condiciones de optimalidad para los problemas primal y dual. La dirección de Newton (2.12) calculada para $\mu = 0$, debe por tanto, apuntar aproximadamente hacia el óptimo de estos problemas. En general, se puede recorrer poco a lo largo de esta dirección sin que una de las componentes de x ó s se torne negativa y consecuentemente, el progreso obtenido a lo largo de este recorrido es poco.

Por otro lado, la solución de (2.11) para un $\mu > 0$, es un punto perteneciente a la trayectoria central. En este caso, la dirección de Newton (2.12) debe apuntar aproximadamente hacia la trayectoria central , es decir hacia el interior del octante no negativo. Esperando asi que se pueda recorrer un mayor camino a lo largo de esta dirección antes que se alcance la frontera del octante.

Se considera entonces, que la dirección (2.12), calculada para algún $\mu > 0$, se obtiene al desviar hacia el interior del octante no negativo, la dirección que apunta hacia el óptimo del problema. El objetivo de este desvio es mantenerse alejado de la frontera del octante de manera que se permita un mayor paso a lo largo de la dirección calculada, sin salirse de la región factible del problema.

Cosideremos un punto dado (x, u, s), luego a partir de (2.12) tendremos la dirección de centralización que camina a partir de dicho punto; esta dirección es obtenida cuando μ asume el valor dado por:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i s_i = \frac{x^T s}{n}. \tag{2.19}$$

Debemos observar que la brecha de dualidad asociada a las soluciones x y (u, s) de los problemas primal y dual, es dado por

$$c^T x - b^T u = c^T x - x^T A^T u = (c - uA)^T x = x^T s, \tag{2.20}$$

verificamos entonces, de (2.19) que la brecha asociada a x y (u, s) puede ser escrita como $n\mu$, cuando μ asume el valor que le es atribuido en el cálculo de la dirección de

centralización a partir del punto (x, u, s).

Notemos ahora que la dirección de centralización apunta aproximadamente hacia el punto de la trayectoria central (x_μ, u_μ, s_μ) que satisface la relación $x_i s_i = \mu$ para todo $i = 1, ..., n$. La brecha de dualidad asociada a este punto central, dado por $x_\mu^T s_\mu$, es por tanto, igual a $n\mu$ y igual a la brecha asociada al punto de partida (x, u, s).

Concluimos entonces que la dirección de centralización apunta hacia el interior del octante no negativo. Siendo mas especificos, esta apunta aproximadamente hacia el punto central que tiene la misma brecha de dualidad que el punto de donde partimos. Se espera asi que se pueda caminar razonablemente en la dirección de la centralización antes de encontrar la frontera del octante no negativo. Por otro lado, esperamos también que poca o ninguna reducción se obtenga en la brecha de dualidad a lo largo de esta dirección.

Finalmente, considerando al mismo tiempo los objetivos de disminuir la brecha de dualidad y mantenerse lejos de la frontera del octante no negativo, los algoritmos primales - duales parten para cada iteración de un punto dado (x, u, s) y caminan por la siguiente dirección; obtenida de la solución del siguiente sistema

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau\mu e - XSe \end{bmatrix}, \tag{2.21}$$

donde

$$\mu = \frac{x^T s}{n} \text{ y } \tau \in [0, 1]$$

τ puede tomar diferentes valores los cuales caracterizaran diferentes algoritmos pertenecientes a la familia de los algoritmos de puntos interiores primales - duales; como veremos en el capitulo 4 de esta tesis.

Capítulo 3

COMPLEJIDAD DE ALGORITMOS

Una nueva área de investigación surgio en los últimos 80 años, en la frontera entre la Matemática Aplicada, la Lógica y la Ciencia de la Computación. Aún cuando no posee un nombre universalmente acepto, su denominación más difundida es "Complejidad Computacional", Su objetivo es determinar la existencia o no de algoritmos eficientes para ciertos problemas de optimización. En términos generales el análisis de la complejidad computacional de los algoritmos tiene como uno de sus objetivos determinar la existencia o no de algoritmos eficientes para ciertos problemas de optimización estableciendo cuales son los factores que hacen que un problema sea de fácil o dificil solución.

En este sentido, un aspecto importante es la generación de métodos o algoritmos eficientes para obtener una solución óptima, o una suficientemente buena. La eficiencia computacional de un algoritmo queda basicamente caracterizada por su:

a) Tiempo de ejecución (segundos o minutos de *CPU* utilizados para resolver un problema dado).

b) Requerimiento de memoria.

La complejidad computacional se refiere a la existencia y caracterización de algoritmos

en los terminos de eficiencia ya señalados. En esta tesis haremos referencia solamente al tiempo de ejecución ya que es una medida más sencilla de analizar, aunque los conceptos que se presentaran son tambien aplicables a los requerimientos de la memoria.

En este capítulo se establecen las bases y resultados esenciales de la teoría de complejidad computacional, poniendo énfasis en el caso del problema de programación lineal presentando las principales conclusiones que se han obtenido en relación al algoritmo simplex.

En la sección 3.1 definimos el tamaño de un problema lineal, asi como estudiamos la relación entre los vértices y la solución óptima de un problema lineal; en la sección 3.2 definimos el algoritmo polinomial asi como damos tambien la noción de tiempo de ejecución; en la sección 3.3 distinguimos dos enfoques distintos para el estudio de la complejidad: un analisis en el peor caso y un analisis en el caso promedio, y en la sección 3.4 discutiremos la complejidad algorítmica en el caso de la programación lineal; y finalmente en la sección 3.5 analizaremos el comportamiento del algoritmo simplex en el caso promedio.

3.1 Tamaño de un problema lineal

Dados $A \in R^{m\times n}, b \in R^m$ y $c \in R^n$ el problema de programación lineal es el siguiente problema de optimización

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

La función lineal $c^T x$ es llamada la función objetivo, y $x \in R^n$ es llamado vector de variable de decisión.

El conjunto $\mathcal{F} = \{x : Ax = b, x \geqslant 0\}$ es llamado el conjunto factible o región factible, el cual es una región poliedral convexa. Un punto $x \in \mathcal{F}$ es llamado un punto factible,

y un punto factible x^* es llamado una solución óptima si $c^T x^* \leqslant c^T x$ para todo punto factible x.

Si existe una sucesión $\{x^k\}$ tal que, x^k es factible y $c^T x^k \longrightarrow -\infty$, entonces el problema lineal es no acotado.

Definición 3.1:

Sea

$$Z = \left\{(A, b, c) / A \in R^{m \times n}, b \in R^m, c \in R^n\right\},$$

definiremos una concretización del problema lineal como un elemento del conjunto Z.■

Definición 3.2: El conjunto solución del problema lineal se representa de la siguiente manera:

$$\mathcal{M} = \left\{x \in \mathcal{F} : c^T x \leqslant c^T y; \forall y \in \mathcal{F}\right\}.■$$

Será bastante útil hacer una discusión entre un problema y una concretización de este problema. Por ejemplo el problema de programación lineal es un problema, en cambio

$$\begin{aligned} &\text{minimizar} && 2x + 3y \\ &\text{sujeto a} && \\ & && x + y \leqslant 1, \\ & && x, y \geqslant 0. \end{aligned}$$

es una concretización del problema de programación lineal.

Asi que en términos generales podemos afirmar lo siguiente:

Definición 3.3: Un problema de optimización lineal es definido como la reunión de sus concretizaciones.

Ahora sin perdida de generalidad asumiremos que los coeficientes de la matriz A, del vector b y del vector c son todos enteros, pues cada problema de programación lineal puede ser facilmente transformado en un problema equivalente[1] con coeficientes enteros.

[1] Dos problemas de optimización son equivalentes si tienen el mismo conjunto de soluciones óptimas.

Definición 3.4: Para cada entero n

$$\text{tamaño}\,(n) := 1 + \lceil \log_2 (|n| + 1) \rceil ,$$

donde el primer 1 es colocado debido a que se necesita un bit para almacenar el signo de n, el tamaño (n) representa el número de bits que se necesita para codificar n en sistema binario.(La notación $\lceil \alpha \rceil$ denota el menor número entero mayor o igual que α).

Definición 3.5: Dado v un vector de orden $p \times 1$ y M una matriz de orden $p \times l$, definimos lo siguiente:

$$\begin{aligned} \text{tamaño}\,(v) &: = \sum_{i=1}^{p} \text{tamaño}\,(v_i) , \\ \text{tamaño}\,(M) &: = \sum_{i=1}^{p} \sum_{j=1}^{l} \text{tamaño}\,(m_{ij}) , \end{aligned}$$

a continuación definiremos el tamaño de un problema lineal.

Definición 3.6:(*tamaño de un problema lineal*)

$$\text{tamaño (problema lineal)} := \text{tamaño}\ (A) + \text{tamaño}\ (b) + \text{tamaño}\ (c) .$$

Ahora daremos una definición más conveniente para poder hablar del tamaño de un problema lineal.

Definición 3.7:

$$L := \text{tamaño}\,(\det_{\max}) + \text{tamaño}\,(b_{\max}) + \text{tamaño}\,(c_{\max}) + m + n$$

donde:

$$\begin{aligned} \det_{\max} &:= \max_{A^l} \left(|\det (A^l)| \right), \\ b_{\max} &:= \max_{i} \left(|b_i| \right);\ i = 1, ..., m., \\ c_{\max} &:= \max_{j} \left(|c_j| \right);\ j = 1, ..., n., \end{aligned}$$

y A^l es alguna submatriz cuadrada de A.

Teorema 3.1: Para toda concretización del problema lineal

$$L < \text{tamaño (problema lineal)},$$

para probar este resultado, primero necesitamos el siguiente lema,

Lema 3.1:

1. Si $n \in \mathbb{Z}$ entonces $|n| \leqslant 2^{tamaño(n)-1} - 1$.
2. Si $v \in \mathbb{Z}^n$ entonces $\|v\| \leqslant \|v\|_1 \leqslant 2^{tamaño(v)-n} - 1$.
3. Si $A \in \mathbb{Z}^{n \times n}$entonces $|\det (A)| \leqslant 2^{tamaño(A)-n^2} - 1$.

Prueba:

1) Sabemos que el tamaño $(n) := 1 + \lceil \log_2(|n| + 1) \rceil$,

$\Rightarrow$ tamaño$(n) - 1 := \lceil \log_2(|n| + 1) \rceil$.

Sea $\lceil \log_2(|n| + 1) \rceil = y$

$\Leftrightarrow y \geq \log_2(|n| + 1)$; i.e. $y > \log_2(|n| + 1) \vee y = \log_2(|n| + 1)$

$y > \log_2(|n| + 1)$ $\quad \vee \quad$ $y = \log_2(|n| + 1)$

$\Leftrightarrow \log_2(|n| + 1) < y$ $\quad$ $\Leftrightarrow 2^y = |n| + 1$

$2^{\log_2(|n|+1)} < 2^y$ $\quad$ $2^y - 1 = |n|$ $\quad (\beta)$

$|n| + 1 < 2^y$

$|n| < 2^y - 1$ $\quad (\alpha)$

de (α) y (β) tenemos que:

$|n| \leq 2^y - 1$

$\Rightarrow |n| \leq 2^{tamaño(n)-1} - 1$.

2)

$$1+\|v\| \leqslant 1+\|v\|_1 = 1+\sum_{i=1}^{n}|v_i| \leqslant \prod_{i=1}^{n}(1+|v_i|) \leqslant \prod_{i=1}^{n} 2^{tamaño(v_i)-1} = 2^{tamaño(v)-n},$$

donde utilizamos 1.

3) Sea $a_1, ..., a_n$ las columnas de A. Puesto que $|\det(A)|$ representa el volumen del poliedro generado por $a_1, a_2, ..., a_n$, tenemos

$$|\det(A)| = \prod_{i=1}^{n}\|a_i\|.$$

Asi pues, por 2. tenemos

$$1+|\det(A)| \leqslant 1+\prod_{i=1}^{n}\|a_i\| \leqslant \prod_{i=1}^{n}(1+\|a_i\|) \leqslant \prod_{i=1}^{n} 2^{tamaño(a_I)-n} = 2^{tamaño(A)-n^2}. \blacksquare$$

Ahora probemos el teorema 3.1

Prueba:

Si B es una submatriz cuadrada de A entonces, por definición, $tamaño(B) \leqslant tamaño(A)$. Por otra parte, por el lema 3.1, $1+|\det(B)| \leqslant 2^{tamaño(B)-1}$

por lo tanto

$$\lceil \log(1+|\det(B)|) \rceil \leqslant tamaño(B) - 1 < tamaño(B) \leqslant tamaño(A). \qquad (1)$$

Sea $v \in \mathbb{Z}^p$, entonces

$$tamaño(v) \geqslant tamaño(\max_j |v_j|) + p - 1 = \lceil \log(1+\max_j |v_j|) \rceil + p,$$

por lo tanto,

$$tamaño(b) + tamaño(c) \geqslant \left\lceil \log\left(1+\max_j |c_j|\right)\right\rceil + \left\lceil \log\left(1+\max_i |b_i|\right)\right\rceil + m + n, \quad (2)$$

luego de (1) y (2) obtenemos el resultado deseado. ■

Observación 1:

$\det_{\max} * b_{\max} * c_{\max} * 2^{m+n} < 2^L$, puesto que para cada entero n, $2^{tamaño(n)} > |n|$.

De ahora en adelante usaremos L para indicar el tamaño de una concretización de un problema lineal.

Sabemos que la solución de un problema lineal está en un vértice el cual es una solución óptima.

Asi que, cuando encontramos una solución óptima para un problema lineal, fijamos toda nuestra atención solamente en los vértices.

A continuación enunciaremos un teorema donde se nos asegura que los vértices tienen una representación compacta.

Teorema 3.2: Sea x un vertice del poliedro definido por $Ax = b$, $x \geqslant 0$, entonces,

$$x^T = \left(\frac{p_1}{q}\frac{p_2}{q}....\frac{p_n}{q}\right),$$

donde

$$p_i\,(i = 1, ..., n),\ q \in \mathbb{N},$$

y

$$\begin{aligned} 0 &\leqslant p_i < 2^L, \\ 1 &\leqslant q < 2^L. \end{aligned}$$

Prueba:

Puesto que x es una solución básica factible, existe una base B tal que $x_B = A_B^{-1}b$ y $x_N = 0$.

hacemos $p_j = 0\ \forall j \in \mathbb{N}$, y enfocamos nuestra atención en los x_j tal que $j \in B$.

Sabemos por algebra lineal que

$$x_B = A_B^{-1} b = \frac{1}{\det(A_B)} Cof(A_B)\, b$$

donde $Cof(A_B)$ es la matriz cofactor de A_B. Cada entrada de A_B consiste de un determinante para alguna submatriz de A.

Sea $q = |\det(A_B)|$, entonces q es un entero puesto que A_B tiene componentes enteros, $q \geqslant 1$ puesto que A_B es invertible y $q \leqslant \det_{\max} < 2^L$. finalmente notemos que:

$$p_B = q x_B = |Cof(A_B)\, b|$$

por lo tanto

$$p_i \leqslant \sum_{j=1}^{m} \left| Cof(A_B)_{ij} \right| |b_j| \leqslant m \det_{\max} b_{\max} < 2^L. \blacksquare$$

Teorema 3.3: Sean x_1, x_2 los vertices de $Ax = b$, $x \geqslant 0$.

Si $c^T x_1 \neq c^T x_2$, entonces $\left| c^T x_1 - c^T x_2 \right| > 2^{-2L}$.

Prueba:

Por el teorema 3.2, $\exists\ q_1, q_2$ tal que $1 \leqslant q_1$ y $q_2 < 2^L$, y $q_1 x_1$, $q_2 x_2 \in \mathbb{N}^n$. Además

$$\begin{aligned}
\left| c^T x_1 - c^T x_2 \right| &= \left| \frac{q_1 c^T x_1}{q_1} - \frac{q_2 c^T x_2}{q_2} \right|, \\
&= \left| \frac{q_1 q_2 \left(c^T x_1 - c^T x_2 \right)}{q_1 q_2} \right|, \\
&\geqslant \frac{1}{q_1 q_2}, \text{ dado que } c^T x_1 - c^T x_2 \neq 0,\ q_1, q_2 \geqslant 1, \\
&> \frac{1}{2^L 2^L} = 2^{-2L}, \text{ dado que } q_1, q_2 < 2^L. \blacksquare
\end{aligned}$$

Corolario 3.1: Supongamos que

$$z = \min \left\{ c^T x : \underbrace{Ax = b,\ x \geqslant 0}_{\text{Poliedro } \mathcal{F}} \right\}$$

Supongamos que x es factible en $\mathcal{F}$, tal que $c^T x \leqslant z + 2^{-2L}$. Entonces, algún vértice x' tal que $c^T x' \leqslant c^T x$ es una solución óptima del problema lineal.

Prueba:

Supongamos que x' no es el óptimo. Entonces, existe un vértice óptimo x^*, tal que $c^T x^* = z$; puesto que x' no es el óptimo $c^T x \neq c^T x^*$ y por el teorema 3.3

$$\Longrightarrow c^T x' - c^T x^* > 2^{-2L}$$

$$\begin{aligned}
\Longrightarrow \quad & c^T x' > c^T x^* + 2^{-2L}, \\
& = z + 2^{-2L}, \\
& \geqslant c^T x, \qquad \text{por la definición de } x \\
& \geqslant c^T x', \qquad \text{por la definición de } x' \\
\Longrightarrow \quad & c^T x' > c^T x',
\end{aligned}$$

lo cual es una contradicción.■

Lo que este corolario nos indica es que necesitamos calcular la función objetivo con un error menor que 2^{-2L}. Si hallamos un vértice que está dentro de ese margen de error, entonces este será el óptimo.

3.2 Algoritmo Polinomial

Los problemas de optimización son resueltos por algoritmos. En este punto hay una distinción muy importante los algoritmos son diseñados para problemas (i.e, el método simplex es un algoritmo para la programación lineal). Sin embargo, son aplicados para

concretizaciones especificas.

Decimos que un algoritmo resuelve un problema si este termina en tiempo finito y produce la respuesta correcta para toda concretización del problema. Por supuesto es normal que cuando el algortimo es aplicado para concretizaciones de gran tamaño, el tiempo de terminación es mayor. Por esta razón el tiempo de ejecución de un algoritmo es usualmente expresado como una función del tamaño de la concretización para el cual este es aplicado.

A fin de tratar de hacer esto mas preciso necesitamos definir exactamente lo que es un algoritmo y la noción de tiempo de ejecución.

Definición 3.8: Un algoritmo es una lista de instrucciones para resolver un problema, dada cualquier concretización de cierto problema un algoritmo realiza un número finito de operaciones sobre los datos de la concretización.

Un algoritmo para resolver dicho problema determina que

$$\mathcal{M} = \left\{ x \in \mathcal{F} : c^T x \leqslant c^T y,\ \forall y \in \mathcal{F} \right\},$$

el conjunto solución del problema lineal es vacio o genera una solución x; x es tal que $x \in \mathcal{M}$ o x "está cercano" a $\mathcal{M}$ en algun sentido, en cuyo caso es llamado una solución aproximada.

Definición 3.9: Sea A un algoritmo para resolver un problema de optimización. El tiempo de ejecución $T(L)$ de A es el máximo número de operaciones elementales en el computador que requiere A para resolver una concretización del problema de tamaño L.

Definición 3.10: Sean $f : N \longrightarrow \mathbb{R}^+$ y $g : \mathbb{N} \longrightarrow \mathbb{R}^+$ decimos que $g(n) \in O(f(n))$, si existe una constante $c > 0$ y un $n_0 \in \mathbb{N}$ tal que

$$g(n) \leqslant cf(n);\ \forall n \geqslant n_0$$

Definición 3.11: Un algoritmo para un problema de optimización es polinómico si el número de operaciones aritméticas (efectuadas con precisión finita) necesarias para hallar una solución óptima exacta de una concretización del problema de optimización está acotada superiormente por una función polinómica de L. Es decir

$$\tilde{T}(L) = O(P(L))$$

Definición 3.12: Un algoritmo para un problema de optimización es exponencial, si dicho algoritmo no es polinómico.

En la siguiente tabla se compara el crecimiento de algunas funciones polinomiales y otras no polinomiales. Sólo damos valores aproximados, ya que estos son suficientes para ver el orden de crecimiento de funciones polinomiales y no polinomiales.

L	L^2	L^3	L^{100}	2^L	$L!$
2	4	8	1×10^{30}	4	2
5	25	1×10^2	8×10^{69}	32	1×10^2
10	1×10^2	1×10^3	1×10^{100}	1×10^3	4×10^6
50	3×10^3	1×10^5	8×10^{169}	1×10^{15}	3×10^{64}
100	1×10^4	1×10^6	1×10^{200}	1×10^{30}	9×10^{157}

Tabla 3.1: Comparación del orden de crecimiento

Si L representa el tamaño de algún problema, la ventaja de los algoritmos polinomiales podra ser apreciada comparando al crecimiento de L^2 versus 2^L. Si A_1 y A_2 son dos algoritmos para un problema y si además suponemos que la complejidad del algoritmo

A_1 es $O(L^2)$ y la del algoritmo A_2 es $O(2^L)$.

Entonces de acuerdo a la definición 3.11 esto significa que

tiempo de ejecución del algoritmo $A_1 \leqslant c_1 L^2$,

y

tiempo de ejecución del algoritmo $A_2 \leqslant c_2 2^L$.

Luego, para un problema de tamaño $L = 50$, asumiendo $c_1 = c_2 = 1$, el algoritmo A_1requerirá no más de 2500 operaciones y el algoritmo A_2 podria requerir no más de 1×10^{15}.

Existen dos anomalias con el sistema de evaluación del tiempo de ejecución de un algoritmo dado por la definición 3.11.

1°) debemos tratar de obtener cotas polinomiales de grado menor; un algoritmo polinomial $O(L^{100})$ podria no ser de valor más practico que el algoritmo A_2.

2°) las magnitudes de los coeficientes de las cotas polinomiales deben ser relativamente pequeñas.

Si $c_1 = 2^{50}$ y $c_2 = 1$ entonces, aún cuando el tiempo de ejecución del algoritmo A_2 se incrementa con L mucho más rápido que en el algoritmo A_1 desde un punto de vista práctico A_1 no es manejable para algún valor $L \geqslant 1$, no obstante el algoritmo A_2 es manejable para valores pequeños de L.

Afortunadamente, en la práctica esto no suele ocurrir, es decir cuando se encuentra algoritmos polinomiales, esto son de grado menor y con cotas polinomiales relativamente pequeñas.

Las razones por las cuales se insiste en la generación de algoritmos polinomiales son basicamente las siguientes.

Las dos primeras son empíricas:

a) Hasta ahora, los algoritmos polinomiales descubiertos tienen grado y coeficiente razonables.

b) Aun cuando en los casos en que el primer algoritmo polinomial hallado no fue tan eficiente como se esperaba, la experiencia muestra que rapidamente surgiran para el mismo problema algoritmos progresivamente mejores. En el caso de la programación lineal el algoritmo polinomial de Kachyan no muy eficiente fue sucedido en 1984 por el de Karmarkar, que es tambien polinomial aparentemente comparable en eficiencia con el simplex en los problemas reales.

La tercera razón es de índole teórica, y mas importante.

c) Los algoritmos polinomiales poseen ciertas propiedades de "cerradura" que hacen posible el tratamiento matemático. Esto no ocurre con cualquier otro tipo de algoritmo sobre los cuales probablemente no sea posible probar ningún teorema interesante.

d) finalmente, los casos de algoritmos exponenciales exitosos son escasos (uno de ellos lo constituye el algoritmo simplex de programación lineal al que haremos referencia más adelante)

3.3 Análisis en el peor caso y en el caso promedio

Como se observo en la sección 3.2 el estudio de los algoritmos polinomiales es un objetivo básico de la teoría de complejidad computacional. En general, se asocia a los algoritmos polinomiales con una clase de problemas "eficientemente resueltos", es decir, se considera que los algoritmos polinomiales son eficientes y los no polinomiales son ineficientes.

La teoría de complejidad caracteriza la eficiencia de un algoritmo midiendo el tiempo necesario para ejecutar el algoritmo como una función del tamaño de la concretización, que el computador necesita para la ejecución del algoritmo.

Para el estudio de la complejidad se pueden distinguir dos enfoques distintos: un analisis en el peor caso y un análisis en el caso promedio. En esta sección estudiaremos brevemente ambos enfoques.

3.3.1 Analisis en el peor caso

Sea X un problema de optimización determinado y Z el conjunto de sus concretizaciones. Sea $\hat{Z}(L)$ el subconjunto de Z formado por las concretizaciones de Z de tamaño L y con $\hat{Z}_j(L)$ denotamos a un elemento cualquiera de este conjunto. Sea $\tilde{T}_j(L)$ el tiempo de ejecución del algoritmo para el elemento $\hat{Z}_j(L) \in \hat{Z}(L)$. Tenemos la siguiente definición:

Definición 3.13: (Comportamiento en el peor caso).

El comportamiento en el peor caso es definido como el maximo tiempo requerido por cualquier concretización posible, es decir:

$$T(L) := máx\left\{\tilde{T}_j(L) \ / \ j \in J\right\},$$

$T(L)$ se llamara el tiempo de ejecución para un X, concretizaciones de tamaño L, i.e, para concretizaciones en $Z(L)$. Se debe reconocer en la definición del tiempo de ejecución, reside la mayor debilidad de la teoría de conplejidad, por que $T(L)$ es la medida más pesimista posible para el tiempo de ejecución.

3.3.2 Análisis en el caso promedio

Sea X un problema de optimización y Z el conjunto de sus concretizaciones. Sea $\hat{Z}(L)$ el subconjunto de Z formado por las concretizaciones de Z de tamaño L y con $\hat{Z}_j(L)$ denotamos a un elemento cualquiera de este conjunto. Sea $P\left(\hat{Z}_j(L)\right)$ la probabilidad de que $\hat{Z}_j(L)$ ocurra, y sea $\tilde{T}_j(L)$ el tiempo de ejecución del algoritmo para el elemento $\hat{Z}_j(L) \in \hat{Z}(L)$ tenemos la siguiente definición:

Definición 3.14: (Comportamiento en el caso promedio $A(L)$)

El comportamiento en el caso promedio $A(L)$ es definido como el tiempo requerido por todas las concretizaciones de X, es decir:

$$A(L) = \sum_{j \in J} P\left(\hat{Z}_j(L)\right) \tilde{T}_j(L).$$

Sin embargo cabe mencionar que esta cantidad es dificil de calcular para muchos problemas de optimización.

3.3.3 Análisis en el peor caso vs. Análisis en el caso promedio

En el analisis del peor caso, se debe estudiar la concretización más desfavorable para su desempeño de modo que se asegure que, para cualquier oncretización del problema, el tiempo de ejecución está acotado por alguna función de su tamaño. Sin embargo, estas concretizaciones son de rara ocurrencia y no son representativas de una concretización típica del problema. Es decir que en este tipo de análisis no es posible dar una imagen real de cuando un gran porcentaje de concretizaciones de un tamaño dado puede ser rapidamente resuelto y sólo un porcentaje pequeño requiere de considerablemente más tiempo. En estas situaciones, serian preferibles medidas tales como el análisis en el caso promedio.

Pero medidas que requieren una distribución de posibilidades de las concretizaciones, parecen ser más dificiles de analizar, ya que elegir la distribución de posibilidades y medir la probabilidad de ocurrencia de las diversas concretizaciones no es facil de hacer.

La función de distribución de probabilidad, está determinada por la experiencia y/o información especial acerca del problema concreto que se está estudiando.

Además el análisis del caso promedio casualmente es complicado y son necesarias simplificaciones y suposiciones para facilitar el análisis y el tratamiento matemático del comportamiento del algoritmo en promedio, o en relación a sus concretizaciones más típicas.

3.4 Complejidad Algoritmica: El caso de la programación lineal

En esta sección discutiremos brevemente la performance del algoritmo simplex en el peor caso, así como el caso promedio; destacandose tambien el hecho de que el algoritmo

simplex no es polinomial.

Cuando Dantzig presento por primera vez el algoritmo simple, la reacción intuitiva de la comunidad de investigadores fue que este algoritmo no iba a demostrar ser muy eficiente. Por su naturaleza, el método simplex se desliza sobre las aristas de un poliedro y no por el interior del poliedro. De hecho, ocasionalmente en la práctica se observa que este método opera de manera eficiente, para casi todos los problemas prácticos, empiricamente se ha observado que este método lleva a cabo, aproximadamente, $\frac{3m}{2}$ iteraciones, y raras veces más de $3m$ iteraciones, en donde m es el número de restricciones del problema.

Puesto que el algoritmo está efectivamente atrapado en el aspecto potencialmente combinatorio de los $\binom{n}{m}$ vértices que el algoritmo puede recorrer posiblemente como:

$$\binom{n}{m} = \frac{n!}{n!\,(n-m)!} = \left(\frac{n}{m}\right)\left(\frac{n-1}{m-1}\right)\left(\frac{n-2}{m-2}\right)\cdots\left[\frac{n-(m-1)}{m-(m-1)}\right] \geqslant \left(\frac{n}{m}\right)^m$$

En efecto, cada factor es de la forma

$$\frac{n-i}{m-i}; \qquad 0 \leqslant i \leqslant m-1.$$

Como

$$\frac{n-i}{m-i} > \frac{n}{m}; \qquad i = 0, 1, ..., n-1,$$

entonces tenemos

$$\binom{n}{m} = \frac{n!}{n!\,(n-m)!} = \prod_{i=0}^{m-1} \frac{n-i}{m-i} \geqslant \left(\frac{n}{m}\right)^m$$

Si $n \geqslant 2m$, entonces

$$\binom{n}{m} \geqslant \left(\frac{n}{m}\right)^m \geqslant 2^m$$

Luego el algoritmo simplex seria un algoritmo polinomial (Si el número de iteraciones fuera siempre del orden de $O(m)$) ó un algoritmo exponencial (si el número de iteraciones fuera algunas veces $\binom{n}{m}$ y $n \geqslant 2m$).

3.4.1 Comportamiento en el peor caso

Aunque el número de puntos extremos del conjunto factible puede crecer exponencialmente con respecto al número de variables y restricciones, se ha observado en la práctica que el método simplex tiene solamente $O(m)$ pivotes para alcanzar una solución óptima. Sin embargo, desafortunadamente, esta observación práctica no es cierta para todo problema de programacion lineal. A continuación describiremos una familia de problemas para los cuales se requiere un número exponencial de pivotes.

Debemos tener en cuenta que para los problemas no degenerados (Ver [11]) el método simplex se mueve siempre de un vértice a otro vértice adyacente, en cada instante se mejora el valor de la función costo (Ver [11]).

Ahora describimos un poliedro que tiene un número exponencial de vértices, elegimos ir de un vértice a otro adyacente de manera que se tenga un menor costo. Establecido el poliedro, entonces el método simplex (mediante una regla de pivoteamiento que busca esta trayectoria) necesita un número exponencial de pivotes.

Consideremos el siguiente cubo en $\mathbb{R}^n$; definido por las siguientes restricciones

$$0 \leq x_i \leq 1, \ i = 1, ..., n.$$

Este cubo tiene 2^n vértices (para cada i, podemos elegir cualquiera de las dos restricciones $0 \leq x_i$ ó $x_i \leq 1$, la que este activa). Además, existe una trayectoria que recorre siempre todo el contorno del cubo y que llega a cada vértice exactamente una vez; llamaremos a esta trayectoria, *trayectoria extendida.* Esta puede ser construida según el proceso ilustrado en la figura 3.1.

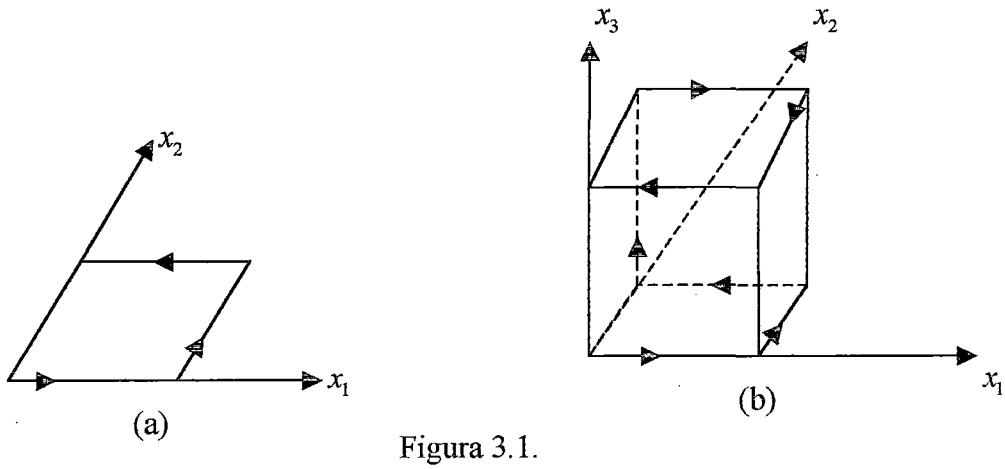

Figura 3.1.

(a) Una trayectoria extendida p_2 en el cubo de dimensión 2.

(b) Una trayectoria extendida p_3 en el cubo de dimensión 3.

Ahora introduciremos la función costo - x_n. La mitad de los vértices del cubo tienen costo cero y la otra mitad tiene un costo de -1. Por lo tanto, el costo no puede decrecer estrictamente para cada traslación a lo largo de la trayectoria extendida.

Si elegimos algún $\varepsilon \in (0, \frac{1}{2})$ y consideramos las perturbaciones del cubo definido por las restricciones,

$$\varepsilon \leq x_1 \leq 1, \tag{3.1}$$

$$\varepsilon x_{i-1} \leq x_i \leq 1 - \varepsilon x_{i-1}, \ i = 2, ..., n. \tag{3.2}$$

Entonces podemos verificar que la función costo decrece estrictamente para cada traslación a lo largo de la trayectoria extendida adecuadamente elegida. Si empezamos el método simplex en el primer vértice de la trayectoria extendida y si nuestra regla de pivoteamiento nos lleva siempre al siguiente vértice sobre la trayectoria, entonces el método simplex requiere $2^n - 1$ pivotes, el siguiente teorema establece esto.

Teorema 3.4.- Consideremos el siguiente problema de programación lineal

$$\begin{aligned} \text{minimizar} \quad & -x_n \\ \text{s.a.} \quad & \\ & \varepsilon \leq x_1 \leq 1, \\ & \varepsilon x_{i-1} \leq x_i \leq 1 - \varepsilon x_{i-1}, \quad i = 2, ..., n. \end{aligned}$$

donde $\varepsilon \in \left(0, \frac{1}{2}\right)$.

Entonces:

(a) El conjunto factible tiene 2^n vértices.

(b) Los vértices pueden ordenarse de manera que sean adyacentes y un vértice tiene un costo menor que el vértice anterior.

(c) Existe una regla de pivoteamiento en la cual el método simplex requiere $2^n - 1$ cambios de base antes de terminar.

Prueba: Ver [12].

3.5 Comportamiento del algoritmo simplex en el caso promedio

A continuación presentaremos, brevemente algunos resultados acerca del compòrtamiento del algoritmo simplex en el caso promedio y que, en cierto modo, validan su efectividad en la práctica.

La evidencia empírica mostraba que un problema de programación lineal con n variables y m restricciones, requeria entre m y $3m$ iteraciones para ser resuelto con el algoritmo simplex. Este número de iteraciones era considerado lo suficientemente bajo para hacer del algoritmo simplex, un algoritmo eficiente y efectivo.

Aún los ejemplos pesimistas de Klee y Minty, no lograron opacar el entusiasmo por el algoritmo simplex. Sin embargo, estudiar el comportamiento del algoritmo simplex en el

caso promedio es difícil, y los resultados obtenidos son influenciados por la accesibilidad y tratabilidad de los instrumentos matematicos apropiados.

En esta sección asumiremos que los problemas de programación lineal están en la forma:

$$\begin{aligned} &\text{maximizar} \quad c^T x \\ &\text{sujeto a} \\ &\qquad Ax \leqslant b, \end{aligned}$$

donde $A \in \mathbb{R}^{m \times n}$. La fila i-esima de A es denotada por a_i^T. Asumiremos tambien que un punto inicial factible x_0, es dado (para ciertos resultados el vector b es escogido como $b = e \equiv (1, ..., 1)^T$ y asi x_0 será automaticamente factible para estos problemas).

El primer teorema que demostró que, en promedio, el algoritmo simplex converge en un número de iteraciones que es polinomial en m y n fue descubierto por Borgward en 1982. Su teorema es presentado a continuación sin prueba. Este teorema asume que los coeficientes en el problema de programación lineal son escogidos aleatoriamente en $\mathbb{R}^n - \{0\}$. El vector del lado derecho b, del problema de programación lineal considerado en el teorema no es aleatorio; es el vector $e = (1, ..., 1)^T$. Notese que no hay restricciones de no negatividad sobre las variables, es decir no se pide que $x \geqslant 0$.

Teorema 3.5: Consideramos un problema de programación lineal de la forma

$$\begin{aligned} &\text{maximizar} \quad c^T x \\ &\text{sujeto a} \\ &\qquad Ax \leqslant e. \end{aligned}$$

donde c y $a_1, ..., a_m$ las columnas de A, son escogidas aleatoriamente y están uniformemente distribuidos en $\mathbb{R}^n - \{0\}$. Si el algoritmo Simplex es aplicado a este problema, el número esperado de iteraciones está acotado por:

$$17n^3 m^{\frac{1}{n-1}} \blacksquare$$

Prueba: ver [11]

Este resultado establece el comportamiento polinomial en el caso promedio del algoritmo simplex, pero la cota obtenida no correspondía a la observada en la práctica, esto es, entre m y $3m$ iteraciones.

Un resultado con una conclusión más satisfactoria fue obtenido independientemente por Haimovich y Adler en 1983. Ellos usan una forma del problema de programación lineal diferente a Borwardt. (el lado derecho b es ahora también aleatorio).

Teorema 3.6: Consideremos un problema de programación lineal de la forma:

$$\begin{aligned} &\text{maximizar} \quad c^T x \\ &\text{sujeto a} \\ &\qquad\qquad Ax \leqslant b. \end{aligned}$$

donde c, b y los coeficientes de A son escogidos aleatoriamente. Si el algoritmo simplex es aplicado a este problema, entonces el número esperado de iteraciones está acotado por:

$$n\left(\frac{m-n+2}{n+1}\right).$$

Prueba: Ver [11]

3.6 Un teorema general de complejidad para los algoritmos de trayectoria central

A continuación estableceremos un teorema que nos da un resultado general de complejidad para los algoritmos de trayectoria central.

Teorema 3.7: Sea $\varepsilon \in (0,1)$. Supongamos que nuestro algoritmo genere una sucesión

de iteraciones que satisfaga

$$\mu_{k+1} \leq (1 - \frac{\delta}{n^w})\mu_k, \ k = 0, 1, 2, ... \tag{3.3}$$

para alguna constante positiva δ y w. Sunpogamos tambien que el punto de partida (x^0, u^0, s^0) satisface

$$\mu_0 \leq \frac{1}{\varepsilon^k} \tag{3.4}$$

para alguna constante positiva k. Entonces existe un índice K con

$$K = O(n^w \, |\log \varepsilon|),$$

tal que

$$\mu_k \leq \varepsilon, \text{ para todo } k \geq K.$$

Prueba:

Aplicando logaritmo a ambos miembros en (3.3), tenemos:

$$\log \mu_{k+1} \leq \log(1 - \frac{\delta}{n^w}) + \log \mu_k,$$

usando nuevamente logaritmo y (3.4), se tiene

$$\log \mu_k \leq k \log(1 - \frac{\delta}{n^w}) + \log \mu_0 \leq k \log(1 - \frac{\delta}{n^w}) + k \log \frac{1}{\varepsilon},$$

al calcular la función logaritmo

$$\log(1 + \beta) \leq \beta, \text{ para todo } \beta > -1$$

tenemos que:

$$\log \mu_k \leq k(\frac{-\delta}{n^w}) + k \log \frac{1}{\varepsilon},$$

por lo tanto, el criterio de convergencia $\mu_k \leq e$ se satisface al tener

$$k(\frac{-\delta}{n^w}) + k \log \frac{1}{\varepsilon} \leq \log \varepsilon.$$

Esta inecuación se cumple para toda k que satisface

$$k \geq K = (1+k)\frac{n^w}{\delta} \log \frac{1}{\varepsilon},$$

asi la prueba queda completa.■

Luego hemos observado en el teorema 3.7 que si la reducción en μ para cada iteración depende de la dimensión n en una trayectoria determinada, y si además la medida de la dualidad inicial no es demasiada larga, entonces el algoritmo tiene complejidad polinomial.

Esto nos será de mucha utilidad al analizar la complejidad de los algoritmos de trayectoria central en el capitulo siguiente.

Capítulo 4

ALGORITMOS DE TRAYECTORIA CENTRAL

En este capitulo, que constituye el capítulo central de nuestro trabajo de tésis, presentamos una familia de algoritmos conocidos en la literatura de la optimización lineal como algoritmos basados en la trayectoria central o de manera abreviada como algoritmos de trayectoria central.

En la sección 4.1 de éste capítulo comentaremos sobre la existencia y unicidad de la trayectoria central, así como analizaremos una medida de proximidad entre dicha trayectoria central y puntos generados por los algoritmos de trayectoria central. En la sección 4.2 estudiaremos una variante de dicho esquema algorítmico conocida como de pasos cortos. En la sección 4.3 describiremos también una variante llamada algoritmo predictor - corrector. En la sección 4.4 describiremos otra variante de dicho algoritmo llamada de pasos largos; y finalmente en la sección 4.5 analizaremos el algoritmo de trayectoria central no factible que se caracteriza, entre otras cosas, por no requerir de un punto inicial (estrictamente factible).

4.1 Algoritmos de trayectoria central

En el capítulo 2 sección 2.1, bajo la condición de que $\mathcal{F}$ es compacto mostramos que para algún $\mu > 0$, el problema (2.3) tenia una única solución óptima sobre $\mathcal{F}^0$. Además, cuando se tiende a cero, la curva formada por toda la minimización es continua en μ y converge a la única solución x^* del problema (2.1).

Ahora en el caso de que el conjunto $\mathcal{F}$ no sea compacto nos definiremos un $\mathcal{R}_\mu$, llamado "conjunto de nivel factible", como sigue:

$$\mathcal{R}_\mu := \{x \in \mathbb{R}^n \ / \ Ax = b;\ x \geq 0,\ f(x,\mu) \leq f(x,\mu)\},$$

y queda demostrado que $\mathcal{R}_\mu$ es compacto, ver [13]; es así que se tiene que el problema de optimización

$$\begin{aligned} &\text{minimizar} \quad f(x,\mu) \\ &\text{sujeto a} \\ &\qquad x \in \mathcal{R}_\mu \end{aligned}$$

tiene una solución (x_μ, u_μ, s_μ). En vista de la real definición del conjunto $\mathcal{R}_\mu$, se deduce que éste vector también es solución del problema de barrera primal (2.3), ver [13].

En éste capítulo estudiaremos los algoritmos de trayectoria central que se caracterizan por seguir esta trayectoria en la dirección en que μ decrece. Los algoritmos siguen la trayectoria en el sentido en que para cada una de sus iteraciones son generados puntos (x^k, u^k, s^k) que no necesariamente pertenecen a la trayectoria, pero que se localizan próximos a ella. Estos puntos son estrictamente positivos y satisfacen las dos primeras ecuaciones de (2.11), en cuanto a la tercera ecuación, no necesariamente es satisfecha de manera exacta. Siendo asi, que para medir la proximidad entre tales puntos y la trayectoria central damos la siguiente definición.

Definición 4.1: Dados $(x, u, s) \in \mathcal{L}^0$ y $\mu > 0$, la proximidad entre los puntos (x, u, s) y (x_μ, u_μ, s_μ) es medida por:

$$\delta(x, s, \mu) := \frac{1}{\mu} \|XSe - \mu e\| \tag{4.1}$$

Los métodos de trayectoria central generan una sucesión de puntos que se situan en una vecindad de la trayectoria caracterizada por δ. Dado $\alpha \in (0, 1)$, definimos esta vecindad como:

$$\mathcal{N}(\alpha) := \bigcup_{\mu \in (0,\infty)} \{(x, u, s) \in \mathcal{L}^0 \ / \ \delta(x, s, \mu) \leq \alpha\} \tag{4.2}$$

Notemos que dado en punto (x, u, s) en la trayectoria central, la relación $XSe = \mu e$ es satisfecha para algún $\mu > 0$, y, consecuentemente, $x^T s = n\mu$. Esta igualdad indica la relación entre los puntos de la trayectoria y de la brecha de dualidad asociado a ellas. Una relación análoga también puede ser obtenida para puntos pertenecientes a la vecindad $\mathcal{N}(\alpha)$. Para estos puntos, la brecha de dualidad dado por $x^T s$ se relaciona con δ por la expresión:

$$x^T s \leq (n + \delta(x, s, \mu)\sqrt{n})\mu \tag{4.3}$$

Para verificar ésta expresión basta multiplicar por e^T, a la relación $\frac{xse}{\mu} = e + \beta$, donde $\|\beta\| = \delta(x, s, \mu)$, y aplicar la desigualdad de Cauchy - Schwartz. La importancia de esta relación está en el hecho de que ella nos dá un buen criterio de parada para los algoritmos.

A continuación presentaremos un método que nos permite inicializar el algoritmo de trayectoria central que encnentra una solución $e-$ óptima después de un número polinomial de iteraciones sin usar números muy grandes ("$bigM$" constante).

4.1.1 Métodos Autoduales

Consideremos el siguiente problema de programación lineal

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

y su respectivo dual

$$\begin{aligned} \text{maximizar} \quad & b^T u \\ \text{sujeto a} \quad & \\ & A^T u + s = c, \\ & s \geq 0. \end{aligned}$$

Dado un punto inicial posiblemente factible (x^0, u^0, s^0) con $x^0 > 0$ y $s^0 > 0$, consideremos el problema:

$$\left.\begin{array}{lllll} \text{minimizar} & ((x^0)^T s^0 + 1)\theta & & & \\ \text{sujeto a} & & & & \\ & Ax - b\tau & +\bar{b}\theta & & = 0 \\ & -A^T u \quad + c\tau & -\bar{c}\theta - s & & = 0 \\ & b^T u - c^T x & +\bar{z}\theta - k & & = 0 \\ & -\bar{b}^T u + \bar{c}x - \bar{z}\tau & & & = -((x^0)^T s^0 + 1) \\ & x \geq 0; \quad \tau \geq 0; & s \geq 0; & k \geq 0, & \end{array}\right\} P_{A-D}$$

donde:

$\bar{b} = b - Ax^0$; $\bar{c} = c - A^T u^0 - s^0$; $\bar{z} = c^T x^0 + 1 - b^T u^0$

P_{A-D} (problema auto - dual).

Vemos que el problema lineal es auto - dual, quiere decir, su dual es equivalente

consigo mismo. Notemos que

$$(x, u, s, \tau, \theta, k) = (x^0, u^0, s^0, 1, 1, 1),$$

es una solución factible interior del problema P_{A-D}. Puesto que tanto su problema primal y su dual son problemas factibles, estos tienen soluciones óptimas. Puesto que el dual es equivalente al primal, el valor óptimo de P_{A-D} es cero.

Establecemos que el algoritmo de trayectoria central encuentra una solución óptima $(x^*, u^*, s^*, \tau^*, \theta^*, k^*)$ que satisface

$$\theta^* = 0,\ x^* + s^* > 0,\ \tau^* + k^* > 0,$$
$$(s^*)^T x^* = 0,\ \tau^* k^* = 0,$$

tal solución satisface la condición conocida como la complementariedad estricta ver [12].

Ahora tenemos el siguiente resultado:

Corolario 4.1: Sea $(x^*, u^*, s^*, \tau^*, \theta^*, k^*)$ una solución óptima del problema P_{A-D} que satisface las siguientes condiciones:

$$\theta^* = 0,\ x^* + s^* > 0,\ \tau^* + k^* > 0,$$
$$(s^*)^T (x^*) = 0,\ \tau^* k^* = 0.$$

a) El problema (2.1) tiene una solución óptima si y sólo sí $\tau^* > 0$. En este caso $\frac{x^*}{\tau^*}$ es una solución óptima del problema (2.1) y $(\frac{u^*}{\tau^*}, \frac{s^*}{\tau^*})$ es una solución óptima del problema (2.2).

b) El problema (2.1) no tiene una solución óptima si y sólo sí $k^* > 0$. En este caso, si $c^T x^* < 0$, entonces el problema primal (2.1) es no acotado y el problema dual (2.2) es no factible; si $-b^T u^* < 0$, entonces el problema dual (2.2) es no acotado y el problema primal (2.1) es no factible; si $c^T x^* < 0$ y $-b^T u^* < 0$, entonces tanto el problema (2.1) como el problema (2.2) son no factibles.

Usando este corolario se tiene que el algoritmo de trayectoria central aplicado al problema P_{A-D}, inicializado con la siguiente solución interior $(x^0, u^0, s^0, 1, 1, 1)$, soluciona adecuadamente los problemas de programación lineal (2.1) y (2.2), esto tiene algunas ventajas:

1°) Esto resuelve, problemas de programación lineal sin la necesidad de asumir la existencia de soluciones factibles, interiores factibles ó soluciones óptimas.

2°) Esto puede empezar en alguna solución factible ó no.

3°) Sus exigencias computacionales por iteración son comparables con otros métodos de trayectoria central.

4°) Esto encuentra una solución $\epsilon-$ óptima en tiempo polinomial usando alguna "*big* M" constante.

4.2 El algoritmo de trayectoria central de pasos cortos

A continuación presentaremos algoritmos que generan puntos siempre en la vecindad $\mathcal{N}(\alpha)$. Estos algoritmos, llamados algoritmos de trayectoria central de pasos cortos, adoptan valores próximos a 1 para τ. De ésta forma, ahora tomamos el paso $(\Delta x, \Delta u, \Delta s)$ a partir de un punto en $\mathcal{N}(\alpha)$, el nuevo punto alcanzado también pertenecerá a la vecindad.

Para entender el comportamiento del algoritmo de trayectoria central de pasos cortos, debemos observar que, si a cada iteración, el punto de partida está en $\mathcal{N}(\alpha)$, entonces el punto (x, u, s) está próximo al punto de la trayectoria central (x_μ, u_μ, s_μ). Ahora tomamos el valor de τ próximo a 1, tenemos $\tau\mu \approx \mu$ y, por tanto, (x, u, s) también estará próximo a $(x_{\tau\mu}, u_{\tau\mu}, s_{\tau\mu})$. Esa proximidad garantiza que la dirección de Newton (2.21) sea una buena aproximación para la dirección que apunta hacia la trayectoria central y,

consecuentemente, que el nuevo punto obtenido pertenezca también a la proximidad de la trayectoria definida por $\mathcal{N}(\alpha)$.

Una vez garantizada que todas las iteraciones del algoritmo pertenecen a $\mathcal{N}(\alpha)$, concluímos que para cada iteración, se generan soluciones viables para los problemas primal y Dual, cuya brecha de dualidad puede ser estimada por (4.3).

A continuación presentamos el algoritmo primal - dual de trayectoria central de pasos cortos que fue introducido por Kojima, Mizumo e Yoshise y Monteiro y Adler independientemente. El análisis de complejidad presentado a continuación se obtiene del segundo grupo de autores. En este análisis verificamos que la complejidad del algoritmo es de $\sqrt{n}\log\frac{1}{\varepsilon}$, donde ε mide la precisión de la solución obtenida.

Algoritmo 4.2: (trayectoria central de pasos cortos)

Dados: $\varepsilon > 0$, $\alpha = 0.4$, $\tau = 1 - \frac{1}{\sqrt{n}}$, $(x^0, u^0, s^0) \in \mathcal{N}(\alpha)$;

$k := 0$,

Repita

hacemos $\tau_k = \tau$, $\mu_k = \dfrac{x^{k^T} s^k}{n}$;

Calcule la dirección de Newton (2.21), osea resuelva el sistema

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta u^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau_k \mu_k e - X^k S^k e \end{bmatrix},$$

Hacemos

$(x^{k+1}, u^{k+1}, s^{k+1}) := (x^k, u^k, s^k) + (\Delta x^k, \Delta u^k, \Delta s^k)$,

$k := k + 1$,

Hasta que

$$\mu_k < \varepsilon.$$

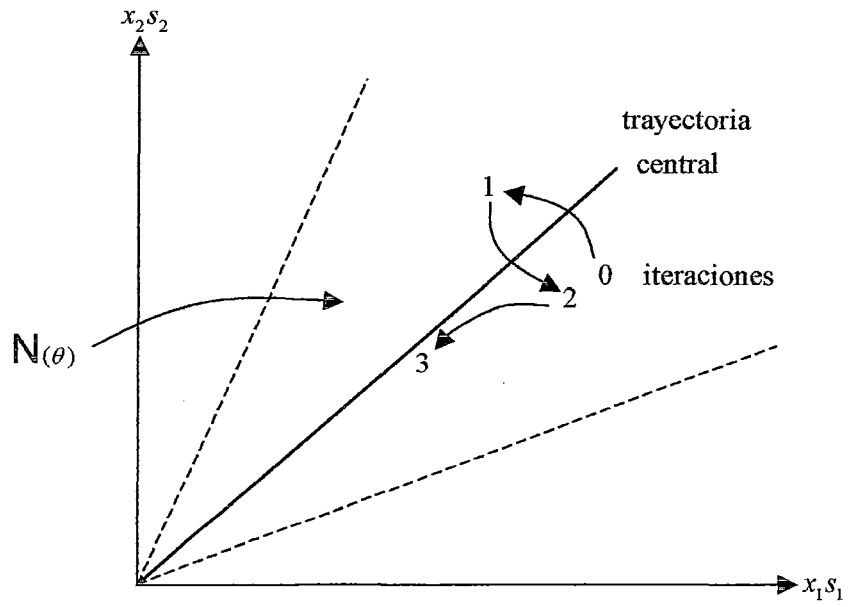

Figura 4.1: iteraciones del algoritmo de pasos cortos graficado en el plano (xs).

En el lema 4.2 verificamos que la medida de la brecha de dualidad, μ, es reducida de forma lineal para cada iteración del algoritmo de trayectoria central de pasos cortos. Este resultado es fundamental para que comprobemos la complejidad polinomial del algoritmo. El resultado del siguiente lema será utilizado en su demostración.

Lema 4.1: La dirección $(\Delta x, \Delta u, \Delta s)$ definida por (2.21) satisface la siguiente ecuación:

$$\Delta x^T \Delta s = 0.$$

Prueba:

De las dos primeras ecuaciones de (2.21) tenemos:

$$\begin{aligned} A^T \Delta u &= -\Delta s, \\ -A^T \Delta u &= \Delta s, \end{aligned}$$

$$A\Delta x = 0.$$

Por lo tanto,

$$\begin{aligned}\Delta x^T \Delta s &= \Delta x^T(-A^T \Delta u), \\ &= -(A\Delta x)^T \Delta u, \\ &= 0.\end{aligned}$$

Lema 4.2: Sea $(x, u, s) \in \mathcal{N}(\alpha)$, $(\Delta x, \Delta u, \Delta s)$ el paso de Newton dado por (2.21), $(\widetilde{x}, \widetilde{u}, \widetilde{s}) := (x, u, s) + \sigma(\Delta x, \Delta u, \Delta s)$, para $\sigma \in [0, 1]$ y $\widetilde{\mu} := \dfrac{\widetilde{x}^T \widetilde{s}}{n}$. Entonces

$$\widetilde{\mu} = (1 - \sigma(1 - \tau))\mu. \tag{4.4}$$

Prueba:

$$\begin{aligned}n\widetilde{\mu} &= \widetilde{x}^T \widetilde{s}, \qquad (4.5)\\ &= (x + \sigma\Delta x)^T (s + \sigma\Delta s), \\ &= x^T s + \sigma\Delta x^T s + \sigma x^T \Delta s + \sigma^2 \Delta x^T \Delta s, \\ &= x^T s + \sigma(\Delta x^T s + x^T \Delta s),\end{aligned}$$

donde en la última igualdad se utiliza el resultado del lema anterior.

De la última ecuación de (2.21), tenemos:

$$S\Delta x + X\Delta s = \tau\mu e - XSe. \tag{4.6}$$

Multiplicando esta expresión por e^T y observando que $\mu e^T e = \mu n$, tenemos:

$$s^T\Delta x + x^T\Delta s = \tau n\mu - x^T s = (\tau - 1)n\mu. \tag{4.7}$$

De (4.4) y (4.7), tenemos que:

$$n\widetilde{\mu} = (1 - \sigma(1 - \tau))n\mu.$$

Finalmente, dividimos esta última ecuación por n, y obtenemos el resultado del lema.■

Con éste resultado que se ha obtenido, comprobamos la reducción lineal del parámetro μ, siempre que un paso sea dado en la dirección de Newton. Esta reducción es proporcional al tamaño de paso, el cual es medido por el parámetro $\sigma \in [0, 1]$. En el caso particular del algoritmo de trayectoria central de pasos cortos, un paso unitario es dao en la dirección de Newton para cada iteración. En este caso, al sustituir el valor $\sigma = 1$ en el resultado del lema 4.2, verificamos fácilmente que la reducción del parámetro μ, para cada iteración del algoritmo, es dada por:

$$\widetilde{\mu} = \tau\mu. \tag{4.8}$$

Teorema 4.1: Supongamos que $(x^0, s^0, \mu_0) \in \mathcal{N}(\alpha)$ y

$$\mu_0 \leq \frac{1}{\varepsilon\beta}, \tag{4.9}$$

para alguna constante positiva β.

Entonces, el algoritmo de trayectoria central de pasos cortos para en $O(\sqrt{n}\log\frac{1}{\varepsilon})$ iteraciones.

Prueba:

Utilizando la expresión (4.8) y el valor de τ en el algoritmo, tenemos:

$$\begin{aligned} \mu_k &= (1 - \frac{1}{\sqrt{n}})\mu_{k-1}, \\ &= (1 - \frac{1}{\sqrt{n}})^k \mu_0. \end{aligned}$$

De esta manera, el algoritmo para cuando k es tal que

$$(1 - \frac{1}{\sqrt{n}})^k \mu_0 < \varepsilon.$$

Aplicando la función logaritmo a esta relación, tenemos

$$k \log(1 - \frac{1}{\sqrt{n}}) + \log \mu_0 < \log \varepsilon.$$

Considerando (4.9), observamos que ésta última desigualdad de satisface si

$$k \log(1 - \frac{1}{\sqrt{n}}) + \beta \log \frac{1}{\varepsilon} < \log \varepsilon,$$

como $\log(1+\gamma) \leq \gamma$, para todo $\gamma > -1$, esta ultima desigualdad se cumple si

$$-k\frac{1}{\sqrt{n}} \leq \log \varepsilon - \beta \log \frac{1}{\varepsilon},$$

o, equivalentemente

$$-k\frac{1}{\sqrt{n}} \leq -(1+\beta) \log \frac{1}{\varepsilon}.$$

El criterio de convergencia del algoritmo, $\mu_k < \varepsilon$, se satisface entonces si

$$k \geq \sqrt{n}(1+\beta)\log\frac{1}{\varepsilon}.\blacksquare$$

Una vez demostrada la complejidad polinomial del algoritmo, nos queda verificar que al tomarse un paso en la dirección de Newton (2.21), a partir de un punto dado $(x, u, s) \in \mathcal{N}(\alpha)$, la nueva iteración $(x,u,s) + (\Delta x, \Delta u, \Delta s)$ será también un punto perteneciente a $\mathcal{N}(\alpha)$.

A continuación formalizaremos una cota superior para la norma $\|\Delta X \Delta S e\|$. Esta cota constituye una parte importante para éste análisis. Para entender tal importancia, debemos recordar que para cada iteración del algoritmo, partimos de un punto (x,u,s) y calculamos una aproximación para el paso que nos llevaria al punto (x_μ, u_μ, s_μ) de la trayectoria central. Esta aproximación queda establecida por la linealización de la tercera ecuación de (2.12), la cual da origen al paso de Newton (2.21). Al realizar esta linealización, el término $\Delta X \Delta S e$ es justamente el término descado. Consecuentemente, la eficiencia del paso de Newton estará asociada al tamaño de su norma. Al acotar la norma garantizamos que la aproximación dada por el paso de Newton, nos lleva a un punto próximo a la trayectoria central, donde ésta aproximidad es medida por la vecindad $\mathcal{N}(\alpha)$.

El siguiente lema formaliza ésta idea.

Lema 4.3: Sea $(x,u,s) \in \mathcal{N}(\alpha)$; $(\widetilde{x},\widetilde{u},\widetilde{s}) := (x,u,s) + \sigma(\Delta x, \Delta u, \Delta s)$ para $\sigma \in [0,1]$, donde $(\Delta x, \Delta u, \Delta s)$ es el paso de Newton dado por (2.21), y $\widetilde{\mu} := \dfrac{\widetilde{x}^T\widetilde{s}}{n}$. Entonces

$$\left\|\widetilde{X}\widetilde{S}e - \widetilde{\mu}e\right\| \leq (1-\sigma)\alpha\mu + \sigma^2\|\Delta X \Delta S e\|.$$

Prueba:

$$\left\|\widetilde{X}\widetilde{S}e - \widetilde{\mu}e\right\| = \|XSe + \sigma S\Delta X e + \sigma X \Delta S e + \sigma^2 \Delta X \Delta S e - \widetilde{\mu}e\|$$

$$
\begin{aligned}
&= \left\|XSe + \sigma(\tau\mu e - XSe) + \sigma^2\Delta X\Delta Se - \widetilde{\mu}e\right\| && (4.10)\\
&= \left\|XSe + \sigma(\tau\mu e - XSe) + \sigma^2\Delta X\Delta Se - (1-\sigma(1-\tau))\mu e\right\| && (4.11)\\
&= \left\|(1-\sigma)(XSe - \mu e) + \sigma^2\Delta X\Delta Se\right\| \\
&\leq (1-\sigma)\left\|XSe - \mu e\right\| + \sigma^2\left\|\Delta X\Delta Se\right\| \\
&\leq (1-\sigma)\alpha\mu + \sigma^2\left\|\Delta X\Delta Se\right\| && (4.12)
\end{aligned}
$$

donde (4.10) utiliza la tercera ecuación de (2.21), (4.11) utiliza el resultado del lema 4.2 y (4.12) se obtiene del hecho que $(x, u, s) \in \mathcal{N}(\alpha)$.■

Ahora pasaremos a verificar que la medida $\|\Delta X\Delta Se\|$ es pequeña y suficiente para garantizar que el punto $(\widetilde{x}, \widetilde{u}, \widetilde{s}) \in \mathcal{N}(\alpha)$. Este resultado se demostrará en el teorema 4.4. A continuación presentaremos una serie de lemas que nos servirán para la demostración de dicho teorema.

Lema 4.4: Sea $u, v \in \mathbb{R}^n$ tales que $u^T v = 0$, entonces

$$\|UVe\| \leq \frac{1}{\sqrt{8}}\|u+v\|^2.$$

Prueba:

Consideremos la igualdad

$$u_i v_i = \frac{1}{4}((u_i+v_i)^2 - (u_i - v_i)^2),$$

para todo $i = 1, ..., p$, tenemos:

$$
\begin{aligned}
\|UVe\|^2 &= \sum_{i=1}^{p}(u_i v_i)^2 \\
&= \frac{1}{16}\sum_{i=1}^{p}((u_i+v_i)^2 - (u_i-v_i)^2)^2
\end{aligned}
$$

$$\leq \frac{1}{16}\sum_{i=1}^{p}((u_i+v_i)^4+(u_i-v_i)^4) \tag{4.13}$$

$$\leq \frac{1}{16}\left[(\sum_{i=1}^{p}(u_i+v_i)^2)^2+(\sum_{i=1}^{p}(u_i-v_i)^2)^2\right] \tag{4.14}$$

$$= \frac{1}{16}(\|u+v\|^4+\|u-v\|^4)$$

$$= \frac{1}{16}(2\|u+v\|^4) \tag{4.15}$$

$$= \frac{1}{8}(\|u+v\|^4) \tag{4.16}$$

donde:

(4.13) se verifica puesto que si tenemos que $a,\ b \geq 0$ entonces $(a-b)^2 \leq a^2+b^2$; (4.14) utiliza la relación

$$\sum_{i=1}^{p} a_i^2 \leq (\sum_{i=1}^{n} a_i)^2,$$

para $a_i \geq 0$, $i = 1, ..., p$, el cual es un caso particular de la desigualdad de Holder.

(4.15) se verifica porque u y v son ortogonales ($u^Tv = 0$), y por lo tanto $\|u+v\| = \|u-v\|$.■

Lema 4.5: Sea $(x, u, s) \in \mathcal{N}(\alpha)$ y $(\Delta x, \Delta u, \Delta s)$, el paso de Newton dado por (2.21). Entonces

$$\|\Delta X \Delta S e\| \leq \frac{\alpha^2+n(1-\tau)^2}{\sqrt{8}(1-\alpha)}\mu.$$

Prueba:

Iniciamos la demostración haciendo un cambio de escala en la tercera ecuación de (2.21), con la intención de escribirla de forma mas conveniente.

El cambio de escala se obtiene al multiplicar la ecuación por la matriz $(XS)^{-1/2}$. Considerando D, la matriz diagonal cuyos elementos de la diagonal son dados por $\sqrt{\frac{x_i}{s_i}}$

para todo $i = 1, ..., n$, osea, $D = S^{-1/2}X^{1/2}$, el resultado de éste producto puede ser escrito como

$$D^{-1}\Delta X + D\Delta S = (XS)^{-1/2}(-XSe + \tau\mu e) \tag{4.17}$$

observando que $\Delta X\Delta S = (D\Delta X)(D^{-1}\Delta S)$ y utilizando (4.17) y los resultados de los lemas 4.1 y 4.4, tenemos:

$$\begin{aligned}
\|\Delta X\Delta Se\| &= \left\|(D\Delta X)(D^{-1}\Delta S)e\right\| \\
&\leq \frac{1}{\sqrt{8}}\left\|D\Delta X + D^{-1}\Delta S\right\|^2 \\
&= \frac{1}{\sqrt{8}}\left\|(XS)^{-1/2}(-XSe + \tau\mu e)\right\|^2 \\
&= \frac{1}{\sqrt{8}}\sum_{i=1}^{n}\frac{(-x_is_i + \tau\mu)^2}{x_is_i} \\
&\leq \frac{\sum_{i=1}^{n}(-x_is_i + \tau\mu)^2}{\sqrt{8}(1-\alpha)\mu}
\end{aligned}$$

$$\|\Delta X\Delta Se\| = \frac{\|XSe - \tau\mu e\|^2}{\sqrt{8}(1-\alpha)\mu}, \tag{4.18}$$

donde la última desigualdad se cumple porque $(x, u, s) \in \mathcal{N}(\alpha)$.

Consecuentemente, $|x_is_i - \mu| \leq \alpha\mu$, y $x_is_i \geq (1-\alpha)\mu$, para todo $i = 1, ..., n$.

Ahora tenemos que:

$$\begin{aligned}
\|XSe - \tau\mu e\|^2 &= \|(XSe - \mu e) + (1-\tau)\mu e\|^2 \\
&= \|XSe - \mu e\|^2 + 2(1-\tau)\mu e^T(XSe - \mu e) + (1-\tau)^2\mu^2 e^T e \\
&= \|XSe - \mu e\|^2 + 2(1-\tau)\mu(x^Ts - \mu n) + (1-\tau)^2\mu^2 n \\
&= \|XSe - \mu e\|^2 + (1-\tau)^2\mu^2 n
\end{aligned}$$

$$\|XSe - \tau\mu e\|^2 \leq \alpha^2\mu^2 + (1-\tau)^2\mu^2 n. \tag{4.19}$$

donde la última igualdad se cumple porque $\mu = \dfrac{x^T s}{n}$, y la última desigualdad se cumple porque $(x, u, s) \in \mathcal{N}(\alpha)$. Sustituyendo (4.19) en (4.18) obtenemos el resultado del lema.■

Lema 4.6: Sea $(x, u, s) \in \mathcal{N}(\alpha)$; $(\widetilde{x}, \widetilde{u}, \widetilde{s}) := (x, u, s) + \sigma(\Delta x, \Delta u, \Delta s)$ para $\sigma \in [0, 1]$, donde $(\Delta x, \Delta u, \Delta s)$ es el paso de Newton dado por (2.21), y $\widetilde{\mu} := \dfrac{\widetilde{x}^T\widetilde{s}}{n}$. Entonces, considerando que α y τ toman valores dados en el algoritmo 4.2, tenemos:

$$\left\|\widetilde{X}\widetilde{S}e - \widetilde{\mu}e\right\| \leq \alpha\widetilde{\mu}.$$

Prueba:

$$\begin{aligned}
\left\|\widetilde{X}\widetilde{S}e - \widetilde{\mu}e\right\| &\leq (1-\sigma)\alpha\mu + \sigma^2 \|\Delta X \Delta S e\| && (4.20)\\
&\leq (1-\sigma)\alpha\mu + \sigma^2 \tfrac{\alpha^2 + n(1-\tau)^2}{\sqrt{8}(1-\alpha)}\mu && (4.21)\\
&\leq (1-\sigma)\alpha\mu + \sigma^2\tau\alpha\mu && (4.22)\\
&\leq (1-\sigma+\sigma\tau)\alpha\mu && (4.23)\\
&= \alpha\widetilde{\mu}. && (4.24)
\end{aligned}$$

donde:

(4.20) usa el resultado del lema 4.3,

(4.21) usa el resultado del lema 4.5,

(4.22) se puede verificar fácilmente cuando α y τ toman los valores dados en el algoritmo 4.2,

(4.23) se verifica porque $\sigma \in [0, 1]$ y

(4.24) utiliza el resultado del lema 4.2.■

Teorema 4.2: Sea $(x, u, s) \in \mathcal{N}(\alpha)$; y $(\Delta x, \Delta u, \Delta s)$, el paso de Newton dado por (2.21). Entonces

$$(\widetilde{x}, \widetilde{u}, \widetilde{s}) := (x, u, s) + \sigma(\Delta x, \Delta u, \Delta s) \in \mathcal{N}(\alpha)$$

para todo $\sigma \in [0, 1]$.

Prueba:

El resultado del lema 4.6 indica que $\delta(\widetilde{x}, \widetilde{s}, \widetilde{\mu}) \leq \alpha$. Luego nos queda solo demostrar que $(\widetilde{x}, \widetilde{s}, \widetilde{\mu}) \in \mathcal{L}^0$.

Las restricciones $A\widetilde{x} = b$ y $A^T\widetilde{u} + \widetilde{s} = c$ son fácilmente verificables ya que, de (2.21), tenemos que $A\Delta x = 0$ y $A^T\Delta u - \Delta s = 0$, y finalmente, la no negatividad de $\widetilde{x}$ y $\widetilde{s}$ es garantizada al considerarse que si $\delta(\widetilde{x}, \widetilde{s}, \widetilde{\mu}) \leq \alpha$, entonces $|\widetilde{x}_i\widetilde{s}_i - \widetilde{\mu}| \leq \alpha\widetilde{\mu}$, osea, $\widetilde{x}_i\widetilde{s}_i \geq (1-\alpha)\widetilde{\mu} = (1-\alpha)(1-\sigma(1-\tau))\mu > 0$, puesto que $\alpha, \sigma, \tau \in [0,1]$. Por lo tanto, $(\widetilde{x},\ \widetilde{s}) > 0$ para todo $\sigma \in [0,1]$.■

4.3 El algoritmo predictor - corrector

En (2.21) definimos la dirección de Newton para $\tau \in [0,1]$. Observamos que cuando τ toma los valores extremos 0 y 1, la dirección de Newton es conocida respectivamente como dirección escala afin y dirección de centralización.

La primera apunta aproximadamente para el óptimo de los problemas primal y dual, pero al caminar en ella es probable que nos aproximemos de manera rápida a la frontera del octante no negativo. Realmente, si tomamos un paso unitario en la dirección escala afin a partir de un punto en $\mathcal{N}(\alpha)$, el nuevo punto puede que no satisfaga las restricciones de no negatividad de las variables x y s. Por otro lado, si tomamos un paso unitario en la dirección de centralización a partir de un punto en $\mathcal{N}(\alpha)$, tenemos la garantia de que el nuevo punto también pertenecerá a $\mathcal{N}(\alpha)$, osea, el será viable y estará lejos de la frontera del octante. En tanto, este paso mantiene aproximadamente constante a la brecha de dualidad.

En el algoritmo de trayectoria central de pasos cortos le dimos a τ un valor menor que 1, caso contrario podria perder la garantia dada por la dirección de centralización. Osea, continuamos exigiendo que al tomar un paso unitario en la dirección de Newton a partir de un punto en $\mathcal{N}(\alpha)$, el nuevo punto tambien pertenece a la vecindad. Esta exigencia hace que el valor de τ continue muy próximo a 1, y consecuentemente, la convergencia del algoritmo en dirección al óptimo del problema sea, en general, lenta.

Mejores resultados se han obtenido en la práctica con la aplicación del llamado algoritmo predictor - corrector. En éste algoritmo trabajamos con dos vecindades de trayectoria central definidas por (4.2) para dos diferentes valores de α. Utilizaremos $\alpha_1 = 0.25$ y $\alpha_2 = 0.5$, de manera que la primera vecindad es un subconjunto de la segunda. Entonces utilizaremos dos diferentes tipos de iteraciones, de la siguiente forma:

Iteraciones Impares: Constituyen el llamado paso corrector en el cual partimos de un punto en $\mathcal{N}(\alpha_2)$ y tomamos un paso unitario en la dirección de centralización. Es posible mostrar que el nuevo punto obtenido pertenece a $\mathcal{N}(\alpha_1)$. Este paso tiene como objetivo entonces, centralizar.

Iteraciones pares: Constituyen el llamado paso predictor en el cual partimos de un punto en $\mathcal{N}(\alpha_1)$ y caminamos en la máxima dirección escala - afin, sin salirnos de la vecindad $\mathcal{N}(\alpha_2)$ especialmente, partimos del punto $(x, u, s) \in \mathcal{N}(\alpha_1)$ y tomamos el nuevo punto $(\widetilde{x}, \widetilde{u}, \widetilde{s}) := (x, u, s) + \sigma(\Delta x, \Delta u, \Delta s)$, donde $(\Delta x, \Delta u, \Delta s)$ es el paso de Newton (2.14) calculado para $\tau = 0$ y $\sigma \in [0, 1]$ asumiendo el mayor valor para el cual $(\widetilde{x}, \widetilde{u}, \widetilde{s}) \in \mathcal{N}(\alpha_2)$. Este paso tiene como objetivo, disminuir la brecha de dualidad.

El algoritmo predictor corrector mantiene la complejidad de $O(\sqrt{n} \log \frac{1}{\varepsilon})$ del algoritmo de trayectoria central de pasos cortos.

Algoritmo (4.3) predictor - corrector

Dados $\varepsilon > 0$, $\alpha = 0.25$, $\tau = 1 - \frac{1}{\sqrt{n}}$, $(x^0, u^0, s^0) \in \mathcal{N}(0.25)$

$k := 0$,

Repita

Hacemos

$\tau_k = \tau$, $\mu_k = \dfrac{X^{k^T} S^k}{n}$

(paso predictor)

Hacemos

$\tau_k = 0$ y hallamos una solución $(\Delta x^k, \Delta u^k, \Delta s^k)$ del sistema lineal

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta u^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k S^k e \end{bmatrix};$$

y sea $\tau_k = 1 - \dfrac{1}{\sqrt{n}}$ tal que:

$$(x^k, u^k, s^k) + \tau(\Delta x^k, \Delta u^k, \Delta s^k) \in \mathcal{N}(0.5)$$

Por otro lado (paso corrector)

Hacemos $\tau_k = 1$ y hallamos una solución $(\Delta x^k, \Delta u^k, \Delta s^k)$ del sistema lineal

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta u^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k S^k e + \mu_k e \end{bmatrix}$$

Hacemos

$$(x^{k+1}, u^{k+1}, s^{k+1}) := (x^k, u^k, s^k) + \tau_k(\Delta x^k, \Delta u^k, \Delta s^k)$$

$k := k + 1$

hasta que

$\mu_k < \varepsilon$.

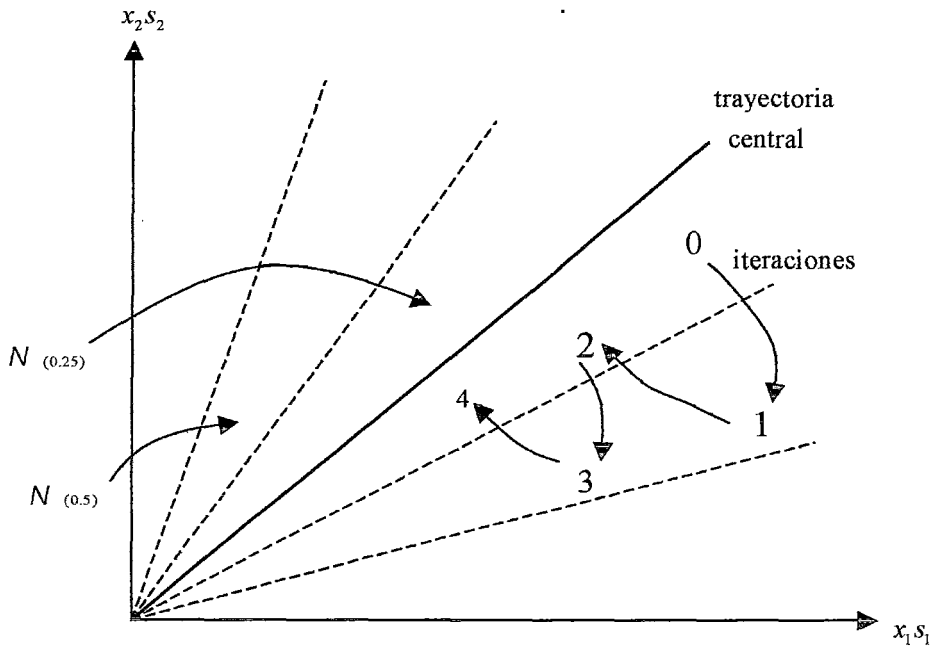

Figura 4.2 : *iteraciones del algoritmo predictor corrector graficado en el plano* (xs).

Nuestro análisis del algoritmo predictor corrector es breve porque la mayor parte del trabajo ya se ha realizado en el análisis del algoritmo de pasos cortos.

El comportamiento de cada paso predictor es descrito en el siguiente lema, el cual halla una cota inferior sobre la longitud de sus pasos y por lo tanto una estimación de la reducción en μ.

Lema 4.7: supongamos que $(x,\ u,\ s) \in \mathcal{N}(0.25)$ y sea $(\Delta x,\ \Delta u,\ \Delta s)$ el paso de newton calculado en (2.21) cuando $\tau = 0$. Entonces $(\widetilde{x},\ \widetilde{u},\ \widetilde{s}) \in \mathcal{N}(0.5)$ para todo $\sigma \in [0, \overline{\sigma}]$ donde

$$\overline{\sigma} = \min\left(\frac{1}{2}, \left(\frac{\mu}{8\,\|\Delta x \Delta S e\|}\right)^{1/2}\right), \tag{4.25}$$

por lo tanto, el paso predictor tiene una longitud menor que $\overline{\sigma}$ y el nuevo valor de μ es a lo más $(1-\overline{\sigma})\mu$.

Prueba:

De (4.12), tenemos

$$\begin{aligned}
\left\|\widetilde{X}\widetilde{S}e-\widetilde{\mu}e\right\| &\leq (1-\sigma)\left\|XSe-\mu e\right\|+\sigma^2\left\|\Delta X\Delta Se\right\| \\
&\leq (1-\sigma)\left\|XSe-\mu e\right\|+\frac{\mu}{8\left\|\Delta x\Delta Se\right\|}\left\|\Delta X\Delta Se\right\| \quad \text{de (4.25)}, \\
&\leq \frac{1}{4}(1-\sigma)\mu+\frac{1}{8(1-\sigma)}(1-\sigma)\mu \quad \text{puesto que } (x,s,u)\in\mathcal{N}(0.25), \\
&\leq \frac{1}{4}(1-\sigma)\mu+\frac{1}{4}(1-\sigma)\mu \quad \text{puesto que } \sigma\leq\frac{1}{2}, \\
&\leq \frac{1}{2}\widetilde{\mu} \qquad \text{por (4.4), con } \tau=0.
\end{aligned}$$

Por lo tanto, el punto $(\widetilde{x},\ \widetilde{u},\ \widetilde{s})$ satisface la condición de proximidad para $\mathcal{N}(0.5)$.

Ahora para verificar que $(\widetilde{x},\ \widetilde{u},\ \widetilde{s})\in\mathcal{L}_0$ bastará revisar la prueba del teorema 4.2.■

El lema 4.5 puede usarse para hallar una cota inferior sobre $\overline{\sigma}$. Hacemos $\alpha=0.25$ y $\tau=0$ en dicho lema, obteniendo

$$\frac{\mu}{8\left\|\Delta x\Delta Se\right\|}\geq\frac{2^{3/2}(1-0.25)}{8((0.25)^2+n)}=\frac{3\sqrt{2}}{1+16n}\geq\frac{0.16}{n},$$

puesto que $n\geq 1$.

Por lo tanto, de (4.25) tenemos

$$\overline{\sigma}\geq\min\left(\frac{1}{2},\left(\frac{0.16}{n}\right)^{1/2}\right)=\frac{0.4}{\sqrt{n}}.$$

Dado que los pasos predictor son elegidos en incluso un índice de iteraciones, esta cota implica que

$$\mu_{k+1}\leq(1-\frac{0.4}{\sqrt{n}})\mu_k,\ k=0,2,4,... \tag{4.26}$$

Los pasos corrector son descritos en el siguiente lema, el cual nos muestra que ellos retornan a algun punto en $\mathcal{N}(0.5)$ al interior de la vecindad $\mathcal{N}(0.25)$ sin cambiar el valor

de μ.

Lema 4.8: supongamos que $(x,\ u,\ s) \in \mathcal{N}(0.5)$ y sea $(\Delta x,\ \Delta u,\ \Delta s)$ el paso de newton calculado en (2.21) cuando $\sigma = 1$, entonces tenemos que

$$(\widetilde{x}_1, \widetilde{u}_1, \widetilde{s}_1) := (x, u, s) + (\Delta x, \Delta u, \Delta s) \in \mathcal{N}(0.25),$$

$\widetilde{\mu}_1 = \mu$.

Prueba:

sustituyendo $\tau = 1$ en (4.4), obtenemos inmediatamente que $\widetilde{\mu}_1 = \mu$ (de hecho, $\widetilde{\mu} = \mu$ para todo $\sigma \in [0, 1]$).

Ahora combinando (4.12) y el lema 4.5 tenemos

$$\left\| \widetilde{X}\widetilde{S}e - \widetilde{\mu}e \right\| \leq (1-\sigma)\alpha\mu + \sigma^2 \left[\frac{\alpha^2 + n(1-\tau)^2}{\sqrt{8}(1-\alpha)} \right] \mu, \tag{4.27}$$

sustituyendo $\alpha = 0.5$, $\sigma = 1$ y $\tau = 1$ en (4.27) encontramos que

$$\left\| \widetilde{X}_1\widetilde{S}_1 e - \widetilde{\mu}_1 e \right\| \leq \frac{\mu}{4} = \frac{\widetilde{\mu}_1}{4},$$

por lo tanto, $(\widetilde{x}_1, \widetilde{u}_1, \widetilde{s}_1)$ satisface las condiciones de proximidad para $\mathcal{N}(0.25)$. Finalmente para completar la prueba sera suficiente verificar que $(\widetilde{x}_1, \widetilde{u}_1, \widetilde{s}_1)$ es estrictamente factible también, pero esta prueba es idéntica a la prueba desarrollada en el teorema 4.2 de esta tésis.■

Como podemos ver de este lema las iteraciones corrector parten del valor de la medida dual μ inalterable. Asi también, ya que las iteraciones predictor logran una reducción sustancial en $\mu(4.26)$, puede probarse que se obtiene la misma clase de complejidad polinomial como para los algoritmos de pasos cortos.

Teorema 4.3: Dado $\varepsilon > 0$, supóngase que el punto inicial $(x^0, u^0, s^0) \in \mathcal{N}(0.25)$ en el

algoritmo predictor corrector tenemos

$$\mu_0 \leq \frac{1}{\varepsilon^k},$$

para algún constante positiva k. Entonces existe un índice K con $K = O\left(\sqrt{n}\log\frac{1}{\varepsilon}\right)$ tal que

$$\mu_k \leq \varepsilon, \text{ para todo } k \geq K.$$

Prueba:

Combinando (4.26) con el lema 4.8, tenemos

$$\mu_{k+2} = \mu_{k+1} \leq \left(1 - \frac{0.4}{\sqrt{n}}\right)\mu_k; \; k = 0, 2, 4, ...,$$

por lo tanto, la reducción exigida (3.3) del teorema 3.7 es casi satisfecha cuando hacemos $\delta = 0.4$ y $w = 0.5$, excepto que la reducción en μ ocurre sobre un intervalo de dos iteraciones, en lugar de exactamente una. Pero la prueba del teorema 3.7 puede adaptarse fácilmente usando esta condición poco diferente en μ sin afectar la conclusión del teorema.■

4.4 El algoritmo de trayectoria central de pasos largos

La utilización de la norma euclidiana en la definición de δ en (4.1) restringe bastante el conjunto de puntos que pertenecen a la vecidad $\mathcal{N}(\alpha)$. De la misma forma para los valores de α próximos a 1, el conjunto de puntos en $\mathcal{L}^0$ es bastante mayor que el conjunto de puntos en $\mathcal{N}(\alpha)$. Como resultado, los algoritmos que permanezcan siempre en esta vecindad, pueden caminar poco en cada iteración en dirección al óptimo.

Algunos algoritmos prácticos utilizan otras normas para definir vecindades de la tra-

yectoria central mucho más grandes. Las dos vecindades mas interesantes son:

$$\mathcal{N}_\infty(\gamma) := \bigcup_{\mu\in(0,\infty)} \left\{(x,u,s) \in \mathcal{L}^0 \;/\; \delta_\infty(x,s,\mu) \leq \gamma\right\},$$

donde

$$\delta_\infty(x,s,\mu) := \frac{1}{\mu}\left\|XSe - \mu e\right\|_\infty,$$

y

$$\mathcal{N}_{-\infty}(\gamma) := \bigcup_{\mu\in(0,\infty)} \left\{(x,u,s) \in \mathcal{L}^0 \;/\; \delta_{-\infty}(x,s,\mu) \leq \gamma\right\},$$

donde

$$\delta_{-\infty}(x,s,\mu) := \frac{1}{\mu}\left\|XSe - \mu e\right\|_{-\infty}, \tag{4.28}$$

y dado $v \in \mathbb{R}^n$, tenemos por definición que $\|v\|_{-\infty} \leq \beta$ si y solamente si $v_i \geq -\beta$ para todo $i = 1, ..., n$.

Observamos fácilmente que para un γ dado estas dos vecindades contienen un número de puntos viables de los problemas primal y dual considerablemente mayor que $\mathcal{N}(\alpha)$. La vecindad $\mathcal{N}_{-\infty}(\gamma)$, en particular, se torna bien próxima de $\mathcal{L}^0$ a medida que γ tiende a 1.

Los algoritmos de trayectoria central de pasos largos utilizan estas vecindades mucho más grandes. En ellas, el parámetro τ recibe un valor menor del que habia recibido en el algoritmo de trayectoria central de pasos cortos, por lo tanto el decrecimiento de la brecha de dualidad es más rápido en cada iteración.

Como ellos tienen mas espacio para trabajar, el progreso en dirección al óptimo es más rápido. Ahora tomamos un valor pequeño para τ, aunque no garantizamos que el punto resultante de un paso unitario en la dirección de Newton (2.21) continue perteneciendo a la vecindad de la trayectoria. En este caso, utilizamos un procedimiento parecido al descrito en el paso predictor del algoritmo predictor - corrector, en el cual realizamos una

busqueda lineal que establece el mayor paso permitido en la dirección de Newton, para que el nuevo punto alcanzado permanezca en la vecindad. Los algoritmos de trayectoria central de pasos largos tienen una mayor perfomance en la práctica que los algoritmos de pasos cortos.

Su complejidad aumenta para $O(n \log \frac{1}{\varepsilon})$ iteraciones.

Algoritmo 4.4 (*trayectoria central de pasos largos*)

Dados $\gamma, \tau_{\min}, \tau_{\max}$ con $\gamma \in (0,1)$, $0 < \tau_{\min} < \tau_{\max} < 1$, y $(x^0, u^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$

$k := 0$,

Repita

elegimos

$\tau_k \in [\tau_{\min}, \tau_{\max}]$

hallamos una solución $(\Delta x^k, \Delta u^k, \Delta s^k)$ del sistema lineal

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta u^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau_k \mu_k e - X^k S^k e \end{bmatrix},$$

elegimos σ_k como el valor mas grande de σ en $[0,1]$ tal que

$$(\widetilde{x}^k,\ \widetilde{u}^k,\ \widetilde{s}^k) \in \mathcal{N}_{-\infty}(\gamma).$$

Hacemos

$$(x^{k+1},\ u^{k+1},\ s^{k+1}) = (\widetilde{x}_k^k,\ \widetilde{u}_k^k,\ \widetilde{s}_k^k)$$

$k := k + 1$

hasta que

$\mu_k < \varepsilon$.

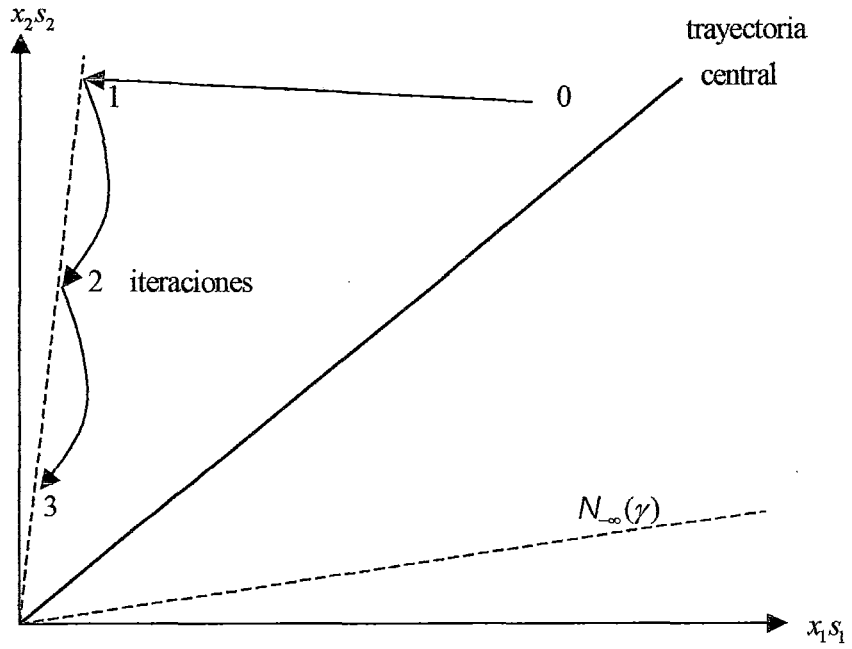

Figura 4.3: Iteraciones del algoritmo de pasos largos graficado en el plano (xs).

La típica conducta del algoritmo es ilustrada en la figura 4.3 como muestra la figura (hecho que se confirmará en el análisis) la cota mínima τ min sobre el parámetro central garantiza que cada búsqueda de dirección empieza a desplazarse fuera de la frontera de $\mathcal{N}_{-\infty}(\gamma)$ y dentro del interior de su vecindad.

El lema 4.9 y el teorema 4.4 nos muestran una cota inferior sobre σ_k y una correspondiente estimación de la reducción en μ para cada iteración. El teorema 4.5 establece el usual resultado de complejidad polinomial.

Lema 4.9: Si $(x, u, s) \in \mathcal{N}_{-\infty}(\gamma)$, entonces

$$\|\Delta X \Delta S e\| \leq 2^{-3/2}(1 + 1/\gamma)n\mu.$$

Prueba:

Como en la prueba del lema 4.5, tenemos que

$$\|\Delta X \Delta S e\| \leq 2^{-3/2} \left\|(XS)^{-1/2}(-XSe + \tau\mu e)\right\|^2$$

desarrollando, el cuadrado de la norma euclidiana y usando las siguientes relaciones $x^T s = n\mu$ y $e^T e = n$, obtenemos,

$$\begin{aligned}
\|\Delta X \Delta S e\| &\leq 2^{-3/2} \left\|-(XS)^{1/2} + \tau\mu(XS)^{-1/2}e)\right\|^2 \\
&\leq 2^{-3/2} \left[X^T S - 2\tau\mu e^T e + \tau^2\mu^2 \sum_{i=1}^{n} \frac{1}{X_i S_i}\right] \\
&\leq 2^{-3/2} \left[X^T S - 2\tau\mu e^T e + \tau^2\mu^2 \frac{n}{\gamma\mu}\right] \text{ puesto que } X_i S_i \geq \gamma\mu \\
&\leq 2^{-3/2} \left[1 - 2\tau + \frac{\tau^2}{\gamma}\right] n\mu \\
&\leq 2^{-3/2}(1 + 1/\gamma)n\mu,
\end{aligned}$$

así la prueba queda completa.

Teorema 4.4: Dados los parámetros $\gamma, \tau_{\min}$ y $\tau_{\max}$ en el algoritmo de pasos largos, existe una constante δ independiente de n tal que

$$\mu_{k+1} \leq (1 - \frac{\delta}{n})\mu_k$$

para todo $k \geq 0$.

Prueba:

Empezaremos por probar que

$$\left(\widetilde{X}^k, \widetilde{U}^k, \widetilde{S}^k\right) \in \mathcal{N}_{-\infty}(\gamma) \text{ para todo } \sigma \in \left[0,\ 2^{3/2}\gamma\frac{1-\gamma}{1+\gamma}\frac{\tau_k}{n}\right], \tag{4.29}$$

además deduciremos que σ_k está acotado inferiormente como sigue:

$$\sigma_k \geq 2^{3/2}\frac{\tau_k}{n}\gamma\frac{1-\gamma}{1+\gamma}. \tag{4.30}$$

Para algún $i = 1, 2, ..., n$ tenemos del lema 4.9 que

$$\left|\Delta x_i^k \Delta s_i^k\right| \leq \left\|\Delta X^k \Delta S^k e\right\|_2 \leq 2^{-3/2}(1+1/\gamma)n\mu_k. \tag{4.31}$$

Usando (4.6) tenemos de $x_i^k s_i^k \geq \gamma\mu_k$ y de (4.31) que

$$\begin{aligned}
\widetilde{X}_i^k, \widetilde{S}_i^k &= (x_i^k + \sigma\Delta x_i^k)(s_i^k + \sigma\Delta s_i^k) \\
&= x_i^k s_i^k + \sigma(x_i^k \Delta s_i^k + s_i^k \Delta x_i^k) + \sigma^2 \Delta x_i^k \Delta s_i^k \\
&\geq x_i^k s_i^k(1-\sigma) + \sigma\tau_k\mu_k - \sigma^2\left|\Delta x_i^k \Delta s_i^k\right| \\
&\geq \gamma(1-\sigma)\mu_k + \sigma\tau_k\mu_k - \sigma^2 2^{-3/2}(1+1/\gamma)n\mu_k.
\end{aligned}$$

Por otra parte de (4.4) tenemos que

$$\widetilde{\mu}_k = (1-\sigma(1-\tau_k))\mu_k.$$

De estas últimas dos fórmulas, podemos ver que la condición de proximidad

$$\widetilde{X}_i^k, \widetilde{S}_i^k \geq \gamma\widetilde{\mu}_k,$$

se satisface, siempre que

$$\gamma(1-\sigma)\mu_k + \sigma\tau_k\mu_k - \sigma^2 2^{-3/2}(1+1/\gamma)^{n\mu_k} \geq \gamma(1-\sigma+\sigma\tau_k)\mu_k,$$

ordenando la expresión, obtenemos

$$\sigma\tau_k\mu_k(1-\gamma) \geq \sigma^2 2^{-3/2} n\mu_k(1+1/\gamma),$$

la cual es verdadera si

$$\sigma \leq \frac{2^{3/2}}{n}\tau_k\gamma\frac{1-\gamma}{1+\gamma},$$

luego queda probado que $(\widetilde{X}^k, \widetilde{U}^k, \widetilde{S}^k)$ satisface las condiciones de proximidad para $\mathcal{N}_{-\infty}(\gamma)$ cuando σ se mantiene en el rango establecido en (4.29). Además del teorema 4.2 queda claro que $(\widetilde{X}^k, \widetilde{U}^k, \widetilde{S}^k) \in \mathcal{L}^0$ para todo σ en el rango dado. Así pues queda probado (4.29) y por lo tanto (4.30).

Completamos la prueba del teorema estimando la reducción en μ sobre el $k-$ ésimo paso; de (4.4) y (4.30), tenemos

$$\mu_{k+1} = (1 - \sigma_k(1-\tau_k))\mu_k$$

$$\leq \left(1 - \frac{2^{3/2}}{n}\gamma\frac{1-\gamma}{1+\gamma}\tau_k(1-\tau_k)\right)\mu_k, \tag{4.32}$$

ahora, la función $\tau(1-\tau)$ es una función cuadrática cóncava de τ, asi que sobre algún intervalo dado la función conseguirá su mínimo valor en uno de los últimos puntos. Por lo tanto, tenemos

$$\tau_k(1-\tau_k) \geq \min\left\{\tau_{\min}(1-\tau_{\min}), \tau_{\max}(1-\tau_{\max})\right\}, \text{ para todo } \tau_k \in [\tau_{\min}, \tau_{\max}]$$

la prueba queda completa al sustituir esta estimación en (4.31) y hacer

$$\delta = 2^{3/2}\gamma\frac{1-\gamma}{1+\gamma}\min\left\{\tau_{\min}(1-\tau_{\min}), \tau_{\max}(1-\tau_{\max})\right\}. \blacksquare$$

El resultado de complejidad para el algoritmo de pasos largos es una consecuencia inmediata del teorema 4.4 y del teorema 3.7.

Teorema 4.5: Dados $\varepsilon > 0$ y $\gamma \in (0,1)$, supongamos que el punto inicial $(x^0, u^0, s^0) \in$

$\mathcal{N}_{-\infty}(\gamma)$ en el algoritmo de pasos largos tenemos

$$\mu_0 \leq \frac{1}{\varepsilon^k},$$

para alguna constante positiva k. Entonces existe un índice K con $K = O(n \log \frac{1}{\varepsilon})$ tal que

$$\mu_k \leq \varepsilon, \text{ para todo } k \geq K.$$

4.5 Algoritmos de trayectoria central no factible

Los algoritmos de trayectoria central que se han presentado hasta ahora necesitan una solución inicial (x^0, u^0, s^0), tal que cumplan que

$$Ax^0 = b,\ x^0 > 0 \quad \text{y} \quad A^T u^0 + s^0 = c,\ s^0 > 0.$$

Quiere decir que para inicializar los algoritmos, necesitamos de una solución factible y estrictamente positiva tanto para el problema primal, como para el problema dual. Sin embargo, es posible utilizar el método de Newton para desarrollar algoritmos que parten para cada iteración de puntos no factibles. La única condición dada sobre el punto de partida de estos algoritmos es la positividad de x y s.

Consideremos los residuos primal y dual asociados a un punto (x, u, s), definidos respectivamente por:

$$\rho_b := Ax - b \quad \text{y} \quad \rho_c := A^T u + S - c.$$

La dirección de Newton se aproxima a la dirección que nos lleva al punto de la trayectoria central (x_μ, u_μ, s_μ), a partir de un punto (x, u, s), y es dada por la solución del siguiente sistema lineal:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta s \end{bmatrix} = \begin{bmatrix} \rho_c \\ \rho_b \\ \mu e - XSe \end{bmatrix} \tag{4.33}$$

donde las dos primeras componentes del lado derecho de la ecuación (2.12) dejan de ser nulas para tomar el valor de los residuos asociados al punto de partida.

Consideremos entonces algoritmos que parten, para cada iteración, de un punto (x, u, s) y caminan en la dirección de Newton, dada por la solución de (4.33).

Observamos que al caminar en esta dirección, estamos al mismo tiempo interesados en una centralización y una disminución de los residuos primal y dual. Vemos que, si un paso unitario es dado en la dirección de Newton, el nuevo punto encontrado será primal y viable, ya que, por la ecuación (4.33) y por la definición de ρ_c y ρ_b, tenemos:

$$\begin{aligned} A^T(u + \Delta u) + (s + \Delta s) &= A^T u + A^T \Delta u + s + \Delta s \\ &= A^T u + s + A^T \Delta u + \Delta s \\ &= A^T u + s - \rho_c = c \end{aligned}$$

y

$$A(x + \Delta x) = Ax + A\Delta x = Ax - \rho_b = b.$$

A partir de un punto primal y dual viable, sucesivos pasos en la dirección de Newton, nos llevaran siempre a puntos que también son viables, puesto que es este caso tenemos que $\rho_c = \rho_b = 0$ en (4.33).

Los algortimos de la trayectoria central infactibles parten, para cada iteración, de un punto (x, u, s) tal que $x > 0$ y $s > 0$, y siguen la dirección (4.33) manteniéndose siempre en una vecidad de la trayectoria central. Esta vecindad es una extensión de la vecindad $\mathcal{N}_{-\infty}(\gamma)$, que fue definida para el algoritmo de la trayectoria central de pasos largos, que contiene puntos que violan las restricciones $Ax = b$ y $A^T u + s = c$. En la vecindad extendida, las normas de las residuos primal y dual son limitadas por una constante μ. De

esta manera cuando $\mu \to 0$, los residuos se hacen cada vez menores y nos aproximamos, a la región viable de los problemas primal y dual. La definición de la vecindad extendida es dada por:

$$\mathcal{N}_{-\infty}(\gamma,\ \beta) := \bigcup_{\mu\in(0,\infty)} \left\{ \delta_{-\infty}(x,s,\mu) \leq \alpha, \frac{\|(\rho_b,\rho_c)\|}{\|(\rho_b^0,\rho_c^0)\|} \leq \frac{\mu}{\mu_0}\beta,\ (x,\ s) > 0 \right\},$$

donde $\gamma \in (0,1)$, $\beta \geq 1$, $\delta_{-\infty}(x,s,\mu)$ está definida en (4.28) y ρ_b^0 y ρ_c^0 son los residuos relativos en el punto de partida del algoritmo, (x^0,u^0,s^0).

Los algoritmos de trayectoria central no factibles generan una solución para los problemas primal y dual, (x^k,u^k,s^k), para el cual $\mu^k \leq e$, en no mas de $O(n^2 \log \frac{1}{e})$ iteraciones.

Algoritmo (4.5) trayectoria central no factible

Dados γ, β, $\tau_{\min}, \tau_{\max}$, con $\gamma \in (0,1)$; $\beta \geq 1$, y $o \leq \tau_{\min} \leq \tau_{\max} \leq 0.5$;

elegimos $(x^0,u^0,\ s^0)$ con $(x^0,\ s^0) > 0$;

para $k = 0$

elegimos $\tau_k \in [\tau_{\min}, \tau_{\max}]$ y resolvemos

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta u^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -\rho_c^k \\ -\rho_b^k \\ -X^k S^k e + \tau_k \mu_k e \end{bmatrix}, \tag{4.34}$$

elegimos σ_k como el valor extendido de σ en $[0,1]$ tal que

$$(\widetilde{X}^k,\ \widetilde{U}^k,\ \widetilde{S}^k) \in \mathcal{N}_{-\infty}(\gamma,\beta), \tag{4.35}$$

y bajo la siguiente condición

$$\widetilde{\mu}_k \leq (1 - 0.01\sigma)\mu_k; \tag{4.36}$$

hacemos

$$(X^{k+1},\ U^{k+1},\ S^{k+1}) = (\widetilde{X}^k,\ \widetilde{U}^k,\ \widetilde{S}^k);$$

$k := k + 1$

hasta que

$\mu_k < \varepsilon$.

Del lema 4.2 usaremos la siguiente notación

$$(\widetilde{X},\ \widetilde{U},\ \widetilde{S}) \ :\ = (X,\ U,\ S) + \sigma(\Delta X,\ \Delta U,\ \Delta S);$$

$$\widetilde{\mu} \ :\ = \frac{\widetilde{X}^T\widetilde{S}}{n}.$$

A continuación introduciremos un concepto útil en nuestra discusión basada en el valor extendido llamada cantidad escalar v_k definida por

$$v_k = \prod_{j=0}^{k-1}(1-\sigma_j) \qquad\qquad (v_0 = 1).$$

Puesto que las dos primeras componentes de la función F en (2.12) son lineales, tenemos

$$\begin{aligned}(\rho_b^k,\ \rho_c^k) &= (1-\sigma_{k-1})(\rho_b^{k-1},\ \rho_c^{k-1}) \\ &= (1-\sigma_{k-1})...(1-\sigma_0)(\rho_b^0,\ \rho_c^0) \\ &= v_k(\rho_b^0,\ \rho_c^0), \end{aligned} \tag{4.37}$$

mientras tanto, como $(X^k,\ U^k,\ S^k)$ pertenece a $\mathcal{N}_{-\infty}(\gamma,\beta)$, tenemos de (4.37) que

$$v_k\frac{\|(\rho_b^0,\ \rho_c^0)\|}{\mu_k} = \frac{\|(\rho_b^k,\ \rho_c^k)\|}{\mu_k} \leq \beta\frac{\|(\rho_b^0,\ \rho_c^0)\|}{\mu_0}.$$

Siempre que $(\rho_b^0,\ \rho_c^0) \neq 0$, de esta desigualdad tenemos que

$$v_k \leq \frac{\beta\mu_k}{\mu_0}. \tag{4.38}$$

En el caso factible de que $(\rho_b^0, \rho_c^0) = 0$, todas las iteraciones $(X^k,\ U^k,\ S^k)$ son estrictamente factibles. El algoritmo de trayectoria central no factible entonces se reduce al caso del algoritmo de trayectoria central de pasos largos visto en la sección 4.4 de éste capítulo.

Para el análisis que haremos asumiremos que punto inicial (x^0, u^0, s^0) es no factible.

4.5.1 Convergencia del algoritmo de trayectoria central no factible

La convergencia de éste algoritmo se prueba al mostrar que hay una constante $\overline{\sigma} > 0$ tal que $\sigma_k \geq \overline{\sigma}$ para todo k.

Por la condición (4.36), tenemos que

$$\mu_{k+1} \leq (1 - 0.01\sigma_k)\mu_k \leq (1 - 0.01\overline{\sigma})\mu_k, \text{ para todo } k \geq 0, \tag{4.39}$$

de tal manera que la sucesión $\{\mu_k\}$ de brecha de dualidad converga $Q-$ lineal a cero. Por (4.37), tenemos también que

$$\left\|(\rho_b^{k+1},\ \rho_c^{k+1})\right\| \leq (1 - \overline{\sigma})\left\|(\rho_b^k,\ \rho_c^k)\right\|, \tag{4.40}$$

de tal forma que la sucesión de normas residual también converge a cero.

Esta discusión se formaliza en los siguientes teoremas de convergencia global.

Teorema 4.6: La sucesión $\{\mu_k\}$ generada por el algoritmo de trayectoria central infactible converge $Q-$ lineal a cero, y la sucesión de normas residual $\left\{\left\|(\rho_b^k,\ \rho_c^k)\right\|\right\}$ converge $R-$ lineal a cero.

Prueba:

La complejidad polinomial del algoritmo se obtiene del hecho que la cota inferior $\overline{\sigma}$ sobre la longitud del paso es una función polinomial inversa de n si elegimos como punto de partida

$$(x^0, u^0, s^0) = (\zeta\varepsilon, 0, \zeta\varepsilon), \tag{4.41}$$

donde ζ es un escalar para el cual

$$\|(x^*, s^*)\|_\infty \leq \zeta, \tag{4.42}$$

para alguna solución primal - dual (x^*, u^*, s^*). Usualmente no conocemos $\|(x^*, s^*)\|_\infty$ ya que no conocemos soluciones óptimas previamente. Si embargo, estas condiciones son aún relevantes para la práctica computacional por la siguiente razón: computacionalmente hablando, los puntos de partida bien centrados para los cuales la proporción

$$\frac{\|(\rho_b^0,\ \rho_c^0)\|}{\mu_0}, \tag{4.43}$$

es la pequeña dirección hacia la convergencia más rápida, hacen que los puntos mal centrados estan mas cercanos al conjunto solución. El punto (4.41) satisface ese criterio. Está perfectamente centrado y la proporción (4.43) varia como $\frac{1}{\zeta}$.

A continuación expresaremos exactamente el resultado de complejidad en el siguiente teorema.

Teorema 4.7: Dado $\varepsilon > 0$. Supongamos que usamos el siguiente punto de partida $(x^0, u^0, s^0) = (\zeta\varepsilon, 0, \zeta\varepsilon)$, donde ζ se define como $\zeta = \min_{i=1,\ldots,n} \min(x_i^0,\ s_i^0)$. Supongamos que el valor de ζ para nuestro particular problema de programación lineal satisface por ejemplo

$$\zeta^2 \leq \frac{C}{\varepsilon^k},$$

para alguna constante positiva C y k. Entonces existe un índice K

$$K = O(n^2 |\log \varepsilon|),$$

tal que las iteraciones $\{(x^k, u^k, s^k)\}$ generadas por el algoritmo de trayectoria central no factible satisfacen

$$\mu_k < \varepsilon, \text{ para todo } k \geq K.$$

Prueba: Ver [13].

Capítulo 5

RESULTADOS NUMÉRICOS

En este capítulo nuestro objetivo es presentar algunas implementaciones de los algoritmos de la familia simplex, de los de la trayectoria central, asi como también usaremos los algoritmos elipsoidales de Khachian de cortes centrales y de cortes profundos para hacer comparaciones en cuanto a la perfomance numérica (número de iteraciones) de estos algoritmos, estos puntos se tratará en la sección 5.1.

Asi también estableceremos la estabilidad numérica entre el algoritmo simplex y los algoritmos de trayectoria central, punto que será visto finalmente en la sección 5.2.

5.1 Perfomance numérica

Los algoritmos de la familia simplex que consideramos es éste capítulo son:

- Algoritmos simplex primal fase 2: Utilizaremos este algoritmo cuando ya conocemos alguna solución básica viable para el problema primal. Asi, partimos de una solución básica viable inicial, buscando alcanzar una solución óptima, caso exista alguna. Por esto decimos que este algoritmo está asociado a optimalidad. En estas implementaciones, tanto por motivos de convergencia del algoritmo simplex, como por simplicidad entre las diversas maneras de obtener una nueva solución básica viable, optamos por la regla del menor índice o regla de Bland (Ver [2]).

- Algoritmos simplex primal fases 1 y 2: Utilizaremos este algoritmo cuando no conocemos cualquier solución básica viable para el problema primal. Este algoritmo utiliza el método de las dos fases. El problema está en encontrar una solución básica viable inicial y un nuevo problema de programación lineal, denominado problema fase 1 o, equivalentemente, problema de viabilidad. Asi, inicialmente utilizaremos el algoritmo simplex primal fase 2 que nos ofrece una solución óptima para el problema de viabilidad. Con esa solución óptima, tenemos un criterio de parada para el algoritmo simplex, certificando inviabilidad del problema de programación lineal original, o obteniendo una solución básica viable inicial para el problema primal. En este último caso, volvemos a utilizar el algoritmo simplex fase 2, ahora para el problema primal, después de la sustitución de posibles variables artificiales como variables básicas.

- Algoritmo simplex dual: Utilizaremos este algoritmo cuando conocemos alguna solución básica viable para el problema lineal dual. Básicamente, el simplex dual es equivalente a aplicar el algoritmo simplex para el problema dual; resolviendo, asi, el problema primal. El algoritmo simplex primal mantiene la viabilidad primal y las condiciones de holgura complementarias, procurando llegar a la viabilidad dual de una solución óptima para el primal. De forma similar, el algoritmo simplex dual mantiene la viabilidad dual y las condiciones de holguras complementarias, procurando llegar a la viabilidad primal de una solución óptima para el problema dual que, por el teorema de dualidad (Ver [2]), será una solución óptima para el primal.

- Algoritmo simplex primal - dual: Utilizaremos este algoritmo cuando conocemos una solución viable para el problema dual, no necesariamente básica. Resolver un problema lineal através del simplex primal - dual es semejante a aplicar, directamente, el método de las dos fases (del simplex primal) al problema primal.■

Ahora recordemos el ejemplo de Klee y Minty enunciado en el capitulo 3 de éste trabajo en la sección 3.4 donde confirmamos que el método o algoritmo simplex no es polinomial por lo tanto puede llegar a ejecutar un número exponencial de iteraciones, además mostramos que el algoritmo puede efectuar $2^n - 1$ iteraciones para llegar a una solución óptima. El algoritmo simplex para el ejemplo de Klee y Minty se puede observar en [Bueno].

Respecto al algoritmo elipsoidal de Khachian tanto el de cortes centrales como el de cortes profundos podemos revisar (Ver [2]).

Para observar la perfomance numérica (números de iteraciones), presentaremos a continuación los principales resultados obtenidos de una implementación en Matlab de los algoritmos mencionadas en éste capítulo para el ejemplo de Klee y Minty, o problema (KM), para $n = 2$. Referenciamos este caso particular por $(KM)_2$.

La tabla 5.1 nos muestra el número de iteraciones obtenidos para el problema lineal $(KM)_2$. En este caso, los algoritmos de la familia simplex nos ofrecen una solución óptima (exacta) $\left[\frac{1}{4}, \frac{15}{16}\right]^T$; los algoritmos de Khachian de cortes profundos encontrarán una solución aproximada con presición de orden 10^{-5}, y los algoritmos de trayectoria central de pasos cortos y el algoritmo de trayectoria central infactible predictor - corrector ofrecen una solución con presición de orden 10^{-6}. Observamos que los mismos algoritmos simplex varian entre si en el número de iteraciones. Esto ocurre puesto que cada uno de ellos tiene una manera diferente de resolver el problema, como vimos al principio de esta sección.

En el caso del simplez primal fases 1 y 2, el número de iteraciones

Algoritmo	Iteraciones
Simplex primal fase 2	3
Simplex primal fases 1 y 2	6
Simplex dual	1
Simplex primal - dual	1
Simplex para el ejemplo de Klee y Minty	3
Khachian (cortes centrales)	156
Cortes profundos	44
Trayectoria central pasos cortos	162
Infactible predictor - corrector	9

Tabla 5.1 Número de iteraciones (ejemplo de Klee y Minty con $n = 2$)

de la fase 1 sumadas a las de la fase 2. Además podemos observar que, para el problema $(KM)_2$, la variante de Khachian, cortes profundos , (Ver [2]) realiza un número significativamente menor de iteraciones en comparación con el algoritmo original de Khachian (cortes centrales). El algoritmo de trayectoria central pasos cortos nos muestra su relativa ineficiencia práctica, en el número de iteraciones, debido a su naturaleza: algoritmo de "pasos cortos". Por otro lado, el algoritmo de trayectoria central infactible predictor - corrector llega a una solución óptima aproximada en apenas 9 iteraciones.

n	Algoritmo de Khachian		Algoritmo simplex	predictor - corrector
	Cortes centrales	Cortes profundos		
2	156	44	3	9
3	425	122	7	13
4	875	257	15	16
5	1575	452	31	19
6	2622	721	63	22
7	3959	1 068	127	23
8	5654	1 563	255	24
9	7731	2 138	511	24
10	10 283	2 824	1 023	24
11	13 658	3 686	2 047	24
12	17 336	4 687	4 095	24
13	21 881	5 827	8 191	25
14	26 742	7 161	16 383	25
15	32 407	8 745	32 767	25
16	39 084	10 566	65 535	25
17	46 261	12 455	131 071	25
18	54 410	14 716	262 143	25
19	63 495	17 151	524 287	25
20	73 978	19 754	1 048 575	25

Tabla 5.2 Número de iteraciones para el ejemplo de Klee y Minty.

La tabla 5.2 nos muestra el número de iteraciones obtenidas con la implementación del algoritmo de Khachian (cortes centrales), y de su variante (cortes profundos), del simplex para el ejemplo de Klee y Minty, y del predictor corrector. Pero, es este caso, resolvemos el problema (KM) con el número de variables n de 2 hasta 20.

Observamos, que la implementación del algoritmo simplex para el ejemplo de Klee y Minty es bastante particular, puesto que logra que el número de iteraciones para encontrar una solución óptima del problema (KM) con n variables sea, simepre, $k = 2^n - 1$. Osea, el simplex esta resolviendo un problema lineal en su peor caso. En la tabla 5.2 observamos que, en cuanto al núero de iteraciones, el simplex se mostró más eficiente que los algoritmos de Khachian apenas para los problemas con hasta $n = 12$ variables; y

hasta $n = 4$, si comparamos con el algoritmo de trayectoria central infactible predictor - corrector.

Es por eso que resulta cierto el comentario que mencionan en su articulo R.Marsten y M. Saltzman que dicen: "Una de las más sorprendentes e importantes características de los algoritmos de trayectoria central es que el número de iteraciones requeridas es inversible al tamaño del problema. Ellos requieren entre 20 y 80 iteraciones para ser resueltos. Existe evidencia experimental que indica que el número de iteraciones se incrementa con el logaritmo del número de variables. Dado este comportamiento, la superioridad de un algoritmo de trayectoria central comparado al algoritmo simplex depende de que tan eficiente pueden ser realizados los pasos del algoritmo".

Por lo tanto podemos concluir que los algoritmos de trayectoria central tienen mejor perfomance numérica que el algoritmo simplex.

5.2 Estabilidad numérica

En esta sección presentamos un ejemplo que prueba que el algoritmo de trayectoria central es más estable numéricamente que el algoritmo simplex para una determinada familia de problemas de programación lineal

Considere el siguiente problema de programación lineal:

$$\begin{aligned} \max \quad & z = c^T x \\ \textit{sujeto a} \quad & \\ & Ax \leqslant b \end{aligned}$$

donde

$$A = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \cdots & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & & & \frac{1}{n+2} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \cdots & \frac{1}{2n} \end{bmatrix}$$

$$b_i = \sum_{j=1}^{n} \frac{1}{i+j} \ , i = 1, ..., n$$

$$c_i = \frac{2}{i+1} + \sum_{j-2}^{n} \frac{1}{i+j} \ , i = 1, ..., n$$

Este problema tiene una única solución óptima:

$$x = (1, 1, ..., 1)^T$$

y el problema dual tiene también una solución óptima única:

$$y = (2, 1, ..., 1)^T$$

Cuando este problema es resuelto por el algoritmo simplex los resultados son malos, se alejan de la solución verdadera $x = (1, 1, ..., 1)^T$. La razón es la mala estructura de la matriz A. Al aplicar el algoritmo de trayectoria central a este problema los resultados son mucho mejores.

La **Tabla 5.3** nos muestra las soluciones producidas por el algoritmo de trayectoria central para varios valores de n.

	$n = 5$		$n = 10$		$n = 15$	
	SIMPLEX	Trayectoria Central	SIMPLEX	Trayectoria Central	SIMPLEX	Trayectoria Central
x_1	0.973	0.996	0.948	1.010	1.020	1.010
x_2	1.300	1.010	1.510	0.959	0.802	0.940
x_3	0.000	1.000	0.000	1.010	1.700	1.050
x_4	2.290	0.998	0.000	1.030	0.000	1.050
x_5	0.438	0.993	4.460	1.020	2.180	1.020
x_6			0.000	1.010	0.000	0.999
x_7			0.569	0.998	0.000	0.971
x_8			0.000	0.991	0.000	0.964
x_9			0.000	0.987	6.290	0.966
x_{10}			2.530	0.985	0.000	0.975
x_{11}					0.000	0.988
x_{12}					0.000	1.000
x_{13}					0.000	1.020
x_{14}					0.000	1.030
x_{15}					3.020	1.040

Tabla 5.3 Soluciones producidas por el algoritmo simplex y el algoritmo de trayectoria central para $n \in \{5, 10, 15\}$.

Esta información nos muestra claramente que el algoritmo de trayectoria central es mucho más estable numéricamente que el algoritmo simplex.

Capítulo 6

CONCLUSIONES

A manera de conclusión podriamos decir lo siguiente:

1°) Si bien es cierto, desde un punto de vista teórico, el comportamiento del algoritmo simplex es exponencial, y por lo tanto malo desde el punto de vista de la teoría de complejidad de algoritmos, el algoritmo simplex ha probado ser eficiente en muchas situaciones prácticas convirtiéndose quizás en el algoritmo de tipo exponencial más exitoso que se conozca. Por lo tanto, no creemos que los algoritmos de punto interior, como los descritos en nuestro trabajo de tésis, vayan a desplazar al algoritmo simplex sino que estos algoritmos se utilizarán en problemas especializados de programación lineal, como los que surgen en ciertos problemas de asignación y en ciertos problemas de flujos en grafos.

2°) Si bien la experiencia numérica presentada en nuestro trabajo de tésis no es lo suficientemente extensa para hacer conclusiones definitivas si a podido dejar ver la mayor "robustez" de los algoritmos de punto interior en comparación con el algoritmo simplex. Existen reportes, presentados en revistas internacionales dedicadas al campo de la programación matemática y la optimización numérica, donde se reportan resultados numéricos que evidencian una superioridad de los algoritmos de punto interior cuando el número de restricciones más el número de variables está

en el rango de

$$2\ 000 < n + m < 10\ 000.$$

3°) Lo que si es importante destacar es que estos algoritmos de punto interior, desde un punto de vista teórico y también práctico, han permitido una mejor comprensión del problema de programación lineal. Demostrando con esto que en áreas aparentemente cerradas a la investigación es posible aún generar resultados que constituyen aportes a la teoría. Prueba de esto es los más de 3 000 artículos y trabajos de tésis doctorales que fueron apareciendo desde el año 1984 hasta hace relativamente poco.

4°) Finalmente, espero que éste trabajo de tésis sirva como referencia en el campo de la optimización lineal a los estudiantes de la Facultad de Ciencias Naturales y Matemática de la Universidad Nacional del Callao ya que éste es un tema sobre el cual aún hay mucho más que decir y que ha sido poco trabajado en las universidades del país.

Capítulo 7

BIBLIOGRAFIA

[1] **de Almeida D.C.**, *Diccionário de Expressões idiomáticas Inglês - Português*, Editorial Atlas, São Paulo, 1997.

[2] **Apostol T.M.**, *Análisis Matemático*, segunda edición, Editorial Revertí. S.A, Barcelona, 1996.

[3] **Frisch K.R**, *The Logarithmic Potential Method of Convex Programming*, Memorandum, University Institute of Economics, Oslo, Norway, 1955.

[4] **Goberna M. A., V. Jornet V. y Puente R.**, *Optimización lineal teoría, métodos y modelos*, Editorial Mc Graw - Hill, Interamericana de España, S.A.U, 2004.

[5] **Gonzaga C.C.**, *Algoritmos de Pontos interiores em programacão linear*, Escola Brasileira de Otimizacão, Universidade Federal do Rio de Janeiro, 1989.

[6] **den Hertog D**, *Interior point approach to linear Quadratic and conex programming algorithms and complexity*, Kluwer Academic Publishen.

[7] **Maculan N. y Fampa M**, *Optimización lineal*, Universidad Federal do Rio de Janeiro, 2004, pp. 123 - 178.

[8] **Nocedal J y Wright S. J.**, *Numerical Optimización*, Springer - Verlag New york, Berlin, Heidelberg, 1999.

[9] **Padberg M**, *Lincar Optimización and cxtesions*, Springer - Verlag, Berlin, Heidelberg, Germany, 1995.

[10] **Ruey - Lin Shen - Cherng Fang,** *On the Generalized Path-following Methods for linear Programming*, NCSUOR Report N°261, January 1992.

[11] **Sofer A. and Nash S.**, *Linear and no Linear Programming*, Editorial Mc Graw Hill, 1996.

[12] **Tsitsiklis J. and D. Bersimas D**, Introduction to Linear Optimization, Editorial Athenea, 2003.

[13] **Wright S.J.**, *Primal - Dual Interior - Point Meihods*, Society For Industrial Appied Mathematica (SIAM), philadelphia, 1997, pp. 83 - 124.

[14] **Zoutendijk G.,** *Mathematical programming methods*, North - Holland Publishing Company, 1976.