



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS
MAESTRÍA EN CIENCIAS (MATEMÁTICAS)

**Implementación en MatLab del Método de Planos de Corte
Central Acelerado.**

TESIS

PARA OBTENER EL TÍTULO DE
MAESTRO EN CIENCIAS

PRESENTADA POR
Rolando Roberto Linaldi Guzmán

DIRECTORA DE TESIS
Lidia Aurora Hernández Rebollar

Puebla, Pue.

Abril de 2016



BUAP

DR. JOSÉ ENRIQUE BARRADAS GUEVARA
SECRETARIO DE INVESTIGACIÓN Y
ESTUDIOS DE POSTGRADO, FCFM-BUAP
P R E S E N T E:

Por este medio le informo que el(la) C:

ROLANDO ROBERTO LINALDI GUZMÁN

estudiante de la Maestría en Ciencias (Matemáticas), ha cumplido con las indicaciones que el Jurado le señaló en el Coloquio que se realizó el día 21 de enero de 2016, con la tesis titulada:

"Implementación en Mat Lab del Método de Planos de Corte Central Acelerado"

Por lo que se le autoriza a proceder con los trámites y realizar el examen de grado en la fecha que se le asigne.

A T E N T A M E N T E.
H. Puebla de Z. a 15 de marzo de 2016

DR. FERNANDO MACÍAS ROMERO
COORDINADOR DEL POSTGRADO
EN MATEMÁTICAS.

Cep. Archivo.
DR. JEBG/mtrv



Facultad
de Ciencias
Físico Matemáticas

Av. San Claudio y 18 sur, edif. 111 A,
Ciudad Universitaria, Col. San
Manuel, Puebla, Pue. C.P. 72570
01 (222) 229 55 00 Ext. 7550 y 7552

Para Tio Cheque y Tia Berta

Agradecimientos

Agradezco a mis padres, Roberto Linaldi Dolores y Gabriela Guzmán Aranda, por todo su apoyo a través de toda mi existencia.

A mis hermanos sanguíneos, Vladimir Illescas Guzmán y Orlando Linaldi Paez, por ser mis compañeros de vida.

A la fruta más linda y rica de mi huerto, la flor más hermosa de mi jardín, mi bordón más ligero, mi musa y mi escaramuza, Karina Isidro Mora.

A la Doctora Lidia Aurora Hernández Rebollar, mi segunda madre, tutora y asesora, que nunca dudó en ayudarme y apoyarme por muy loca que fuera mi idea o decisión.

A los doctores Maxim Todorov y Enrique González, por instruirme en el mundo de la optimización, verdaderos expertos en la materia.

A mis hermanos no sanguíneos, que me han seguido la corriente en mi muy particular forma de ver la vida, Luis (el más veterano), Fio (mi hermana menor postiza), JJ (el estadista, ¿quién en su sano juicio quiere ser estadista?), Cheques (el gigante amistoso), Mon (todo un vato), Omar (el magazo), Ozka (Don Disperso) y Sam (desde “Shihuahua”, el hombre que casi conoció a Michi Panero).

Un especial agradecimiento a mis sinodales, los doctores Jiménez Pozo, Bustamante González, Slavisa y Mendoza Torres, por el tiempo que invirtieron en mi formación como estudiante y por sus consejos, revisiones, correcciones y observaciones que hicieron de esta tesis un mejor documento.

A mi jefa, Angélica Perez, por confiar en mí, corregirme y hacer de mí un mejor editor.

A todos los que directa o indirectamente me apoyaron para que pudiera llegar hasta aquí.

A los que no mencioné, no por falta de reconocimiento, si no por falta de memoria.

Y por último, pero no menos importante, al Consejo Nacional de Ciencia y Tecnología, que sin su apoyo no hubiera sido posible este proyecto, la Maestría en ciencias (matemáticas), y su conclusión, esta tesis.

Índice general

1. Introducción	1
2. La programación lineal semi-infinita	5
2.1. El problema de la programación lineal semi-infinita	5
2.2. Teoría preliminar	6
2.2.1. Conjuntos convexos	6
2.2.2. Consistencia y optimalidad	9
3. Métodos de planos de corte central	17
3.1. Discretizando mediante planos de corte	17
3.2. El método de planos de corte central	18
3.2.1. El algoritmo del método de PCC	20
3.3. El método de planos de corte central acelerado	25
3.3.1. El algoritmo del método de planos de corte central ace- lerado	25
4. Pruebas numéricas	37
5. Conclusiones	43
A. Programa en MatLab del método de PCC	45
B. Programa en MatLab del método de PCCA	49
Bibliografía	53

Capítulo 1

Introducción

La Programación Semi-Infinita (PSI) es una extensión natural de la programación matemática ordinaria y ha sido estudiada desde principios de los años sesentas.

Varios enfoques han sido propuestos para la solución numérica del problema de la Programación Lineal Semi-Infinita (PLSI). Un estudio puede ser encontrado en [4], junto con una visión integral de la teoría de la PLSI y sus motivaciones.

De acuerdo al mencionado estudio, los algoritmos para la PLSI pueden ser clasificados en cinco categorías:

- (A) Métodos de discretización (mediante rejillas y mediante planos de corte);
- (B) Métodos de reducción local;
- (C) Métodos de intercambio;
- (D) Métodos tipo simplex;
- (E) Métodos de descenso.

La posición de cada familia en la lista anterior corresponde, estrictamente hablando, a su reputación de eficiencia computacional en el tratamiento numérico de los problemas de PLSI. Es importante mencionar que los métodos de tipo (A) pueden ser aplicados bajo condiciones más bien leves, en contraste con las de tipo (B), las cuales requieren, al menos, la propiedad C^1 para los coeficientes del sistema de restricciones.

En [8] se presenta otra clasificación de los métodos para resolver los programas semi-infinitos: métodos de discretización, métodos de tipo semi-continuo y métodos de tipo continuo, propuesta por Hettich en 1979. En esta tesis nos

enfocaremos en la primera clasificación, específicamente en los métodos de discretización.

En un trabajo previo, desarrollado en los estudios de licenciatura del autor de la presente, revisamos la variedad de métodos propuesta por Goberna en [4]. En esta ocasión centraremos nuestra atención en los métodos de Planos de Corte Central (PCC) por las buenas propiedades que presenta su algoritmo. Dichas propiedades son su aplicación bajo condiciones suaves, antes mencionadas, y el hecho de que produce una sucesión convergente de puntos que se encuentran en el interior de la región factible. Sin embargo, a pesar de su clasificación en la categoría (A), ha sido observada una lenta convergencia en ejemplos prácticos, presentados en el ejemplo 11.2 de [4], y corroborados en la tesis de licenciatura del autor.

Como objetivo de este trabajo, valoramos el desempeño del algoritmo presentado en [1], que propone un proceso de aceleramiento del algoritmo del método de PCC presentado en [4], a través de su programación en MatLab y su aplicación a algunos ejemplos. El código de los programas realizados en MatLab se presenta en los apéndices A y B.

Objetivo

El objetivo de esta tesis es valorar el desempeño del algoritmo de planos de corte central acelerado, a través de su programación en MatLab y su aplicación en algunos ejemplos seleccionados.

Revisión de la literatura

Como ya lo mencionamos antes, la mayor parte del trabajo de esta tesis se dedica al estudio de un algoritmo presentado por Bruno Betto en [1]. Este autor menciona que su interés en los métodos de la PLSI ha sido motivado por la necesidad de procedimientos computacionales efectivos para el análisis robusto de los procesos bayesianos cuando la distribución a priori pertenece a una clase de momento generalizado de medidas.

En el mismo artículo mencionado arriba, Betto propone un aceleramiento para el método de PCC, cuyo algoritmo fue propuesto por Elzinga en 1975. Una primera idea del aceleramiento se presenta en [8] y se desarrolla en [7]. En el primer capítulo de este trabajo se da una introducción de lo que es la

programación semi-infinita y de su procedencia, así como los objetivos que se buscan cumplir con esta tesis.

En el segundo capítulo presentamos conceptos básicos de la Programación Lineal (PL) y de la PLSI, extraídos de [4], [5] y [6], que forman la base para poder establecer teoremas importantes para el capítulo 3.

En el tercer capítulo se presenta el método de PCC (tomado de [4]) y su aceleramiento (extraído de [1]), los algoritmos de dichos métodos y los teoremas importantes relacionados con la finalización finita y la razón de convergencia de éstos. La demostración de los teoremas de este capítulo se desarrolló a partir de las que se presentan en [4] y [7].

Para iniciar los algoritmos de los métodos de PCC y de PCCA es necesario contar con un punto factible inicial, por lo que se revisaron los métodos para la solución de sistemas de desigualdades lineales, presentados en [2] y [3], contando con el asesoramiento del autor de dichos artículos, el Dr. González Gutiérrez.

Por último, en el cuarto capítulo se presenta la solución de problemas, tomados de [4], que sirven para valorar el desempeño de los algoritmos creados en MatLab.

Capítulo 2

La programación lineal semi-infinita

2.1. El problema de la programación lineal semi-infinita

Un problema de Programación Lineal Semi-Infinita consiste en minimizar una función lineal, llamada *función objetivo*, definida en un espacio de dimensión finita pero sujeta a un número infinito de restricciones lineales. Una extensión natural del problema de PL al problema de PLSI puede obtenerse mediante el reemplazo del conjunto de índices $\{1, 2, \dots, m\}$, del caso finito, por un conjunto arbitrario T , y permitiendo sólo un número finito de variables asociadas, diferentes de cero.

Ahora definiremos la forma general del *problema primal* de la PLSI, a quien llamaremos (P):

$$\begin{aligned} (P) \quad & \text{Inf} \quad c'x \\ \text{s.a.} \quad & a_t'x \geq b_t, \quad t \in T, \end{aligned}$$

donde $c \neq 0_n$ y x pertenecen a $\mathbb{R}^n$, T es un conjunto de índices arbitrario (posiblemente infinito), $a_t = a(t) = (a_1(t), \dots, a_n(t))$ es una función definida de T a $\mathbb{R}^n$ y $b_t = b(t)$ es una función que va de T a $\mathbb{R}$.

El *conjunto factible* de (P), F , es un conjunto cerrado convexo en $\mathbb{R}^n$ (dado que es la intersección de una familia de semiespacios cerrados), tal vez

vacío o $\mathbb{R}^n$, para el cual el *sistema lineal semi-infinito* (abreviado SLSI) $\{a'_t x \geq b_t, t \in T\}$ proporciona una representación externa. Si (P) es *consistente* (es decir, que tenga al menos una solución, $F \neq \emptyset$), su *valor óptimo* $v(P)$ puede ser un número real o $-\infty$. En el primer caso, (P) se dice que es *acotado* y en contraste con el caso de la programación lineal ordinaria, la existencia de una solución óptima no está garantizada; en otras palabras, un problema acotado de PLSI no necesita ser *resoluble*. Si (P) es *inconsistente*, $v(P) = \infty$ por definición.

2.2. Teoría preliminar

2.2.1. Conjuntos convexos

En esta sección se introduce una familia de conjuntos, llamados *convexos*, y algunas de sus propiedades básicas, ya que son de gran utilidad en la construcción y análisis de los modelos de programación matemática y en particular, de programación lineal.

En este trabajo se usarán superíndices para denotar a diferentes vectores, mientras que los subíndices se usarán para denotar las componentes de un vector.

Definición 2.1. $C \subset \mathbb{R}^n$ es **convexo** si $(1 - \lambda)x^1 + \lambda x^2 \in C$, cualesquiera que sean $x^1, x^2 \in C$ y $\lambda \in [0, 1]$. La **dimensión** de $C \neq \emptyset$ es la de su envoltura afín, es decir, $\dim C := \dim \text{aff} C$.

En otras palabras, un conjunto $C \subseteq \mathbb{R}^n$ es convexo si, para cualquier pareja de puntos $x^1, x^2 \in C$, el segmento de recta que une estos dos puntos está también en C .

Por convenio, $\emptyset$ también es convexo y $\dim \emptyset = -1$.

Recordemos que una *combinación lineal* (CL) de los vectores $x^1, x^2, \dots, x^p \in \mathbb{R}^n$ es una expresión de la forma $\sum_{i=1}^p \alpha_i x^i$, con $\alpha_i \in \mathbb{R}$, $i = 1, \dots, p$. Dicha CL se dice que es *no negativa* cuando $\alpha_i \geq 0$, para todo $i = 1, \dots, p$. Si $\sum_{i=1}^p \alpha_i = 1$ la CL se dice que es *afín*. Una CL es *convexa* cuando es no negativa y afín. Así, una combinación lineal convexa de un número finito de

puntos, $x^1, \dots, x^p$, se define como el punto

$$x = \sum_{i=1}^p \alpha_i x^i, \quad \text{donde } \alpha_i \geq 0, \quad i = 1, \dots, p; \quad \sum_{i=1}^p \alpha_i = 1.$$

Proposición 2.2. Sea $C \neq \emptyset$ subconjunto de $\mathbb{R}^n$. C es convexo si y sólo si C contiene todas las combinaciones lineales convexas de elementos de C .

Definición 2.3. La **envoltura convexa** de un conjunto $X \subset \mathbb{R}^n$ es el menor conjunto convexo que lo contiene. Se denota por $\text{conv}X$ y es la intersección de todos los conjuntos convexos que contienen a X .

Observemos que $\text{conv}\emptyset = \emptyset$.

Proposición 2.4. Si $X \neq \emptyset$, $\text{conv}X$ es el conjunto de todas las combinaciones lineales convexas de puntos de X .

Ahora pasaremos a definir un concepto muy usado en la programación lineal, que es el de cono.

Definición 2.5. Un conjunto K , tal que $0_n \in K \subset \mathbb{R}^n$ es un **cono** si $\lambda x \in K$, cualesquiera que sean $x \in K$ y $\lambda \geq 0$.

En otras palabras, son conos aquellos conjuntos de $\mathbb{R}^n$ que contienen a todas las semirectas que parten de 0_n y que pasan por un punto del mismo. Cualquier subespacio vectorial es un cono (aunque 0_n no sea un vértice del mismo). El único conjunto acotado que también es cono es $\{0_n\}$.

Definición 2.6. Un conjunto K es **cono convexo** si es cono y conjunto convexo a la vez.

Proposición 2.7. Sea $\emptyset \neq K \subset \mathbb{R}^n$. K es un cono convexo si y sólo si contiene todas las combinaciones lineales no negativas de elementos de K .

Definición 2.8. La **envoltura cónica** (se sobreentiende que convexa) de un conjunto $X \subset \mathbb{R}^n$ es el menor cono convexo que contiene a X . Se denota por $\text{cone}X$ y es la intersección de todos los conos convexos que contienen a X .

Se puede observar que $\text{cone}\emptyset = \{0_n\}$.

Proposición 2.9. Si $X \neq \emptyset$, $\text{cone}X$ es el conjunto de todas las combinaciones lineales no negativas de puntos de X .

Definición 2.10. Dado un sistema de inecuaciones

$$\sigma = \{a_i^T x \geq b_i, i = 1, \dots, m\},$$

el **conjunto factible** de σ se define como

$$F := \{x \in \mathbb{R}^n | a_i^T x \geq b_i, i = 1, \dots, m\}.$$

Definición 2.11. Se define como **interior relativo** de un conjunto convexo $X \subseteq \mathbb{R}^n$, que será denotado como $\text{rint}X$, al interior de X en la topología relativa del menor subconjunto afín que contiene a X . Es decir $\text{rint}X = \{x \in \text{aff}(X) | \exists \epsilon > 0, (x + B_0(\epsilon)) \cap (\text{aff}(X)) \subset X\}$.

Otro concepto relacionado con el anterior es el de *frontera relativa*, que es el conjunto diferencia $\text{cl}X \setminus \text{rint}X$, es decir, la clausura de X menos el interior relativo de X .

Definición 2.12. El conjunto $\{x^1, \dots, x^p\} \subset \mathbb{R}^n$, con $p \leq n$, es **afinmente independiente** (AI) cuando $\left\{ \begin{pmatrix} x^1 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} x^p \\ 1 \end{pmatrix} \right\} \subset \mathbb{R}^{n+1}$ es linealmente independiente (LI).

Teorema 2.13 (Caratheodory, 1911). Sea $\emptyset \neq X \subset \mathbb{R}^n$. Se cumple:

1. Toda combinación lineal no negativa de elementos de X , con $X \neq \{0_n\}$, es también combinación lineal no negativa de un subconjunto LI de X (con un máximo de n elementos).
2. Toda combinación lineal convexa de puntos de X también es combinación lineal convexa de un subconjunto AI de X (con un máximo de $n + 1$ puntos).

Ahora definiremos los llamamos *conos de primer y segundo momento* de σ .

$$M := \text{cone}\{a_t, t \in T\} \quad y \quad N := \text{cone}\left\{ \begin{pmatrix} a_t \\ b_t \end{pmatrix}, t \in T \right\},$$

Así como también el *cono característico* de σ ,

$$K := \text{cone}\left\{ \begin{pmatrix} a_t \\ b_t \end{pmatrix}, t \in T; \begin{pmatrix} 0_n \\ -1 \end{pmatrix} \right\}.$$

Otro cono importante por definir es el polar, que se usará en el lema de Farkas para conos, lema que usaremos en la siguiente subsección.

Definición 2.14. Dado un cono convexo $K \subset \mathbb{R}^n$, se define su **cono polar** (positivo) como

$$K^o = \{y \in \mathbb{R}^n \mid x'y \geq 0, \forall x \in K\}.$$

Lema 2.15 (de Farkas para conos). Si K es un cono convexo cerrado, entonces $K^{oo} = K$.

Definición 2.16. Una inecuación $a'x \geq b$ es **consecuencia** del sistema $\{a'_i x \geq b_i, i \in I\}$ si toda solución de este sistema satisface dicha inecuación.

Lema 2.17 (extendido de Farkas). La inecuación $a'x \geq b$ es una consecuencia del sistema $\{a'_t x \geq b_t, t \in T\}$ si y sólo si

$$\begin{pmatrix} a \\ b \end{pmatrix} \in \text{clcone} \left\{ \begin{pmatrix} a_t \\ b_t \end{pmatrix}, t \in T; \begin{pmatrix} 0_n \\ -1 \end{pmatrix} \right\}.$$

2.2.2. Consistencia y optimalidad

Decimos que una sucesión $\{x^r\} \in \mathbb{R}^n$ es una *solución asintótica* del sistema $\sigma = \{a'_i x \geq b_i, i = 1, \dots, m\}$ si, para todo $t \in T$,

$$\liminf_r a'_t x^r \geq b_t,$$

donde $+\infty$ es permisible como solución.

Si x es el punto límite de una sucesión que satisface la condición anterior, entonces, para todo $t \in T$, tenemos

$$b_t \leq \liminf_r a'_t x^r = \lim_r a'_t x^r = a'_t x,$$

tal que x será una solución ordinaria de σ . Además σ es consistente si y sólo si existe una solución asintótica acotada, de acuerdo al siguiente argumento. Si $\{x^r\}$ es una solución asintótica acotada de σ , entonces $\text{cl}\{x^r, r = 1, 2, \dots\}$ es compacto, y existirá una subsucesión convergente cuyo punto límite será una solución de σ . Por lo tanto, es evidente que si $x \in \mathbb{R}^n$ es una solución ordinaria de σ , entonces la solución constante $x^r = x, r = 1, 2, \dots$, es una solución asintótica de este sistema.

Así, podemos ver que todo sistema consistente tendrá soluciones asintóticas, y algunos sistemas inconsistentes tendrán la misma propiedad.

Un sistema lineal semi-infinito σ que no tiene una solución asintótica es

llamado *fuertemente inconsistente*. Si σ no tiene una solución ordinaria, pero al menos tiene una solución asintótica, será llamado *asintóticamente inconsistente*.

Teorema 2.18. *Dado un SLSI σ , las proposiciones siguientes se cumplen:*

1. σ es consistente si y sólo si $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \notin clN$;
2. σ es asintóticamente inconsistente si y sólo si $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \in (clN) \setminus N$;
3. σ es fuertemente inconsistente si y sólo si $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \in N$

Por otra parte, en los enunciados anteriores, N puede ser reemplazado por K y obtener los mismos resultados. Esto lo podemos asegurar por el lema siguiente.

Lema 2.19. *Las relaciones siguientes entre N , K y sus clausuras permanecen:*

1. $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \in N$ si y sólo si $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \in K$
2. $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \in clN$ si y sólo si $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \in clK$

El criterio de clasificación posterior se basa en el rendimiento óptimo de los problemas de optimalidad, que se presentan a continuación, asociados con los conos N y K . Para ello comenzaremos definiendo el problema siguiente:

$$\begin{aligned} (PN) \quad & \inf \|x\| \\ \text{s.a.} \quad & \begin{pmatrix} x \\ 1 \end{pmatrix} \in N. \end{aligned}$$

y (PK) definido en una forma similar, reemplazando N por K (justificado por el lema anterior).

Dado un problema de programación (P) , $v(P)$ denotará el valor óptimo de la función objetivo de (P) .

Teorema 2.20. *Dado un sistema σ , los enunciados siguientes se cumplen:*

1. σ es consistente si y sólo si $v(PN) > 0$;
2. σ es asintóticamente inconsistente si y sólo si $v(PN) = 0$ y (PN) es insoluble;

3. σ es fuertemente inconsistente si y sólo si $v(PN) = 0$ y (PN) es soluble.

De igual manera que antes, (PN) puede ser reemplazado por (PK) en las proposiciones del teorema anterior.

Ahora pasaremos a otro criterio de clasificación de σ basado en el concepto de rayo ascendente.

Un vector $\omega \in \mathbb{R}^n$ es llamado *rayo ascendente* para el sistema σ si para cada escalar μ existe un subsistema finito $\sigma(\mu) := \{a'_t x \geq b_t, t \in S(\mu)\}$, con $S(\mu) \subset T$, tal que $\omega'x \geq \mu$, para toda solución x de $\sigma(\mu)$, es decir, $\sigma(\mu)$ es inconsistente o $\omega'x \geq \mu$ es una consecuencia de $\sigma(\mu)$.

Si esto último es cierto para todo $\mu \in \mathbb{R}$, por el lema de Farkas y observando que el cono característico de $\sigma(\mu)$ está contenido en K , tenemos que $\{(\omega_\mu), \mu \in \mathbb{R}\} \subset K$.

Por lo tanto K contiene la línea vertical que pasa por ω .

Representaremos por A al conjunto de rayos ascendentes para σ . Por lo que A es un cono convexo.

Teorema 2.21. *Dado un sistema σ , los enunciados siguientes se cumplen:*

1. σ es consistente si y sólo si $A = \emptyset$.
2. σ es asintóticamente inconsistente si y sólo si $\emptyset \neq A \neq \mathbb{R}^n$. Más aún, en este caso, $\text{rint}M \subset A \not\subseteq M$.
3. σ es fuertemente inconsistente si y sólo si $A = \mathbb{R}^n$.

Teorema 2.22. *Sean σ_1 y σ_2 dos sistemas lineales semi-infinitos con conos característicos K_1 y K_2 y conjuntos solución F_1 y F_2 , respectivamente. Entonces*

1. $F_2 \subset F_1$ si y sólo si $\text{cl}K_1 \subset \text{cl}K_2$;
2. σ_1 y σ_2 son equivalentes si y sólo si $\text{cl}K_1 = \text{cl}K_2$.

Ahora consideraremos al vector de coeficientes de la función objetivo c , también llamado vector de costo, como un parámetro y analizaremos las propiedades de $v : \mathbb{R}^n \rightarrow [-\infty, +\infty]$, recalando que v toma el valor de $+\infty$

cuando $F = \emptyset$.

También consideraremos el mapeo del conjunto optimal $F^* : \mathbb{R}^n \rightarrow 2^{\mathbb{R}^n}$, que asigna a cada $c \in \mathbb{R}^n$ el conjunto óptimo (posiblemente vacío) del problema

$$P(c) \quad \inf c'x \quad s.a. \quad x \in F,$$

donde F es el conjunto factible (fijo), obtenido como conjunto solución del sistema dado $\sigma = \{a'_t x \geq b_t, t \in T\}$.

El problema $P(c)$ es *discretizable* si existe una solución para los subproblemas finitos

$$P_k(c) \quad \inf \quad c'x \\ s.a. \quad a'_t x \geq b_t \quad t \in T_k.$$

Siendo T_k un subconjunto finito de T , para $k = 1, 2, \dots$, y tal que sus valores óptimos satisfacen

$$v(c) = \lim_k v_k(c),$$

donde $v_k(\cdot)$ es la función del valor óptimo del problema $P_k(\cdot)$ (cuyo sistema de restricciones está asociado con el subconjunto de índices finito T_k).

En particular, $P(c)$ se dice que es *reducible* si existe un conjunto finito $S \subset T$ tal que $v(c) = v_S(c)$, donde $v_S(c)$ es el valor óptimo del problema de programación lineal ordinaria

$$P_S(c) \quad \inf \quad c'x \\ s.a. \quad a'_t x \geq b_t \quad t \in S.$$

Un problema $P(c)$ inconsistente será reducible si y sólo si σ es fuertemente inconsistente. Por otra parte si $v(c) = -\infty$, $P(c)$ es trivialmente reducible. Cuando $P(c)$ no es discretizable, aún puede ser posible aproximar $v(c)$ mediante la resolución de una sucesión de problemas de programación lineal, obtenida a partir de $P(c)$.

Esto es por medio de dos modificaciones alternativas: a través de una perturbación suficientemente pequeña de la función objetivo o intersectando su conjunto factible con una vecindad politópica del origen suficientemente grande.

Decimos que $P(c)$ es débilmente discretizable si existe $d \in \mathbb{R}^n$ tal que el problema perturbado $P(c + \alpha d)$ es discretizable para $\alpha \in]0, \alpha_0]$, con $\alpha_0 > 0$, y

$$v(c) = \lim_{\alpha \searrow 0} v(c + \alpha d).$$

$P(c)$ se dice *regularizable* si existe un politopo C , con $0_n \in \text{int}C$, tal que el problema de PLSI

$$\begin{aligned} P(c, \rho) \quad & \inf \quad c'x \\ \text{s.a.} \quad & a_t'x \geq b_t \quad t \in T, \\ & x \in \rho C \end{aligned}$$

es discretizable cuando $\rho > \rho_0$, con $\rho_0 > 0$, y cualquier representación lineal de ρC que tomemos, también cumple que

$$v(c) = \lim_{\rho \rightarrow +\infty} v(c, \rho).$$

Si $P(c)$ es débilmente discretizable para el vector d (regularizable para el politopo C) y el $\lim_k \alpha_k = 0$ ($\lim_k \alpha_k = \infty$), $\alpha_k > 0$ ($\rho_k > \rho_0$), $k = 1, 2, \dots$, existirá una subsección $\{T_k\}$ de subconjuntos finitos de T tal que $\lim_k v_k(c + \alpha_k d) = v(c)$ ($\lim_k v_k(c, \rho_k) = v(c)$ respectivamente).

En otras palabras, $v(c)$ puede ser aproximado, en ambos casos, a través de un proceso diagonal, que implica subproblemas finitos modificados.

Tabla 1

σ	c	$P(c)$	$\delta(c)$
Fuertemente inconsistente		Discretizable	$0 \Leftrightarrow c \in M$
Asintóticamente inconsistente	$c \in A$	Discretizable	0
	$c \in (clM) \setminus A$	Débilmente discretizable (Pero no discretizable)	$+\infty$
	$c \notin clM$	Ni siquiera débilmente discretizable	$+\infty$
Consistente	$c \in rintM$	Discretizable	0
	$c \in rbdM$	Débilmente discretizable y discretizable $\Leftrightarrow C_1 = C_2$	$0 \Leftrightarrow C_1 = C_2$
	$c \notin clM$	Discretizable	0

$$C_1 = cl[(\{c\} \times \mathbb{R}) \cap K] \text{ y } C_2 = (\{c\} \times \mathbb{R}) \cap (clK).$$

Teorema 2.23. *Cualquier problema de PLSI que tenga la forma $P(c)$ es regularizable. Por otra parte, éste es discretizable (o al menos débilmente discretizable) dependiendo de las propiedades de consistencia que cumpla α y la posición de c respecto a M y a A , las cuales se presentan en la Tabla 1 (tabla que también determina la existencia del hueco dual).*

El teorema anterior establece que para $c \in \text{rint}M$, el problema consistente $P(c)$ será resoluble y discretizable.

Ahora nos enfocaremos en aproximar soluciones óptimas del problema original por medio de una sucesión de soluciones óptimas, $\{x^k\}$, de subproblemas finitos $P_k(c)$, definidos a través de una sucesión de subconjuntos de índices $\{T_r\}$.

Representaremos por F_r al conjunto factible de $P_k(c)$.

Si $F = \bigcap_{k=1}^{\infty} F_k$, se dice que $\{P_k(c)\}$ es una *sucesión manejable* (s.m.) de subproblemas finitos de $P(c)$.

Si la sucesión $\{T_k\}$ es expansiva, decimos que la s.m. asociada de subproblemas es monótona.

Con esto, se puede ver que si $\{T_k\}$ define una s.m. de subproblemas de $P(c)$, la sucesión $\{T_k^1\}$, con $T_k^1 := T_1 \cup T_2 \cup \dots \cup T_k$, $k = 1, 2, \dots$, define una s.m. monótona.

La existencia de una s.m. de subproblemas es garantizada por la separabilidad de $\mathbb{R}^{n+1}$. Por otro lado, si $x^k \in F_k^*(c)$, es decir, si x^k es una solución óptima de $P_k(c)$, y $\{P_k(c)\}$ es una s.m. de subproblemas, decimos que $\{x^k\}$ es una *sucesión manejable de puntos* asociada.

Estas sucesiones manejables de puntos constituyen las sucesiones de aproximación natural en cualquier procedimiento de discretización típico.

Teorema 2.24. *Si $P(c)$ es consistente y $c \in \text{rint}M$, existirá una sucesión manejable de puntos convergente a una sucesión óptima de $P(c)$. Más precisamente, si $\{P(c)\}$ es una s.m. monótona particular de subproblemas de $P(c)$, existirá una s.m. $\{x^{k_r}\}$ de soluciones óptimas de los subproblemas correspondientes $P_{k_r}(c)$, $r = 1, 2, \dots$, que converge a una solución óptima $\bar{x}$ de $P(c)$.*

Corolario 2.25. *Si $c \in \text{int}M$, toda s.m. de puntos para $P(c)$ asociado con una s.m. monótona de subproblemas tiene puntos de acumulación que son necesariamente puntos óptimos de este problema. Por otro lado, si $c \in \text{rint}M$*

y la última propiedad se cumple entonces tenemos que $\text{int}M \neq \emptyset$.

Teorema 2.26. Si $F \neq \emptyset$, los siguientes enunciados son equivalentes:

1. F es acotado;
2. $\begin{pmatrix} 0_n \\ -1 \end{pmatrix} \in \text{int}K$, o equivalentemente

$$\text{int}K = \left\{ \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^{n+1} \mid a'x > b, \forall x \in F \right\};$$

3. $M = \mathbb{R}^n$;
4. existe un subsistema finito de σ cuyo conjunto factible es acotado;
5. $\cup\{\text{supp}\lambda, \lambda \in 0^+\Lambda\} = T$, y la suposición de rango completo se cumple.

Corolario 2.27. Dado un problema de PLSI consistente (P) , las siguientes condiciones son equivalentes:

1. F^* es un conjunto acotado no vacío;
2. (P) es acotado y los conjuntos de nivel no vacíos de (P) son acotados;
3. existe un subproblema finito de (P) cuyos conjuntos de nivel no vacíos son acotados;
4. $c \in \text{int}M$;
5. existe $\varepsilon > 0$ tal que el conjunto óptimo del problema perturbado (P_1) $\inf(c^1)'x$ s.a. x sea una solución para σ , es no vacío y acotado si $\|c - c^1\| < \varepsilon$.

Capítulo 3

Métodos de planos de corte central

3.1. Discretizando mediante planos de corte

Los métodos de planos de corte originalmente fueron creados para solucionar problemas de la Programación No Lineal (PNL), por lo que comenzaremos hablando de dichos problemas.

Definiremos al problema convexo (PC) de PNL como sigue:

$$\begin{aligned} (PC) \quad & \inf \quad c'x \\ \text{s.a.} \quad & x \in F = \{x \in \mathbb{R}^n | g_i(x) \leq 0, i = 1, \dots, m\}, \end{aligned}$$

donde todas las funciones g_i , $i = 1, \dots, m$, son convexas en $\mathbb{R}^n$.

La iteración número r del método de planos de corte comienza con un conjunto convexo poliédrico $F_r \supset F$, luego calcula una solución óptima x^r del problema auxiliar de Programación Lineal (PL)

$$\begin{aligned} & \inf \quad c'x \\ \text{s.a.} \quad & x \in F_r, \end{aligned}$$

y define $F_{r+1} = \{x \in F_r | a'x \geq b\}$ para algún semiespacio que contiene a F , $a'x \geq b$, cuya frontera separa x^r de F , a menos que $x^r \in F$ (en este caso x^r es una solución óptima de (PC)).

El corte de factibilidad $a'x \geq b$ puede obtenerse de diferentes formas, una de ellas puede ser la siguiente: mediante cualquier semiespacio de la forma $g_i(x^r) + d'(x - x^r) \leq 0$, con $g_i(x^r) > 0$ y $d \in \mathbb{R}^n$ elegido arbitrariamente en $\partial g_i(x^r)$.

La distancia de x^r a tal semiespacio es $\|d\|^{-1}g_i(x^r) > 0$, y ésta es una medida de la profundidad del corte.

Cheney y Goldstein (1959), e independientemente Kelley (1960), propusieron seleccionar i como el índice de la restricción más violada por x^r y $d = \nabla g_i(x^r)$, bajo la suposición previa de la diferenciabilidad de $g_1, \dots, g_m$.

Esta idea puede adaptarse a la PLSI mediante la selección de cortes de factibilidad entre las restricciones de (P) , es decir, tomando un $t \in T$ tal que $a'_t x \geq b_t$ es una de la restricciones más violadas por x^r .

Si el sistema de restricciones ha sido previamente normalizado, es decir, $\|a_t\| = 1$, para toda $t \in T$, el proceso consiste en calcular el corte más profundo.

3.2. El método de planos de corte central

El algoritmo de PCC intenta resolver un problema común de los algoritmos de discretización: la terminación finita en puntos no factibles.

Por otro lado, intenta resolver un problema particular del algoritmo de planos de corte: que los planos de corte generados tienden a ser paralelamente cercanos al plano tangente a la región factible en el punto óptimo, provocando que los puntos óptimos de las iteraciones siguientes se localicen en vértices muy lejanos, creando con esto inestabilidad y acumulamiento.

Hettich y Zencke (1982) probaron que, bajo una cierta suposición de regularidad relacionada a la condición de Slater, la unicidad fuerte implica la convergencia superlineal de una gran clase de algoritmos.

Un inconveniente de los métodos de discretización es que el conjunto de índices del problema primal, en este caso T , debe tener buenas propiedades, por ejemplo, que $T = \{t \in \mathbb{R} | 0 \leq t \leq 1\}$, y generalmente fallan cuando dicho T no es acotado.

El método de planos de corte central (o de punto interior, o de incrustación) de Elzinga y Moore (1975) fue diseñado para eliminar los inconvenientes en el tratamiento numérico del problema (PC).

Ahora daremos una breve descripción de su iteración número r .

Sea F_r un politopo que contiene algún conjunto de nivel diferente del vacío de (PC) . Esto es sólo posible cuando F^* es un conjunto acotado diferente del vacío, pero F no necesita estar en F_r .

Sea x^r el centro de la bola más grande contenida en F_r (su radio ρ_r será cero si y sólo si $\text{int}F_r = \emptyset$).

Esencialmente, el siguiente politopo F_{r+1} se construye a partir de F_r como sigue:

1. Si $x^r \in F$, tomamos $F_{r+1} = \{x \in F_r | c'x \leq c'x^r\}$; es decir, se realiza un *corte objetivo* sobre F_r .
2. Si $x^r \notin F$ entonces $F_{r+1} = \{x \in F_r | a'x \geq b\}$, donde $a'x \geq b$ es un *corte factible*, es decir, $a'x \geq b$ para todo $x \in F$ y $a'x^r < b$. Este corte factible no necesita corresponder a la restricción más violada.

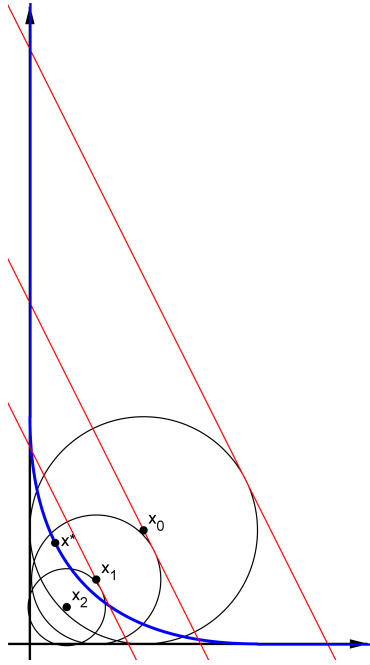


Figura 3.1: Cortes objetivos.

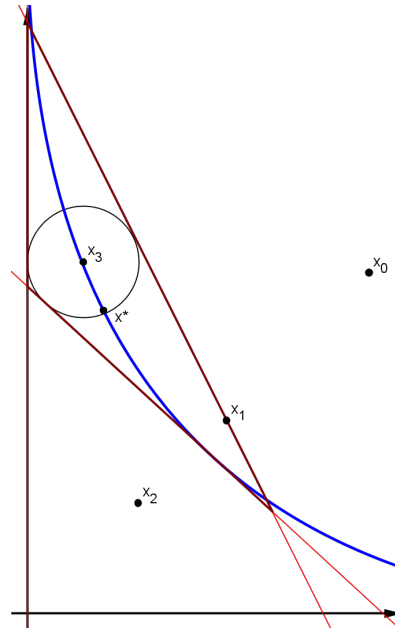


Figura 3.2: Cortes factibles.

En ambos casos, F_{r+1} es otra vez un politopo que contiene un conjunto de nivel diferente del vacío de (PC) .

Para evitar la acumulación de cortes en los pasos siguientes, pueden agregarse reglas de borrado, las cuales eliminarán cortes previos que no afecten significativamente el progreso del algoritmo en iteraciones posteriores.

3.2.1. El algoritmo del método de PCC

En el algoritmo del método de planos de corte central para el problema de PLSI (P), todos los cortes factibles serán una restricción $a'_t x \geq b_t$, para algún $t \in T$, tal que los politopos siguientes tendrán la forma

$$F_r = \{x \in \mathbb{R}^n | c'x \leq \alpha_r; a'_t x \geq b_t, \quad p.a. \quad t \in T_r\},$$

y una bola maximal contenida en F_r será $x_r + \rho_r(clB)$ para alguna solución óptima (x_r, ρ_r) , del problema auxiliar de PL

$$\begin{aligned} (Q_r) \quad & \text{Sup} \quad \rho \\ & s.a. \quad c'x + \|c\|\rho \leq \alpha_r, \\ & a'_t x - \|a_t\|\rho \geq b_t, \quad t \in T_r. \end{aligned}$$

Consecuentemente, el algoritmo resuelve una sucesión de problemas de PL asociados con pares de rejillas T_r y escalares α_r .

Algoritmo 1. Sean z^0 un punto tal que $a'_t z^0 \geq b_t$ para todo $t \in S$ y $c'z^0 \geq v(P)$, una tolerancia $\varepsilon \geq 0$ y un escalar $\beta \in]0, 1[$.

Inicializamos el índice de iteración $r = 0$ y definimos $\alpha_0 = c'z^0$ y $T_0 = S$.

Paso 1: Calcular una solución óptima (x^r, ρ_r) de

$$\begin{aligned} (Q_r) \quad & \text{Sup} \quad \rho \\ & s.a. \quad c'x + \|c\|\rho \leq \alpha_r, \\ & a'_t x - \|a_t\|\rho \geq b_t, \quad t \in T_r. \end{aligned}$$

Si $\rho_r \leq \varepsilon$ entonces se detiene.

De otra manera, ir al Paso 2.

Paso 2. Calcular $s_r = \inf_{t \in T} g(t, x^r)$.

Si $s_r \geq 0$, reemplazar r por $r + 1$, tomar $\alpha_{r+1} = c'x^r$ y $T_{r+1} = T_r$.

Entonces ir al Paso 1.

Si $s_r < 0$, encontrar un índice $t_r \in T$ tal que $g(t_r, x^r) < 0$, reemplazar r por $r + 1$, tomar $\alpha_{r+1} = \alpha_r$ y $T'_{r+1} = T_r \cup \{t_r\}$; entonces ir al Paso 3.

Paso 3. Sea S_r el conjunto de índices $t \in T'_{r+1}$ tal que $a'_t x \geq b_t$ es un corte factible introducido en alguna iteración $p < r$, que $\rho_r \leq \beta \rho_p$ y que $a'_t x^r - \|a_t\| \rho_t > b_t$.

Es decir que la distancia de x^r al hiperplano $a'_t x = b_t$ es mayor que ρ_t .

Ahora tomemos

$$T_{r+1} = T'_{r+1} \setminus S_r,$$

y vamos al Paso 1.

Hay algunos detalles que hay que aclarar y demostrar respecto a elementos del algoritmo que acabamos de presentar.

Notemos que la condición $c'z^0 \geq v(P)$ implica la consistencia de (P) y entonces la compacidad de $F^* \neq \emptyset$.

Por otra parte, ya que $F^* \subset F_r \subset F_0$, F_r es un politopo para todo $r \in \mathbb{N}$, tal que (Q_r) tiene una solución óptima (x^r, ρ_r) , con $\rho_r \geq 0$, es decir, el algoritmo está bien definido.

Observar también que $\{\alpha_r\}$ es una sucesión no creciente, ya que, en el Paso 2, $\alpha_{r+1} = \alpha_r$ o $\alpha_{r+1} = c'x^r \leq \alpha_r - \|c\| \rho_r < \alpha_r$. Por otro lado, ésta es acotada inferiormente por $v(P)$ pues los cortes objetivos son realizados a través de puntos factibles.

Para probar que $\{\rho_r\}$ tiende a cero, necesitamos mostrar que esta sucesión es no creciente, a pesar de las reglas de borrado.

El efecto de las reglas de borrado en el Paso 2 es eliminar, en la representación lineal de F_{r+1} , una restricción redundante (un corte objetivo anterior), tal que esta regla preserve la contracción de $\{F_r\}$.

El efecto de las reglas de borrado del Paso 3 es el que sigue: una vez que se determina un nuevo corte en el Paso 2, se eliminan ciertas restricciones (cortes factibles viejos), en la representación lineal de F_r . Ninguna de estas restricciones será cercana a la bola máxima inscrita $x^r + \rho_r(clB)$. Luego se agrega el nuevo corte para obtener F_{r+1} .

Probaremos ahora que borrar una de estas restricciones no altera el tamaño de la bola escrita más grande en F_r mediante el principio de inducción, y así la sucesión $\{\rho_r\}$ será no creciente.

Sea $F_r = \{x \in \mathbb{R}^n | c'_u x \geq d_u, u \in U\}$, $|U| < \infty$, con $c_u \neq 0_n$, para todo $u \in U$. Asumimos que la distancia de x^r al hiperplano $c'_v x = d_v$ es mayor que ρ_r .

Entonces

$$c'_u x^r - \|c_u\| \rho_r \geq d_u \quad \forall u \in U,$$

con la desigualdad estricta para el caso $u = v$. Ahora asumimos que

$$z^r + \mu_r(clB) \subset G_r := \{x \in \mathbb{R}^n | c'_u x \geq d_u, u \in U \setminus \{v\}\},$$

para ciertos z^r y $\mu_r > \rho_r$ obtendremos una contradicción. Dado que la última bola es incluida en G_r pero no en F_r , tenemos

$$c'_u x^r - \|c_u\| \mu_r \geq d_u \quad \forall u \in U \setminus \{v\},$$

donde $c'_v z_v - \|c_v\| \mu_r < d_v$.

Notemos que $x(\lambda) = (1 - \lambda)x^r + \lambda z^r$, con $\lambda \in [0, 1]$. Como

$$\lim_{\lambda \searrow 0} c'_v x(\lambda) > d_v + \|c_v\| \rho_r,$$

existe $\bar{\lambda} \in]0, 1[$ y $\varepsilon_1 > 0$ tal que

$$c'_v x(\bar{\lambda}) > d_v + (\rho_r + \varepsilon_1) \|c_v\|.$$

Por otro lado, existe un $\varepsilon_2 > 0$ tal que, para todo $u \in U \setminus \{v\}$,

$$c'_u x(\bar{\lambda}) > d_u + [(1 - \bar{\lambda})\rho_r + \bar{\lambda}\mu_r] \|c_u\| > d_u + (\rho_r + \varepsilon_2) \|c_u\|.$$

Entonces, definiendo $\varepsilon_3 = \min\{\varepsilon_1, \varepsilon_2\}$, llegamos a la conclusión contradictoria de que $x(\bar{\lambda}) + (\rho_r + \varepsilon_3)(clB) \subset F_r$.

Teorema 3.1. *Si el algoritmo anterior no termina entonces $\lim_r \rho_r = 0$.*

Demostración. Acabamos de probar, antes de enunciar este teorema, que $\{\rho_r\}$ es convergente, mediante la demostración de que se trata de una sucesión no creciente con todos sus términos mayores o iguales que cero. Ahora obtendremos una contradicción asumiendo que $\bar{\rho} := \lim_r \rho_r > 0$.

Dado que $\{\rho_r\}$ es una sucesión no creciente que converge a $\bar{\rho} < \beta^{-1}\bar{\rho}$, existe un $p \in \mathbb{N}$ tal que $\bar{\rho} < \rho_r < \rho_p < \beta^{-1}\bar{\rho}$, para todo $r \geq p$, así que $\rho_r > \beta\rho_p$ (de otra manera, tendríamos $\beta^{-1}\bar{\rho} < \beta^{-1}\rho_r < \rho_p < \beta^{-1}\bar{\rho}$).

De esta manera, para r mayor que cierto $p \in \mathbb{N}$, la regla de borrado del Paso 3 está inactiva, por lo que, $\{T_r\}$ llega a ser expansiva y $\{F_r\}$ contractiva.

Ahora probaremos que $\{x^r\}$ no contiene subsucesiones de Cauchy. Contradicción requerida ya que $\{x^r\} \subset F_0$ es acotada.

Sea $k > r \geq p$. Dado que $x^r + \rho_r(clB) \subset F_r \subset F_{r+1}$, (x^r, ρ_k) es factible para (Q_{r+1}) .

Dos casos podrían surgir en la iteración r (Paso 2):

- (a) Si $x^r \in F$, el corte objetivo $c'x \leq c'x^r$ fue realizado sobre F_r . Entonces $c'x^r + \|c\|\rho_k \leq \alpha_{r+1} = c'x^r$ y, por la desigualdad de Cauchy-Schwartz tenemos

$$\|c\|\rho_k \leq c'(x^r - x^k) \leq \|c\|\|x^r - x^k\|,$$

así que $\bar{\rho} \leq \rho_k \leq \|x^r - x^k\|$.

- (b) Si $x^r \notin F$, un corte factible $a'_t x \geq b_t$, $t \in T_{r+1} \subset T_k$ fue agregado a F_r , con $a'_t x^r < b_t$. Entonces $a_t \neq 0$ y $a'_t x^k - \|a_t\|\rho_k \geq b_t$, así que

$$\|a_t\|\rho_k < a'_t(x^r - x^k) \leq \|a_t\|\|x^r - x^k\|,$$

y otra vez ocurre que $\bar{\rho} \leq \rho_k \leq \|x^k - x^r\|$. ✓

El siguiente lema nos asegura que, bajo ciertas condiciones, el Algoritmo 1 converge a un punto óptimo del problema.

Lema 3.2. *Asumimos que $\sigma = \{a'_t x \geq b_t, t \in T\}$ tiene un Punto Fuerte de Slater y que a_t es acotada, con $t \in T$. Entonces*

1. *si $\varepsilon = 0$, el algoritmo 1 termina en la errésima iteración, para algún $r = 0, 1, \dots$ y con $\alpha_r = v(P)$, o*
2. *éste no termina, y $\lim_r \alpha_r = v(P)$.*

Demostración. Para comenzar notemos que la hipótesis implica que $\text{int}(F) \neq \emptyset$.

Sea $y^1 \in \mathbb{R}^n$ y $\eta > 0$ tal que $a'_t y^1 \geq b_t + \eta$, para todo $t \in T$.

1. Asumimos que $\rho_i > 0$, con $i = 0, 1, \dots, r-1$ y $\rho_r = 0$. Sabemos que $\alpha_t \geq v(P)$ por definición de α_r . Si $\alpha_r > v(P)$ entonces existirá un $y^2 \in F$ tal que $c'y^2 < \alpha_r$. Consideremos

$$x(\theta) := (1 - \theta)y^1 + \theta y^2, \quad \theta \in]0, 1[.$$

Ya que $\lim_{\theta \nearrow 1} c'x(\theta) = c'y^2 < \alpha_r$ y $a'_t x(\theta) \geq b_t + (1 - \theta)\eta$, pues $y^2 \in F$, implica que $a_t y^2 \geq b_t$.

Ahora, junto con la desigualdad $a'_t y^1 \geq b_t + \eta$ hacen la desigualdad deseada, para toda $\theta \in]0, 1[$ y toda $t \in T$.

Por lo que, podemos asumir la existencia de un Punto Fuerte de Slater para σ , $\bar{x}$, tal que $c'\bar{x} < \alpha_r$.

Sean $\delta > 0$ y $\eta > 0$ tal que $a'_t \bar{x} \geq b_t + \delta$ y $\eta > \|a_t\|$, para toda $t \in T$. Entonces, para toda $t \in T$, tenemos

$$a'_t \bar{x} - \mu^{-1} \delta \|a_t\| \geq a'_t \bar{x} - \delta \geq b_t,$$

tal que $\mu^{-1} \delta$ es una cota inferior de la distancia de $\bar{x}$ a $a'_t x = b_t$. Por otro lado $\|c\|^{-1}(\alpha_r - c' \bar{x} > 0)$ es la distancia de $\bar{x}$ a $c' x = \alpha_r$. Por lo tanto

$$0 < \min\{\mu^{-1} \delta, \|c\|^{-1}(\alpha_r - c' \bar{x})\} \leq \rho_r,$$

lo que contradice la finalización finita en la r -ésima iteración.

2. Si el Algoritmo 1 no termina, la sucesión no creciente infinita $\{\alpha_r\}$ tendrá un límite $\bar{\alpha}$.

Si $\bar{\alpha} > v(P)$, podemos reemplazar α_r por $\bar{\alpha}$ en el argumento previo, tal que existe un $\bar{x} \in F$, con $c' \bar{x} < \bar{\alpha}$, y un $\bar{\rho} > 0$, tal que la distancia de $\bar{x}$ a $c' x = \bar{\alpha}$ (por lo tanto, $c' x = \alpha_r \geq \bar{\alpha}$) y a $a'_t x = b_t$, con $t \in T$, es al menos $\bar{\rho}$.

Esto significa que $0 < \bar{\rho} \leq \rho_r$, para todo r , en contradicción con el resultado del teorema anterior, que asegura $\lim_r \rho_r = 0$.

Por lo tanto, como $\bar{\alpha}$ no es mayor que $v(P)$, tiene que ser igual. ✓

Finalmente, el teorema siguiente establece la convergencia de los puntos generados por el Algoritmo 1 a las soluciones óptimas de (P) .

Teorema 3.3. *Bajo las mismas condiciones del lema anterior, si $z^0 \in F$ entonces cierto z^r será su último elemento (y z^r también será una solución óptima de (P) si $\varepsilon = 0$) o los puntos de acumulación de $\{z^r\}$ existen y son soluciones óptimas.*

Demostración. Distinguiremos los dos casos del lema anterior, observando que $c' z^r = \alpha_r$, con $r = 0, 1, 2, \dots$

1. Si el algoritmo termina en un número r de pasos, entonces $c' z^r = v(P)$ y z^r está en F^* .
2. De otra manera, $\{z^r\}$ es una sucesión infinita incluida en un conjunto acotado F_0 , tal que ésta contiene una subsucesión que converge a un punto factible (pues $\{z^r\} \subset F$, con F cerrado). Dado que $\lim_r c' z^r = v(P)$, cualquier punto de acumulación de $\{z^r\}$ es también un punto óptimo. ✓

3.3. El método de planos de corte central acelerado

En el algoritmo de PCC se agrega un corte objetivo $c'x \leq c'x_r$ a un cierto conjunto de restricciones cada vez que se encuentra un punto factible x_r .

En el de PCCA se agrega un corte objetivo más profundo hasta que el subproblema de PL relacionado se vuelve no factible. En este caso, se obtiene una cota inferior para I o un nuevo punto factible para (P) .

La convergencia del algoritmo de PCCA se prueba esencialmente bajo las mismas condiciones del algoritmo de PCC.

La razón de convergencia entre los puntos factibles del algoritmo de PCCA es aún lineal, pero es la mitad de la razón de convergencia del algoritmo de PCC.

3.3.1. El algoritmo del método de planos de corte central acelerado

Ahora definiremos al problema de PLSI (P) como sigue:

$$(P) \quad I = \inf c'x, \quad x \in \mathbb{R}^n \\ a'(y)x \geq b(y), y \in Y \subset \mathbb{R}^m.$$

Esto es con base en la nomenclatura propuesta por Breto. Para continuar presentaremos una descripción formal del algoritmo de PCCA, estableciendo previamente las variables a usar en dicho algoritmo.

Sea $\beta_0 \geq I$ dado (por ejemplo $\beta_0 = c'z_0$, donde z_0 es un punto factible de (P)) y sea $S \subset Y$ un conjunto finito que genera el subproblema finito de (P) obtenido al reemplazar $F = \{x \in \mathbb{R}^n | a(y)'x \geq b(y), y \in Y\}$ por $\hat{F}_0 = \{x \in \mathbb{R}^n | a(y)'x \geq b(y), y \in S\}$, y que tiene conjuntos de nivel acotados no vacíos (lo cual implica $I > -\infty$).

Tal conjunto S existe si y sólo si el conjunto de soluciones óptimas de (P) , F^* , es un conjunto acotado no vacío.

Algoritmo 2. Definimos $Y_0 = S$, $r = 0$, $\alpha_0 = -\infty$, $\tau_0 = \beta_0 - h$, $t = 0$ y también se eligen $\varepsilon \geq 0$, $h > 0$, $\zeta > 0$, $\gamma \in]0, 1[$, $\xi \in]0, 1[$, $\omega \in]0, 1[$.

Paso 1. Intentar resolver el problema de PL

$$\begin{aligned} (Q_r) \quad & \text{Sup} \quad \rho, \quad x \in \mathbb{R}^n, \quad \rho \in \mathbb{R} \\ & \text{s.a.} \quad c'x + \|c\|\rho \leq \tau_r, \\ & a'(y)x - \|a(y)\|\rho \geq b(y), \quad \forall y \in Y_r. \end{aligned}$$

Si (Q_r) es no factible, fijar $t = 0$, $\alpha_r = \tau_r$, eliminar las marcas asignadas hasta ahora en el Paso 2 e ir al Paso 4.

Si (Q_r) es factible, calcular (x^r, ρ_r) , una solución óptima (con $\rho_r \geq 0$, necesariamente), e ir al Paso 2.

Paso 2. Calcular $s_r = \inf_{y \in Y} s(y, x^r)$, donde $s(y, x^r)$ es la función de holgura $a(y)'x - b(y)$.

Si $s_r \geq 0$, entonces redefinir $Y_{r+1} = Y_r$, $\alpha_{r+1} = \alpha_r$, $\beta_{r+1} = c'x^r$, reemplazar r por $r + 1$, definir $t = 0$, eliminar las marcas asignadas hasta ahora y

-Si $\alpha_r = -\infty$, definir $\tau_r = \beta_r - h$ e ir al Paso 1;

-Si $\alpha_r > -\infty$, ir al Paso 4.

Si $s_r < 0$, entonces encontrar y^r tal que $s(y^r, x^r) \leq -\omega$ o $s(y^r, x^r) \leq \omega s_r$, redefinir $\alpha_{r+1} = \alpha_r$, $\beta_{r+1} = \beta_r$ y $\tilde{Y}_{r+1} = Y_r \cup \{y^r\}$;

-Si $t = 0$, redefinir $\tau_{r+1} = \tau_r - \zeta$ si $\alpha_r = -\infty$ o $\tau_{r+1} = \alpha_r + \xi(\tau_r - \alpha_r)$ de otra manera, $\tilde{\tau}_{r+1} = \tau_r$, $t = 1$ e ir al Paso 3.

-Si $t = 1$, definir $\tau_{r+1} = \tilde{\tau}_r$, $t = 0$ y $Y_{r+1} = \tilde{Y}_{r+1}$, marcar y^r como imborrable, reemplazar r por $r + 1$ e ir al Paso 1.

Paso 3. Sea S_r el conjunto de puntos $y \in Y_r$ los cuales han sido introducidos en alguna iteración $p < r$ y actualmente no son marcados como imborrables, con $\rho_r \leq \gamma \rho_p$ y $a'(y)x^r - \|a(y)\|\rho_r > b(y)$. Entonces, definir $Y_{r+1} = \tilde{Y}_{r+1} \setminus S_r$, reemplazar r por $r + 1$ e ir al Paso 1.

Paso 4. Si $\beta_r - \alpha_r \leq \varepsilon$, parar ($I \in (\alpha_r, \beta_r]$); de otra manera, redefinir $\tau_r = (\alpha_r + \beta_r)/2$ e ir al Paso 1.

Si definimos $h = \zeta = 0$, entonces el algoritmo de PCCA se reduce esencialmente al de PCC sin criterio de finalización. Ya que, en este caso, $\tau_r = \beta_r$ siempre y por lo tanto (Q_r) es factible, mientras que $\alpha_r = -\infty$, para todo r , así que el criterio de paro del Paso 4 nunca ocurrirá.

En el algoritmo original de PCC, cualquier y^r que cumpla $s(y^r, x^r) < 0$ puede ser elegido en el Paso 2, cuando $s_r < 0$. En el de PCCA, esto no será suficiente para asegurar la finalización del algoritmo ya que ahora pueden haber cortes objetivos que cumplan $\tau_r < I$. Notemos que el Paso 2 requiere la solución de un subproblema de optimización global (aunque no necesitemos un

mínimo global de $s(y, x^r)$), como es típico de la programación semi-infinita, asumiremos que un algoritmo de solución adecuada está disponible para dicha situación, el cual es el caso, por ejemplo, si Y es un intervalo acotado en $\mathbb{R}^m$.

El papel de la variable t en el algoritmo de PCCA es evitar que $\tau_r = \tau_{r+1}$, cuando tanto (Q_r) , como (Q_{r+1}) , son factibles; esto preveé el posible estancamiento del algoritmo cuando, para algún r , $\tau_r = I$, como se puede ver en la prueba del Teorema 2.1.

Ahora presentaremos algunas observaciones del algoritmo de PCCA:

1. En el Paso 1, como en el algoritmo de PCC, se buscan el centro y el radio de la esfera maximal inscrita en un politopo determinado por la intersección de un conjunto de nivel de $c'x$ con $F_r = \{x \in \mathbb{R}^n | a'(y)x \geq b(y), y \in Y_r\}$ son buscados, pero ahora, esta intersección puede ser vacía. Sin embargo, definiendo $\hat{F}_r = \{x \in F_r | c'x \leq \beta_r\}$, tenemos que $\hat{F}_r \subset \hat{F}_0$ (los puntos en S nunca son borrados en el Paso 3) es acotado (en realidad compacto, ya que es cerrado), por la acotación de $\hat{F}_0$, y no vacío.
2. Como en el algoritmo de PCC, la sucesión $\{\beta_r\}$ es no creciente; ya que, β_r es actualizado sólo en el Paso 2, ajustando $\beta_{r+1} = c'x^r \leq \beta_r - \|c\|\rho_r \leq \beta_r$. Más aún, $\beta_{r+1} \geq I$ pues, en su actualización en el Paso 2, x^r es factible para (P) , mientras que $\beta \geq I$ por definición.
3. La sucesión $\{\alpha_r\}$ es no decreciente, puesto que si $\alpha_{r+1} \neq \alpha_r$ entonces $\alpha_{r+1} = \tau_r > \alpha_r$.
4. Se cumple $\alpha \leq I$ para cada r , pues $\alpha_r > I$ podría implicar que $\alpha_r > c'x$, $x \in F^*$, pero esto contradice el hecho de que, por construcción, $\alpha_r < c'x$ para cualquier $x \in F_r \supset F^*$.
5. Podemos ver que en ambos algoritmos el paso 3 coincide.
6. El criterio de paro en el Paso 4 del algoritmo de PCCA permite controlar la precisión a la que I es aproximado, a diferencia del basado en el tamaño del radio ρ_r en el algoritmo original de PCC.
7. En lugar de la h fija, también puede ser considerada una sucesión positiva $\{h_j\}$, siempre que $\sum_j h_j = \infty$.

8. Partiendo del supuesto de S, los valores τ_r , para los cuales (Q_r) es factible, son acotados inferiormente por $\inf\{c'x : x \in \hat{F}_0\} > -\infty$.
9. En el Paso 4 es posible generalizar la elección de τ , considerando $\tau = \alpha_r + \phi(\beta_r - \alpha_r)$, con $\phi \in]0, 1[$ que no depende de r . Cuando ϕ es mas pequeño, mayor es la reducción en β_r una vez que (Q_r) es factible y $s_r \geq 0$. Entonces, sin embargo, mayor es también la posibilidad de obtener un (Q_r) no factible, tal que $\phi = 1/2$ puede ser visto como un valor de compensación.

Ahora veamos los teoremas que aseguran la convergencia del Algoritmo 2 presentados originalmente en [1].

Teorema 3.4. *Si $\mu = \sup_{y \in Y} \|a(y)\| < +\infty$ y si existe un punto $\hat{x}$ tal que $a'(y)\hat{x} \geq b(y) + \eta$ para algún y y algún $\eta > 0$ (es decir, se cumple la Condición Fuerte de Slater) entonces el algoritmo de PCCA con $\varepsilon = 0$ concluye que*

$$\lim_{r \rightarrow +\infty} \alpha_r = I = \lim_{r \rightarrow +\infty} \beta_r,$$

y cuando $\varepsilon > 0$ el algoritmo de PCCA termina en un número finito de pasos.

Demostración.

Esta prueba es basada en argumentos similares a los de la prueba de convergencia del algoritmo 1 (Teorema 2.1), así como en el teorema uno de [7].

Sea $\alpha = \lim_{r \rightarrow \infty} \alpha_r$ y $\beta = \lim_{r \rightarrow \infty} \beta_r$, donde, por lo visto en las observaciones anteriores, $\alpha \leq I \leq \beta$.

Ahora buscaremos una contradicción asumiendo que $\alpha < \beta$.

Si $\alpha_r = -\infty$, para algún r , entonces β_r no se puede actualizar infinitas veces.

De otra manera, para un r suficientemente grande, el problema (Q_r) se volvería no factible, causando la actualización de α_r a un valor finito, cuando $\tau_r \leq \beta_r - h$ y, por lo tanto, β_r es disminuido por lo menos h en cada actualización al valor $c'x^r \leq \tau_r$ (visto en la observación 8).

Si α_r es actualizado al menos una vez, entonces tanto α_r como β_r no se pueden actualizar un número infinito de veces, ya que después de la primera actualización de α_r cualquier modificación posterior de α_r o de β_r causa la

reducción de la diferencia $\beta_r - \alpha_r$ por un factor $\leq \max\{0,5, 1 - \xi/2\} < 1$.

Por lo tanto, en ambos casos, el algoritmo definitivamente cicla a través del Paso 1 con un estado de factibilidad para (Q_r) , y del Paso 2 con $s_r < 0$, agregando nuevos puntos y_r y quizá descartando viejos puntos en el Paso 3, mientras que τ_r cambia entre dos valores, digamos $\bar{\tau}$ y $\underline{\tau}$.

Asumiremos que tal circunstancia ocurre para $r \geq \bar{r}$. Probaremos que $\rho_r \rightarrow 0$ cuando $r \rightarrow \infty$.

Si $\bar{\rho}_r = \max\{\rho_r, \rho_{r+1}\}$, entonces $\bar{\rho}_r$ es no creciente, para $r \geq \bar{r}$.

En efecto, si $\tau_r = \bar{\tau}$ y ninguna cancelación ocurre en la iteración $r + 1$, podemos ver que

$$\rho_{r+1} \leq \rho_r$$

y

$$\rho_{r+2} \leq \rho_r,$$

entonces

$$\rho_{r+2} \leq \rho_r + 1,$$

de modo que

$$\bar{\rho}_r \geq \rho_{r+1},$$

mientras ocurra que

$$\rho_{r+1}^- = \rho_{r+1}.$$

Por lo tanto,

$$\rho_{r+1}^- \geq \bar{\rho}_r,$$

si ninguna cancelación ocurre ya sea en r o en $r + 1$.

Supongamos que en la iteración número r , con $\tau_r = \bar{\tau}$ y $\tilde{y} \in Y_r$, el elemento $\tilde{y}$ que fue introducido en la iteración $p < r$, con $\rho_r < \gamma\rho_p$ y $a'(\tilde{y})x^r - \|a(\tilde{y})\|\rho_r > b(\tilde{y})$, es descartado.

Ahora consideremos el problema

$$\begin{aligned} (\tilde{Q}_r) \quad & \sup \rho \\ \text{s.a.} \quad & c'x + \|c\|\rho \leq \bar{\tau} \\ & a'(y)x - \|a(y)\|\rho \geq b(y), \quad \forall y \in Y_r \setminus \tilde{y} \end{aligned}$$

y definamos a $(\tilde{x}^r, \tilde{\rho}_r)$ como una solución óptima de $(\tilde{Q}_r)$.

Como (x^r, ρ_r) es factible para $(\tilde{Q}_r)$, entonces $\rho_r \leq \tilde{\rho}_r$.

Supongamos que

$$\rho_r < \tilde{\rho}_r;$$

esto implica que

$$(\tilde{x}^r, \tilde{\rho}_r)$$

no es factible para (Q_r) , tal que

$$a'(\tilde{y})x^r - \|a(\tilde{y})\|\tilde{\rho}_r < b(\tilde{y}),$$

mientras que

$$a'(\tilde{y})x^r - \|a(\tilde{y})\|\rho_r > b(\tilde{y}).$$

Definiendo

$$(x(\lambda), \rho(\lambda)) = \lambda(\tilde{x}^r, \tilde{\rho}_r) + (1 - \lambda)(x^r, \rho_r),$$

con $\lambda \in [0, 1]$, entonces para un $\lambda > 0$ suficientemente pequeño tenemos

$$a'(\tilde{y})x(\lambda) - \|a(\tilde{y})\|\rho(\lambda) \geq b(\tilde{y}),$$

tal que $(x(\lambda), \rho(\lambda))$ es factible para (Q_r) , con $\rho(\lambda) > \rho_r$, contra la optimalidad de ρ_r para Q_r .

Por lo tanto $\rho_r = \tilde{\rho}_r$.

Análogamente, la igualdad anterior se cumple por inducción si en Q_r , S_r toma el lugar de $\{\tilde{y}\}$.

Cuando las regiones factibles de Q_{r+1} y Q_{r+2} están contenidas en la de $\tilde{Q}_r$, se tiene que $\rho_{r+1} \leq \rho_r$ y $\rho_{r+2} \leq \rho_r$, de manera que, como antes vimos, tenemos finalmente que $\bar{\rho}_r \geq \bar{\rho}_{r+1}$.

Probamos que $\bar{\rho}_r \rightarrow 0$ y, por lo tanto, $\lim_{r \rightarrow \infty} \rho_r = 0$, asumiendo por contradicción que $\bar{\rho} = \lim_{r \rightarrow \infty} \bar{\rho}_r > 0$.

Como $\gamma \in]0, 1[$, entonces $\bar{\rho} < \gamma^{-1}\bar{\rho}$, tal que, para un r y un q suficientemente grandes, con $r \geq q \geq \bar{r}$, tenemos $\bar{\rho} \leq \bar{\rho}_r \leq \bar{\rho}_q < \gamma^{-1}\bar{\rho}$ y, por lo tanto,

$$\gamma\rho_q \leq \gamma\bar{\rho}_q < \bar{\rho} < \bar{\rho}_r.$$

Consecuentemente, en el Paso 3, como $\bar{\rho}_r = \rho_r$, nunca son borrados los puntos y introducidos en cualquier iteración $p \geq q$.

Sean k y r cualquier par de enteros, tales que $k > r \geq q$ y $\tau_r = \tau_k = \bar{\tau}$. Entonces el Paso 2 da

$$Y_{r+1} = \tilde{Y}_{r+1} = Y_r \cup \{y^r\}$$

con $a'(y^r)x^r < b(y^r)$, mientras que en la iteración k tenemos

$$a'(y^r)x^k \geq \|a(y^r)\|\rho_k + b(y^r).$$

Entonces,

$$a(y^r) \neq 0_n$$

y

$$0 < \bar{\rho} \leq \bar{\rho}_k = \rho_k \leq \frac{a'(y^r)}{\|a'(y^r)\|}(x^k - x^r) \leq \|x^k - x^r\|.$$

Así que $\{x^r, r \geq q \mid \tau_r = \bar{\tau}\}$ no contiene sucesiones de Cauchy, contradiciendo el hecho de que los puntos x^r caen en el conjunto compacto $\hat{F}_0$.

Por lo tanto $\bar{\rho} = \lim_{r \rightarrow \infty} \bar{\rho}_r = 0$.

Consideremos ahora la ubicación de $\bar{\tau}$ respectivo a I .

Si $\bar{\tau} > I$, existe

$$\tilde{x} \in F = \{x \in \mathbb{R}^n \mid a'(y)x > b(y), y \in Y\}$$

tal que

$$\bar{\tau} > c'\tilde{x}.$$

Definiendo

$$x(\lambda) = \lambda\tilde{x} + (1 - \lambda)\hat{x},$$

con $\lambda \in [0, 1]$, entonces como

$$c'x(\lambda) < \bar{\tau},$$

para λ suficientemente cerca a 1 y

$$a'(y)x(\lambda) \geq b(y) + (1 - \lambda)\eta,$$

existe un punto $\bar{x} = x(\bar{\lambda}) \in F$, para cualquier $y \in Y$, que satisface la Condición Fuerte de Slater con $\delta = (1 - \bar{\lambda})\eta > 0$ y tal que $\bar{\tau} > c'\bar{x}$.

Como $\mu < \infty$, tenemos

$$a'(y)\bar{x} - \delta\mu^{-1}\|a(y)\| \geq a'(y)\bar{x} - \delta \geq b(y),$$

para algún $y \in Y$ y para algún $\rho \leq \delta\mu^{-1}$.

Por otra parte, como $c'\bar{x} < \bar{\tau}$, tenemos que $c'\bar{x} + \|c\|\rho \leq \bar{\tau}$, para algún $\rho \leq (\bar{\tau} - c'\bar{x})\|c\|^{-1}$.

Por lo tanto, si $\bar{\rho} = \min\{\delta\mu^{-1}, (\bar{\tau} - c'\bar{x})\|c\|^{-1}\}$, resulta que $(\bar{x}, \bar{\rho})$ es factible para (Q_r) , para algún r , tal que $\rho_r = \bar{\rho}$, lo que implica que un número infinito de veces $\rho_r \geq \bar{\rho}$, contradiciendo que $\lim \rho_r = 0$.

Si $\bar{\tau} \leq I$, entonces $\underline{\tau} < I$; sea $r_i \geq \bar{r}$, con $1 = 1, 2, \dots$, tal que $\tau_{r_i} = \underline{\tau}$ y $\hat{Y}_{r_i} = Y_0 \cup \{y^{r_1}\} \cup \dots \cup \{y^{r_{i-1}}\}$ y consideremos el problema

$$\begin{aligned} (P_{r_i}) \quad & \inf c'x \\ \text{s.a.} \quad & a'(y)x \geq b(y), \quad \forall y \in Y_{r_i}. \end{aligned}$$

Los puntos en Y_{r_i} , marcados en el Paso 2, nunca son borrados en el Paso 3, así que x^{r_i} es factible para (P_{r_i}) y $c'x^{r_i} \leq \tau_{r_i} - \|c\|\rho_{r_i} \leq \underline{\tau} < I$.

Cuando los puntos y_{r_i} satisfacen la Condición de Hu, podemos presentar los mismos argumentos de la prueba del Teorema 1 de [7] y obtener $c'x^{r_i} \rightarrow I$, lo que contradice que $c'x^{r_i} \leq \underline{\tau} < I$. ✓

Teorema 3.5. *Bajo las mismas condiciones del teorema anterior, si $\beta_0 > I$ y $\varepsilon = 0$, entonces el algoritmo de PCCA genera una sucesión $\{z^s\}$, con $s \geq 1$, de puntos en F .*

- Si la sucesión $\{z^s\}$ es finita, entonces su último elemento es una solución óptima de (P) .
- Si la sucesión $\{z^s\}$ es infinita, entonces sus puntos de acumulación son solución óptima de (P) .

Demostración.

Como $\beta_0 > I$, el hecho de que $\beta_r \rightarrow I$, como una consecuencia del teorema anterior, implica que β_r es actualizado al menos una vez.

La primera actualización de β_r ocurre para $r = r_1$, la segunda para $r = r_2$ y así sucesivamente.

Entonces, $\beta_{r_i+1} = c'x^{r_i}$, con s_{r_i} , es decir $x^{r_i} \in F$.

Ahora definamos $z^i = x^{r_i}$.

Si $\{z^i\}$ es una sucesión finita, y z^l es su último elemento, entonces necesariamente $I = \beta_{r_l+1} = c'x^{r_l} = c'z^l$.

Si no es finita, como $z^i \in \hat{F}_0$ compacto, entonces los puntos de acumulación existen y cualquiera de ellos, digamos z^* , es tal que $c'z^* = \lim c'z^i = \lim \beta_{r_i} = I$. ✓

El teorema siguiente establece la razón de convergencia de $\{c'z^s\}$, refinando el resultado obvio de que $c'z^{s+1} - I \leq \frac{1}{2}(z^s - I)$ tan pronto como $\alpha_s > -\infty$.

Teorema 3.6. *Bajo las mismas condiciones del Teorema 3.4, asumiendo que $\varepsilon = 0$, existe $\psi > 0$ tal que para cualquier $s \geq 1$*

$$c'z^{s+1} - I \leq (1 - \psi)(c'z^s - I). \quad (3.1)$$

Por otra parte, si s es suficientemente grande tal que $\alpha_s > -\infty$, entonces

$$c'z^{s+1} - I \leq \frac{1}{2}(1 - \psi)(c'z^s - I). \quad (3.2)$$

Demostración. Sean $\hat{x}$, η , μ como en el Teorema 2.4.

Cuando $\hat{x} \notin F^*$ tenemos que $c'\hat{x} > I$.

De otra manera, se tiene que

$$\tilde{x} = \hat{x} - \eta\mu^{-1}c/\|c\| \in F$$

y

$$c'\tilde{x} = I - \eta\mu^{-1}\|c\| < I.$$

Para $x^* \in F^*$ fijo, definimos $x(\lambda) = x^* + \lambda(\hat{x} - x^*)$.

Para cualquier $\lambda \in]0, 1]$, $x(\lambda)$ es un punto Fuerte de Slater con constante $\lambda\eta$.

Para $s \geq 1$, sea r_s el índice de la iteración en la cual z^s fue obtenida.
Para s fija, si $c'z^{s+1}$ es igual a I , entonces tanto (2.1), como (2.2), se cumplen.

Asumimos que $c'z^{s+1} > I$.

Si $c'\hat{x} < \tau_{r_{s+1}}$, entonces para algún $\lambda \in]0, 1]$ tenemos que $c'x(\lambda) < \tau_{r_{s+1}}$.

Si $c'\hat{x} \geq \tau_{r_{s+1}}$, entonces $c'x(\lambda) < \tau_{r_{s+1}}$, para $0 < \lambda < (\tau_{r_{s+1}} - I)(c'\hat{x} - I)^{-1}$.

Definiendo

$$\lambda^s = \nu(\tau_{r_{s+1}} - I)(c'\hat{x} - I)^{-1},$$

$0 < \nu < I$ es suficientemente pequeña.

Como $\tau_{r_{s+1}} - I$ está acotado superiormente por $\beta_0 - I$, tenemos que $\lambda^s \leq 1$, cuando $c'\hat{x} < \tau_{r_{s+1}}$.

De otra forma, tenemos que $\lambda^s \leq (\tau_{r_{s+1}} - I)(c'\hat{x} - I)^{-1}$.

Por otra parte, cuando $\rho \leq \lambda^s \eta \mu^{-1}$,

$$\begin{aligned} c'x(\lambda^s) &= I + \nu(\tau_{r_{s+1}} - I)(c'\hat{x} - I)^{-1}(c'\hat{x} - I) = I + \nu(\tau_{r_{s+1}} - I), \\ a'(y)x(\lambda^s) - \|a(y)\|\rho &\geq b(y), \end{aligned}$$

con $y \in Y$, y por lo tanto

$$c'x(\lambda^s) + \|c\|\rho \leq \tau_{r_{s+1}},$$

cuando

$$\rho \leq (\tau_{r_{s+1}} - c'x(\lambda^s))\|c\|^{-1} = (1 - \nu)(\tau_{r_{s+1}} - I)\|c\|^{-1}.$$

Esto implica que en la iteración r_{s+1} , cuando z^{s+1} es obtenido, definiendo

$$\psi = \min\{\nu\eta\mu^{-1}(c'\hat{x} - I)^{-1}, (1 - \nu)\|c\|^{-1}\},$$

por lo tanto,

$$\|c\|\rho_{r_{s+1}} \geq \psi(\tau_{r_{s+1}} - I).$$

Por otra parte, tenemos que

$$c'z^{s+1} + \|c\|\rho_{r_{s+1}} \leq \tau_{r_{s+1}},$$

esto es

$$\|c\|\rho_{r_{s+1}} \leq \tau_{r_{s+1}} - c'z^{s+1},$$

tal que

$$\tau_{r_{s+1}} - c'z^{s+1} \geq \psi(\tau_{r_{s+1}} - I)$$

y, por lo tanto,

$$(\tau_{r_{s+1}} - I)(1 - \psi) \geq c'z^{s+1} - I.$$

Por esto, como, para cualquier s , se tiene por definición

$$\tau_{r_{s+1}} \leq c'z^s,$$

obtenemos que

$$c'z^{s+1} - I \leq (1 - \psi)(c'z^s - I),$$

lo que prueba (2.1).

Sin embargo, en cuanto $\alpha_s > -\infty$, se tiene que

$$\tau_{r_{s+1}} \leq \frac{1}{2}(I + c'z^s)$$

y, por lo tanto,

$$c'z^{s+1} - I \leq \frac{1}{2}(c'z^s - I),$$

probando (2.2).✓

Observemos que (2.1) coincide con la razón de convergencia del algoritmo de PCC, pero (2.2) dice que la razón de convergencia del Algoritmo de PC-CA es la mitad del de PCC.

Capítulo 4

Pruebas numéricas

En [4] se presentan ejemplos, los cuales sirven para ilustrar el desempeño de los métodos presentados en el mismo (el método de discretización mediante rejillas, el método de discretización mediante planos de corte, el método de planos de corte central).

Bruno Breto, en [1], usa uno de los ejemplos que se presentan en [4] para comparar sus resultados con los de Goberna. Aquí presentaremos dos de los ejemplos empleados por Goberna, para poder mostrar el desempeño de los programas que creamos en MatLab.

Dichos ejemplos satisfacen las condiciones que requieren ambos métodos para su ejecución, como se muestra en [4] y [1].

Ejemplo 4.1. *Consideremos el siguiente problema:*

$$\begin{aligned} (P^1) \quad & \text{Inf} \quad 2x_1 + x_2 \\ \text{s.a.} \quad & tx_1 + (1-t)x_2 \geq t - t^2, \quad t \in [0, 1]. \end{aligned}$$

El conjunto factible de (P^1) , F^1 , es la intersección de las epígrafes de un continuo de funciones lineales con pendientes negativas. F^1 puede ser identificado de la envoltura de un grupo de líneas o, equivalentemente, mediante la solución singular de una ecuación diferencial Clairaut cuya solución general es $y = cx + c/(c-1)$ (véase [4], página 4), con $c < 0$, obteniendo a través de un cambio de parámetros: $c = t/(t-1)$. Por lo tanto, la ecuación es $y = xy' + y'/(y'-1)$, y su solución singular, después de la eliminación del parámetro, es el arco astroide $x^{1/2} + y^{1/2} = 1$, con $x > 0$ e $y > 0$. De aquí se

tiene que F^1 sea la epígrafe de una función convexa

$$h(x_1) = \begin{cases} +\infty & (x_1 < 0), \\ 1 + x_1 - 2(x_1)^{1/2} & (0 \leq x_1 \leq 1), \\ 0 & (1 < x_1), \end{cases}$$

así como la suma algebraica de la envoltura convexa de los puntos extremos de F^1 con el cono convexo generado por las dos direcciones extremas (como se puede ver en la Figura 4.1).

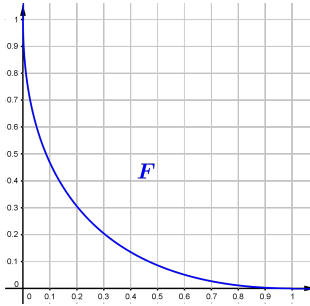


Figura 4.1: Región factible de (P^1) , F^1 .

Por otro lado, es importante mencionar que (P^1) tiene una solución óptima única, $x^* = (1/9, 4/9)^T$, con $v(x^*) = 2/3$.

Este ejemplo lo hemos resuelto con el algoritmo programado en MatLab del método de PCC, el cual se presenta en el Apéndice A, con una tolerancia de 10^{-7} . Se requirieron de 30 iteraciones para obtener el punto óptimo, las últimas 10 se presentan en la Figura 4.2.

El mismo ejemplo se resolvió también con un algoritmo basado en el método de PCCA y programado en MatLab. Se requirió el mismo número de iteraciones que en el algoritmo del método de PCC para obtener el óptimo y las últimas 10 iteraciones se muestran en la Figura 4.3.

A pesar de presentar el mismo número de iteraciones, los últimos valores que muestra el algoritmo del método de PCCA se acercan más al valor óptimo que los entregados por el del método de PCC, es decir, que cada vecindad alrededor de estos puntos, propuesta en cada iteración, es más pequeña que las presentadas por el algoritmo del método de PCC.

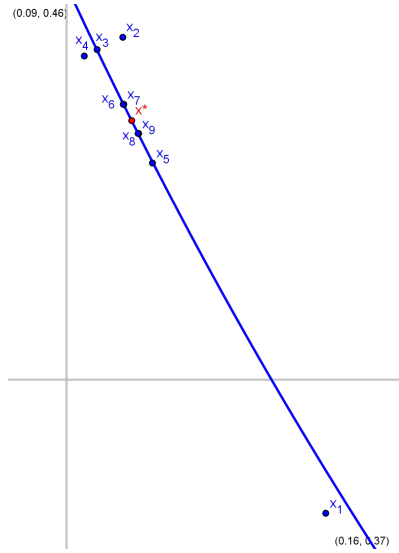


Figura 4.2: Últimas iteraciones del método de PCC.

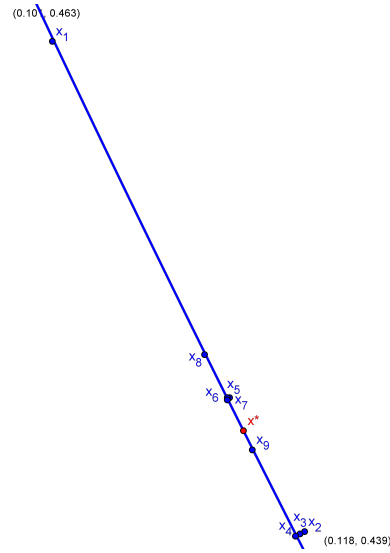


Figura 4.3: Últimas iteraciones del método de PCCA.

Otro aspecto a resaltar es que los valores expuestos por el método de PCCA son más cercanos a la frontera de la región factible, generando con esto cortes factibles que acercan de mejor manera cada F_r a F , acelerando así la convergencia al óptimo.

Ejemplo 4.2. Consideremos el cálculo de polinomios, de grado menor que n , de menor L_1 -aproximación a $\tan(t)$, con $t \in [0, 1]$. Formularemos el problema como la minimización de $c'x$, con

$$c_i = \int_0^1 t^{i-1} dt = i^{-1}, \quad i = 1, \dots, n,$$

para todos los polinomios factibles $P_n(t) = \sum_{i=1}^n x_i t^{i-1}$, es decir,

$$\begin{aligned} (P^n) \quad & \inf \sum_{i=1}^n i^{-1} x_i \\ \text{s.a.} \quad & \sum_{i=1}^n t^{i-1} x_i \geq \tan t, \quad t \in [0, 1]. \end{aligned}$$

Como una consecuencia del Teorema 3 en [9], este problema tiene una única solución óptima, para todo n natural, que puede ser calculada explícitamente resolviendo un sistema lineal obtenido de las condiciones específicas de optimalidad dadas en el Teorema 4 del artículo mencionado antes.

Tomaremos como primer ejemplo $n = 2$, por lo que, el problema a resolver es:

$$(P^2) \inf \quad x_1 + \frac{1}{2}x_2$$

$$s.a. \quad x_1 + tx_2 \geq \tan(t), \quad t \in [0, 1].$$

La solución de este problema es $x^* = (0, \tan(1))^T$, con $v(P^2) = \tan(1)/2$. Lo hemos resuelto, igualmente que el anterior, con ambos algoritmos, con una tolerancia de 10^{-7} .

Con el algoritmo del método de PCC se realizaron 32 iteraciones, en comparación con el de PCCA, en el que se realizaron 24.

En este ejemplo se pudieron notar cambios al modificar los parámetros propuestos para el método de PCCA. Por ejemplo, al modificar el valor de ξ , de 0.5 a 0.75 o 0.25, el número de iteraciones del algoritmo se incrementa, e incluso llega a provocar un estancamiento del mismo. La influencia de los parámetros en el desempeño del algoritmo se presenta a detalle en [1].

Ahora tomaremos $n = 8$, por lo que el problema es:

$$(P^8) \quad \inf \quad \sum_{i=1}^8 i^{-1}x_i$$

$$s.a. \quad \sum_{i=1}^8 t^{i-1}x_i \geq \tan(t), \quad t \in [0, 1].$$

Como ya habíamos mencionado antes, (P^8) admite una solución óptima única y cumple la condición fuerte de Slater, sin embargo, para $n = 8$, la estimación numérica del punto óptimo es más difícil, por lo que es un buen problema para evaluar el desempeño de los algoritmos de PLSI. Breto, en [1], reporta que corrió su algoritmo bajo las siguientes condiciones: $S = \{0, 1\} \cup \{\text{seis elementos al azar de } (0, 1)\}$, $\beta_0 = \sum_{i=1}^8 i^{-1}$, $\varepsilon = 10^{-10}$, $h = \beta_0/2$, $\zeta = 1$, $\gamma = 0.5$, $\xi = 0.5$, $\omega = 10^{-4}$. Se tomaron estas mismas condiciones para correr el algoritmo programado en Matlab con el fin de hacer una comparación. En

[1] se reporta que el algoritmo acelerado se detuvo cuando alcanzó la iteración número 39, mientras que nuestro algoritmo lo hizo en 35 iteraciones.

Como conclusión de este capítulo, al revisar los resultados de los ejemplos presentados aquí, se observó que los algoritmos presentan una sucesión de puntos tanto factibles, como no factibles, pero en el algoritmo del método de PCC se presentaron en su mayoría puntos no factibles, lo que dificulta al algoritmo determinar una ε -optimización.

Otra observación importante fue que el tamaño de la tolerancia no influyó negativamente en la finalización en un número finito de pasos de ambos algoritmos, aunque sí es un factor importante para determinar el número de iteraciones por realizar.

Por otro lado, se pudo observar en el segundo ejemplo, cuando $n = 2$, que es muy notoria la reducción de iteraciones al usar el algoritmo acelerado (de 32 iteraciones del método de PCC a 24 del método de PCCA).

Algo importante que destacar es que se debe poner cuidado en la declaración de los valores iniciales de los parámetros propuestos en el método acelerado, pues estos son importantes para el desarrollo y la convergencia del algoritmo. El rendimiento del algoritmo se ve modificado al cambiar los valores iniciales incrementando el número de iteraciones realizadas, impidiendo la finalización en un número finito de pasos, creando estancamientos o generando errores computacionales.

Finalmente, en los algoritmos programados para esta tesis se usó la función *linprog* de MatLab para resolver los subproblemas finitos de cada iteración, por lo que es importante tomar en cuenta los posibles errores de origen en su programación. Otra función usada en los programas es la de *fminbnd*. Ésta ha presentado algunos errores de programación en su origen, los cuales han sido presenciados por el Dr. Enrique González Gutiérrez, y que deben tomarse en cuenta. El problema de encontrar el mínimo de una función en un intervalo compacto no fue objeto de estudio en esta tesis, por lo que decidimos usar la función proporcionada por MatLab para resolverlo.

Capítulo 5

Conclusiones

El objetivo de esta tesis fue valorar el desempeño del método de planos de corte central acelerado propuesto por Breto en [1]. Para este fin, se inició con la presentación de una revisión de conceptos básicos que sustentan los teoremas de convergencia del método en prueba. En el capítulo 2 se desarrolló la teoría básica y en el capítulo 3 se presentaron dos algoritmos, el de planos de corte central, propuesto por Elzinga en 1975 y el de planos de corte central acelerado, propuesto por Bruno Breto. En este mismo capítulo se presentaron los teoremas de convergencia para los dos algoritmos y se demostró que la razón de convergencia del algoritmo acelerado es la mitad de la del algoritmo original. En el capítulo 4 se presentaron los resultados que se obtuvieron al aplicar los algoritmos a tres ejemplos tomados de la literatura, así como el análisis del desempeño de los mismos para compararlos. Los algoritmos se programaron en Matlab y el código se puede encontrar en los apéndices de esta tesis. En general, el desempeño del algoritmo del método de PCCA fue mejor que el del método de PCC porque alcanzó el punto óptimo en un número menor de iteraciones en dos de los tres ejemplos, en uno de ellos el número de iteraciones fue el mismo en ambos algoritmos, pero se debe destacar que estas iteraciones presentaron valores más cercanos al óptimo y a la frontera de la región factible en el algoritmo del método de PCCA que en el de PCC. Además, se observó que el PCCA alcanzó el óptimo a través de una sucesión de puntos en su mayoría factibles, tal y como lo reportó Breto en [1], mientras que el algoritmo del método de PCC utiliza mayormente puntos no factibles. La programación del algoritmo de PCCA y su aplicación nos permitió observar la importancia de los valores iniciales de los parámetros que permiten el aceleramiento del PCC original, por lo que en el análisis del

desempeño se hizo una reflexión al respecto.

A pesar de que se logró valorar el desempeño de los algoritmos presentados en la tesis satisfactoriamente, quedaron puntos por tratar. Por ejemplo, el diseño de una interfaz que permita el ingreso de las funciones pertinentes, los parámetros iniciales y la salida de los resultados. Otro punto es la determinación en cada problema de valores iniciales óptimos que propicien un buen desempeño computacional del algoritmo, evitando así parámetros obsoletos para ciertos problemas. Además, como las funciones auxiliares pueden afectar el desempeño de los algoritmos, éstas podrían programarse por separado para un mayor control de las mismas. Por último, sería recomendable analizar los tiempos de cómputo del algoritmo para detectar pasos que pudieran entorpecer algún proceso.

Apéndice A

Programa en MatLab del método de PCC

A continuación se presenta el script desarrollado en MatLab del algoritmo del método de PCC.

```
format long
clear
c=[1,.5];

T0=[0,1];
AT=[at(T0(1));at(T0(2))];
z0=linprog(zeros(length(AT),1),-AT,-[bt(T0(1));bt(T0(2))])
f=[0;0;-1];
alpha=c*z0;
beta=.5;
T=[0,1;1,1];
r=1;
tol=.0000001;

gt=@(x)(@(t)x*[t^0,t^1]'-tan(t));

disp('Aqui comienza el algoritmo del MPCC')
long=1;
while(r<1000)
r
```

```
for i=1:length(T)+1
    if(i==1)
        B(i)=alpha;
    else
        B(i)=-bt(T(i-1));
        aux=at(T(i-1));
    end

    for j=1:length(c)+1
        if(i==1)
            if(j==length(c)+1)
                A(i,j)=norm(c);
            else
                A(i,j)=c(j);
            end
        else
            if(j==length(c)+1)
                A(i,j)=norm(aux);
            else
                A(i,j)=-aux(j);
            end
        end
    end
end
end

[x,fval,exitflag]=linprog(f,A,B');
% Notese que x es un vector columna

if(exitflag~=1)
    disp('El problema no tiene solucion mediante este algoritmo.')
    xr
    break
end

ro=x(length(x));

if(ro>tol)
```

```

for i=1:length(x)
    if(i~=length(x))
        xr(i)=x(i,:);
    end
end

xr

ron(r)=ro;

taux=fminbnd(gt(xr), 0, 1);

%Tomar en cuenta que 0 y 1 son los extremos del intervalo T

sr=g(xr,taux);

if(sr>=0)
    r=r+1;
    alpha=c*xr';

else

    tr=taux;

    T1=T;
    T1(length(T1)+1,:)=[tr,r];

    %Aquí calcularemos Sr y después T_{r+1}
    n=1;
    while(n<=length(T1))
        if(at(T1(n))*xr'-norm(at(T1(n)))*ron(T1(n,2))>bt(T1(n))
            && ro<=beta*ron(T1(n,2)))

            T1(n,:)=[];
        else
            n=n+1;
        end
    end
end

```

```
T=T1;
clearvars T1;
r=r+1;
end
else
    disp('La solucion aproximada del problema de PLSI es:')
    xr
    break
end

end
```

Apéndice B

Programa en MatLab del método de PCCA

Aquí se presenta el script desarrollado en MatLab del algoritmo del método de PCCA.

```
format long
clear all
c=[1,.5];
T0=[0;1];
AT=[at(T0(1));at(T0(2))];
z0=linprog(zeros(length(AT),1),-AT,-[bt(T0(1));bt(T0(2))])
f=[0;0;-1];

beta=c*z0;
tol=.0000001;

h=.00001;
zeta=1;
gamma=.5;
xi=.5;
omega=.0001;
T=[0,1,0;1,1,0];

r=1;
alpha=-Inf;
```

```
tao=beta-h;
t=0;

gt=@(x)(@(y)x*[1,y]'-tan(y));

t1=T(1):10^(-5):T(2);
disp('Aqui comienza el algoritmo del MPCC Acelerado')

long=1;
while(r<10^3)
    r
    tao
    %Aqui se define A y B
    for i=1:length(T)+1
        if(i==1)
            B(i)=tao;
        else
            B(i)=-bt(T(i-1));
            aux=at(T(i-1));
        end

        for j=1:length(c)+1
            if(i==1)
                if(j==length(c)+1)
                    A(i,j)=norm(c);
                else
                    A(i,j)=c(j);
                end
            else
                if(j==length(c)+1)
                    A(i,j)=norm(aux);
                else
                    A(i,j)=-aux(j);
                end
            end
        end
    end
end
```

```
%Aqui se intenta resolver Qr
[x,fval,exitflag]=linprog(f,A,B');

if(exitflag~=1)
    t=0;
    alpha=tao;
    T(:,3)=0;

    if(beta-alpha<=tol)
        disp('I esta entre alpha y beta')
        alpha

        beta
        break
    else
        tao=(alpha+beta)/2;
    end

else% (exitflag==1)

    %Definir xr y rho
    ro=x(length(x));

    for i=1:length(x)
        if(i~=length(x))
            xr(i)=x(i,:);
        end
    end
    xr

    ron(r)=ro;

    taux=fminbnd(gt(xr), 0, 1);
    %Tomar en cuenta que 0 y 1 son los extremos del intervalo T

    sr=g(xr,taux)

    %Aqui ya se tiene el valor de sr
```

```

if(sr>=0)
    beta=c*xr';
    r=r+1;
    t=0;
    T(:,3)=0;

    if(alpha== -Inf)
        tao=beta-h;
    else
        if(beta-alpha<=tol)
            disp('I esta entre alpha y beta')
            alpha

            beta

            break
        else
            tao=(alpha+beta)/2;
        end
    end
else %(sr<0)

    %Encontrar yr
    k=1;
    while(g(xr,t1(k))>-omega && g(xr,t1(k))>omega*sr)
        yr=t1(k);
        k=k+1;
    end
    %yr encontrado

    T1=T;
    T1(length(T1)+1,:)= [yr,r,0];

    if(t==0)
        if(alpha== -Inf)
            tao=tao-zeta;
        else

```

```

        tao=alpha+xi*(tao-alpha);
    end
    tao1=tao;
    t=1;
    %Paso 3
    n=1;
    while(n<=length(T1))
        if(at(T1(n))*xr'-norm(at(T1(n)))*ron(T1(n,2))>bt(T1(n))
        && ro<=gamma*ron(T1(n,2)) && T1(n,3)~=1)

            T1(n,:)=[];
        else
            n=n+1;
        end

    end

    T=T1;
    clearvars T1;
    r=r+1;
    %termina el Paso 3
else
    if(t==1)
        tao=tao1;
        t=0;
        T=T1;
        T(length(T1),3)=1;
        clearvars T1;
        r=r+1;
    end
end
end

end

end
end

```


Bibliografía

- [1] Betro, Bruno. *An accelerated Central Cutting Plane algorithm for lineal semi-infinite programming*, Mathematical Programing, Ser. A 101: 479-495 (2004).
- [2] E. Gonzalez-Gutierrez, L. Hernandez Rebollar, and Maxim I. Todorov, *Rate of convergence of a class of numerical methods solving linear inequality systems*, Optimization, Vol. 60- 7 pp. 947-957. Septiembre 2011. ISSN: 0233-1934.
- [3] E. Gonzalez-Gutierrez, L. Hernandez Rebollar, and Maxim I. Todorov. *Relaxation Methods for Solving Linear Inequality Systems: Converging Results*, Publicado en lÁnea TOP, DOI 10.1007/s11750-011-0234-4. Octubre 2011.
- [4] Goberna, M. A., Lopez, M. A., *Linear Semi-infinite Optimization*, Edit. John Wiley & Sons, 1998.
- [5] Goberna, M. A., et al, *Optimización lineal. Teoría, métodos y modelos*, Edit. Mc Graw Hill, 2004.
- [6] Rockafellar R. T. *Convex Analysis*, Princenton Landmark in Mathematics. USA, 1970.
- [7] Hu, H., *A one-phase algorithm for semi-infinite linear programming*, Math. Program. 46, 85-103 (1990).
- [8] Hu, H., *Semi-Infinite Programming*. TECHNICAL REPORT SOL 89-2, Stanford University. Marzo 1989.
- [9] Bojanic, R., Devore, R., *On polynomials of best one-sided approximation*, L'Enseignement Mathématique (Ser. 2), 1966.