



UNIVERSIDAD AUTÓNOMA
DEL ESTADO DE MORELOS

CHICAp | *Centro de Investigación en
Ingeniería y Ciencias Aplicadas*

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

CENTRO DE INVESTIGACIÓN EN INGENIERÍA
Y CIENCIAS APLICADAS

METODO SIMPLEX KKT PARA LA SOLUCIÓN DEL ANÁLISIS DE
LA SUJECCIÓN POR MANOS ROBÓTICAS

TESIS PARA OBTENER EL GRADO DE:

DOCTOR EN INGENIERÍA Y CIENCIAS APLICADAS
OPCIÓN TERMINAL TECNOLOGÍA ELÉCTRICA

PRESENTA:

ALEJO MOSSO VAZQUEZ

ASESOR: DR. DAVID JUÁREZ ROMERO

ASESOR: DR. MARCO ANTONIO CRUZ CHÁVEZ

CUERNAVACA MORELOS

JUNIO DE 2012

Agradecimientos

Deseo agradecer profundamente a mis hijos Vicente, Andrei, Jesús y Alejandro, y por supuesto a mi amada esposa por la compañía de todos ellos en el curso de mi vida. Ellos han sido mi inspiración y motivación continuas durante mis estudios.

A mis padres Vicenta Vázquez Hernández, in memoriam, y Victorio Mosso Candia de quienes soy heredero de la cultura del amor por la familia y del progreso por desempeño.

A mis honorables asesores de este trabajo de tesis, Dr. David Juárez Romero y Dr. Marco Antonio Cruz Chávez, por su amplio sentido de solidaridad.

Al Centro de Investigación en Ingeniería y Ciencias Aplicadas (CIICAP) de la Universidad Autónoma del Estado de Morelos que me adoptó como alumno y por su ejemplo de investigación pujante.

Al Consejo Nacional de Ciencia y Tecnología (CONAYT) por el apoyo financiero proveído oportunamente durante mis estudios doctorales.

También deseo agradecer al Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Cuernavaca, en donde recibí el apoyo suficiente en el inicio de mi ruta de la investigación.

A la sociedad humana, cada vez más compleja y atribulada, pero hermosa, por la que vale la pena cualquier esfuerzo.

Resumen

En la modelación de procesos robóticos se necesita ejecutar algoritmos en el mínimo tiempo de manera que se disponga de decisiones oportunas y óptimas para dirigir los movimientos de componentes mecánicos del robot.

El tema central de esta tesis es el diseño de una nueva metodología basada en la identificación de restricciones superfluas para resolver eficientemente un problema de programación lineal que modela la sujeción de un objeto rígido por medio de manos robóticas de cuatro dedos.

En la formulación de un problema de programación lineal para el análisis de la sujeción por manos robóticas aparece una gran cantidad de restricciones no-atadas en la solución óptima. Si algunas de estas restricciones pudieran ser suprimidas, la dimensión del problema sería disminuida y el esfuerzo computacional reducido. Los métodos comunes para la identificación de restricciones no-atadas no son confiables en la práctica debido al cómputo excesivo requerido para su implementación. En esta tesis se desarrolla una nueva metodología para identificar restricciones superfluas, redundantes y no-atadas, y en consecuencia para mejorar la eficiencia del algoritmo Simplex en la solución de un problema de programación lineal para análisis de la sujeción. Se proponen dos métodos complementarios para identificar restricciones superfluas por medio de proyecciones ortogonales sobre el gradiente de la función objetivo, uno de ellos se ejecuta antes de que comience el algoritmo Simplex y el otro se ejecuta mientras el algoritmo evoluciona. El método Simplex que será desarrollado en esta tesis se nombra como *método Simplex KKT* porque preserva el *cono Karush-Kuhn-Tucker* mientras las restricciones superfluas son suprimidas. El *método Simplex KKT* se obtiene por agregación de los procedimientos computacionales requeridos al conocido *método Simplex revisado*. Los resultados experimentales sobre problemas primales de programación lineal para análisis de la sujeción muestran un decrecimiento monótonico de la dimensión del problema, alcanzando una reducción global por arriba del 50%.

R é s u m é

Modeling robotic processes requires the execution of algorithms in the minimum time in order to make timely and optimum decisions to lead the movement of the mechanical components of the robot.

The main subject of this thesis is the design of a new methodology based on the identification of superfluous constraints to efficiently solve a linear programming problem that models the grasp of a rigid object by means of robotic hands of four fingers.

In formulating a linear programming problem for grasping analysis of robotic hands there appears many nonbinding constraints at the optimal solution. If some of these constraints could be suppressed, the dimension of the problem would be decreased and the computational effort reduced. Common methods for identification of nonbinding constraints are not guaranteed in practice because of the excessive computations required to implement them. In this paper a new methodology is introduced to identify superfluous constraints, nonbinding and redundant, and consequently to improve the efficiency of the Simplex algorithm in solving a linear programming problem for grasping analysis. Two complementary methods are proposed to identify superfluous constraints by orthogonal projections onto the gradient of the objective function, one of them performs before the Simplex algorithm starts and the other performs while the algorithm progresses. The Simplex method that will be developed in this paper is named as the *KKT Simplex method* because it preserves the *Karush-Kuhn-Tucker cone* while superfluous constraints are suppressed. The *KKT Simplex method* is obtained by aggregating the required computational procedures into the well-known *revised Simplex method*. Numerical experiments on primal linear programming problems for grasping analysis show a monotonic decreasing of the problem dimension, reaching a global reduction higher than 50%.

Índice

| | |
|---|----|
| Agradecimientos | i |
| Resumen | ii |
| Capítulo 1: Introducción | 1 |
| 1.1 Trabajos previos en la mejora de la eficiencia del método Simplex | 2 |
| 1.1.1 Criterio angular | 2 |
| 1.1.2 Identificación de restricciones redundantes | 2 |
| 1.1.3 Resultados en redundancias de Thompson | 3 |
| 1.1.4 Resultados en redundancias de Telgen | 4 |
| 1.1.5 Trabajos previos en restricciones no atadas | 6 |
| 1.1.6 Contraste de los trabajos previos con la metodología propuesta | 7 |
| 1.2 Formulación del problema de programación lineal | 7 |
| 1.3 Objetivo de la tesis | 8 |
| 1.4 Alcances y limitaciones | 8 |
| 1.5 Organización de la tesis | 9 |
| Capítulo 2: Migración de un punto interior a un vértice frontera | 10 |
| 2.1 Tratamiento de variables libres | 10 |
| 2.1.1 Tratamiento de variables no negativas | 11 |
| 2.1.2 Tratamiento de variables libres | 12 |
| 2.1.3 Procedimiento de migración al vértice frontera | 14 |
| 2.1.4 Observaciones al mecanismo de variables libres | 16 |
| 2.1.5 Observaciones al mecanismo de vértice frontera | 16 |
| 2.2 Ejemplo numérico | 17 |
| 2.2.1 Prueba paso a paso de los mecanismos | 17 |
| 2.2.2 Análisis de resultados | 19 |
| 2.3 Conclusiones del capítulo | 20 |

| | |
|---|-----|
| Capítulo 3: Identificación de restricciones superfluas por criterio angular | 22 |
| 3.1 Definición de clases de restricciones | 23 |
| 3.2 Partición por restricciones superfluas | 26 |
| 3.3 El Cluster como fenómeno de polarización de restricciones | 28 |
| 3.4 Identificación de restricciones superfluas | 32 |
| 3.5 Problema de superfluidad por cono KKT | 43 |
| 3.6 Algoritmo Simplex KKT | 44 |
| 3.7 Conclusiones | 46 |
| Capítulo 4: Algoritmo simplex KKT | 48 |
| 4.1 Problema de identificación de restricciones superfluas remanentes | 50 |
| 4.2 Coordenadas KKT para identificación de restricciones superfluas | 53 |
| 4.3 Mecanismo de identificación de restricciones superfluas por criterio de coordenadas | 69 |
| 4.4 Supresión de restricciones superfluas por criterio de coordenadas | 77 |
| 4.5 Procedimiento de pivoteo y actualización del algoritmo simplex KKT | 80 |
| 4.6 Algoritmo simplex KKT | 81 |
| 4.7 Conclusiones del capítulo | 84 |
| Capítulo 5: Resultados experimentales | 87 |
| 5.1 Modelo de sujeción por manos robóticas | 89 |
| 5.2 Criterios de análisis de la sujeción | 92 |
| 5.3 Análisis de la sujeción por programación lineal | 94 |
| 5.4 Resultados numéricos | 98 |
| 5.5 Conclusiones del capítulo | 105 |
| Capítulo 6 Conclusiones generales y trabajos futuros | 106 |
| 6.1 Conclusiones generales | 106 |
| 6.2 trabajos futuros | 107 |
| Referencias | 109 |

Lista de figuras

- 2.1 Migración desde un punto interior p de un Polígono \mathcal{P} hasta un vértice $x^\#$ en su frontera.
- 2.2 Polígono \mathcal{P} de soluciones factibles
- 3.1 Migración $p \rightarrow x^\# \rightarrow x^*$
- 3.2 Polígono de soluciones factibles caracterizado por (a) ∇f_1 y por (b) ∇f_2
- 3.3 Fenómeno de polarización
- 3.4 Cono de direcciones factibles \mathcal{K}_{FAC}^t , cono de mejora \mathcal{K}_{MEJORA} y cono *KKT* \mathcal{K}^* asociados al vértice x^t .
- 3.5 Medida angular de un vector ∇h_j respecto de ∇f_1
- 3.6 Polarización angular de restricciones
- 3.7 Cono de direcciones factibles \mathcal{K}_{FAC}^t y cono de mejora \mathcal{K}_{MEJORA}
- 3.8 Conos \mathcal{K}_{MEJORA} , \mathcal{K}_{FAC}^* y \mathcal{K}^* , y la condición óptima del vértice x^*
- 3.9 Vecindad $\mathcal{N}_\delta(x^*)$, $\delta > 0 \ni (x - x^*) \in \mathcal{K}_{FAC}^* \forall x \in (\mathcal{P} \cap \mathcal{N}_\delta(x^*))$
- 3.10 Gradiente ∇h_1 excluido del *Cono KKT* \mathcal{K}^*
- 3.11 Medida angular $\text{Cos}(\theta_j)$ de la j -ésima restricción
- 4.1 Migración $p \rightarrow x^\# \rightarrow x^*$
- 4.2 Polígono \mathcal{P} de soluciones factibles en R^2
- 4.3 Especificación de eje ξ de *coordenadas KKT*
- 4.4 Geometría de las coordenadas para $\mathcal{N} = \{10 \ 9 \ 11 \ 8 \ 7 \ 1 \ 2 \ 6\}$
- 4.5 Coordenadas KKT $\xi_j, j \in \mathcal{N}^t$
- 4.6 Gradiente ∇h_1 excluido del *Cono KKT* \mathcal{K}^*
- 4.7 Supresión de coordenadas superfluas
- 4.8 Cambio de signo de la coordenada ξ_4 , de negativa a positiva
- 5.1 The Shadow Dexterous Hand
- 5.2 Esquema simplificado de sujeción
- 5.3 Criterio de rayo emisor
- 5.4 Dualidad entra cascos convexos $H(\mathcal{G})$ y polítopos convexos $\mathcal{P}(\mathcal{G})$
- 5.5 Evolución de la dimensión de cuatro problemas de programación lineal debido a la supresión por criterio angular.
- 5.6 Evolución de la dimensión de cuatro problemas de programación lineal debido a la supresión por criterio de coordenadas.
- 5.7 Evolución de la dimensión de cuatro problemas de programación lineal debido a la acción conjunta del criterio angular y del criterio de coordenadas.
- 5.8 Comparación de los cambios de dimensión por medio de los tres criterios de supresión de restricciones superfluas respecto del algoritmo simplex revisado.
- 5.9 Complejidades normalizadas de los tres criterios de supresión de restricciones superfluas comparada con la del algoritmo simplex revisado.

Lista de tablas

- 2.1 Procedimiento de optimalidad del algoritmo simplex revisado que determina el índice $k_{ENT} = k$ de la variable no básica x_{Nk} no negativa.
- 2.2 Condiciones de incremento de la función objetivo f para determinar el índice k de la variable no básica x_{Nk} .
- 2.3 Mecanismo de variables libres.
- 2.4 Procedimiento de optimalidad que incluye los mecanismos de variables libres y de vértice frontera.
- 3.1 Registro de clusters de restricciones $\mathcal{C}(f_k)$, $k = 1, 2, 3$
- 3.2 Fenómeno de polarización: (a) por fuente eléctrica en un transistor MOSFET y (b) por gradiente ∇f_k y por ángulo del cono KKT $\mathcal{K}^*(f_k)$.
- 3.3 Comparación de la polarización eléctrica y la numérica
- 3.4 Algoritmo simplex KKT
- 4.1 Valores de *Coordenadas KKT* sobre los ejes ξ y ζ
- 4.2 Criterio de coordenadas para identificación de restricciones superfluas
- 4.3 Valores permisibles de ξ cuando $x_{Nk}(\xi)$ crece.
- 4.4 Fórmulas de las coordenadas $\xi_j, j = 1, \dots, |\mathcal{N}^t|$ para el caso $x_{Nk} + \xi$
- 4.5 Valores permisibles de ξ cuando $x_{Nk}(\xi)$ decrece
- 4.6 Fórmulas de las coordenadas $\xi_j, j = 1, \dots, |\mathcal{N}^t|$ para el caso $x_{Nk} - \xi$
- 4.7 Resumen de fórmulas para el cálculo de las *coordenadas KKT*
- 4.8 Procedimiento de factibilidad del algoritmo simplex KKT
- 4.9 Combinaciones de los índices k^+ y k^-
- 4.10 Mecanismo de identificación de restricciones superfluas \mathcal{F}_{restr}^t y de variables y parámetros básicos superfluos \mathcal{F}_{var}^t .
- 4.11 Datos de entrada para el mecanismo de supresión de la tabla 15
- 4.12 Mecanismo de supresión de restricciones superfluas y de variables y parámetros superfluos identificados por criterio de coordenadas
- 4.13 Datos de entrada para el procedimiento de actualización y pivoteo
- 4.14 Procedimiento de pivoteo y actualización del diccionario
- 4.15 Datos de entrada al algoritmo simplex KKT
- 4.16 Algoritmo simplex KKT
- 4.17 Parámetros y variables sujetos a pivoteo y supresión
- 5.1 Algoritmo para análisis de la sujeción por manos robóticas.

Capítulo 1

Introducción

Optimización es un concepto que se refiere a un principio que se encuentra en los fundamentos de análisis de problemas de decisión para la supervivencia y la evolución. La optimización es parte de toda actividad de los sistemas biológicos, desde las formas de vida más simples, como lo es la bacteria *Escherichia Coli* que detecta azúcar y la alcanza dirigiendo el movimiento de sus propulsores, hasta las complejas actividades del homo sapiens. En las últimas décadas se han desarrollado metodologías de optimización que permiten al ser humano solucionar problemas tecnológicos complejos. Muchos problemas ya han sido organizados en clases con sus correspondientes métodos de solución. La optimización trata los aspectos teóricos y computacionales de los programas matemáticos, como la programación lineal, la programación no lineal, la programación convexa, la programación entera, entre otras. En esta tesis se tratarán solamente modelos de programación lineal.

La manipulación por manos robóticas requiere de planteamientos de problemas de optimización. En particular, se ha observado en problemas de sujeción por manos robóticas la necesidad de determinar una decisión de sujeción en tiempos reducidos por razones de operación en tiempo real. Se ha reportado en la literatura que un modelo computacional apropiado para esta aplicación es la programación lineal. Los resultados de solución publicados a la fecha con este método son razonables pero, como se mostrará en este trabajo de tesis podrán ser mejorados substancialmente. Se ha observado también que el modelo de programación lineal para sujeción por manos robóticas exhibe una gran cantidad de restricciones no atadas. Si gran parte de estas restricciones pudieran ser identificadas y suprimidas, se reduciría la dimensión del problema de programación lineal y los tiempos para determinar la decisión de sujeción serían también reducidos. En esta tesis se propone una nueva metodología basada en la identificación de restricciones superfluas, redundantes y no atadas, para mejorar la eficiencia del *método Simplex*. La identificación se logra por medio de variables obtenidas por proyección ortogonal de vectores asociados a las restricciones del modelo de programación lineal.

Este capítulo está organizado como sigue: La sección 1 presenta una revisión de los trabajos previos al presentado en esta tesis acerca de los métodos para mejorar la eficiencia del *método Simplex*. En la sección 2 se formula el problema de programación lineal que se abordará en esta tesis. En la sección 3 se presentan los objetivos y en la 4 los alcances y limitaciones de la tesis. En la sección 5 se expone la organización de esta tesis.

1.1 Trabajos previos en mejora de la eficiencia del método Simplex

Se han realizado grandes esfuerzos para mejorar la eficiencia de solución de un problema de programación lineal después de la aparición del método Simplex de Dantzig [Dantzig 1983, Dantzig et al. 2003]. El algoritmo de punto interior desarrollado por Karmarkar (1984) supera al algoritmo Simplex en problemas grandes de programación lineal. El algoritmo del elipsoide de Khachian [Khachiyan 1979; Bland et al. 1981] fue establecido para resolver problemas de programación lineal en tiempo polinomial. A pesar de estos grandes avances, existen investigaciones en desarrollo para mejorar la eficiencia del *algoritmo Simplex* por medio de un apropiado procesamiento de las restricciones del problema de programación lineal. Por tanto, en esta sección se revisará las investigaciones previas basadas en tres métodos diferentes que mejoran la eficiencia del *algoritmo Simplex*:

- a) Criterio angular
- b) Identificación de restricciones redundantes
- c) Identificación de restricciones no atadas

1.1.1 Criterio angular

La idea principal del criterio angular es procesar con preferencia las restricciones cuyos gradientes son vecinas angulares del gradiente de la función objetivo.

Trigos-Salazar et al. (2002, 2005) introdujo el *Simplex Cosine Method*, el cual alcanza eficientemente la solución óptima en la solución de problemas libres de redundancias. Corley et al. (2005) desarrolló el *Cosine Simplex Algorithm*, el cual tiene un buen desempeño en la solución de problemas de programación lineal con un número de restricciones no mayor a diez. Junior et al. (2005) sugiere como base inicial para arrancar el algoritmo Simplex un vértice que es más cercano al vértice óptimo que la base canónica; Junior reporta una eficiencia del orden del 33% cuando su algoritmo es suministrado por una mejor base. Yeh-Wei-Chang et al. (2009) introdujo el *simple direct cosine Simplex algorithm*, el cual implementa el criterio angular para seleccionar la variable entrante en lugar de la regla tradicional del coeficiente más negativo usada en el *método Simplex*; el algoritmo propuesto reduce el número de iteraciones en la mayoría de los problemas probados.

Como se mostrará en esta tesis, el amplio ángulo del *cono Karush-Kunh-Tucker* es la mayor dificultad enfrentada por los métodos basados en el criterio angular. Esta dificultad es superada en esta tesis por medio de la preservación del *cono Karush-Kunh-Tucker* mientras se suprimen restricciones superfluas.

1.1.2 Identificación de restricciones redundantes

En términos generales, algo es redundante si puede ser omitido *sin* afectar al sistema de referencia. Es decir, la redundancia es una condición que indica exceso de información. Las

restricciones redundantes hacen el problema más grande de lo necesario, por lo que su supresión en el problema mejora la eficiencia del proceso de solución. En términos más precisos, la restricción $\mathbf{g}_k^T \mathbf{x} \leq b_k$ es redundante si y solo si el polítopo \mathcal{P} de soluciones factibles descrito por $\{\mathbf{x} | \mathbf{g}_j^T \mathbf{x} \leq b_j, j = 1, 2, \dots, m\}$ es idéntico al polítopo definido por $\{\mathbf{x} | \mathbf{g}_j^T \mathbf{x} \leq b_j, j = 1, 2, \dots, m, j \neq k\}$.

El problema de redundancia en programación lineal se abordó inicialmente por la identificación de la condición redundante de una restricción a la vez. Se propuso [Boot (1962)] probar la redundancia de una sola restricción verificando la infactibilidad del polítopo \mathcal{P} modificado por el cambio de una desigualdad a una igualdad de la restricción a probar. Posteriormente, se propuso checar la redundancia de una restricción en cada iteración del algoritmo simplex [Thompson (1966), Karwan (1983), Telgen (1986)]. Se presentó un notable trabajo sobre el estado del arte del problema de redundancia y del problema de restricciones no atadas a principios de la década de los 80's [Karwan (1983)]. En esta obra, se analizan la mayoría de los procedimientos conocidos para identificar restricciones redundantes en programación lineal. De entre estos procedimientos, el más general requiere dar solución a un problema de programación lineal formulado para cada restricción del problema original. Sin embargo, el método más eficiente para la identificación de redundancias conocido a la fecha consiste en verificar la condición redundante de varias restricciones mientras se ejecuta el algoritmo simplex. Este método es conocido como *criterio del renglón* [Gal (1983), Gal (1995)], cuya propiedad sobresaliente es su eficiencia computacional porque solo revisa los signos de los renglones de matrices del diccionario en cada iteración del algoritmo simplex. Existen otros métodos que explotan la dualidad entre cascos convexos y polítopos convexos [Dula (1994)]. La idea de estos métodos es identificar los puntos interiores del casco convexo que corresponden a restricciones redundantes en el polítopo convexo dual. La dificultad principal de estos métodos es su complejidad computacional [Fukuda(2004)]. Existen también métodos probabilísticos para la detección de redundancias [Feng (1999)], los cuales son aplicables a problemas de programación lineal y no lineal. Las propiedades principales de estos métodos son su eficiencia computacional pero la falta de completitud en la detección de las redundancias.

1.1.3 Resultados en redundancia de Thompson

Thompson et al (1966) presentó un método para identificar la condición redundante de una restricción en cada iteración simplex. Antes de formular este problema de redundancia consideremos algunas definiciones y terminologías relativas al *cono Karush-Kuhn-Tucker*. Sea la variable de holgura en un vértice \mathbf{x}^t en la iteración t :

$$u_j(\mathbf{x}^t) = b_j - \mathbf{g}_j^T \mathbf{x}^t, j = 1, 2, \dots, m \quad (1)$$

El vértice \mathbf{x}^t tiene asociadas " n " variables de holgura nulas: $u_k(\mathbf{x}^t) = 0$, donde $k \in \mathcal{M} = \{1, 2, \dots, m\}$. Notar que las restricciones con holguras nulas en \mathbf{x}^t forman el conjunto de restricciones atadas: $\mathcal{N}_{ata}^t = \{k | u_k(\mathbf{x}^t) = 0\}$. Para las restricciones: $h_k(\mathbf{x}^t) = \mathbf{g}_k^T \mathbf{x}^t - b_k \leq 0$ correspondientes a las holguras nulas, se tienen " n " gradientes: $\nabla h_k(\mathbf{x}^t) = \mathbf{g}_k$. Estos gradientes definen el *cono Karush-Kuhn-Tucker* en \mathbf{x}^t , el cual se especifica como sigue:

$$\mathcal{K}^t = \{x \in R^n \mid x = \sum_{k \in \mathcal{M}} \lambda_k \nabla h_k(x^t) = E\lambda, \lambda_k \geq 0\} \quad (2)$$

Donde $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]^T$ y E es la matriz de los "n" gradientes linealmente independientes que delimitan el cono \mathcal{K}^t . Sea \tilde{E} la matriz de los gradientes que no delimitan el cono \mathcal{K}^t .

Teorema [Thompson (1966)]: Una restricción $g_i^T x \leq b_i$ donde $g_i \in \tilde{E}$ es redundante si $E^{-1}g_i$ no tiene componentes negativos.

Demostración[Thompson (1966)]: El vector $g_i \in \tilde{E}$ puede ser expresado como combinación lineal de los "n" gradientes linealmente independientes en E que delimitan el cono \mathcal{K}^t :

$$g_i = Ez \quad (3)$$

Donde $z = [z_1, z_2, \dots, z_n]^T$. En general $z_j \in R, j = 1, 2, \dots, n$. Despejando z se tiene:

$$z = E^{-1}g_i \quad (4)$$

Pero si se obtiene $z_j \geq 0, j = 1, 2, \dots, n$, entonces g_i es una combinación lineal positiva de los gradientes en E que delimitan el cono \mathcal{K}^t . Por tanto, $g_i^T x \leq b_i$ es una restricción redundante detectada en el vértice corriente x^t . ■

En la figura 2 se nota que la restricción 11 es redundante porque su hiperplano no delimita el polítopo \mathcal{P} . Se aprecia también un cono KKT \mathcal{K}^t en el vértice x^t delimitado por ∇h_8 y ∇h_9 . Dado que $\nabla h_{11} \in \mathcal{K}^t$, se escribe la siguiente combinación lineal positiva:

$$\nabla h_{11} = \lambda_8 \nabla h_8 + \lambda_9 \nabla h_9, \lambda_8, \lambda_9 \geq 0 \quad (5)$$

El teorema confirma la redundancia de la restricción 11. La restricción $h_{11} \notin \mathcal{N}^t$ pero $\nabla h_{11} \in \mathcal{K}^t$ por lo que $E^{-1}g_i$ no tiene componentes negativos. Por tanto la restricción h_{11} es redundante.

El resultado de Thompson expuesto en el teorema evalúa la condición redundante de una restricción en un vértice corriente x^t . Cada restricción $h_k(x^t) = g_k^T x^t - b_k \leq 0$ cuyo gradiente está en la matriz \tilde{E} , se pone evaluando $E^{-1}g_k, k = 1, 2, \dots, (m - n)$ y $g_k \in \tilde{E}$. Es decir, se tiene resolver el sistema $g_i = Ez$. Además, esta solución debe obtenerse en cada vértice x^t hasta llegar al vértice óptimo x^* . El resultado de Thompson es computacionalmente impráctico, pero es teóricamente atractivo. La dificultad observada es que el sistema $g_i = Ez$ tiene que ser resuelto adicionalmente al diccionario.

1.1.4 Resultados en redundancias de Telgen y Gal

Telgen (1983) presenta los teoremas que dieron origen al método conocido como *criterio del renglón* propuesto por Tomas Gal (1995) para identificar la condición redundante de una restricción en cada iteración simplex. El método es computacionalmente práctico

porque solo revisa los signos de los renglones de matrices del diccionario en cada iteración del algoritmo simplex. El método fue desarrollado originalmente sobre un problema de programación lineal de variables no negativas, y así se expone enseguida.

Consideremos el sistema de restricciones en variables básicas y no básicas:

$$N\mathbf{x}_N + B\mathbf{x}_B = \mathbf{b} \quad (6)$$

Definición [Telgen (1983)]: La restricción $\mathbf{g}_k^T \mathbf{x} \leq b_k$ es redundante respecto del polítopo $\mathcal{P}^+ = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{g}_j^T \mathbf{x} \leq b_j, x_j \geq 0, j = 1, 2, \dots, m\}$ si y solo si

$$\hat{u}_k = \min_x \{u_k(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}^+\} \geq 0 \quad (7)$$

Donde $u_k(\mathbf{x}) = b_k - \mathbf{g}_k^T \mathbf{x}$ es la variable de holgura de la restricción $\mathbf{a}_k^T \mathbf{x} \leq b_k$ especificada en un punto $\mathbf{x} \in \mathcal{P}^+$. Si $\hat{u}_k = 0$, la restricción es llamada débilmente redundante, y si $\hat{u}_k > 0$, la restricción es llamada fuertemente redundante. ■

Multiplicando (6) por B^{-1} se tiene el sistema equivalente

$$\gamma \mathbf{x}_N + I \mathbf{x}_B = \beta \quad (8)$$

Donde

$$\begin{aligned} \gamma &= B^{-1}N = [\gamma_1, \gamma_2, \dots, \gamma_n] \in \mathbb{R}^{m \times n} \\ \beta &= B^{-1}\mathbf{b} = [\beta_1, \beta_2, \dots, \beta_m]^T \in \mathbb{R}^m \\ \mathbf{x}_N &= [x_{N1}, x_{N2}, \dots, x_{Nn}]^T \in \mathbb{R}^n \\ \mathbf{x}_B &= [x_{B1}, x_{B2}, \dots, x_{Bm}]^T \in \mathbb{R}^m \end{aligned} \quad (10)$$

La matriz γ es usualmente referida como el tableau simplex contraído. El teorema que sigue explora la condición de redundancia dada en (8) y los signos de los elementos de los renglones de γ y β .

Teorema [Telgen (1983)]: La restricción $\mathbf{g}_k^T \mathbf{x} \leq b_k$ es redundante respecto del polítopo \mathcal{P}^+ de soluciones factibles si existe una solución básica en la que la variable de holgura $u_k = x_{Br}$ con $\gamma_{rj} \leq 0$ para todo j .

Demostración [Telgen (1983)]: El r -ésimo renglón de (9) se escribe como sigue:

$$x_{Br} = \beta_r - \sum_{j=1}^n \gamma_{rj} x_{Nj} \quad (11)$$

Se supone $\beta_r \geq 0$ y $\gamma_{rj} \leq 0$ para todo j , y que las variables del problema de programación lineal son no negativas. Dado que el valor de x_{Nj} puede solo ser no negativo, se tendrá:

$$\sum_{j=1}^n \gamma_{rj} x_{Nj} \leq 0$$

Finalmente, dado que $\beta_r \geq 0$ se tendrá $u_k = x_{Br} \geq 0$. ■

El r - ésimo renglón en el cual u_k es básica corresponde al criterio del renglón de un problema de programación lineal. El siguiente teorema trata explícitamente el criterio del renglón y provee las condiciones necesarias y suficientes sobre redundancia.

Teorema [Gal en Telgen (1983)]: La restricción $\mathbf{g}_k^T \mathbf{x} \leq b_k$ es redundante respecto del polítopo \mathcal{P}^+ de soluciones factibles si y solo si en alguna solución básica factible se tiene que la variable de holgura $u_k = x_{Br}$ con $\gamma_{rj} \leq 0$ para todo j .

Demostración si [Gal en Telgen (1983)]: Si $\gamma_{rj} \leq 0$ para todo j , entonces:

Se sabe que en toda solución básica se tiene $\beta_r \geq 0$.

Dado que $x_{Nj} \geq 0$ y que $\gamma_{rj} \leq 0$ para todo j , de (19) se sigue $u_k = \beta_r \geq 0$.

Demostración solo si [Gal en Telgen (1983)]: Solo si $u_k \geq 0$, entonces:

Considerando el criterio del renglón que se explica enseguida entonces, si $u_k \geq 0$, en la solución óptima se debe tener $\gamma_{rj} \leq 0$ para todo j y $\beta_r \geq 0$. ■

Criterio del renglón [Gal en Karwan (1983)]: El resultado de la minimización expresada en (16) es un vértice \mathbf{x}^* del polítopo \mathcal{P}^+ de soluciones factibles. La minimización (16) se ejecuta por medio del algoritmo simplex. Luego, si la k - ésima restricción es redundante, la holgura $u_k(\mathbf{x})$ es variable básica en todo vértice de \mathcal{P}^+ . La idea clave es que el r - ésimo renglón de la matriz $\boldsymbol{\gamma}$ y del vector $\boldsymbol{\beta}$ puede ser considerado como el gradiente ∇f de la función objetivo a minimizar por el algoritmo simplex. Por tanto, cuando se alcanza el vértice óptimo \mathbf{x}^* , el r - ésimo renglón contiene solo elementos no negativos [Gal en Karwan (1983)].

1.1.5 Trabajos previos en restricciones no atadas

Los trabajos acerca de la identificación de restricciones no atadas en programación lineal no son abundantes. Las principales referencias son [Thompson (1966), Karwan (1983)], en ellas se exponen las ideas básicas de identificación de restricciones no atadas. Los métodos de identificación propuestos a la fecha no son eficientes en la práctica.

En [Thompson (1966)] se reporta que G. B. Dantzig (1955) sugirió que se podía prever que algunas restricciones podían ser no atadas y que otras podían estar atadas a la solución óptima. Observó que las holguras de las restricciones no atadas podían ser llevadas a la base óptima. Notemos que las holguras de las restricciones atadas al vértice óptimo deben ser variables no básicas. Thompson (1966) explica en un teorema el pasaje de las holguras por la base mediante un concepto de trayectoria convexa. La dificultad en este teorema es que no hay forma de determinar una trayectoria convexa, excepto en problemas en R^2 . El teorema se transcribe enseguida sin demostración.

Teorema [Thompson (1966)]: Mientras se sigue una trayectoria convexa de soluciones durante la optimización de una función objetivo sujeta a restricciones lineales no degeneradas, si una variable inicia en la base, abandona la base, y luego retorna a la base, tal variable estará presente en la base óptima. ■

1.1.6 Contraste de los trabajos previos con la metodología propuesta

La metodología propuesta para la identificación de restricciones superfluas toma en cuenta la información contenida en los gradientes e hiperplanos asociados a las restricciones, mientras los métodos corrientes están basados solo en la información de los gradientes de las restricciones. Para mejorar la eficiencia del *método Simplex*, nuestra metodología usa la identificación y supresión de restricciones superfluas mientras los métodos corrientes usan el criterio angular. Además, para garantizar la confiabilidad de los métodos de identificación, nuestra metodología usa métodos de preservación del cono Karush-Kuhn-Tucker, mientras que los métodos corrientes de identificación no procuran la confiabilidad. Los resultados experimentales en problemas de programación lineal para análisis de la sujeción con 800 restricciones muestran un decrecimiento monótonico de la dimensión del problema alcanzado una reducción global por arriba del 50%.

1.2 Formulación del problema de programación lineal

El problema de programación lineal que será abordado en esta tesis se formula como sigue:

$$\begin{aligned} & \underset{x}{\text{maximizar}} f(x) = \mathbf{c}^T \mathbf{x} \\ & \text{subjecta a: } \mathbf{a}_j^T \mathbf{x} + s_j = b_j, j = 1, 2, \dots, m \\ & L_{Ni} \leq x_i \leq U_{Ni}, i = 1, \dots, n, L_{Bj} \leq s_j \leq U_{Bj}, j = 1, 2, \dots, m \end{aligned} \quad (12)$$

Aquí los elementos del vector $\mathbf{x} = [x_1, \dots, x_n]^T$ son las variables de decisión del problema, $L_{Ni} = -\infty, U_{Ni} = \infty, s_j = (b_j - \mathbf{a}_j^T \mathbf{x}) \geq 0, j = 1, 2, \dots, m$ son las variables de holgura, $L_{Bj} = 0, U_{Bj} = \infty, f: R^n \rightarrow R$ es la función objetivo, $h_j: R^n \rightarrow R$ donde $h_j(\mathbf{x}) = \mathbf{a}_j^T \mathbf{x} - b_j \leq 0, j = 1, 2, \dots, m$ son funciones de restricciones lineales, $\|\nabla h_j\| = \|\mathbf{a}_j\| = 1$ por simplicidad, y las constantes b_1, \dots, b_m son los límites de las restricciones. Las restricciones lineales en (12) determinan el polítopo de soluciones factibles:

$$\mathcal{P} = \{\mathbf{x} \in R^n \mid \mathbf{a}_j^T \mathbf{x} \leq b_j, j = 1, 2, \dots, m\} \quad (13)$$

Un vector \mathbf{x}^* es llamado óptimo del problema (12) si produce el mayor valor de la función objetivo de entre el resto que satisfacen las restricciones. Supondremos que \mathcal{P} es no vacío y acotado, el problema de programación lineal es no degenerado, el número m de restricciones es mayor que el número n de variables de decisión, $b_j > 0$, y el origen de R^n es un punto interior de \mathcal{P} . Cuando el espacio vectorial está limitado a R^6 , el problema de programación lineal (12) modela el análisis de la sujeción por manos robóticas.

El *método Simplex* procesa el problema de programación lineal (2) escrito en la siguiente forma canónica:

$$\begin{aligned} & \text{maximize } f = \mathbf{c}_N^T \mathbf{x}_N + \mathbf{c}_B^T \mathbf{x}_B \\ & \text{subjected to: } N\mathbf{x}_N + B\mathbf{x}_B = \mathbf{b} \end{aligned} \quad (14)$$

Donde \mathbf{x}_N es un vector de variables no básicas $x_{Ni}, i = 1, \dots, n$, \mathbf{x}_B es un vector de variables básicas $x_{Bj}, j = 1, \dots, m$, $N = [N_1, \dots, N_n]$ es la matriz no básica, $B = [B_1, \dots, B_m]$ es la matriz básica. Comparando el problema (14) con el (12) se tiene: $\mathbf{x}_N = [x_1, \dots, x_n]^T$, $\mathbf{x}_B = [s_1, \dots, s_m]^T$, $\mathbf{c}_N = \mathbf{c}$, $\mathbf{c}_B = \mathbf{0}$, $N = [\mathbf{a}_1, \dots, \mathbf{a}_m]^T$, y $B = I$. Cuando la matriz cuadrada B es no singular, f y \mathbf{x}_B puede ser registrada por lo que se conoce como diccionario:

$$\begin{aligned} & \mathbf{x}_B = B^{-1}\mathbf{b} - B^{-1}N\mathbf{x}_N \\ & f = \mathbf{c}_B^T B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1}N)\mathbf{x}_N \end{aligned} \quad (15)$$

En cada iteración, el *algoritmo Simplex* procesa el diccionario (15) por actualización de una solución básica factible $\{\mathbf{x}_N, \mathbf{x}_B\}$.

1.3 Objetivo de la tesis

El objetivo de este trabajo de tesis es desarrollar una metodología basada en la identificación de restricciones superfluas para mejorar la eficiencia del *algoritmo Simplex* en la solución de modelos de programación lineal para análisis de la sujeción por manos robóticas.

Se desarrollarán los métodos de identificación de restricciones superfluas por medio de coordenadas determinadas de la información contenida en las restricciones del problema de programación lineal que se desea resolver. Dada la estructura del problema, se desarrollarán los mecanismos computacionales para tratar las variables libres.

1.4 Alcances y limitaciones

Los problemas de programación lineal que serán tratados en este trabajo de tesis tienen las características siguientes:

- (i) El problema de programación lineal abordado en esta tesis está definido en el espacio R^n , por lo que el problema para análisis de la sujeción es un caso particular por estar definido en R^6 .
- (ii) El punto de inicio del *algoritmo Simplex* es un punto interior del polítopo de soluciones factibles. Si el punto de inicio es un vértice en la frontera del polítopo, la solución por medio del algoritmo Simplex se simplifica.
- (iii) El número de restricciones que pueden definirse en un problema de sujeción por cuatro dedos robóticos pueden ir desde 12 hasta más de 1000.

- (iv) El polítopo de soluciones factibles del problema de programación lineal es no vacío y acotado.
- (v) El problema de programación lineal es no degenerado. Esta limitación se establece por facilidad de análisis, sin embargo se puede relajar por medio de la supresión de restricciones que causan la degeneración.

El método de solución del problema de programación planteado es el *método Simplex revisado* presentado por Chvatal (1983), el cual será modificado por agregación de los procedimientos computacionales de identificación de restricciones superfluas.

1.5 Organización de la tesis

La metodología desarrollada en esta tesis tiene el propósito de hacer eficiente el *método Simplex* por medio de la identificación y supresión de restricciones superfluas. Los métodos de identificación y el método Simplex resultante serán presentados en esta tesis como sigue.

En el capítulo 2 se desarrolla un método de identificación de restricciones superfluas, que estará basado en un *método de coordenadas gradiente*. En el capítulo 3 se desarrolla un método alternativo de identificación de restricciones superfluas basado en un *método de coordenadas Simplex*. En este mismo capítulo se presenta el algoritmo Simplex producto de la metodología propuesta. En el Capítulo 5 se presentan resultados experimentales que evalúan la eficiencia y robustez del método desarrollado. En el capítulo 6 se presentan las conclusiones generales de esta tesis.

Capítulo 2

Migración de un Punto Interior a un Vértice Frontera

En este capítulo se desarrolla un procedimiento de migración desde un punto interior p de un polígono convexo \mathcal{P} hasta un vértice $x^\#$ en su frontera como se muestra en la figura 1. Esta migración debe ser resuelta en problemas de programación lineal que modelan el análisis de la sujeción por manos robóticas [Dan 2001]. Porque la sujeción es un problema práctico, la migración se debe resolver en un mínimo número de pasos procesando al mismo tiempo variables libres.

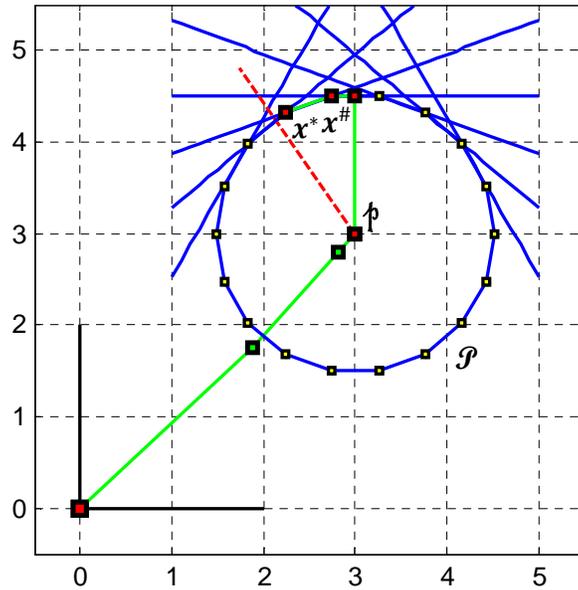


Fig. 1: Migración desde un punto interior p de un Polígono \mathcal{P} hasta un vértice $x^\#$ en su frontera

El *algoritmo Simplex revisado* [Chvatal 1983] resuelve problemas de programación lineal con variables de decisión no negativas y con punto de inicio un vértice en la frontera del polígono de soluciones factibles. En este capítulo se adapta el *algoritmo Simplex revisado* para que resuelva problemas de programación lineal de variables libres con punto de inicio en el interior del polígono, como el mostrado en la figura 1.

La sección 1 presenta una revisión de las formas de tratar las variables libres y del cálculo de un vértice en la frontera del polígono de soluciones factibles. En la sección 2 se presenta un ejemplo en R^3 para evaluar paso a paso el cálculo del vértice frontera. La sección 3 presenta las conclusiones del capítulo.

2.1 Tratamiento de variables libres

El algoritmo simplex revisado requiere que las variables libres sean transformadas en variables no negativas. Hay dos formas usuales para realizar ésta transformación. La primera forma sustituye una variable libre por dos variables no negativas, mientras que la segunda forma expresa una variable libre como combinación lineal de las variables no negativas, la cual se cancela por sustitución en las ecuaciones donde aparezca [Lumberger, 2008]. La segunda forma no aplica porque todas sus variables de decisión son libres. La primera forma puede aplicar con la desventaja de duplicar el número de variables de decisión.

2.1.1 Tratamiento de variables no negativas

Consideremos el procedimiento de optimalidad del algoritmo simplex revisado que procesa variables no negativas. El procedimiento selecciona de entre los elementos del vector $\mathbf{x}_N = [x_{N1}, x_{N2}, \dots, x_{Nn}]^T$ una variable x_{Nk} que al ser modificada incrementa la función objetivo:

$$f(\mathbf{x}_N) = \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{x}_N \quad (1)$$

El vector $\mathbf{x}_N = [x_{N1}, x_{N2}, \dots, x_{Nn}]^T$ es nulo por definición. Sus componentes pueden ser variables de decisión o variables de holgura en virtud del procedimiento de pivoteo. El algoritmo simplex revisado requiere que todas las variables de decisión sean no negativas y, dado que las de holgura son también no negativas por definición, la forma de incrementar la función objetivo es por incremento de $x_{Nk} \in \mathbf{x}_N$ por un escalar $t > 0$. Es decir, $x_{Nk} \leftarrow (x_{Nk} + t)$ siempre que su correspondiente coeficiente satisfaga la siguiente condición [Chvatal, 1983]:

$$(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) > 0 \quad (2)$$

La tabla 1 muestra el procedimiento del algoritmo simplex revisado que determina el índice $k_{ENT} = k$ de la variable no básica x_{Nk} que incrementará el valor de la función objetivo.

| |
|--|
| <p>Ejecutar para $k = 1, 2 \dots n$ lo que sigue:</p> <ul style="list-style-type: none"> Si $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) > 0$ <ul style="list-style-type: none"> ○ $k_{ENT} = k$ ○ Salir del lazo $k = 1, 2 \dots n$ |
|--|

Tabla 1: Procedimiento de optimalidad del algoritmo simplex revisado que determina el índice $k_{ENT} = k$ de la variable no básica x_{Nk} no negativa.

Si se insiste en usar el algoritmo simplex revisado en problemas de programación lineal de variables libres, se pueden usar métodos para convertir las variables libres en variables no negativas [Lumberger, 2008]. De otra forma, habrá que modificar el procedimiento de optimalidad para que se acepten variables libres.

2.1.2 Tratamiento de variables libres

Se propone un método que trate directamente las variables libres por medio de una modificación al procedimiento de optimalidad del algoritmo simplex revisado.

Consideremos los siguientes hechos:

- Las variables de decisión son libres $-\infty < x_i < \infty, i = 1, 2, \dots, n$, mientras que las variables de holgura son no negativas por definición $0 \leq u_j < \infty, j = 1, 2, \dots, m$.
- Un componente x_{Nk} del vector de variables no básicas \mathbf{x}_N , puede ser variable de decisión $x_i, i \in \{1, 2, \dots, n\}$ o variable de holgura $u_j, j \in \{1, 2, \dots, m\}$. Por tanto, la variable no básica x_{Nk} puede adoptar valores positivos o negativos. Es decir, el valor nulo de x_{Nk} puede emigrar hacia un valor positivo o negativo. Notar que el índice $k \in \{1, 2, \dots, n\}$.

Por consiguiente, la función objetivo (8) puede ser incrementada por modificación de una variable no básica x_{Nk} :

$$x_{Nk}(t) = x_{Nk} \pm \xi \quad (3)$$

donde $\xi > 0$ y $x_{Nk} = 0$ del lado derecho para todo k . Se puede demostrar que este cambio modifica al vector de variables no básicas en la forma siguiente:

$$\mathbf{x}_N(t) = \mathbf{x}_N \pm \mathbf{e}_k \xi \quad (4)$$

donde \mathbf{e}_k es el vector canónico con la unidad en su k -ésima coordenada y $\mathbf{x}_N = 0$ del lado derecho. Sustituyendo (4) en las expresiones de f y de \mathbf{x}_B del diccionario se tiene:

$$\begin{aligned} f &= \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{x}_N \\ f(t) &= \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) (\mathbf{x}_N \pm \mathbf{e}_k \xi) \\ &= \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{x}_N \pm (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{e}_k \xi \\ &= f \pm (\mathbf{c}_N^T \mathbf{e}_k - \mathbf{c}_B^T B^{-1} N \mathbf{e}_k) \xi \\ &= f \pm (c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) \xi \end{aligned} \quad (5)$$

$$\begin{aligned} \mathbf{x}_B &= B^{-1} \mathbf{b} - B^{-1} N \mathbf{x}_N \\ \mathbf{x}_B(t) &= B^{-1} \mathbf{b} - B^{-1} N (\mathbf{x}_N \pm \mathbf{e}_k \xi) \\ &= B^{-1} \mathbf{b} - B^{-1} N \mathbf{x}_N \mp B^{-1} N \mathbf{e}_k \xi \\ &= \mathbf{x}_B \mp B^{-1} N_k \xi \end{aligned} \quad (6)$$

El simple cambio $x_{Nk} = x_{Nk} \pm \xi$ puede hacer que la función objetivo sea incrementada como se indica en (5) bajo ciertas condiciones del coeficiente $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k)$, y bajo las siguientes condiciones de factibilidad de $x_{Nk}(\xi)$ y $\mathbf{x}_B(\xi)$:

$$L_{Nk} < (x_{Nk} \pm \xi) < U_{Nk}$$

$$\mathbf{L}_B \leq (\mathbf{x}_B \mp B^{-1}\mathbf{N}_k\xi) < \mathbf{U}_B \quad (7)$$

La cantidad " $\xi > 0$ " se calcula tal que las condiciones (7) se satisfagan, y se calcularán en el procedimiento de factibilidad del algoritmo simplex más adelante. En (7) se necesita el índice k para precisar las cantidades x_{Nk} , L_{Nk} , U_{Nk} y la k -ésima columna \mathbf{N}_k de la matriz no básica. Por otro lado, el índice k se determina por evaluación del signo del coeficiente $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k)$ que hará que la función objetivo $f(t)$ sea incrementada, como se muestra enseguida para el caso de variables libres que se está tratando.

El impacto del incremento $(x_{Nk} \pm \xi)$ sobre la función objetivo $f(t)$ se muestra en la tabla 2 con el propósito de determinar el índice k de la variable no básica x_{Nk} , y de deducir por tanto el mecanismo de variables libres. La columna 1 muestra los posibles valores de la variable entrante x_{Nk} , mientras que la 6 muestra sus correspondientes incrementos. La columna 2 muestra los límites de x_{Nk} , la 3 muestra solo sus límites inferiores. La columna 4 muestra la condición positiva o negativa del coeficiente $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k)$ dependiendo del valor de la variable entrante x_{Nk} .

| x_{Nk} | $[L_{Nk}, U_{Nk}]$ | L_{Nk} | Coeficiente | Indicador de signo | $x_{Nk} \pm \xi$ |
|----------|----------------------|-----------|--|--------------------|------------------|
| s_j | $[0, +\infty]$ | 0 | $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k) > 0$ | $k^+ = k$ | $x_{Nk} + \xi$ |
| x_i | $[-\infty, +\infty]$ | $-\infty$ | $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k) > 0$ | $k^+ = k$ | $x_{Nk} + \xi$ |
| x_i | $[-\infty, +\infty]$ | $-\infty$ | $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k) < 0$ | $k^- = k$ | $x_{Nk} - \xi$ |

Tabla 2: Condiciones de incremento de la función objetivo f para determinar el índice k de la variable no básica x_{Nk} .

El impacto del incremento $(x_{Nk} \pm \xi)$ se observa en los renglones de la tabla 2 como sigue:

- El renglón 2 se lee como sigue: Si $x_{Nk} = s_j$, lo que se detecta por $L_{Nk} = 0$, se elige $(x_{Nk} + \xi)$ siempre que el k -ésimo coeficiente $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k) > 0$.
- El renglón 3 se lee como sigue: Si $x_{Nk} = x_i$, lo que se detecta por $L_{Nk} = -\infty$, se elige $(x_{Nk} + t)$ siempre que el k -ésimo coeficiente $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k) > 0$.
- El renglón 4 se lee como sigue: Si $x_{Nk} = x_i$, lo que se detecta por $L_{Nk} = -\infty$, se elige $(x_{Nk} - \xi)$ siempre que el k -ésimo coeficiente $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k) < 0$.

De los renglones 2 y 3 se deduce que se tiene que elegir $(x_{Nk} + \xi)$ cuando el coeficiente $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k) > 0$, ya sea que $L_{Nk} = 0$ ó que $L_{Nk} = -\infty$. Del renglón 4 se deduce que se tiene que elegir $(x_{Nk} - \xi)$ cuando $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k) < 0$ y $L_{Nk} = -\infty$. Dado que $x_{Nk} = 0$ por definición, la condición $L_{Nk} = -\infty$ se puede escribir como: $x_{Nk} > L_{Nk}$.

Por consiguiente, la función objetivo $f(t)$ dada en (11) es incrementada por identificación del signo del k -ésimo coeficiente $(c_{Nk} - \mathbf{c}_B^T B^{-1}\mathbf{N}_k)$ y por el límite inferior de la variable no básica x_{Nk} mientras el índice k se varía $k = 1, 2, \dots, n$:

$$\begin{aligned} & (c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) > 0 \text{ para el caso } x_{Nk} + \xi \\ & x_{Nk} > L_{Nk} \text{ y } (c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) < 0 \text{ para el caso } x_{Nk} - \xi \end{aligned} \quad (8)$$

Las condiciones (8) para identificar el k – ésimo coeficiente de la variable no básica x_{Nk} permiten tratar las variables libres directamente incluyéndolas en el procedimiento de optimalidad del algoritmo simplex revisado.

La tabla 3 muestra el mecanismo de variables libres que determina el índice k de la variable no básica x_{Nk} . Comparando el procedimiento de la tabla 1 con el de la tabla 3 notamos que el algoritmo simplex revisado y adaptado, acepta las variables de decisión como variables libres por la condición adicional: $(x_{Nk} > L_{Nk})$ y $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) < 0$.

| |
|--|
| $k^+ = 0, k^- = 0$ Ejecutar para $k = 1, 2 \dots n$ lo que sigue: Si $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) > 0$ ○ $k^+ = k$ ○ Salir del lazo $k = 1, 2 \dots n$ Si $(x_{Nk} > L_{Nk})$ y $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) < 0$ ○ $k^- = k$ ○ Salir del lazo $k = 1, 2 \dots n$ Terminar si $k^+ = 0$ y $k^- = 0$, y declarar la solución corriente como solución óptima. |
|--|

Tabla 3: Mecanismo de variables libres.

2.1.3 Procedimiento de migración al vértice frontera

Como se hizo notar anteriormente, todas las variables de decisión están contenidas en \mathbf{x}_N al inicio del algoritmo simplex. Por tanto, el mecanismo propuesto de vértice frontera consiste de las dos acciones siguientes:

- Sacar del vector $\mathbf{x}_N = [x_1, x_2, \dots, x_n]^T$ cada una de sus variables (una a la vez) durante las primeras " n " iteraciones del simplex. Durante estas iteraciones, las dos condiciones de variables libres dadas en (8) podrán satisfacerse.
- Asegurar que ninguna variable de decisión regrese al vector \mathbf{x}_N después haber salido de él. Por tanto, a partir de la iteración $(n + 1)$, el vector \mathbf{x}_N deberá estar constituido solo por variables de holgura.

Que cada variable de decisión $x_i, i = 1, \dots, n$ salga del vector $\mathbf{x}_N = [x_1, x_2, \dots, x_n]^T$ en las primeras " n " iteraciones del simplex es relativamente simple porque en general la elección de la variable saliente no es única. Sin embargo, la variable que regresa al vector \mathbf{x}_N es única, por lo que es más difícil asegurar que ninguna variable de decisión regrese a \mathbf{x}_N . Como se verá más adelante en el procedimiento de factibilidad del algoritmo simplex revisado, es posible hacer que la variable que regresa al vector \mathbf{x}_N sea solamente variable básica haciendo uso de la hipótesis de no degeneración del problema de programación

lineal. La implementación del mecanismo de vértice frontera se presenta como parte del procedimiento de optimalidad de la tabla 4.

Los mecanismos de variables libres y de vértice frontera modificarán el procedimiento de optimalidad del algoritmo simplex revisado. El procedimiento modificado se denota como *procedimiento de optimalidad de vértice frontera*, el cual se presenta en la tabla 4.

| |
|--|
| <p>Datos de entrada: $\mathbf{c}_N, \mathbf{c}_B, N, B^{-1}, \mathbf{x}_N, \mathbf{L}_N$ $Canon = [1, 2, \dots, n]$ $t^\# = n$ $k^+ = 0, k^- = 0$ Si $t < t^\#$ Ejecutar para $k = 1, 2 \dots n$ lo que sigue: Si $k \in Canon$ <u>VERTICE FRONTERA:</u> Si $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) > 0$ <u>VARIABLES LIBRES:</u> <ul style="list-style-type: none"> ○ $k^+ = k$ ○ Suprimir k del conjunto <i>Canonicas</i> ○ Salir del lazo $k = 1, 2 \dots n$ Si $(x_{Nk} > L_{Nk})$ y $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) < 0$ <u>VARIABLES LIBRES:</u> <ul style="list-style-type: none"> ○ $k^- = k$ ○ Eliminar k del conjunto <i>Canonicas</i> ○ Salir del lazo $k = 1, 2 \dots n$ En otro caso ($t \geq t^\#$) <u>DESPUÉS DE VERTICE FRONTERA:</u> Ejecutar para $k = 1, 2 \dots n$ lo que sigue: Si $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) > 0$ <ul style="list-style-type: none"> ○ $k^+ = k$ ○ Salir del lazo $k = 1, 2 \dots n$ Si $(x_{Nk} > L_{Nk})$ y $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) < 0$ <ul style="list-style-type: none"> ○ $k^- = k$ ○ Salir del lazo $k = 1, 2 \dots n$ Terminar si $k^+ = 0$ y $k^- = 0$, y declarar la solución corriente como solución óptima.</p> |
|--|

Tabla 4: Procedimiento de optimalidad que incluye los mecanismos de variables libres y de vértice frontera.

El procedimiento de optimalidad de vértice frontera de la tabla 4 puede ser expresado computacionalmente por medio de la siguiente función:

$$[k^+, k^-] = \text{Optimalidad}(\mathbf{c}_N, \mathbf{c}_B, N, B^{-1}, \mathbf{x}_N, \mathbf{L}_N, Canon, t^\#, k^+, k^-) \quad (16)$$

Los parámetros de salida k^+, k^- se usarán en los procedimientos de factibilidad y pivoteo del *algoritmo Simplex*.

2.1.4 Observaciones al mecanismo de variables libres

- El mecanismo de variables libres determina el índice $k_{ENT} = k$ de la variable no básica x_{Nk} que hará crecer la función objetivo. El mecanismo se codifica en la tabla 4 mediante las dos instrucciones: “Si $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) > 0$ ” y “Si $(x_{Nk} > L_{Nk})$ y $(c_{Nk} - \mathbf{c}_B^T B^{-1} N_k) < 0$ ”, las cuales permiten valores negativos para la variable no básica entrante, es decir: $x_{Nk} \leftarrow (x_{Nk} \pm \xi)$ con $\xi > 0$.

2.1.5 Observaciones al mecanismo de vértice frontera

En el procedimiento de optimalidad de la tabla 4 se ha definido el conjunto "*Canonicas*" cuyos elementos son los índices de las variables de decisión asociadas a los ejes canónicos del espacio vectorial R^n . Es decir, los índices de los elementos del vector de variables básicas $\mathbf{x}_N = [x_1, x_2, \dots, x_n]^T$ al inicio del algoritmo simplex son los siguientes:

$$Canonicas = \{1, 2, \dots, n\}$$

- En una iteración del algoritmo simplex revisado, el mecanismo de vértice frontera elige mediante la instrucción: " $k \in Canonicas$ " el índice de una variable canónica que será variable entrante. Que esta variable entrante se elija en las primeras " n " iteraciones del algoritmo se codifica con la instrucción: "Ejecutar para $k = 1, 2 \dots n$ lo que sigue".
- La instrucción: "si $k \in Canonicas$ " asegura que se elija una variable de decisión, mientras que la instrucción: "eliminar k del conjunto *Canonicas*", asegura que $Canonicas \rightarrow \emptyset$. Por tanto, la estrategia de sacar cada variable de decisión del vector $\mathbf{x}_N = [x_1, x_2, \dots, x_n]^T$ hasta agotarlas permite emigrar desde el origen de R^n en el interior del polítopo \mathcal{P} hasta un vértice $\mathbf{x}^\#$ en su frontera en solo " n " iteraciones del algoritmo simplex.
- Al finalizar la iteración " n ", el vector \mathbf{x}_N estará constituido por solo variables de holgura teniéndose: $\mathbf{x}_N = [s_{j_1}, s_{j_2}, \dots, s_{j_n}]^T$ con $s_{j_k} \in \{s_1, s_2, \dots, s_m\}$ y $s_{j_k} = 0, k = 1, 2, \dots, n$. Que estas variables sean nulas significa que habiendo comenzado en el origen de R^n , el algoritmo simplex ha alcanzado un vértice $\mathbf{x}^\#$ sobre la frontera del polítopo \mathcal{P} de soluciones factibles en " n " pasos. El vértice $\mathbf{x}^\#$ puede ser considerado como el vértice de inicio de la fase 2 del método simplex revisado. Por tanto, el mecanismo que lleva desde el origen de R^n hasta $\mathbf{x}^\#$ es equivalente a la fase 1 del método simplex.
- El algoritmo simplex propuesto comienza en el origen de R^n en el interior del polítopo \mathcal{P} , llega a un vértice $\mathbf{x}^\#$ en la frontera de \mathcal{P} y continúa por la frontera hasta llegar al vértice óptimo \mathbf{x}^* . Es decir, el algoritmo propuesto incluye las dos fases del método simplex sin distinguirlas porque el mecanismo de vértice frontera es parte del procedimiento de optimalidad del algoritmo simplex.
- Una propiedad notable del mecanismo de vértice frontera propuesto es que usa la función objetivo del problema original de programación, lo que hace que el vértice frontera $\mathbf{x}^\#$ esté angularmente cercano del vértice óptimo \mathbf{x}^* .

2.2 Ejemplo numérico

Se presenta un ejemplo numérico de un problema de programación lineal de variables libres con el fin de evaluar los mecanismos de variables libres y de vértice frontera de la tabla 4. El problema es de seis restricciones y está definido en R^3 por facilidad de análisis:

$$\begin{aligned} \underset{x}{\text{maximizar}} \quad & f(x) = \mathbf{c}_N^T \mathbf{x}_N + \mathbf{c}_B^T \mathbf{x}_B \\ \text{subjeto a:} \quad & N\mathbf{x}_N + B\mathbf{x}_B = \mathbf{b} \end{aligned}$$

El algoritmo se inicia con los parámetros y variables siguientes:

$$\begin{aligned} \mathbf{c}_N &= [c_{N1}, c_{N2}, c_{N3}]^T = [-0.5, \quad -1.5, \quad 2]^T \\ \mathbf{c}_B &= [c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5}, c_{B6}]^T \\ &= [0, 0, 0, 0, 0, 0]^T \\ N &= [N_1, \quad N_2, \quad N_3] \\ &= [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6]^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 \end{bmatrix}^T \end{aligned} \quad (16)$$

$$B = I, 6 \times 6$$

$$\mathbf{b} = [1.5, 1.1492, 1.9254, 1.9254, 2.8508, 1.5]^T$$

$$\mathbf{x}_N = \mathbf{x} = [x_1, x_2, x_3]^T = [0, 0, 0]^T, \quad \mathbf{L}_N < \mathbf{x} < \mathbf{U}_N$$

$$\mathbf{x}_B = \mathbf{s} = [s_1, s_2, s_3, s_4, s_5, s_6]^T, \quad \mathbf{L}_B \leq \mathbf{s} < \mathbf{U}_B$$

$$\mathbf{L}_N = [LN_1, LN_2, LN_3] = [-\infty, -\infty, -\infty]$$

$$\mathbf{U}_N = [UN_1, UN_2, UN_3] = [+\infty, +\infty, +\infty]$$

$$\mathbf{L}_B = [LB_1, LB_2, \dots, LB_6] = [0, 0, 0, 0, 0, 0]$$

$$\mathbf{U}_B = [UB_1, UB_2, \dots, UB_6] = [+\infty, +\infty, +\infty, +\infty, +\infty, +\infty]$$

2.2.1 Prueba paso a paso de los mecanismos

Datos de entrada para Iteración 1:

$$\mathbf{c}_N = [c_{N1}, c_{N2}, c_{N3}]^T = [2, -0.5, -2]^T$$

$$\mathbf{c}_B = [c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5}, c_{B6}]^T = [0, 0, 0, 0, 0, 0]^T$$

$$\text{inv}B = I, 6 \times 6$$

$$N = [N_1, N_2, N_3] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 \end{bmatrix}^T = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6]^T$$

$$\mathbf{L}_N = [LN_1, LN_2, LN_3] = [-\infty, -\infty, -\infty]$$

$$\mathbf{x}_N = [x_{N1}, x_{N2}, x_{N3}]^T = [0, 0, 0]^T$$

Ejecutar para $k = 1, 2, 3$ lo que sigue:

$$\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N = [2, -0.5, -2]^T - [0, 0, 0, 0, 0, 0]^T B^{-1} N = [2, -0.5, -2]^T$$

$$\Rightarrow c_{N1} - \mathbf{c}_B^T B^{-1} N_1 = 2 > 0$$

- ⇒
- $kENT = 1$
 - $Canonicas = \{2, 3\}$
 - $Canon_cont = 1$
 - $Salir\ del\ lazo\ k = 1, 2, 3$
- $$\mathbf{x}^1 = [x_1, x_2, x_3]^T = [1.5, 0, 0]^T$$
- $$\mathbf{s}(\mathbf{x}^1) = [0 \ 1.149 \ 1.925 \ 1.925 \ 2.8508 \ 3]^T$$

Datos de entrada para Iteración 2:

$$\mathbf{c}_N = [c_{N1}, c_{N2}, c_{N3}]^T = [0, -0.5, -2]^T$$

$$\mathbf{c}_B = [c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5}, c_{B5}]^T = [2, 0, 0, 0, 0, 0]^T$$

$$invB = I, 6x6 \text{ con la primer columna} = [1, 0, 0, 0, 0, 1]^T$$

$$N = [N_1, N_2, N_3] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 \end{bmatrix}^T = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6]^T$$

$$\mathbf{L}_N = [LN_1, LN_2, LN_3] = [0, -\infty, -\infty]$$

$$\mathbf{x}_N = [x_{N1}, x_{N2}, x_{N3}]^T = [0, 0, 0]^T$$

Ejecutar para $k = 1, 2, 3$ lo que sigue:

$$\begin{aligned} \mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N &= [0, -0.5, -2] - [2, 0, 0, 0, 0, 0]^T B^{-1} N \\ &= [0, -0.5, -2] - [2, 0, 0] \\ &= [-2, -0.5, -2] \end{aligned}$$

$$\Rightarrow (x_{N2} > L_{N2}) \text{ y } (c_{N2} - \mathbf{c}_B^T B^{-1} N_2) < 0$$

- ⇒
- $kENT = 2$
 - $Canonicas = \{3\}$
 - $Canon_cont = 2$
 - $Salir\ del\ lazo\ k = 1, 2, 3$
- $$\mathbf{x}^2 = [x_1, x_2, x_3]^T = [1.5, -1.9254, 0]^T$$
- $$\mathbf{s}(\mathbf{x}^2) = [0 \ 1.1492 \ 0 \ 3.8508 \ 2.8508 \ 3]^T$$

Datos de entrada para Iteración 3:

$$\mathbf{c}_N = [c_{N1}, c_{N2}, c_{N3}]^T = [0, 0, -2]^T$$

$$\mathbf{c}_B = [c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5}, c_{B5}]^T = [2, 0, -0.5, 0, 0, 0]^T$$

$$invB = I, 6x6 \text{ con la columna 1} = [1, 0, 0, 0, 0, 1]^T \text{ y la 3: } -e_3 + e_4$$

$$N = [N_1, N_2, N_3] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 \end{bmatrix}^T = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6]^T$$

$$\mathbf{L}_N = [LN_1, LN_2, LN_3] = [0, 0, -\infty]$$

$$\mathbf{x}_N = [x_{N1}, x_{N2}, x_{N3}]^T = [0, 0, 0]^T$$

Ejecutar para $k = 1, 2, 3$ lo que sigue:

$$\begin{aligned} \mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N &= [0, 0, -2] - [2, 0, -0.5, 0, 0, 0]^T B^{-1} N \\ &= [0, -0.5, -2] - [2, 0.5, 0] \\ &= [-2, 0, -2] \end{aligned}$$

$$\Rightarrow (x_{N3} > L_{N3}) \text{ y } (c_{N3} - \mathbf{c}_B^T B^{-1} N_3) < 0$$

- ⇒
- $kENT = 3$
 - $Canonicas = \phi$
 - $Canon_cont = 3$
 - $Salir\ del\ lazo\ k = 1, 2, 3$
- $$\mathbf{x}^3 = [x_1, x_2, x_3]^T = [1.5, -1.9254, -1.1492]^T$$
- $$\mathbf{s}(\mathbf{x}^3) = [0 \ 0 \ 0 \ 5 \ 4 \ 3]^T$$

El polígono de soluciones factibles $\mathcal{P} = \{x \in \mathbb{R}^n | a_1^T x \leq b_1\} \cap \dots \cap \{x \in \mathbb{R}^n | a_6^T x \leq b_6\}$ y los tres puntos calculados x^1, x^2, x^3 se muestran en la figura 1.

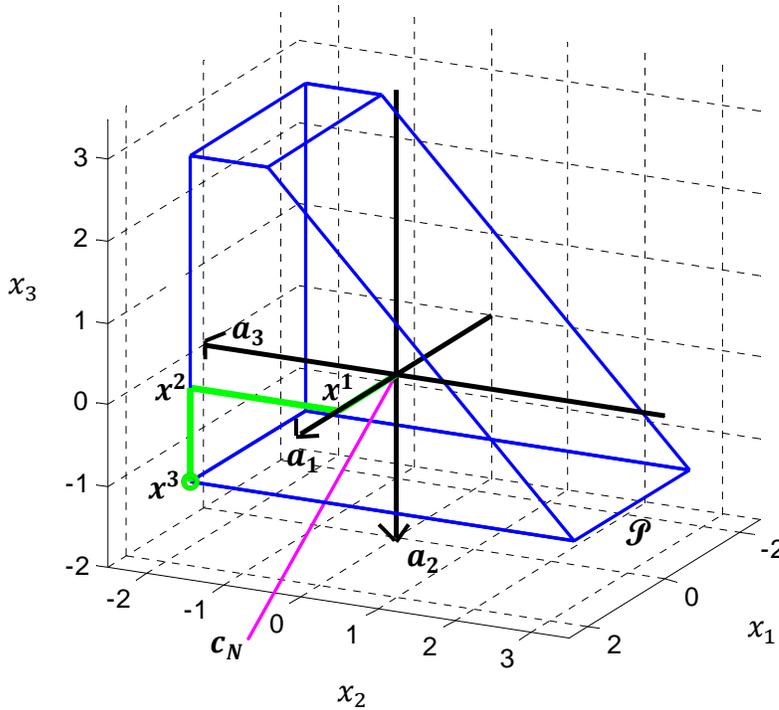


Fig. 2: Polígono \mathcal{P} de soluciones factibles

2.2.2 Análisis de resultados:

1. Como se aprecia en la figura 1, el vértice $x^\#$ en la frontera del polígono \mathcal{P} se alcanza en solo tres pasos desde el origen de \mathbb{R}^3 , es decir $x^\# = x^3$. En este ejemplo, la solución óptima x^* coincide con $x^\# = [1.5, -1.9254, -1.1492]^T$, con lo que se muestra que $x^\#$ es angularmente cercano a x^* .
2. Como se aprecia en la ejecución paso a paso, la naturaleza libre de la variable no básica x_{Nk} se detecta en cada iteración. En la iteración 1 sucedió el incremento: $x_{N1} \leftarrow (x_{N1} + t)$ el cual se detectó por la condición:

$$c_{N1} - c_B^T B^{-1} N_1 = 2 > 0.$$

Mientras que en las iteraciones 2 y 3 sucedieron los decrementos: $x_{N2} \leftarrow (x_{N2} - t)$ y $x_{N3} \leftarrow (x_{N3} - t)$, los cuales fueron detectados por las condiciones:

$$(x_{N2} > L_{N2}) \text{ y } (c_{N2} - c_B^T B^{-1} N_2) < 0$$

$$(x_{N3} > L_{N3}) \text{ y } (c_{N3} - c_B^T B^{-1} N_3) < 0$$

respectivamente.

3. Existe consistencia entre las posiciones de los tres vértices calculados x^1, x^2, x^3 y sus correspondientes variables de holgura:

$$s(x^1) = [0 \quad 1.149 \quad 1.925 \quad 1.925 \quad 2.8508 \quad 3]^T$$

$$s(x^2) = [0 \quad 1.1492 \quad 0 \quad 3.8508 \quad 2.8508 \quad 3]^T$$

$$\mathbf{s}(\mathbf{x}^3) = [0 \quad 0 \quad 0 \quad 5 \quad 4 \quad 3]^T$$

Como se aprecia en la figura 1, dado que el punto \mathbf{x}^1 se localiza sobre una cara del polígono, el vector de variables de holgura $\mathbf{s}(\mathbf{x}^1)$ tiene solo una variable nula. Porque el punto \mathbf{x}^2 se localiza sobre en la intersección de dos caras del polígono, el vector de variables de holgura $\mathbf{s}(\mathbf{x}^2)$ tiene dos variables nulas. El punto \mathbf{x}^3 se localiza en la intersección de tres caras del polígono, por lo que el vector de variables de holgura $\mathbf{u}(\mathbf{x}^3)$ tiene tres variables nulas.

4. Los ceros del origen $\mathbf{x} = [x_1, x_2, x_3]^T = [0, 0, 0]^T$ de R^3 se propagan hacia cada solución $[(\mathbf{x}^j)^T, (\mathbf{s}(\mathbf{x}^j))^T], j = 1, 2, 3$, como se observa en la tabla 5:

| t | $[(\mathbf{x}^t)^T, (\mathbf{s}(\mathbf{x}^t))^T] = [x_1^t, x_2^t, x_3^t \mid s_1^t, s_2^t, s_3^t, s_4^t]$ |
|-----|---|
| 0 | $[0, 0, 0 \mid 1.5, 1.1492, 1.9254, 1.9254, 2.8508, 1.5], \mathbf{x}_N = [x_1^0, x_2^0, x_3^0]^T$ |
| 1 | $[1.5, 0, 0 \mid 0 \quad 1.149 \quad 1.925 \quad 1.925 \quad 2.8508 \quad 3], \mathbf{x}_N = [u_1^1, x_2^1, x_3^1]^T$ |
| 2 | $[1.5, -1.9254, 0 \mid 0 \quad 1.1492 \quad 0 \quad 3.8508 \quad 2.8508 \quad 3], \mathbf{x}_N = [u_1^2, u_3^2, x_3^2]^T$ |
| 3 | $[1.5, -1.9254, -1.1492 \mid 0 \quad 0 \quad 0 \quad 5 \quad 4 \quad 3], \mathbf{x}_N = [u_1^3, u_3^3, u_2^3]^T$ |

Tabla 5: Propagación de los ceros del origen de R^3

5. El vector de variables no básicas \mathbf{x}_N recibe una variable de holgura s_j^t en cada iteración t . Dado que el ejemplo está definido en R^3 , en la iteración $t = 3$ \mathbf{x}_N tiene sus tres componentes con variables de holgura como se nota en la tabla 5.
6. Se ha comprobado que el código de la tabla 4 implementa los mecanismos de variables libres y de vértice frontera para el ejemplo dado. El vértice frontera $\mathbf{x}^\#$ es angularmente cercano al óptimo \mathbf{x}^* y las tres variables libres $-\infty < x_i < \infty, i = 1, 2, 3$ se manipularon directamente en el procedimiento de optimalidad del algoritmo simplex.

2.3 Conclusiones del capítulo

Como se observa de la tabla 5, no es una sorpresa que los "n" ceros del origen $\mathbf{x} = [x_1, x_2, x_3]^T = [0, 0, 0]^T$ de R^n se reacomoden en cada solución: $[(\mathbf{x}^j)^T, (\mathbf{s}(\mathbf{x}^j))^T] = [x_1^j, x_2^j, x_3^j \mid s_1^j, s_2^j, s_3^j, s_4^j], j = 1, 2, \dots$ durante las primeras "n" iteraciones simplex. Solo hay "n" ceros en esta solución cuando se tiene un polígono de soluciones factibles no degenerado. El procedimiento de optimalidad selecciona una variable entrante $x_{Nk} = 0$ de entre los componentes del vector $\mathbf{x}_N = [x_{N1}, x_{N2}, \dots, x_{Nn}]^T$ como candidata para que sea movida de su valor nulo mediante el cambio $x_{Nk} \leftarrow (x_{Nk} \pm \xi)$ con $\xi > 0$. Inmediatamente después, el procedimiento de factibilidad ejecuta este cambio mediante el cálculo de $\xi > 0$ de tal forma que una sola variable básica x_{Br} se haga nula. De esta forma, la variable no básica x_{Nk} se hace distinta de cero al mismo tiempo que una variable básica x_{Br} se hace cero, por lo que el número de ceros en la solución $[x_1^j, x_2^j, x_3^j \mid s_1^j, s_2^j, s_3^j, s_4^j], j = 1, 2, \dots$ se mantiene en "n" siempre que el polígono de soluciones factibles sea no degenerado.

El procedimiento presentado incluye las dos fases del método simplex usual. La fase 1 está codificada implícitamente en el procedimiento de optimalidad del algoritmo simplex frontera como se muestra en la tabla 4. Además de que el vértice frontera $x^\#$ es calculado en " n " pasos, este vértice se encuentra angularmente cercano al vértice óptimo x^* . Este hecho se explica porque el mecanismo de vértice frontera usa la misma función objetivo del problema de programación que se está resolviendo.

El procedimiento de la tabla 4 tiene las dos propiedades siguientes: (i) la aceptación en forma directa de las variables libres sin usar métodos de conversión a variables no negativas y (ii) el cálculo en " n " iteraciones simplex de un vértice en la frontera del polígono de soluciones factibles a partir del origen del espacio R^n en el interior del polígono.

Capítulo 3

Identificación de Restricciones Superfluas por Criterio angular

En la formulación de problemas de programación lineal para análisis de la sujeción se ha observado la existencia de una gran cantidad de restricciones superfluas en la solución óptima, las cuales causan tiempos prolongados de cálculo. Las restricciones superfluas encontradas en estos problemas son eminentemente no atadas y rara vez redundantes.

La pregunta importante que puede ser planteada ahora es que si existe alguna forma de forzar a que las restricciones se manifiesten como restricciones superfluas.

En este capítulo se probará la hipótesis de que es posible someter el conjunto de restricciones de un problema de programación lineal a cierta clase de polarización que las fuerce a agruparse en dos clases: el conjunto de restricciones candidatas a delimitar el *cono Karush-Kuhn-Tucker (cono KKT)* y un conjunto de restricciones que con seguridad son superfluas.

El fenómeno de polarización de restricciones usado aquí es como el de la polarización de cargas eléctricas sujetas a un campo de fuerza. En un problema de programación lineal existen entidades polarizadoras que funcionan como generadoras de un campo de fuerza sobre las restricciones. Las entidades de polarización son el gradiente de la función objetivo ∇f y el *cono KKT*. Las entidades sometidas al fenómeno de polarización son fundamentalmente las restricciones, pero no directamente, sino a través de cantidades derivadas de ellas que son sensibles a la polarización. Las dos cantidades sensibles que serán tratadas en esta tesis son: (i) la *coordenada gradiente* γ_j obtenida por proyección ortogonal del gradiente ∇h_j de la j -ésima restricción sobre ∇f y (ii) la *coordenada Simplex* ξ_j obtenida de la proyección ortogonal de un vector asociado al hiperplano de la j -ésima restricción. En este capítulo se diseñará un método basado en las *coordenadas gradiente* para identificar restricciones superfluas. En el capítulo 5 será desarrollado otro método para identificar restricciones superfluas, pero basado las *coordenadas Simplex*.

La metodología propuesta en el presente capítulo consiste en insertar en el *método Simplex revisado* [Chvatal (1983)] el procedimiento de optimalidad desarrollado en el capítulo 3 en el que se calcula el vértice frontera $\mathbf{x}^\#$, y el procedimiento de identificación de restricciones superfluas basado en las *coordenadas gradiente*. La inclusión de estos procedimientos al *método Simplex revisado* producirá un algoritmo mejorado en eficiencia computacional, el cual se nombrará como *método Simplex KKT*. El algoritmo inicia en un punto interior \mathbf{p} del polítopo \mathcal{P} de soluciones factibles, migra hacia un vértice $\mathbf{x}^\#$ en su frontera y continúa hasta llegar al vértice óptimo \mathbf{x}^* , como se ilustra en la figura 1. El problema de programación lineal sujeto de estudio está definido en el espacio R^n cuyo origen puede ser un punto exterior de \mathcal{P} , como se muestra en la misma figura 1. Si este es el caso, el punto interior \mathbf{p} de \mathcal{P} puede ser calculado por medio del algoritmo de Newton. En el presente capítulo se

supone que el punto interior p es dado, como sucede en el análisis de la sujeción por manos robóticas. Por tanto, el algoritmo generará la siguiente secuencia de vértices: $p, \dots, x^\#, \dots, x^*$.

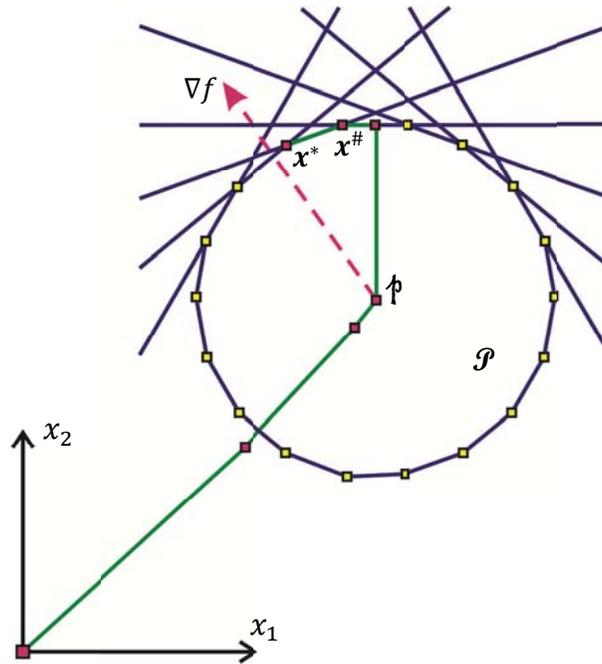


Fig. 1: Migración $p \rightarrow x^\# \rightarrow x^*$

En la sección 1 se definen las clases de restricciones que componen un problema de programación lineal. En la sección 2 se plantea el problema de partición del conjunto de restricciones. En la sección 3 se plantea el fenómeno de polarización de restricciones. En la sección 4 se diseña un mecanismo de identificación de restricciones superfluas por medio del criterio angular. En la sección 5 se analiza el efecto de la supresión de restricciones atadas al óptimo sobre la condición de superfluidad. En la sección 6 se presenta el *algoritmo simplex KKT*. En la sección 7 se presentan las conclusiones del capítulo.

3.1 Definiciones de clases de restricciones

Consideremos el siguiente problema de programación lineal en R^n con las suposiciones y definiciones del capítulo 1:

$$\begin{aligned} & \underset{x}{\text{maximizar}} \quad f(x) = c^T x \\ & \text{subjeto a:} \quad a_j^T x \leq b_j, j = 1, \dots, m \end{aligned} \tag{1}$$

Se enfatiza que el problema de programación lineal (1) es no degenerado por ser una condición necesaria para la identificación de restricciones superfluas y que el polítopo de soluciones factibles es no vacío. Para precisar el concepto de restricciones superfluas se definirán ciertas clases de restricciones que componen el problema, las cuales están referidas al polítopo de soluciones factibles. El polítopo de referencia es el de la figura 2

definido en R^2 por razones de ilustración gráfica de los conceptos asociados, y corresponde al siguiente ejemplo numérico:

$$\begin{aligned} \underset{x}{\text{maximizar}} \quad & f(x) = [-1 \quad -2]^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \text{subjecta a:} \quad & \end{aligned}$$

| <i>Restricciones</i> | <i>Indices</i> |
|--|----------------|
| $0.9359x_1 - 0.3523x_2 + s_1 = 0.7707$ | 1 |
| $0.8869x_1 - 0.4619x_2 + s_2 = 1.3858$ | 2 |
| $0.9998x_1 + 0.0202x_2 + s_3 = 1.5148$ | 3 |
| $0.8601x_1 + 0.5101x_2 + s_4 = 2.7004$ | 4 |
| $0.1709x_1 + 0.9853x_2 + s_5 = 3.4183$ | 5 |
| $-0.6816x_1 + 0.7317x_2 + s_6 = 2.5059$ | 6 |
| $-0.9404x_1 + 0.3401x_2 + s_7 = 1.7007$ | 7 |
| $-0.9999x_1 + 0.0101x_2 + s_8 = 1.2121$ | 8 |
| $-0.9573x_1 - 0.2892x_2 + s_9 = 1.3960$ | 9 |
| $-0.6816x_1 - 0.7317x_2 + s_{10} = 1.9045$ | 10 |
| $-0.9939x_1 - 0.1104x_2 + s_{11} = 2.1082$ | 11 |

$-\infty \leq x_1, x_2 \leq +\infty, s_j \geq 0, j = 1, 2, \dots, 11$

Los gradientes ∇h_j de las restricciones $j = 1, 2, 4, 5, 8, 9, 10, 11$, un vértice inicial \mathbf{x}^0 y dos vértices óptimos \mathbf{x}^* se muestran en la figura 1. Dos problemas de programación lineal con el mismo polígono \mathcal{P} pero con diferentes funciones objetivo se muestran también para estudiar las restricciones respecto de los conjuntos \mathcal{K}_1^* and \mathcal{K}_2^* mostrados también aquí.

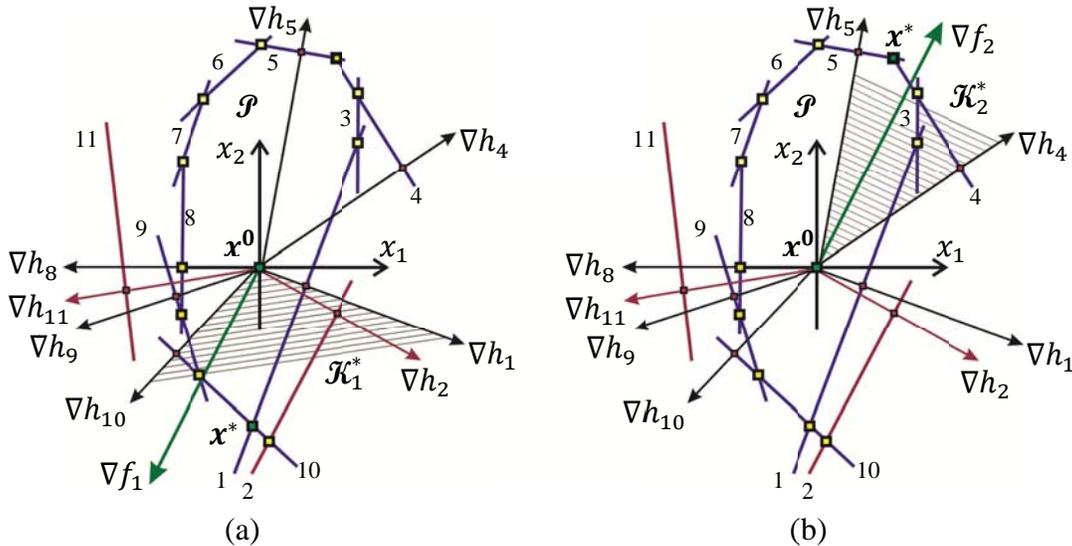


Fig. 2: Polígono de soluciones factibles caracterizado por (a) ∇f_1 y por (b) ∇f_2

El vector gradiente ∇f_1 en la figura 2(a) coincide con la de (2), es decir, $\nabla f_1 = \nabla f = [-1, -2]^T$. En la figura 2 (b) otro vector gradiente $\nabla f_2 = [1, 2]^T$ caracteriza el otro problema.

En las definiciones que siguen se dan diferentes conjuntos de índices que serán usados en esta tesis, los cuales representan conjuntos de restricciones del problema de prueba (2) y que son mostrados en la figura 2.

Definición 1: Sea \mathcal{M} el conjunto de restricciones en un problema de programación lineal dado. Para el problema de prueba se tiene: $\mathcal{M}=\{1,2,3,4,5,6,7,8,9,10,11\}$.

Definición 2: Sea el *cluster de restricciones*, denotado como \mathcal{C} , el conjunto de restricciones en \mathcal{M} ordenadas de acuerdo al ángulo entre el gradiente de cada restricción y el gradiente de la función objetivo. En la figura 2, $\mathcal{C}(\nabla f_1)=\{10, 9, 11, 8, 7, 2, 1, 6, 3, 4, 5\}$ es el cluster referido a ∇f_1 , y $\mathcal{C}(\nabla f_2)=\{5, 4, 3, 6, 1, 2, 7, 8, 11, 9, 10\}$ es el cluster referido a ∇f_2 .

Definición 3: El *conjunto de restricciones redundantes* con respect al polítopo \mathcal{P} de soluciones factibles, denotado por \mathcal{R} , es el conjunto de restricciones que satisfacen la siguiente condición: $\mathbf{g}_k^T \mathbf{x} \leq b_k$ es una restricción redundante si y solo si el polítopo \mathcal{P} es idéntico al conjunto: $\{\mathbf{x} | \mathbf{g}_j^T \mathbf{x} \leq b_j, j = 1, 2, \dots, m, j \neq k\}$. Para el problema de prueba se tiene $\mathcal{R}=\{2,11\}$. Las restricciones redundantes no delimitan \mathcal{P} como se ve en la figura 2.

Definición 4: Sea el *conjunto de restricciones atadas*, denotado por \mathcal{N}_{bind}^* , el conjunto de restricciones cuyas variables de holgura son nulas en el vértice óptimo \mathbf{x}^* : $\mathcal{N}_{bind}^* = \{k | s_k(\mathbf{x}^*) = 0\}$. $\mathcal{N}_{bind}^* = \{1,10\}$ está asociado a la función objetivo f_1 y $\mathcal{N}_{bind}^* = \{4,5\}$ a la f_2 .

Notamos que los gradientes $\nabla h_k(\mathbf{x}^*)$ de las restricciones atadas delimitan el *cono KKT*. En problemas de programación lineal no degenerados las holguras nulas definen restricciones atadas a cualquier vértice \mathbf{x}^t del polítopo. En este caso podremos definir el conjunto de restricciones atadas a un vértice \mathbf{x}^t en la forma siguiente:

$$\mathcal{N}_{ata}^t = \{k | u_k(\mathbf{x}^t) = 0\} \quad (3)$$

Una restricción atada a un vértice corriente \mathbf{x}^t es una restricción cuyo gradiente delimita un cono KKT en \mathbf{x}^t , el cual se denotará como \mathcal{K}^t . Notar que existe un cono \mathcal{K}^t para cada vértice corriente \mathbf{x}^t en el sentido de que \mathbf{x}^t puede ser considerado como vértice óptimo para alguna función objetivo. En la figura 2 las restricciones $\mathbf{g}_8^T \mathbf{x} \leq b_8$ y $\mathbf{g}_9^T \mathbf{x} \leq b_9$ son restricciones atadas al vértice frontera $\mathbf{x}^\#$, teniéndose por tanto:

$$\mathcal{N}_{ata}^\# = \{k | u_k(\mathbf{x}^\#) = 0\} = \{8, 9\}$$

Definición 5: Sea el *Cono Karush-Kuhn-Tucker* (cono KKT), denotado por \mathcal{K}^* , el conjunto de puntos determinados por la combinación lineal positiva de los gradientes $\nabla h_i(\mathbf{x}^*)$ de las restricciones i en \mathcal{N}_{bind}^* :

$$\mathcal{K}^* = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} = \sum_{i \in \mathcal{N}_{bind}^*} \lambda_i \nabla h_i(\mathbf{x}^*), \lambda_i \geq 0\} \quad (4)$$

En la figura 2(a): $\mathcal{K}_1^* = \{\mathbf{x} \in R^2 \mid \mathbf{x} = \lambda_1 \nabla h_1(\mathbf{x}^*) + \lambda_{10} \nabla h_{10}(\mathbf{x}^*), \lambda_1, \lambda_{10} \geq 0\}$ está caracterizado por las restricciones en $\mathcal{N}_{bind}^* = \{1,10\}$ mientras que en la figura 2(b): $\mathcal{K}_2^* = \{\mathbf{x} \in R^2 \mid \mathbf{x} = \lambda_4 \nabla h_4(\mathbf{x}^*) + \lambda_5 \nabla h_5(\mathbf{x}^*), \lambda_4, \lambda_5 \geq 0\}$ está caracterizado por las restricciones en $\mathcal{N}_{bind}^* = \{4,5\}$.

La expresión dada en (3) nos permite explicar la relación que existe entre el cono KKT \mathcal{K}^* y el gradiente de la función objetivo ∇f en el vértice óptimo \mathbf{x}^* por medio de lo que se conoce como las condiciones de optimalidad de Karush-Kuhn-Tucker [Bazaraa 2006]: Si \mathbf{x}^* es una solución óptima para el problema de programación lineal que se está resolviendo, el gradiente de la función objetivo ∇f en \mathbf{x}^* puede ser representado como un punto interior del cono KKT \mathcal{K}^* siempre que los vectores $\nabla h_i(\mathbf{x}^*), i = 1, \dots, n$ sean linealmente independientes:

$$\nabla f = \sum_{i=1}^n \lambda_i \nabla h_i(\mathbf{x}^*), \lambda_i \geq 0 \quad (5)$$

La expresión (4) agrupa las n restricciones que definen la solución óptima \mathbf{x}^* . Para la figura 2 se tiene: $\nabla f = \lambda_1 \nabla h_1(\mathbf{x}^*) + \lambda_{10} \nabla h_{10}(\mathbf{x}^*), \lambda_1, \lambda_{10} \geq 0$.

Definición 6: Sea el *ángulo del cono Karush-Kuhn-Tucker (ángulo del cono KKT)*, denotado por $ang(\mathcal{K}^*)$, la cardinalidad del mínimo conjunto de restricciones en el cluster \mathcal{C} que contiene \mathcal{N}_{bind}^* normalizado por su cardinalidad $|\mathcal{C}|$. En la figura 2(a) $ang(\mathcal{K}_1^*) = 7/11$ se obtiene de $\mathcal{C}(\nabla f_1)$ y de $\mathcal{N}_{bind}^* = \{10, 1\}$, mientras que en la figura 2(b) $ang(\mathcal{K}_2^*) = 2/11$ se obtiene de $\mathcal{C}(\nabla f_2)$ y de $\mathcal{N}_{bind}^* = \{5, 4\}$.

Definición 7: Sea el *conjunto de restricciones superfluas*, denotado por \mathcal{F} , el conjunto de restricciones no atadas que, aunque delimitan el polítopo de soluciones factibles, sobre definen la solución óptima, y el conjunto de restricciones redundantes que no delimitan el polítopo. Las restricciones superfluas pueden ser calculadas en el vértice óptimo \mathbf{x}^* : $\mathcal{F} = \mathcal{M} - \mathcal{N}_{bind}^*$. $\mathcal{F} = \{2, 3, 4, 5, 6, 7, 8, 9, 11\}$ para la función objetivo f_1 , y $\mathcal{F} = \{1, 2, 3, 6, 7, 8, 9, 10, 11\}$ para f_2 .

3.2 Partición por restricciones superfluas

Se prevé que el conjunto de todas las restricciones \mathcal{M} puede ser particionado en el conjunto de las restricciones superfluas y el conjunto de las restricciones atadas al óptimo:

$$\begin{aligned} \mathcal{M} &= \mathcal{F} \cup \mathcal{N}_{ata}^* & (6) \\ &= \{9, 11, 8, 7, 2, 6, 3, 4, 5\} \cup \{1, 10\} \\ &= \{10, 9, 11, 8, 7, 1, 2, 6, 3, 4, 5\} \end{aligned}$$

El conjunto de las atadas \mathcal{N}_{ata}^* es el de las restricciones deseables, mientras que el conjunto de las superfluas \mathcal{F} es el de las restricciones indeseables. En un problema de programación lineal definido en R^n , la solución óptima está determinada solamente por las n restricciones

de \mathcal{N}_{ata}^* . El resto son las restricciones superfluas de \mathcal{F} . Si se conociera a priori el conjunto \mathcal{N}_{ata}^* , se determinaría inmediatamente la solución óptima. El algoritmo simplex puede ser considerado como un procedimiento de simplificación de \mathcal{M} a \mathcal{N}_{ata}^* , con la particularidad de que \mathcal{N}_{ata}^* se determina hasta que el vértice óptimo \mathbf{x}^* es alcanzado. En este capítulo se pretende reducir la cardinalidad del conjunto \mathcal{M} antes de obtener a \mathcal{N}_{ata}^* .

Aunque se conociera a priori un conjunto contenedor del conjunto de restricciones atadas, se insiste en agregar restricciones al sistema de desigualdades para que formen parte del conjunto de restricciones necesarias al problema de programación lineal. Estas restricciones son necesarias en el diseño de estrategias o en toma de decisiones. Otra causa de que las restricciones no atadas y las redundantes son incluidas en la formulación del problema es porque no se sabe a priori cuales son redundantes o no atadas. Para propósitos de optimización, algunas restricciones no atadas pueden ser suprimidas del problema y la información perdida por su supresión puede ser recuperada una vez que la solución óptima haya sido calculada.

La inclusión indiscriminada de restricciones a un sistema de restricciones puede causar tres clases de problemas:

- (i) **Problema de infactibilidad:** consiste en que el polítopo dado de soluciones factibles puede ser vacío.
- (ii) **Problema de redundancia:** consiste en que el polítopo dado de soluciones factibles está contaminado con restricciones redundantes que nunca determinan la solución óptima.
- (iii) **Problema de restricciones no atadas:** consiste en que el polítopo dado de soluciones factibles tiene exceso de información, lo cual produce restricciones no atadas que incrementan la carga computacional.

Los problemas de identificación de restricciones redundantes y no atadas se revisarán en la siguiente sección. El problema de infactibilidad se evitará por medio de la siguiente hipótesis que ya fue impuesta sobre el polítopo de soluciones factibles:

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{g}_1^T \mathbf{x} \leq b_1\} \cap \dots \cap \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{g}_m^T \mathbf{x} \leq b_m\} \neq \phi \quad (7)$$

3.3 El cluster como fenómeno de polarización de restricciones

En esta sección se propone explorar el fenómeno de polarización inducido por el gradiente ∇f de la función objetivo de un problema de programación lineal y la forma de forzar a que las restricciones del problema se manifiesten como superfluas bajo su influencia. Para tal propósito se desplegarán y correlacionarán los datos asociados a clusters de restricciones. Se probará con tres problemas de programación lineal en R^2 definidos con un mismo polítopo \mathcal{P} , el del ejemplo (2), y las siguientes funciones objetivo:

$$\begin{aligned} f_1(\mathbf{x}) &= [-1, -2]^T \mathbf{x} \\ f_2(\mathbf{x}) &= [1, 2]^T \mathbf{x} \\ f_3(\mathbf{x}) &= [1, 0.8]^T \mathbf{x} \end{aligned} \quad (8)$$

La figura 2 muestra el polítopo convexo \mathcal{P} y los gradientes de las funciones objetivo $f_1(\mathbf{x})$ y $f_2(\mathbf{x})$. El polítopo está delimitado por los hiperplanos de las restricciones de la 1 a la 10. Se han propuesto las restricciones 2 y 11 como redundantes para que forme parte de las restricciones superfluas que serán identificadas. La solución óptima es un vértice \mathbf{x}^* en cada problema. El cono KKT $\mathcal{K}^*(f_1)$ de la figura 2(a) está delimitado por los gradientes ∇h_{10} y ∇h_1 , mientras que el cono KKT $\mathcal{K}^*(f_2)$ de la figura 3(b) está delimitado por los gradientes ∇h_4 y ∇h_5 . Los clusters de restricciones son $\mathcal{C}(f_1) = \{10, 9, 11, 8, 7, 1, 2, 6, 3, 4, 5\}$ para el problema con función objetivo f_1 y $\mathcal{C}(f_2) = \{5, 4, 3, 6, 2, 1, 7, 8, 11, 9, 10\}$ para el problema con función objetivo f_2 .

En la tabla 1 se comparan los clusters de restricciones $\mathcal{C}(f_1)$, $\mathcal{C}(f_2)$ y $\mathcal{C}(f_3)$ para determinar si la posición de ∇f_k respecto del polítopo \mathcal{P} afecta la concentración de las restricciones atadas $\mathcal{N}_{ata}^*(f_k)$ dispuestas en $\mathcal{C}(f_k)$ para $k = 1, 2, 3$.

| $f_k(\mathbf{x})$ | $\mathcal{C}(f_k)$ | $\mathcal{N}_{ata}^*(f_k)$ | $\mathcal{N}_{min}(f_k)$ | $ang(\mathcal{K}^*(f_k))$ |
|-------------------|-------------------------------------|----------------------------|--------------------------|----------------------------------|
| f_1 | {10, 9, 11, 8, 7, 1, 2, 6, 3, 4, 5} | {10, 1} | {10, 9, 11, 8, 7, 1} | $ang(\mathcal{K}^*(f_1)) = 6/11$ |
| f_2 | {5, 4, 3, 6, 2, 1, 7, 8, 11, 9, 10} | {5, 4} | {5, 4} | $ang(\mathcal{K}^*(f_2)) = 2/11$ |
| f_3 | {4, 3, 5, 2, 1, 6, 7, 8, 11, 9, 10} | {4, 5} | {4, 3, 5} | $ang(\mathcal{K}^*(f_3)) = 3/11$ |

Tabla 1: Registro de clusters de restricciones $\mathcal{C}(f_k)$, $k = 1, 2, 3$

En la tabla, $\mathcal{N}_{min}(f_k)$ es el conjunto de restricciones de cardinalidad mínima que satisface: $\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}_{min}(f_k) \subseteq \mathcal{C}(f_k)$. La cardinalidad del conjunto $\mathcal{N}_{min}(f_k)$ puede usarse como medida angular del cono KKT: $ang(\mathcal{K}^*(f_k)) = |\mathcal{N}_{min}(f_k)|/|\mathcal{C}(f_k)|$ como se muestra en la columna 5 de la tabla 1. A partir de los datos de la tabla 1 y con los conos $\mathcal{K}^*(f_1)$ y $\mathcal{K}^*(f_2)$ de la figura 2, se deducen enseguida ciertas relaciones entre las cantidades ∇f_k , $\mathcal{C}(f_k)$, $\mathcal{N}_{ata}^*(f_k)$ y $ang(\mathcal{K}^*(f_k))$.

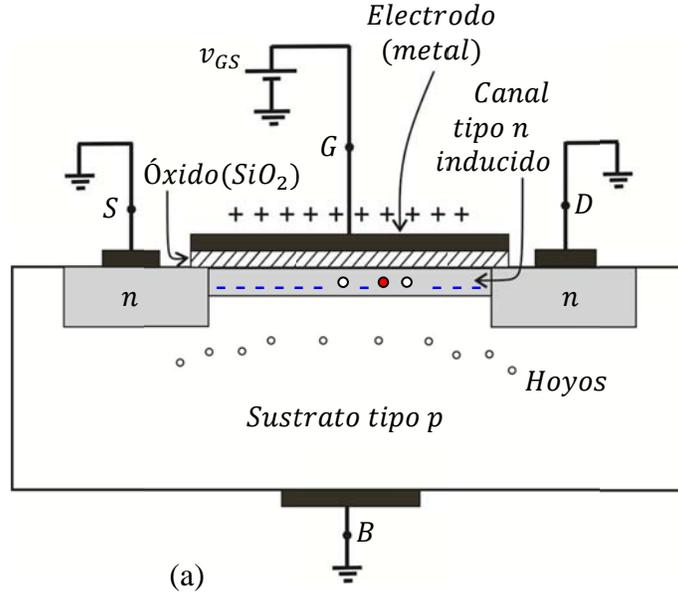
1. **Acumulamiento de restricciones atadas:** El conjunto de restricciones atadas $\mathcal{N}_{ata}^*(f_k)$ está posicionado hacia la izquierda de $\mathcal{C}(f_k)$ por efecto de ∇f_k . Recordar que la restricción "j" está ordenada en $\mathcal{C}(f_k)$ de acuerdo al ángulo entre ∇h_j y ∇f_k .
2. **Dispersión de las restricciones atadas:** Mientras mayor es $ang(\mathcal{K}^*(f_k))$ mayor es la dispersión de las restricciones atadas $\mathcal{N}_{ata}^*(f_k)$ en $\mathcal{C}(f_k)$. Notar que las restricciones de $\mathcal{N}_{ata}^*(f_1) = \{10 \ 1\}$ están más dispersas en el cluster $\mathcal{C}(f_1) = \{10 \ 9 \ 11 \ 8 \ 7 \ 1 \ 2 \ 6 \ 3 \ 4 \ 5\}$ porque $ang(\mathcal{K}^*(f_1)) = 6/11$ es el mayor que el resto de los ángulos. Por el contrario, las restricciones de $\mathcal{N}_{ata}^*(f_2) = \{5 \ 4\}$ están más concentradas en $\mathcal{C}(f_2) = \{5 \ 4 \ 3 \ 6 \ 2 \ 1 \ 7 \ 8 \ 11 \ 9 \ 10\}$ porque $ang(\mathcal{K}^*(f_2)) = 2/11$ es el menor que el resto de los ángulos. Notar en la figura 2 que los ángulos en grados de los conos KKT son $ang(\mathcal{K}^*(f_1)) \cong 90^\circ$ y $ang(\mathcal{K}^*(f_2)) \cong 60^\circ$.
3. **Acumulamiento de restricciones superfluas:** El acumulamiento del conjunto de restricciones atadas $\mathcal{N}_{ata}^*(f_k)$ en las primeras posiciones de $\mathcal{C}(f_k)$ y la dispersión de sus restricciones determinan el acumulamiento de algunas restricciones superfluas en las últimas posiciones de $\mathcal{C}(f_k)$.

Como conclusión se puede decir que (i) el posicionamiento de $\mathcal{N}_{ata}^*(f_k)$ en $\mathcal{C}(f_k)$ es causada por ∇f_k y la dispersión de sus restricciones en $\mathcal{C}(f_k)$ es causada por el ángulo del cono KKT $\mathcal{K}^*(f_k)$, y (ii) dado que las restricciones de $\mathcal{N}_{ata}^*(f_k)$ se posicionan a la izquierda de $\mathcal{C}(f_k)$, las restricciones superfluas se posicionarán a la derecha de $\mathcal{C}(f_k)$.

3.3.1 Fenómeno de polarización de restricciones:

El fenómeno de polarización eléctrica consiste en el acumulamiento en una región de un tipo de cargas eléctricas por la presencia de cargas de signo opuesto en una región próxima. Tal polarización se puede apreciar en el funcionamiento de un transistor de efecto de campo denominado como MOSFET [Sedra, A. S. et al (2006)] cuyo esquema se muestra en la figura 3 (a). El transistor MOSFET consiste de un sustrato de silicio tipo p , con portadores llamados *hoyos* representados en la figura por pequeños círculos, sobre el que se construyen regiones tipo n conectadas a las terminales S y D . Una fuente de voltaje v_{GS} se conecta a las placas de un capacitor formado por una placa metálica y el sustrato. Las placas están separadas por una tercera placa de óxido que funciona como dieléctrico.

El efecto polarizador de v_{GS} se muestra en la figura 3 (a). Cuando $v_{GS} \neq 0$ el canal tipo n es inducido en el cuerpo del sustrato con el propósito de permitir una corriente eléctrica entre las terminales D y S , pero con $v_{GS} = 0$ el canal desaparece. El canal tipo n mostrado se ha formado parcialmente con cargas negativas entre algunos hoyos por debajo de la placa de óxido. La entidad polarizadora es la fuente de voltaje v_{GS} , mientras que las cargas negativas en el sustrato son las entidades sujetas al efecto de la polarización.



(a)

| | | |
|---------------------------|--|----------------------------------|
| $\nabla f_1 = [-1, -2]^T$ | $\mathcal{C}(f_1) = \{10 \ 9 \ 11 \ 8 \ 7 \ 1 \ 2 \ 6 \ 3 \ 4 \ 5\}$ | $ang(\mathcal{K}^*(f_1)) = 6/11$ |
| $\nabla f_2 = [1, 2]^T$ | $\mathcal{C}(f_2) = \{5 \ 4 \ 3 \ 6 \ 2 \ 1 \ 7 \ 8 \ 11 \ 9 \ 10\}$ | $ang(\mathcal{K}^*(f_2)) = 2/11$ |
| $\nabla f_3 = [1, 0.8]^T$ | $\mathcal{C}(f_3) = \{4 \ 3 \ 5 \ 2 \ 1 \ 6 \ 7 \ 8 \ 11 \ 9 \ 10\}$ | $ang(\mathcal{K}^*(f_3)) = 3/11$ |

(b)

Fig. 3: Fenómeno de polarización: (a) por fuente eléctrica en un transistor MOSFET y (b) por gradiente ∇f_k y por ángulo del cono KKT $\mathcal{K}^*(f_k)$.

El efecto polarizador de ∇f_k y del ángulo de $\mathcal{K}^*(f_k)$ se observa en la figura 3 (b). El gradiente ∇f_k atrae las restricciones de $\mathcal{N}_{ata}^*(f_k)$ hacia el lado izquierdo de $\mathcal{C}(f_k)$, pero cada $ang(\mathcal{K}^*(f_k))$ las acerca o las aparta de este lado izquierdo. La dispersión de $\mathcal{N}_{ata}^*(f_1) = \{10 \ 1\}$ en $\mathcal{C}(f_1) = \{10 \ 9 \ 11 \ 8 \ 7 \ 1 \ 2 \ 6 \ 3 \ 4 \ 5\}$ es causada por el amplio valor de $ang(\mathcal{K}^*(f_1)) = 6/11$. Pero la concentración de $\mathcal{N}_{ata}^*(f_2) = \{5 \ 4\}$ en $\mathcal{C}(f_2) = \{5 \ 4 \ 3 \ 6 \ 2 \ 1 \ 7 \ 8 \ 11 \ 9 \ 10\}$ es causada por el bajo valor de $ang(\mathcal{K}^*(f_2)) = 2/11$.

El fenómeno de agrupamiento de cargas eléctricas negativas sujetas al efecto polarizador de una fuente de voltaje v_{GS} puede ser comparado con el agrupamiento de restricciones sujetas al efecto polarizador de ∇f_k y de $ang(\mathcal{K}^*(f_k))$. La tabla 2 muestra algunas correspondencias entre el fenómeno eléctrico y el fenómeno numérico.

| <i>Polarización eléctrica</i> | <i>Polarización numérica</i> |
|---|--|
| Entidad polarizadora: v_{GS} | Entidades polarizadoras: ∇f_k y $ang(\mathcal{K}^*(f_k))$ |
| Canal inducido tipo n | Zona izquierda de $\mathcal{C}(f_k)$ |
| Cargas eléctricas negativas en el canal | Restricciones de $\mathcal{N}_{ata}^*(f_k)$ en la zona izquierda de $\mathcal{C}(f_k)$ |
| Hoyos en el canal | Restricciones superfluas en la zona izquierda de $\mathcal{C}(f_k)$ |
| Hoyos en el sustrato | Restricciones superfluas en la zona derecha de $\mathcal{C}(f_k)$ |

Tabla 2: Comparación de la polarización eléctrica y la numérica

De este análisis se concluye que existe una polarización de las restricciones en el cluster $\mathcal{C}(f_k)$. Aunque la polarización se puede apreciar en $\mathcal{C}(f_k)$, la selección de las restricciones superfluas a partir de $\mathcal{C}(f_k)$, ∇f_k y $ang(\mathcal{K}^*(f_k))$ no es factible. Una buena razón es que $ang(\mathcal{K}^*(f_k))$ solo se conoce hasta que el problema de programación lineal es resuelto. Por tanto la polarización de restricciones en $\mathcal{C}(f_k)$ es de interés conceptual.

3.3.2 Condiciones del conjunto de restricciones superfluas:

Se ha encontrado que el gradiente ∇f_k de la función objetivo y el ángulo del cono KKT $ang(\mathcal{K}^*(f_k))$ son los agentes polarizadores de las restricciones de $\mathcal{C}(f_k)$. La polarización hace que las restricciones atadas $\mathcal{N}_{ata}^*(f_k)$ se acumulen en el extremo izquierdo de $\mathcal{C}(f_k)$ y que las restricciones superfluas se acumulen en el derecho. Sea $\mathcal{F}_{ang}(f_k)$ al conjunto de restricciones superfluas. Para precisar el concepto de $\mathcal{F}_{ang}(f_k)$ se establecen enseguida las condiciones necesarias que debe satisfacer.

Se da por hecho que las restricciones del conjunto $\mathcal{F}_{ang}(f_k)$ se concentran en el extremo derecho de $\mathcal{C}(f_k)$. Se propone por tanto que el conjunto de restricciones superfluas $\mathcal{F}_{ang}(f_k)$ satisfaga las dos condiciones siguientes:

- (i) Condición de superfluidad: $\mathcal{N}_{ata}^*(f_k) \not\subset \mathcal{F}_{ang}(f_k)$
- (ii) Condición de partición: $\mathcal{F}_{ang}(f_k)$ particiona $\mathcal{C}(f_k)$ como sigue

$$\mathcal{C}(f_k) = \mathcal{N}(f_k) \cup \mathcal{F}_{ang}(f_k) \text{ y } \mathcal{N}(f_k) \cap \mathcal{F}_{ang}(f_k) = \phi \quad (9)$$

Dado que las restricciones de $\mathcal{N}(f_k)$ son vecinas angulares de ∇f_k será llamado *vecindad angular* de ∇f_k . Dado que $\mathcal{N}_{ata}^*(f_k) \not\subset \mathcal{F}_{ang}(f_k)$, $\mathcal{N}(f_k)$ debe satisfacer la propiedad siguiente:

$$\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}(f_k) \quad (10)$$

de tal forma que la cardinalidad de $\mathcal{F}_{ang}(f_k)$ sea del mayor valor posible.

La partición (9) sugiere que $\mathcal{F}_{ang}(f_k)$ puede ser suprimido de $\mathcal{C}(f_k)$ siempre que $\mathcal{F}_{ang}(f_k)$ o $\mathcal{N}(f_k)$ sean calculables. En el peor de los casos se puede tener $\mathcal{N}(f_k) = \mathcal{C}(f_k)$ y $\mathcal{F}_{ang}(f_k) = \phi$, aunque se buscará que $\mathcal{F}_{ang}(f_k) \neq \phi$. Notar que la vecindad $\mathcal{N}(f_k)$ puede reemplazar a $\mathcal{C}(f_k)$ para propósitos de optimización siempre que $\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}(f_k)$.

Para ilustrar la formación de la partición (9) consideremos $\mathcal{C}(f_1)$ en la suposición de que $\mathcal{N}_{ata}^*(f_1) = \{\mathbf{10 1}\}$ es conocido. Una partición (no única) es la siguiente:

$$\begin{aligned}\mathcal{C}(f_1) &= \{\mathbf{10} 9 11 8 7 \mathbf{1} 2 6 3 4 5\} \\ &= \{\mathbf{10} 9 11 8 7 \mathbf{1} 2 6\} \cup \{3 4 5\} \\ &= \mathcal{N}(f_1) \cup \mathcal{F}_{ang}(f_1)\end{aligned}$$

Donde

$$\begin{aligned}\mathcal{N}(f_1) &= \{\mathbf{10} 9 11 8 7 \mathbf{1} 2 6\} \text{ con la propiedad } \mathcal{N}_{ata}^*(f_1) \subseteq \mathcal{N}(f_1) \\ \mathcal{F}_{ang}(f_1) &= \{3 4 5\} \text{ con la propiedad } \mathcal{N}_{ata}^*(f_1) \not\subseteq \mathcal{F}_{ang}(f_1)\end{aligned}$$

3.3.3 Formulación del problema de identificación de restricciones superfluas:

Dado el conjunto $\mathcal{C}(f_k)$ de restricciones polarizadas, determinar el conjunto de restricciones superfluas $\mathcal{F}_{ang}(f_k)$ que satisfaga:

- (i) Condición de superfluidad: $\mathcal{N}_{ata}^*(f_k) \not\subseteq \mathcal{F}_{ang}(f_k)$
- (ii) Condición de partición: $\mathcal{C}(f_k) = \mathcal{N}(f_k) \cup \mathcal{F}_{ang}(f_k)$ y $\mathcal{N}(f_k) \cap \mathcal{F}_{ang}(f_k) = \phi$

Estas condiciones aseguran la existencia de $\mathcal{F}_{ang}(f_k)$ y por tanto de la vecindad angular $\mathcal{N}(f_k)$, pero no provee alguna guía para determinar estos conjuntos. En las secciones que siguen se mostrará que $\mathcal{F}_{ang}(f_k)$ es calculable. Ya fue establecido que el cálculo de $\mathcal{F}_{ang}(f_k)$ a partir de $\mathcal{C}(f_k)$, ∇f_k y $ang(\mathcal{K}^*(f_k))$ no es factible porque $\mathcal{K}^*(f_k)$ es conocido sólo hasta que el problema de programación de interés ha sido resuelto.

3.4 Identificación de restricciones superfluas

Aunque el cluster de restricciones $\mathcal{C}(f_k)$ exhibe el fenómeno de polarización debido a ∇f_k y $\mathcal{K}^*(f_k)$, no es factible el cálculo de las restricciones superfluas $\mathcal{F}_{ang}(f_k)$ a partir de $\mathcal{C}(f_k)$ debido a que $\mathcal{K}^*(f_k)$ es conocido sólo hasta que se ha calculado la solución. Además del cono KKT $\mathcal{K}^*(f_k)$, en esta sección se usarán los conos llamados cono de mejora y cono de factibilidad para determinar el conjunto de restricciones superfluas $\mathcal{F}_{ang}(f_k)$.

Se ha observado que los gradientes ∇h_j de las restricciones en $\mathcal{N}(f_k)$ gravitan angularmente en la vecindad de ∇f_k mientras los gradientes de las restricciones en $\mathcal{F}_{ang}(f_k)$ permanecen angularmente apartadas de ∇f_k . Se ha observado también que los

gradientes ∇h_j de las restricciones en $\mathcal{N}(f_k)$ pertenecen al cono llamado cono de mejora. Dado que el cono de mejora es analíticamente descriptible, $\mathcal{N}(f_k)$ es calculable.

Consideremos el problema de programación lineal dado en (2) junto con el gradiente $\nabla f_1 = [-1 \ -2]^T$ de la función objetivo, los gradientes $\nabla h_j, j = 1, \dots, 11$ de cada restricción y el cluster de restricciones $\mathcal{C}(f_1) = \{10 \ 9 \ 11 \ 8 \ 7 \ 1 \ 2 \ 6 \ 3 \ 4 \ 5\}$, los cuales se muestran en la figura 5. En la figura se usa \mathbf{x}^t como vértice de referencia cuya proyección ortogonal sobre ∇f_1 es el punto \mathbf{q}^t . Una línea de referencia Q se hace pasar por los puntos \mathbf{x}^t y \mathbf{q}^t , la cual es perpendicular a ∇f_1 . El inicio de cada vector ∇h_j se ha trasladado desde origen de R^2 hasta el punto \mathbf{q}^t con el propósito de referirlos a la línea Q .

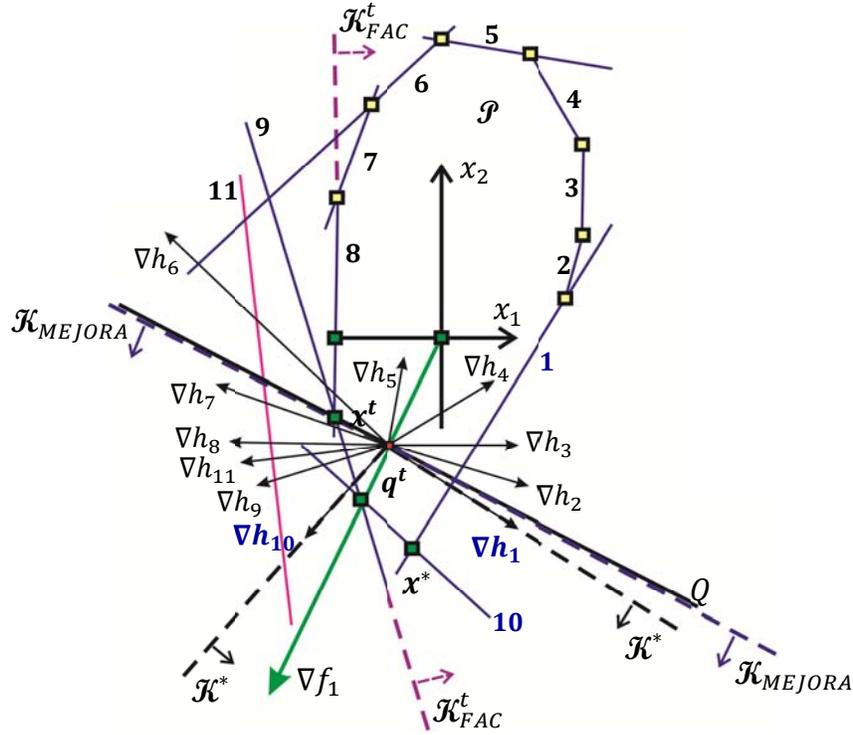


Fig. 4: Cono de direcciones factibles \mathcal{K}_{FAC}^t , cono de mejora \mathcal{K}_{MEJORA} y cono KKT \mathcal{K}^* asociados al vértice \mathbf{x}^t .

Los conceptos de cono de direcciones factibles \mathcal{K}_{FAC}^t y cono de direcciones de mejora \mathcal{K}_{MEJORA} serán precisados más adelante. Estos conos nos facilitarán el análisis de la polarización de las restricciones de $\mathcal{C}(f_1)$ en los dos conjuntos $\mathcal{N}(f_1)$ y $\mathcal{F}_{ang}(f_1)$. Comenzaremos por la definición de distancia angular y continuaremos con las definiciones de los conos.

3.4.1 Distancia angular respecto de la función objetivo

De acuerdo a la definición 2, el cluster de restricciones es el conjunto de restricciones ordenadas por medio del criterio de cercanía angular del gradiente $\nabla h_j, j = 1, 2, \dots, m$ de cada restricción al gradiente ∇f_1 de la función objetivo. Para ejemplificar la cercanía

angular determinemos el ángulo θ_{10} de $\nabla h_{10} = [-0.6783 \quad -0.7348]^T$ respecto de $\nabla f_1 = [-1 \quad -2]^T$ el cual se muestra en la figura 6.

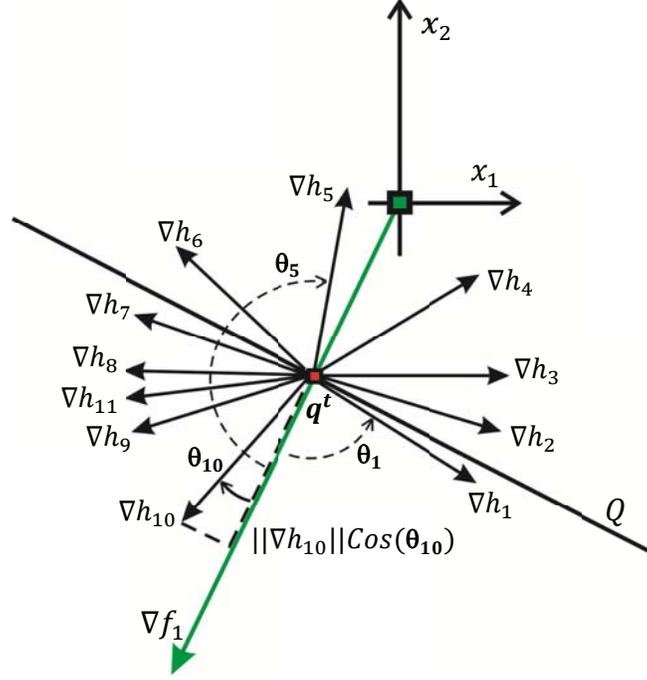


Fig. 5: Medida angular de un vector ∇h_j respecto de ∇f_1

Como medida angular entre ∇h_{10} y ∇f_1 usaremos el coseno del ángulo. El fenómeno de polarización de restricciones se observa cuando esta medida se compara con la línea de referencia Q . De la definición del producto escalar de dos vectores: $(\nabla f_1)^T \nabla h_{10} = \|\nabla f_1\| \cdot \|\nabla h_{10}\| \cos(\theta_{10})$ se tiene:

$$\cos(\theta_{10}) = \frac{(\nabla f_1)^T \nabla h_{10}}{\|\nabla f_1\| \cdot \|\nabla h_{10}\|} = \frac{2.1479}{2.2361} = 0.9606 \quad (11)$$

Como se muestra en la figura 5, el ángulo $\theta_j, j = 1, \dots, 11$ se mide respecto de ∇f_1 aún en $R^n, n \geq 2$. Los rangos de valores de θ_j y de $\cos(\theta_j)$ son los siguientes:

$$\begin{aligned} \theta_j &\in [0, \pi] \\ \cos(\theta_j) &\in [+1, -1] \end{aligned} \quad (12)$$

La posición angular de la línea Q corresponde a $\theta = \pi/2$, o bien a $\cos(\pi/2) = 0$. Notemos que se aprovecha la propiedad par de la función coseno $\cos(\theta) = \cos(-\theta)$ como distancia angular de un vector ∇h_j desde ∇f_1 . Los resultados numéricos de las medidas angulares de las 11 restricciones de la figura 5 se despliegan en la figura 6.

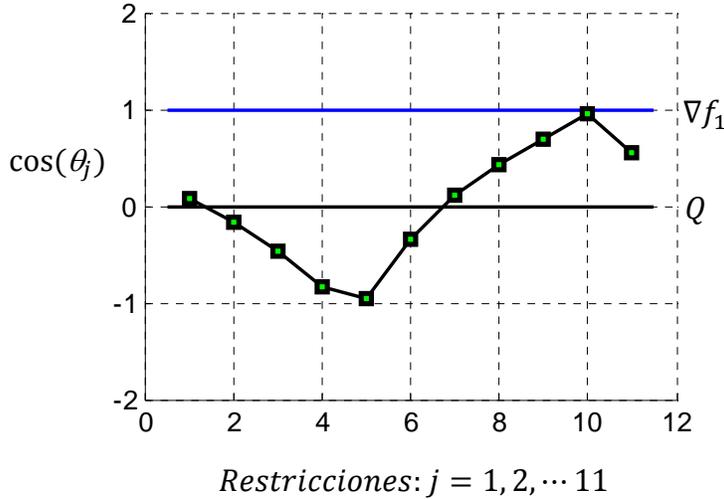


Fig. 6: Polarización angular de restricciones

Considerando los datos numéricos de la tabla en la figura 6, las 11 restricciones pueden ser ordenadas de acuerdo a los valores de $\text{Cos}(\theta_j)$, $j = 1, 2, \dots, 11$, de mayor a menor. Se obtiene por tanto el siguiente cluster de restricciones:

$$\mathcal{E}(f_1) = \{10 \ 9 \ 11 \ 8 \ 7 \ 1 \ 2 \ 6 \ 3 \ 4 \ 5\}$$

De los elementos de $\mathcal{E}(f_1)$ y de la gráfica de la figura 6 se observa que la restricción angularmente más próxima a ∇f_1 es la 10, mientras que la más lejana angularmente es la 5. Se observa también que la línea de referencia Q separa los valores de $\text{Cos}(\theta_j)$ en dos conjuntos produciéndose por tanto un fenómeno de polarización de restricciones. Aunque ∇f_1 es el agente polarizador, la línea Q funciona como discriminante.

3.3.2 Criterio angular por polarización para identificación de restricciones superfluas

De la figura 7 notamos que las medidas angulares $\text{cos}(\theta_j)$, $j = 1, 2, \dots, 11$ son de doble utilidad. Sus valores relativos definen el cluster de restricciones como se acaba de mostrar, mientras que sus signos determinan la polarización de las restricciones en dos conjuntos, como se muestra enseguida.

Consideremos ahora la gráfica de la figura 6 en la suposición de que el conjunto $\mathcal{N}_{ata}^*(f_1)$ es desconocido. Notamos que los signos de las medidas angulares $\text{cos}(\theta_j)$, $j = 1, 2, \dots, 11$ respecto de la línea Q tienen significados relacionados con la vecindad angular $\mathcal{N}(f_1)$ y por tanto con el conjunto de restricciones superfluas $\mathcal{F}_{ang}(f_1)$. Por observación visual de la gráfica y la tabla de la figura 6 notamos que la restricción "j" cuya medida angular $\text{Cos}(\theta_j)$ es positiva pertenece a la vecindad angular $\mathcal{N}(f_1)$, teniéndose: $\mathcal{N}(f_1) = \{1, 7, 8, 9, 10, 11\}$. Se propone por tanto que la vecindad angular $\mathcal{N}(f_k)$ sea calculada como sigue:

$$\mathcal{N}(f_k) = \{j \in \mathcal{M}(f_k) | \cos(\theta_j) \geq 0\} \quad (13)$$

Donde $\mathcal{M}(f_k)$ es el conjunto original de restricciones. Podría usarse $\mathcal{C}(f_k)$ en lugar de $\mathcal{M}(f_k)$, lo cual no es conveniente porque $\mathcal{C}(f_k)$ tiene un costo computacional, por tanto, $\mathcal{C}(f_k)$ será sólo de interés conceptual.

Por su parte, la restricción "i" cuya medida angular $\cos(\theta_i)$ es estrictamente negativa pertenece al conjunto de restricciones superfluas $\mathcal{F}_{ang}(f_1)$. Para la figura 6 se tiene: $\mathcal{F}_{ang}(f_1) = \{2, 3, 4, 5, 6\}$. Se propone por tanto que el conjunto de restricciones superfluas sea calculado como sigue:

$$\mathcal{F}_{ang}(f_k) = \{j \in \mathcal{M}(f_k) | \cos(\theta_j) < 0\} \quad (14)$$

Las expresiones para $\mathcal{N}(f_k)$ y $\mathcal{F}_{ang}(f_k)$ dadas en (12) y (13) respectivamente constituyen el criterio angular por polarización para la identificación de restricciones superfluas, las cuales satisfacen la condición de partición del cluster de restricciones.

El resultado de la polarización para nuestro ejemplo son los conjuntos $\mathcal{N}(f_1)$ y $\mathcal{F}_{ang}(f_1)$. Como se puede observar en la gráfica de la figura 7, la polarización de las restricciones del problema de programación dado es cuantificada por la línea Q . Las restricciones cuyas medidas angulares $\cos(\theta_i)$ están debajo de ésta línea son restricciones superfluas. La polarización de restricciones determinada de esta forma también se observa en el cluster de restricciones $\mathcal{C}(f_1)$. Las restricciones de $\mathcal{N}(f_1)$ se posicionan hacia el lado izquierdo de $\mathcal{C}(f_1)$ mientras que las de $\mathcal{F}_{ang}(f_1)$ se posicionan hacia su lado derecho. Dado que $\mathcal{N}_{ata}^*(f_1) = \{\mathbf{1}, \mathbf{10}\}$, es fácil ver que los conjuntos $\mathcal{N}(f_1)$ y $\mathcal{F}_{ang}(f_1)$ satisfacen las condiciones de partición y de superfluidad:

$$\begin{aligned} \mathcal{C}(f_1) &= \mathcal{N}(f_1) \cup \mathcal{F}_{ang}(f_1) \text{ y } \mathcal{N}(f_1) \cap \mathcal{F}_{ang}(f_1) = \phi \\ \mathcal{N}_{ata}^*(f_1) &\subseteq \mathcal{N}(f_1), \mathcal{N}_{ata}^*(f_1) \not\subseteq \mathcal{F}_{ang}(f_1) \end{aligned}$$

Aunque la clasificación de $\mathcal{C}(f_1)$ en $\mathcal{N}(f_1)$ y $\mathcal{F}_{ang}(f_1)$ se realizó en R^2 , la clasificación es también válida para R^n . Es decir, la medida angular $\cos(\theta_j)$ sigue siendo una curva en un plano a pesar de que los vectores gradientes estén en $R^n, n \geq 2$, lo cual se explica como sigue: Como se observa en la figura 6, la proyección ortogonal de ∇h_j sobre ∇f_1 es:

$$\nabla h_j^\perp = \|\nabla h_j\| \cos(\theta_j) \widehat{\nabla f_1} \quad (15)$$

Por tanto $\cos(\theta_j)$ es una cantidad que se puede medir sobre un eje de los reales alineado con ∇f_1 y origen el punto \mathbf{q}^t . Para un problema de programación lineal dado, ∇f_1 es constante y el eje de los reales alineado con él es fijo. Dado que la proyección ortogonal ∇h_j^\perp está definida en R^n y se encuentra sobre ∇f_1 , la medida angular $\cos(\theta_j)$ es válida en el espacio R^n . Notar que para el espacio R^n , la línea de referencia Q que pasa por los puntos \mathbf{x}^t y \mathbf{q}^t pertenece a un hiperplano. Este hiperplano separa el conjunto de restricciones $\mathcal{N}(f_1)$ del conjunto de restricciones superfluas $\mathcal{F}_{ang}(f_1)$. Este hiperplano es

de interés conceptual, aunque podría ser calculado en términos de \mathbf{q}^t . Es decir, la línea de referencia Q es suficiente para propósitos de polarización de restricciones.

3.3.3 Conos convexos de mejora y de factibilidad

La identificación de los conjuntos $\mathcal{N}(f_k)$ y $\mathcal{F}_{ang}(f_k)$ por medio de la evaluación de los signos de $\cos(\theta_j)$, $j = 1, 2, \dots, m$ respecto de la línea Q es razonable. Sin embargo, no se ha garantizado que estos conjuntos satisfagan la condición de superfluidad:

$$\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}(f_k), \text{ o bien } \mathcal{N}_{ata}^*(f_k) \not\subseteq \mathcal{F}_{ang}(f_k)$$

Definiremos enseguida los conos de direcciones factibles y de direcciones de mejora para intentar formalizar la identificación de $\mathcal{N}(f_k)$ y $\mathcal{F}_{ang}(f_k)$ por medio de (20) y (21) respectivamente. La figura 5 muestra dos conos que son de nuestro interés: el de direcciones factibles \mathcal{K}_{FAC}^t y el de direcciones de mejora \mathcal{K}_{MEJORA} con vértices posicionados en \mathbf{q}^t . En la figura 8, los vértices de \mathcal{K}_{FAC}^t y \mathcal{K}_{MEJORA} han sido desplazados a \mathbf{x}^t para propósitos de comparación. Enseguida se precisan sus definiciones.

Definición 8: Cono de direcciones factibles \mathcal{K}_{FAC}^t

El cono de direcciones factibles del polígono \mathcal{P} en \mathbf{x}^t se denota por \mathcal{K}_{FAC}^t y se define como:

$$\mathcal{K}_{FAC}^t = \{\mathbf{d} \in \mathbb{R}^n \mid \mathbf{d} \neq \mathbf{0}, (\mathbf{x}^t + \lambda \mathbf{d}) \in \mathcal{P} \text{ para todo } \lambda \in (0, \delta) \text{ para algún } \delta > 0\} \quad (16)$$

Cada vector $\mathbf{d} \in \mathcal{K}_{FAC}^t$ distinto de cero es llamado una dirección factible. Para un polígono \mathcal{P} determinado por restricciones lineales como el de la figura 8, el vértice del cono puede estar en un vértice del polígono \mathcal{P} , el \mathbf{x}^t , y sus lados son por tanto hiperplanos que delimitan \mathcal{P} . El vector $(\mathbf{x}^t + \mathbf{d})$ puede salirse de \mathcal{P} pero no $(\mathbf{x}^t + \lambda \mathbf{d})$. Notemos que \mathcal{K}_{FAC}^t es convexo y que $\mathcal{P} \subset \mathcal{K}_{FAC}^t$.

Definición 9: Cono de direcciones de mejora \mathcal{K}_{MEJORA}

El cono de direcciones de mejora de la función objetivo f_k en \mathbf{x}^t se denota por \mathcal{K}_{MEJORA} y se define como sigue:

$$\mathcal{K}_{MEJORA} = \{\mathbf{d} \in \mathbb{R}^n \mid (\nabla f_k)^T \mathbf{d} > 0\} \quad (17)$$

Para el ejemplo en \mathbb{R}^2 que se está analizando, el cono \mathcal{K}_{MEJORA} está ligeramente despegado de la línea Q como se intenta mostrar en la figura 8. Notemos que \mathcal{K}_{MEJORA} es un conjunto convexo.

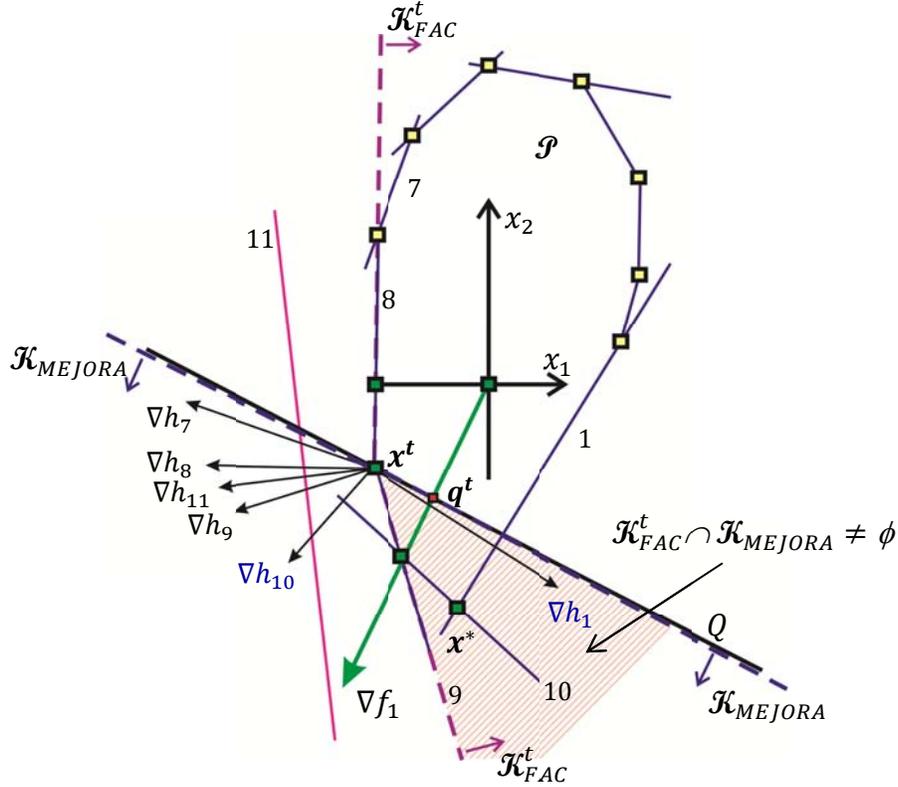


Fig. 7: Cono de direcciones factibles \mathcal{K}_{FAC}^t y cono de mejora \mathcal{K}_{MEJORA}

Notemos en la figura 7 que el cono \mathcal{K}_{FAC}^t depende del vértice \mathbf{x}^t , el cual se encuentra en la frontera de \mathcal{P} . Es decir, \mathcal{K}_{FAC}^t cambia en cada iteración "t". Sin embargo, el cono de mejora \mathcal{K}_{MEJORA} no depende del vértice \mathbf{x}^t porque $\nabla f_k(\mathbf{x}^t) = \nabla f_k$, es decir, no depende de la iteración "t", por lo que este cono está definido aún antes de comenzar la ejecución del algoritmo simplex. Notemos también que $\mathcal{K}_{FAC}^t \cap \mathcal{K}_{MEJORA} \neq \emptyset$ en \mathbf{x}^t que no es el vértice óptimo.

De la figura 8 notamos que los vectores ∇h_j cuyas restricciones "j" pertenecen a la vecindad angular $\mathcal{N}(f_k) = \{j \in \mathcal{C}(f_k) \mid (\nabla f_k)^T \nabla h_j \geq 0\} = \{1, 7, 8, 9, 10, 11\}$ caen en el cono \mathcal{K}_{MEJORA} .

3.3.4 Relación entre vecindad angular y cono de mejora

Se precisa aquí la relación entre la vecindad angular $\mathcal{N}(f_k)$ y el cono \mathcal{K}_{MEJORA} . Comparemos las definiciones de $\mathcal{N}(f_k)$ dada en (20) y de \mathcal{K}_{MEJORA} dada en (24):

$$\mathcal{N}(f_k) = \{j \in \mathcal{M}(f_k) \mid (\nabla f_k)^T \nabla h_j \geq 0\}$$

$$\mathcal{K}_{MEJORA} = \{\mathbf{d} \in \mathbb{R}^n \mid (\nabla f_k)^T \mathbf{d} > 0\}$$

Si \mathbf{d} es un vector que mejora la función objetivo f_k , entonces ∇h_j también la mejora siempre que $(\nabla f_k)^T \nabla h_j > 0$ para $j \in \mathcal{N}(f_k)$. El caso $(\nabla f_k)^T \nabla h_j = 0$ para $j \in \mathcal{N}(f_k)$ se

supondrá que es un accidente. Notar que es seguro que el vector ∇h_j mejora la función objetivo siempre que $(\nabla f_k)^T \nabla h_j > 0$ para $j \in \mathcal{N}(f_k)$, lo que no es seguro es que $j \in \mathcal{N}(f_k)$ implique que ∇h_j delimite el cono KKT.

3.3.5 Condición de superfluidad determinada por la condición de optimalidad

Dado que $\nabla h_j \in \mathcal{K}_{MEJORA}$ para $j \in \mathcal{N}(f_k)$ y considerando que \mathcal{K}_{FAC}^* y \mathcal{K}_{MEJORA} definen el vértice óptimo \mathbf{x}^* [Bazaraa (2006)], se puede plantear la hipótesis de que el cono KKT $\mathcal{K}^* \subset \mathcal{K}_{MEJORA}$, y por tanto que \mathcal{K}^* está delimitado por ∇h_j para $j \in \mathcal{N}(f_k)$. Si se prueba que esta hipótesis es aceptable, se estaría garantizando la condición de superfluidad: $\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}(f_k)$.

Consideremos los conos \mathcal{K}_{FAC}^* y \mathcal{K}_{MEJORA} definidos en el vértice óptimo \mathbf{x}^* , los cuales se han desplazado a \mathbf{x}^* por facilidad de análisis como se muestra en la figura 9:

$$\mathcal{K}_{FAC}^* = \{\mathbf{d} \in \mathbb{R}^n \mid \mathbf{d} \neq \mathbf{0}, (\mathbf{x}^t + \lambda \mathbf{d}) \in \mathcal{P} \text{ para todo } \lambda \in (0, \delta) \text{ para algún } \delta > 0\} \quad (18)$$

$$\mathcal{K}_{MEJORA} = \{\mathbf{d} \in \mathbb{R}^n \mid (\nabla f_k)^T \mathbf{d} > 0\}$$

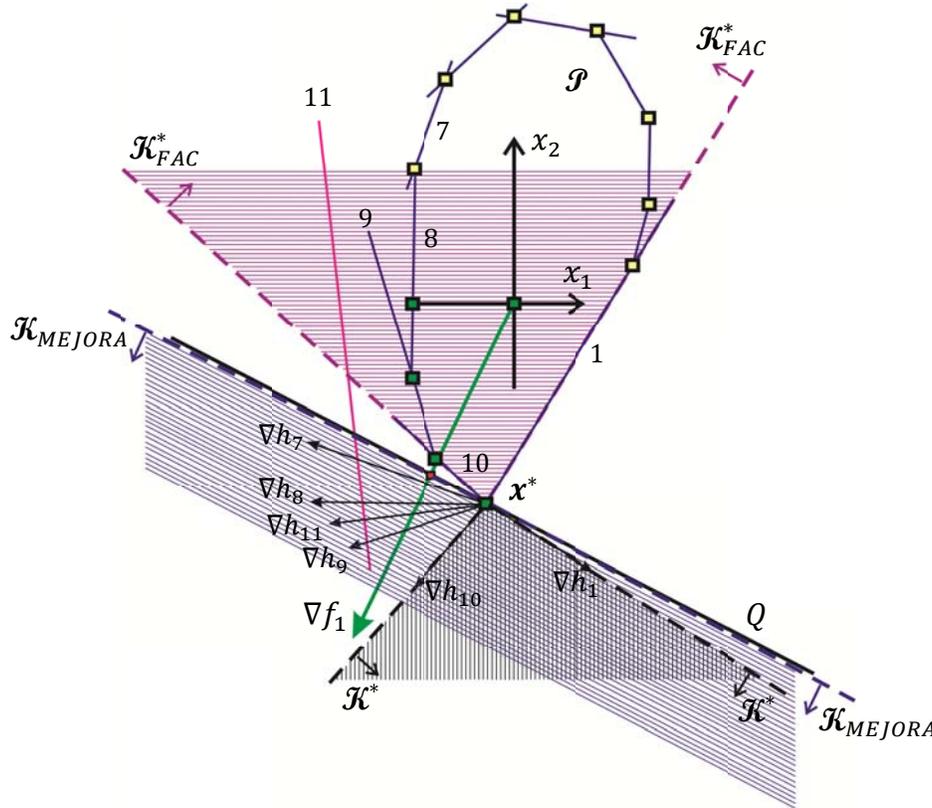


Fig. 8: Conos \mathcal{K}_{MEJORA} , \mathcal{K}_{FAC}^* y \mathcal{K}^* , y la condición óptima del vértice \mathbf{x}^*

De (18) y la figura 8 notamos que un ligero movimiento desde \mathbf{x}^* a lo largo de $\mathbf{d} \in \mathcal{K}_{FAC}^*$ produce un punto en el polígono \mathcal{P} . Mientras que un movimiento similar a lo largo de $\mathbf{d} \in \mathcal{K}_{MEJORA}$ produce un punto que mejora la función objetivo, es decir $(\nabla f_k(\mathbf{x}^*))^T \mathbf{d} > 0$. Como se muestra en el siguiente teorema, si \mathbf{x}^* es un óptimo local y si $(\nabla f_k(\mathbf{x}^*))^T \mathbf{d} > 0$, entonces $\mathbf{d} \notin \mathcal{K}_{FAC}^*$. Es decir, una condición necesaria para optimalidad local es que todo vector \mathbf{d} de mejora no es un vector factible.

Teorema de optimalidad: Condición geométrica de optimalidad [Bazaraa (2006)]:
 Considere el problema de programación lineal dado en (1), el cual se reescribe a continuación:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{maximizar}} \quad f_k(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ & \text{subjeto a: } \mathbf{a}_j^T \mathbf{x} \leq b_j, j = 1, \dots, m \end{aligned}$$

donde $f_k(\mathbf{x}): R^n \rightarrow R$, \mathcal{P} es un conjunto convexo no vacío en R^n , y $\nabla f_k(\mathbf{x}) = \mathbf{c}$. Luego,

1. Si \mathbf{x}^* es una solución óptima local (referirse a las figuras 8 o 9) entonces:

$$\mathcal{K}_{MEJORA} \cap \mathcal{K}_{FAC}^* = \phi$$

2. Por el contrario, si se satisface (referirse a la figura 9) lo siguiente:

- $\mathcal{K}_{MEJORA} \cap \mathcal{K}_{FAC}^* = \phi$
- $(\nabla f_k(\mathbf{x}^*))^T (\mathbf{x} - \mathbf{x}^*) > 0$ implica $f_k(\mathbf{x}) > f_k(\mathbf{x}^*)$. Esta propiedad que se conoce como pseudoconvexidad de f_k en \mathbf{x}^*
- Existe una vecindad $\mathcal{N}_\delta(\mathbf{x}^*)$, $\delta > 0$ tal que $(\mathbf{x} - \mathbf{x}^*) \in \mathcal{K}_{FAC}^*$ para todo $\mathbf{x} \in (\mathcal{P} \cap \mathcal{N}_\delta(\mathbf{x}^*))$.

entonces, \mathbf{x}^* es un óptimo local de f_k .

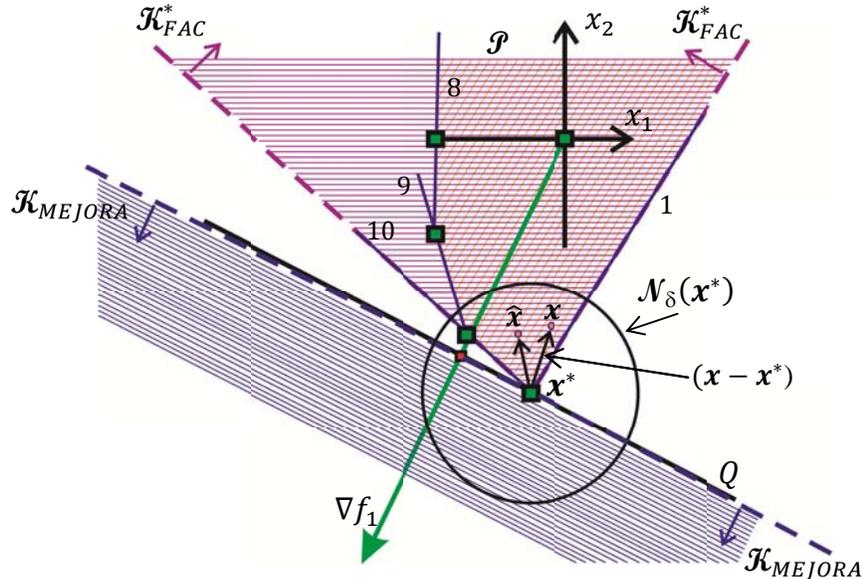


Fig. 9: Vecindad $\mathcal{N}_\delta(\mathbf{x}^*)$, $\delta > 0 \ni (\mathbf{x} - \mathbf{x}^*) \in \mathcal{K}_{FAC}^* \forall \mathbf{x} \in (\mathcal{P} \cap \mathcal{N}_\delta(\mathbf{x}^*))$

3.3.6 Conclusiones derivadas del teorema de optimalidad:

El teorema muestra que \mathbf{x}^* es solución óptima de un problema de programación lineal si y solo si $\mathcal{K}_{MEJORA} \cap \mathcal{K}_{FAC}^* = \emptyset$. Es decir, si $\nabla h_j \in \mathcal{K}_{MEJORA}$ entonces $\nabla h_j \notin \mathcal{K}_{FAC}^*$ y no más.

En otras palabras, la condición: $\nabla h_j \in \mathcal{K}_{MEJORA}$ implica $\nabla h_j \notin \mathcal{K}_{FAC}^*$ asegura que \mathbf{x}^* es solución óptima, pero no asegura que ∇h_j delimite el cono KKT \mathcal{K}^* para $j \in \mathcal{N}(f_k)$. Es decir, no se puede asegurar que $\mathcal{K}^* \subset \mathcal{K}_{MEJORA}$. No se puede garantizar por tanto la condición de superfluidad: $\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}(f_k)$.

Lo que es cierto es que en muchas ocasiones sucede $\mathcal{K}^* \subset \mathcal{K}_{MEJORA}$ y sucede por tanto $\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}(f_k)$. Este es el caso del problema de programación lineal dado en (2) con cluster de restricciones $\mathcal{C}(f_1) = \{10 \ 9 \ 11 \ 8 \ 7 \ 1 \ 2 \ 6 \ 3 \ 4 \ 5\}$ mostrado en la figura 4. Para estos casos se diseñará un mecanismo de identificación de restricciones superfluas.

3.3.7 Mecanismo de identificación de restricciones superfluas por criterio angular

Suponiendo que $\mathcal{K}^* \subset \mathcal{K}_{MEJORA}$, los conjuntos $\mathcal{N}(f_k)$ y $\mathcal{F}_{ang}(f_k)$ satisfacen las condiciones de superfluidad. Si este es el caso, el conjunto de restricciones superfluas podrá ser identificado por medio del mecanismo presentado enseguida.

| |
|--|
| <p>Para el problema de programación lineal dado en (1):</p> <ol style="list-style-type: none"> 1. Calcular: $\{\cos(\theta_j) \mid \cos(\theta_j) = \frac{(\nabla f)^T \nabla h_j}{\ \nabla f\ \cdot \ \nabla h_j\ }, j \in \mathcal{M}\}$ 2. Calcular: <ul style="list-style-type: none"> $\mathcal{N}(f) = \{j \in \mathcal{M} \mid \cos(\theta_j) \geq 0\}$ $\mathcal{F}_{ang}(f) = \{j \in \mathcal{M} \mid \cos(\theta_j) < 0\}$ 3. Sustituir el conjunto \mathcal{M} de todas las restricciones del problema de programación lineal por la vecindad angular $\mathcal{N}(f_k)$. |
|--|

Tabla 3: Mecanismo de identificación de restricciones superfluas por criterio angular.

La sustitución del conjunto \mathcal{M} del problema de programación lineal por la vecindad angular $\mathcal{N}(f_k)$ implica la supresión automática de las restricciones superfluas $\mathcal{F}_{ang}(f_k)$ y la elección de $\mathcal{N}(f_k)$ como dato de entrada al algoritmo simplex. Es decir, el mecanismo de la tabla 3 es un procedimiento computacional que cae en la clase de los métodos conocidos como métodos de pre-solución.

3.3.8 Complejidad del algoritmo después de la supresión por criterio angular

Consideremos un problema de programación lineal en el espacio R^n caracterizado por m restricciones y $(m + n)$ variables, en el cual la matriz básica B es cuadrada y de dimensión m . La complejidad computacional del algoritmo simplex en la optimización de este problema depende del número de formas de elegir los vectores columna de B de entre un total de $(m + n)$ vectores. Es decir, la complejidad del algoritmo al resolver el problema original con m restricciones está dada por la siguiente expresión [Hillier]:

$$C(m) = \binom{m+n}{m} = \frac{(m+n)!}{m!n!} \quad (19)$$

Sea m_{ang} el número de restricciones que quedan después de la supresión de las restricciones superfluas por criterio angular. Por tanto, después de esta supresión, la complejidad computacional del algoritmo simplex queda como sigue:

$$C(m_{ang}) = \binom{m_{ang}+n}{m_{ang}} = \frac{(m_{ang}+n)!}{m_{ang}!n!} \quad (20)$$

Para evaluar los cambios de complejidades por cambio de dimensión $m \rightarrow m_{ang}$, consideremos el problema definido en el espacio R^2 dado en (2). Los datos determinados para este problema se resumen enseguida:

$$\mathcal{C}(f_1) = \{10 \ 9 \ 11 \ 8 \ 7 \ 1 \ 2 \ 6 \ 3 \ 4 \ 5\}$$

$$\mathcal{N}(f_1) = \{10 \ 9 \ 11 \ 8 \ 7 \ 1\} \text{ y } \mathcal{F}_{ang}(f_1) = \{2 \ 6 \ 3 \ 4 \ 5\}$$

Para este ejemplo se tiene $m = |\mathcal{C}(f_1)| = 11$, $m_{ang} = |\mathcal{N}(f_1)| = 6$, y dado que el problema está definido en el espacio R^2 se tiene $n = 2$. Por tanto las complejidades antes y después de la supresión quedan como sigue:

$$C(m) = \binom{m+n}{m} = \binom{13}{11} = 78$$

$$C(m_{ang}) = \binom{m_{ang}+n}{m_{ang}} = \binom{8}{6} = 28$$

La razón de dimensiones por supresión angular se expresa como:

$$\eta_m = \frac{m_{ang}}{m} = \frac{6}{11} = 0.5455 \quad (21)$$

Para el cambio de complejidad $C_{orig} \rightarrow C_{ang}$ se define la siguiente razón de complejidades por supresión angular:

$$\eta_c = \frac{C(m)}{C(m_{ang})} = \frac{28}{78} = 0.3590 \quad (22)$$

Es decir, después de suprimir las restricciones superfluas $\mathcal{F}_{ang}(f_1)$, se preservaron el 54.55% de restricciones de $\mathcal{C}(f_1)$ en la vecindad angular $\mathcal{N}(f_1)$. Mientras que la complejidad del algoritmo se redujo al 35.90% de la complejidad original.

3.5 Problema de superfluidad por cono KKT

En muchas ocasiones sucede que $\mathcal{K}^* \subset \mathcal{K}_{MEJORA}$, por lo que se satisface la condición de superfluidad: $\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}(f_k)$. Pero hay algunos casos excepcionales en que $\mathcal{K}^* \not\subset \mathcal{K}_{MEJORA}$ los cuales se detectan hasta que se determina una solución del problema. Dado que en estos casos $\mathcal{N}_{ata}^*(f_k) \not\subseteq \mathcal{N}(f_k)$ y que $\mathcal{N}(f_k)$ ha sustituido a \mathcal{M} , la solución obtenida no será la óptima buscada.

Enseguida se analiza un caso en el que no se satisface la condición de superfluidad $\mathcal{N}_{ata}^*(f_k) \not\subseteq \mathcal{N}(f_k)$. En la figura 11 hay al menos una restricción cuyo gradiente delimita el cono KKT y al mismo tiempo es suprimido por no pertenecer al cono de mejora. Es decir, parte del cono KKT queda en el exterior del cono de mejora. La figura muestra el polítopo \mathcal{P} del ejemplo dado en (2) arreglado de tal manera que el gradiente ∇h_1 que delimita el cono KKT \mathcal{K}^* definido por $\{\nabla h_1, \nabla h_{10}\}$ se encuentre fuera del cono \mathcal{K}_{MEJORA} .

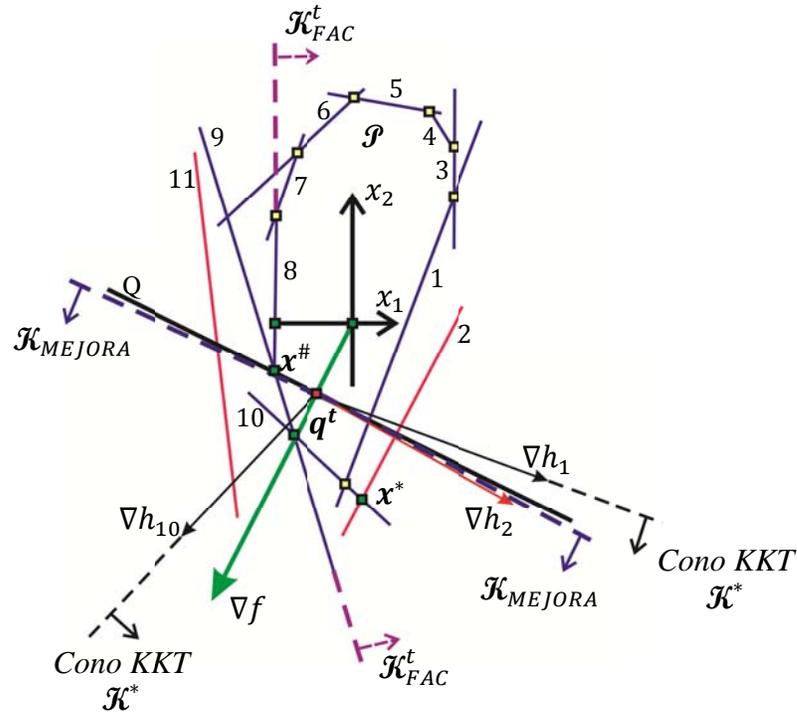


Fig. 10: Gradiente ∇h_1 excluido del Cono KKT \mathcal{K}^*

De acuerdo al criterio angular adoptado hasta este momento, las restricciones seleccionadas como superfluas son las que no se encuentran en el cono de mejora, y entre ellas está la restricción 1 cuyo gradiente ∇h_1 está fuera de éste cono. Los resultados numéricos de las medidas angulares de las 11 restricciones de la figura 11 se despliegan en la figura 12.

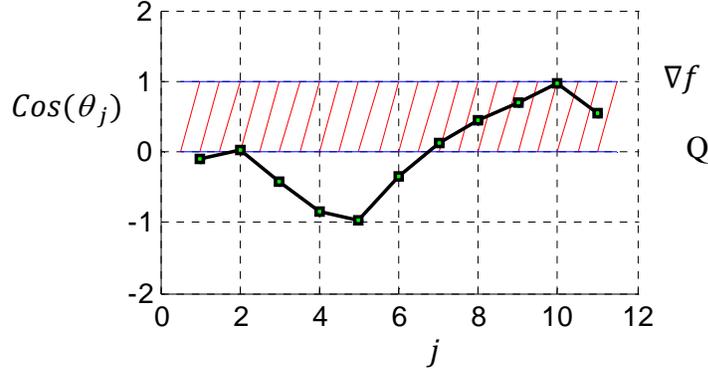


Fig. 11: Medida angular $\text{Cos}(\theta_j)$ de la j –ésima restricción

Aplicando el criterio angular expresado en (21) a las restricciones de la figura 12 se obtiene el conjunto de restricciones superfluas:

$$\mathcal{F}_{ang}(f_k) = \{j \in \mathcal{M}(f_k) | \text{Cos}(\theta_j) < 0\} = \{1, 3, 4, 5, 6\}$$

Este criterio construye $\mathcal{F}_{ang}(f_k)$ con restricciones que no están en el cono \mathcal{K}_{MEJORA} , y entre ellas está la restricción **1** que delimita \mathcal{K}^* . Dado que las restricciones que no están en \mathcal{K}_{MEJORA} se suprimen, la restricción **1** no formará parte de la solución del problema y el algoritmo KKT fallará produciendo una solución \mathbf{x}^* determinada por $\mathcal{N}_{ata}^*(f_k) = \{2, 10\}$, la cual es infactible como se muestra en la figura 11.

El mecanismo de supresión de restricciones superfluas por criterio angular de la tabla 3 es correcto excepto cuando $\mathcal{N}_{ata}^*(f_k) \not\subseteq \mathcal{N}(f_k)$. La pregunta natural que surge ahora es que si existe una forma de detectar las restricciones que están fuera del cono \mathcal{K}_{MEJORA} de manera que al ser insertadas a $\mathcal{N}(f_k)$ se pueda satisfacer la condición de superfluidad: $\mathcal{N}_{ata}^*(f_k) \subseteq \mathcal{N}(f_k)$.

La respuesta es que sí existe una forma de detección de restricciones externas al cono \mathcal{K}_{MEJORA} , la cual será desarrollada en el capítulo 5. Por tanto, el mecanismo de supresión de restricciones superfluas por criterio angular de la tabla 3 será mantenido con fallas en casos excepcionales para ser mejorado más adelante.

3.6 Algoritmo simplex KKT

El algoritmo simplex KKT es el algoritmo simplex revisado como es expuesto por Chvatal (1983), al cual se le ha insertado: (a) el mecanismo de identificación de restricciones superfluas y (b) el procedimiento de optimalidad que calcula el vértice frontera $\mathbf{x}^\#$ en n

iteraciones. El algoritmo simplex KKT se presenta en la tabla 4. El problema de programación lineal de interés fue establecido en (1) y se reescribe enseguida:

$$\begin{aligned} & \text{maximizar } f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ & \text{subjeto a: } \mathbf{a}_j^T \mathbf{x} \leq b_j, j = 1, \dots, m \end{aligned}$$

Calcular la vecindad angular $\mathcal{N}(f)$ por medio del *mecanismo de identificación de restricciones superfluas*.

Ajustar el problema de programación lineal original a las restricciones de $\mathcal{N}(f)$:

$$\begin{aligned} f &= \mathbf{c}_N^T \mathbf{x}_N + \mathbf{c}_B^T \mathbf{x}_B \\ N\mathbf{x}_N + B\mathbf{x}_B &= \mathbf{b} \end{aligned}$$

Construir un conjunto \mathcal{G} de índices de las variables de decisión x_1, x_2, \dots, x_n .

Para cada iteración $t = 1, 2, \dots$ ejecutar los cinco pasos siguientes [Chvatal (1983)]:

Paso 1. Resolver el sistema: $\mathbf{y}^T B = \mathbf{c}_B^T$

Paso 2. Seleccionar una columna entrante [*procedimiento de optimalidad*]:

Si $\mathcal{G} \neq \emptyset$:

Paso 2.1: La columna entrante puede ser una columna \mathbf{N}_k de la matriz no básica N tal que $k \in \mathcal{G}$ y $\mathbf{y}^T \mathbf{N}_k$ es menor que el correspondiente componente de \mathbf{c}_N^T . Si existe tal columna suprimir el índice k del conjunto \mathcal{G} .

Paso 2.2: Si con el paso 2.1 no fue posible seleccionar una columna entrante, la columna entrante puede ser una columna \mathbf{N}_k de la matriz no básica N tal que $k \in \mathcal{G}$ y $\mathbf{y}^T \mathbf{N}_k$ es mayor que el correspondiente componente de \mathbf{c}_N^T . Si existe tal columna suprimir el índice k del conjunto \mathcal{G} .

Paso 2.3 Si no existe columna entrante, entonces la solución corriente es óptima.

En otro caso ($\mathcal{G} = \emptyset$):

la columna entrante puede ser una columna \mathbf{N}_k de la matriz no básica N tal que $\mathbf{y}^T \mathbf{N}_k$ es menor que el correspondiente componente de \mathbf{c}_N^T . Si no existe tal columna, entonces la solución corriente es óptima.

Paso 3. Resolver el sistema: $B\mathbf{d} = \mathbf{N}_k$.

Paso 4. Encontrar el mayor número ξ tal que $\mathbf{x}_B - \xi\mathbf{d} \geq 0$. Si no hay tal ξ , entonces el problema es no acotado. De otra forma, al menos un componente de $\mathbf{x}_B - \xi\mathbf{d}$ es igual a cero y la correspondiente variable x_{B_r} estará abandonando la base.

Paso 5. Hacer el valor de la variable entrante x_{N_k} igual a ξ y reemplazar el valor de x_B por $\mathbf{x}_B - \xi\mathbf{d}$. Reemplace la columna saliente de la matriz básica B por la columna entrante \mathbf{N}_k y reemplace la variable saliente x_{B_r} por la variable entrante x_{N_k} .

Tabla 19: Algoritmo simplex KKT

Los pasos 1,3, 4 y 5 son una transcripción del algoritmo simplex revisado presentado por Chvatal (1983). El paso 2 original ha sido transformado para que se puedan procesar variables libres y, por medio del vaciado del conjunto \mathcal{G} de índices de las variables de decisión, se pueda arribar al vértice frontera $\mathbf{x}^\#$ en n iteraciones. Después de que \mathcal{G} ha sido vaciado, el algoritmo simplex KKT funciona como un algoritmo simplex revisado procesando un problema de dimensión reducida.

3.7 Conclusiones

En este capítulo se ha desarrollado un algoritmo denotado como *algoritmo simplex KKT* para resolver la clase de problemas de programación lineal en el espacio R^n que modelan la sujeción de cuerpos rígidos por manos robóticas. Las suposiciones básicas en estos problemas son: (i) el único punto de inicio conocido es el origen de R^n en el interior del polítopo de soluciones factibles, (ii) el polítopo es no vacío y (iii) el polítopo es no degenerado. El algoritmo simplex KKT es el algoritmo simplex revisado como es expuesto por Chvatal (1983), al cual se le han insertado: (a) el procedimiento de optimalidad desarrollado en el capítulo 3 en el que se calcula el vértice frontera $\mathbf{x}^\#$ en n pasos, y (b) el mecanismo de identificación de restricciones superfluas basado en el criterio angular. Por tanto, el algoritmo simplex KKT inicia en un punto interior del polítopo de soluciones factibles, migra hacia un vértice en su frontera y continúa hasta llegar al vértice óptimo.

De acuerdo a experimentos realizados en R^2 , se prevé que el algoritmo simplex KKT reduzca los tiempos de cálculo de la decisión de sujeción por manos robóticas. El experimento realizado en este capítulo mostró una reducción de la dimensión del problema de programación lineal del orden de 50%.

Se encontró que el mecanismo de identificación de restricciones superfluas basado en el criterio angular es útil solo cuando el cono KKT es subconjunto del cono de mejora, en otro caso el mecanismo falla. La falla sucede porque al menos una de las restricciones atadas al vértice óptimo ha sido suprimida por el mecanismo. Las restricciones suprimidas podrán ser detectadas por medio del criterio de coordenadas que será desarrollado en el capítulo 5.

En este capítulo se desarrolló un concepto de polarización de restricciones el cual nos habilita a identificar restricciones superfluas. Se encontró que las entidades polarizadoras son el gradiente ∇f de la función objetivo y el cono KKT \mathcal{K}^* asociados a un problema de programación lineal. Las entidades sujetas a la polarización son las " m " restricciones del problema de programación lineal. El fenómeno de polarización de las restricciones sucede a través de cantidades derivadas de ellas que son sensibles a la polarización: (i) $\text{Cos}(\theta_j), j = 1, \dots, m$ y (ii) $\xi_j, j = 1, \dots, m$. Donde θ_j es el ángulo entre el gradiente ∇h_j de la j -ésima restricción y ∇f , y ξ_j es una coordenada KKT que será definida en el capítulo 5. La polarización de restricciones por medio de $\text{Cos}(\theta_j)$ dio como resultado el mecanismo de identificación de restricciones superfluas por criterio angular. La polarización por medio de la coordenada KKT ξ_j será explorada en el capítulo 5.

Se comprobó la hipótesis de la existencia de una clase de polarización de restricciones de un problema de programación lineal, la cual está fundamentada en las siguientes cantidades: (i) el gradiente ∇f de la función objetivo y el cono KKT \mathcal{K}^* como entidades polarizadoras y (ii) $\text{Cos}(\theta_j)$ como entidad sensible a la polarización. La polarización fuerza a las restricciones a agruparse en dos clases: (i) el conjunto de restricciones candidatas a delimitar el cono KKT y (ii) el conjunto de restricciones que contiene algunas restricciones superfluas.

Capítulo 5

Algoritmo Simplex KKT

En el capítulo 4 se comprobó la hipótesis de que es posible someter las restricciones de un problema de programación lineal a cierta clase de polarización que las fuerza a agruparse en dos clases: el conjunto de restricciones candidatas a delimitar el *cono Karush-Kuhn-Tucker (cono KKT)* denotado por \mathcal{N} y un conjunto de restricciones superfluas denotado por \mathcal{F}_{ang} . Se encontró que el fenómeno de polarización es causado por el gradiente de la función objetivo ∇f y el cono KKT. Las entidades sometidas al fenómeno de polarización son las restricciones pero a través de cantidades derivadas de ellas las cuales son sensibles a la polarización: (i) el ángulo θ_j del gradiente ∇h_j de la j -ésima restricción respecto de ∇f y (ii) la *coordenada Karush-Kuhn-Tucker (coordenadas KKT)* ξ_j de la j -ésima restricción. En el capítulo 4 se desarrolló un método de identificación de restricciones superfluas basado en el ángulo θ_j , el cual falla cuando no se satisface la condición de superfluidad.

En el presente capítulo se diseñará el método de identificación de restricciones superfluas basado en las coordenadas ξ_j . La idea clave del método es la identificación de restricciones superfluas remanes en la vecindad angular \mathcal{N} y la detección de restricciones en \mathcal{F}_{ang} cuyos gradientes son candidatos a delimitar el cono KKT. La identificación y detección se realizan sobre la trayectoria de vértices desde $x^\#$ hasta x^* mostrada en la figura 1.

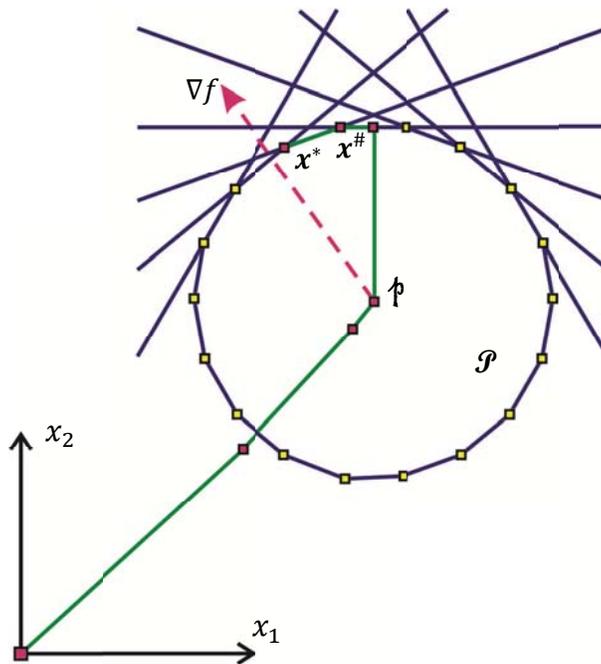


Fig. 1: Migración $p \rightarrow x^\# \rightarrow x^*$

La metodología propuesta en el presente capítulo consiste en insertar en el algoritmo simplex KKT diseñado en el capítulo 4 el mecanismo de identificación de restricciones superfluas basado en el criterio de coordenadas que garantice la condición de superfluidad. El algoritmo resultante suprimirá con seguridad las restricciones superfluas que sean detectadas en cada vértice de la trayectoria de vértices recorrida por el algoritmo, por lo que el algoritmo será mejorado en eficiencia computacional sustancialmente. El algoritmo simplex KKT mejorado reducirá los tiempos de cálculo en problemas de programación lineal que modelan la sujeción por manos robóticas. El algoritmo inicia en un punto interior p del polígono \mathcal{P} de soluciones factibles, migra hacia un vértice $x^\#$ en su frontera y continúa hasta llegar al vértice óptimo x^* , como se ilustra en la figura 1.

En la sección 1 se formula el problema de identificación y supresión de restricciones superfluas por criterio de coordenadas, mientras que en la sección 2 se generan las coordenadas y en la sección 3 se resuelve el problema de identificación. En la sección 4 se resuelve el problema de supresión. En la sección 5 se presenta el procedimiento usual de pivoteo y actualización. En la sección 6 se presenta el algoritmo simplex KKT el cual se verifica en la 7 por medio de un ejemplo numérico. En la sección 8 se presentan las conclusiones del capítulo.

5.1 Problema de identificación de restricciones superfluas remanentes

Para ejemplificar los conceptos acerca de las restricciones superfluas que serán presentados en este capítulo se considerará el ejemplo numérico en R^2 del capítulo 4. La figura 2 muestra el polígono de soluciones factibles y corresponde al siguiente ejemplo numérico:

$$\begin{aligned}
 & \underset{x}{\text{maximizar}} \quad f(x) = [-1 \quad -2]^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
 & \text{sujeto a:} \\
 & \begin{array}{rcllcl}
 0.8523x_1 & - & 0.5230x_2 & + & u_1 & = & 0.9685 & 1 \\
 0.9558x_1 & - & 0.2941x_2 & + & u_2 & = & 1.2254 & 2 \\
 0.9999x_1 & + & 0.0167x_2 & + & u_3 & = & 1.6664 & 3 \\
 0.8619x_1 & + & 0.5070x_2 & + & u_4 & = & 2.5351 & 4 \\
 0.1699x_1 & + & 0.9855x_2 & + & u_5 & = & 3.3981 & 5 \\
 -0.6814x_1 & + & 0.7319x_2 & + & u_6 & = & 2.5238 & 6 \\
 -0.9417x_1 & + & 0.3363x_2 & + & u_7 & = & 1.6817 & 7 \\
 -0.9999x_1 & + & 0.0122x_2 & + & u_8 & = & 1.2194 & 8 \\
 -0.9578x_1 & - & 0.2873x_2 & + & u_9 & = & 1.4367 & 9 \\
 -0.6783x_1 & - & 0.7348x_2 & + & u_{10} & = & 1.9783 & 10 \\
 -0.9937x_1 & - & 0.1118x_2 & + & u_{11} & = & 2.0703 & 11
 \end{array} \quad (1) \\
 & -\infty \leq x_1, x_2 \leq +\infty, \quad u_j \geq 0, j = 1, 2, \dots, 11
 \end{aligned}$$

En el problema (2) x_1, x_2 son las variables de optimización, mientras que $u_j \geq 0, j = 1, 2, \dots, 11$ son las variables de holgura asociadas a cada restricción. La columna de la derecha contiene el índice de cada restricción, mientras que en la figura 2 se señala cada restricción con estos índices. En la figura se muestran también algunos vectores gradientes de restricciones, el vértice frontera $x^\#$ y el vértice óptimo x^* .

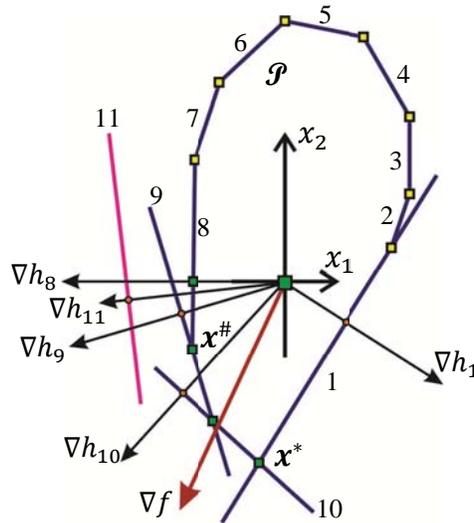


Fig. 2: Polígono \mathcal{P} de soluciones factibles en R^2

En el capítulo 4 se propuso el conjunto de restricciones superfluas \mathcal{F}_{ang} con las condiciones que debe satisfacer:

Condición de superfluidad: $\mathcal{N}_{ata}^* \not\subseteq \mathcal{F}_{ang}$, o bien: $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$

Condición de partición: $\mathcal{C} = \mathcal{N} \cup \mathcal{F}_{ang}$ y $\mathcal{N} \cap \mathcal{F}_{ang} = \emptyset$

Los conjuntos de la partición \mathcal{N} y \mathcal{F}_{ang} se calculan usando el criterio del ángulo θ_j :

$$\begin{aligned}\mathcal{N} &= \{j \in \mathcal{M} \mid \cos(\theta_j) \geq 0\} \\ \mathcal{F}_{ang} &= \{j \in \mathcal{M} \mid \cos(\theta_j) < 0\}\end{aligned}$$

Donde

$$\cos(\theta_j) = \frac{(\nabla f)^T \nabla h_j}{\|\nabla f\| \cdot \|\nabla h_j\|}, j \in \mathcal{M}$$

Para el ejemplo definido en (1) se obtuvo: $\mathcal{N} = \{10, 9, 11, 8, 7, 1\}$, $\mathcal{F}_{ang} = \{2, 6, 3, 4, 5\}$ y el cúmulo de restricciones $\mathcal{C} = \{10, 9, 11, 8, 7, 1, 2, 6, 3, 4, 5\}$.

5.1.1 Formulación del problema de identificación y supresión de restricciones superfluas

Suponiendo que se satisface la condición de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$, las restricciones atadas \mathcal{N}_{ata}^* se encuentran, en general, dispersas en \mathcal{N} y mezcladas con restricciones superfluas. Por tanto \mathcal{N} puede ser particionada como sigue.

$$\mathcal{N} = \mathcal{N}_{ata}^* \cup \mathcal{F}_{rem} \text{ y } \mathcal{N}_{ata}^* \cap \mathcal{F}_{rem} = \emptyset \quad (2)$$

Donde \mathcal{F}_{rem} es el conjunto de restricciones superfluas remanes en \mathcal{N} :

$$\mathcal{F}_{rem} = \{\mathcal{N} - \mathcal{N}_{ata}^*\} \quad (3)$$

El problema de identificación y supresión de restricciones superfluas remanes en \mathcal{N} puede ser formulado como sigue: Se propone identificar y suprimir algunas restricciones superfluas de \mathcal{F}_{rem} mientras el algoritmo simplex KKT evoluciona sobre la trayectoria de vértices: $\{\mathbf{x}^t \mid t = 1, \dots, T^*, \mathbf{x}^1 = \mathbf{x}^\#, \mathbf{x}^{T^*} = \mathbf{x}^*\}$ donde $\mathbf{x}^\#$ es el vértice frontera y \mathbf{x}^* es el vértice óptimo. La supresión de las restricciones identificadas como superfluas implica el cálculo de una secuencia de vecindades angulares \mathcal{N}^t con las propiedades siguientes:

- $\mathcal{N}_{ata}^* \subseteq \mathcal{N}^t$ para cada $t = 1, 2, \dots, T^*$
- $\mathcal{N}^t \rightarrow \mathcal{N}_{ata}^*$ mientras $\mathbf{x}^t \rightarrow \mathbf{x}^*$, $t = 1, 2, \dots, T^*$
- $\mathcal{N}^1 = \mathcal{N}$ asociada a $\mathbf{x}^1 = \mathbf{x}^\#$.

La identificación y supresión correctas de restricciones implica $\mathcal{N}^{t+1} \subseteq \mathcal{N}^t$ y $\mathcal{N}_{ata}^* \subseteq \mathcal{N}^t$ de tal manera que $\mathcal{N}^{T^*} \rightarrow \mathcal{N}_{ata}^*$. No se puede lograr que \mathcal{N}^t sea reducido exactamente a \mathcal{N}_{ata}^* porque solo se suprimen restricciones que han sido identificadas como superfluas durante el recorrido de la trayectoria de vértices.

5.1.2 Metodología para generación de coordenadas

Dada la vecindad angular \mathcal{N} y el vértice frontera $\mathbf{x}^\#$, el problema de identificación y supresión de restricciones superfluas es determinar la vecindad angular $\mathcal{N}^t \rightarrow \mathcal{N}_{ata}^*$ con propiedades dadas en (4). En la iteración "t", la vecindad angular \mathcal{N}^t está particionada como sigue:

$$\mathcal{N}^t = \mathcal{N}_{ata}^* \cup \mathcal{F}_{rem}^t \text{ y } \mathcal{N}_{ata}^* \cap \mathcal{F}_{rem}^t = \emptyset \quad (5)$$

Donde $\mathcal{F}_{rem}^t = \{\mathcal{N}^t - \mathcal{N}_{ata}^*\}$ para cada iteración "t". El conjunto de restricciones superfluas que pueden ser detectadas en \mathcal{F}_{rem}^t se define como sigue:

$$\mathcal{F}_{restr}^t \subseteq \mathcal{F}_{rem}^t \quad (6)$$

El desafío es por tanto calcular \mathcal{F}_{restr}^t .

Por otro lado, el procedimiento de factibilidad del algoritmo simplex, en cualquiera de sus versiones, calcula una coordenada ξ_j para cada $j \in \mathcal{N}^t$. Este procedimiento elige una restricción $r \in \mathcal{N}^t$ cuya coordenada ξ_r sea positiva y la más restrictiva; después de esta elección, el procedimiento selecciona la variable básica x_{B_r} para ser convertida en variable no básica. Después de haber detectado ξ_r , el resto de las coordenadas no se vuelve a usar por el algoritmo simplex. Sin embargo, en este capítulo toda coordenada será procesada porque se prevé que las coordenadas $\xi_j, j \in \mathcal{N}^t$ en cada iteración "t" poseen información de las restricciones superfluas $j \in \mathcal{F}_{rem}^t$.

Por tanto, para precisar los conceptos acerca de la identificación de restricciones superfluas $j \in \mathcal{F}_{rem}^t$ por medio del procesamiento de las coordenadas $\xi_j, j \in \mathcal{N}^t$ se propone la metodología siguiente:

1. Definir los conceptos de coordenadas y eje de coordenadas.
2. Precisar el criterio de identificación de coordenadas superfluas
3. Generar las coordenadas por medio del procedimiento de factibilidad del algoritmo simplex.

5.2 Coordenadas KKT para identificación de restricciones superfluas

En esta sección se generan las coordenadas $\xi_j, j \in \mathcal{N}^t$ para cada iteración "t" con la finalidad de explorar la hipótesis de que estas poseen información acerca de las restricciones superfluas $j \in \mathcal{F}_{rem}^t$.

Dada la vecindad angular $\mathcal{N} = \mathcal{N}^\#$ y el vértice frontera $\mathbf{x}^\#$, el problema principal consiste en desarrollar un mecanismo de identificación de restricciones superfluas por medio de un criterio de coordenadas. En esta sección comenzaremos con el concepto de *eje real de coordenadas KKT* para motivar la generación de *coordenadas KKT* y continuaremos en la siguiente sección con el desarrollo del mecanismo.

5.2.1 Concepto de eje real de coordenadas KKT

Consideremos la trayectoria de vértices: $\{\mathbf{x}^t | t = 1, \dots, T^*, \mathbf{x}^1 = \mathbf{x}^\#, \mathbf{x}^{T^*} = \mathbf{x}^*\}$ donde $\mathbf{x}^\#$ es el vértice frontera y \mathbf{x}^* es el vértice óptimo. A través de los vértices consecutivos \mathbf{x}^t y \mathbf{x}^{t+1} se puede hacer pasar una línea como se ilustra en la figura 4. Esta línea puede convertirse en un eje de números reales, al cual denotaremos por ξ , cuando a la línea se le agrega un origen y se define la dirección en que van los positivos y negativos. Si se desea que este eje aloje las coordenadas que necesitamos, su origen debe elegirse en \mathbf{x}^t , los positivos en la dirección de \mathbf{x}^{t+1} y los negativos en dirección contraria. La figura 11 muestra un polítopo en el espacio R^3 el cual exhibe caras, vértices y el eje ξ de *coordenadas KKT*. Cada cara es un pentágono regular determinado por la intersección de un hiperplano con el polítopo.

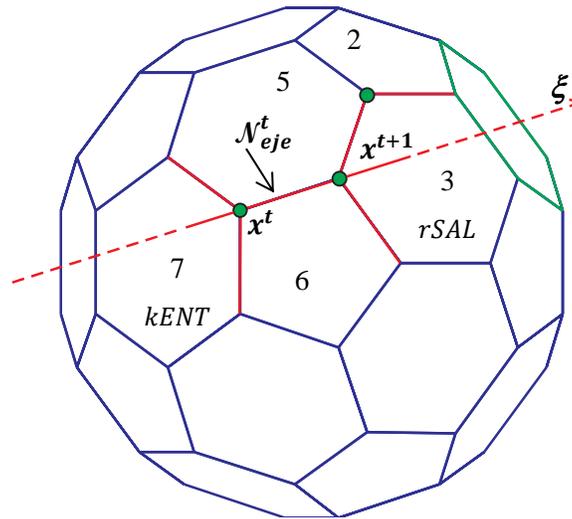


Fig. 3: Especificación de eje ξ de *coordenadas KKT*

El vértice \mathbf{x}^t es la intersección de los hiperplanos 7, 6 y 5, por lo que el conjunto de restricciones atadas a \mathbf{x}^t es $\mathcal{N}_{ata}^t = \{7, 6, 5\}$. Si se ejecuta el procedimiento de optimalidad,

se tendrá el índice $k = 7$ de la variable no básica x_{Nk} que es candidata para entrar a la base, e incrementará por tanto el valor de la función objetivo. Porque $k = 7$ es índice de la variable entrante debe ser suprimido de \mathcal{N}_{ata}^t . Después de la supresión, se tendrá un conjunto de restricciones que determina el *eje ξ de coordenadas KKT*, el cual se define como sigue:

$$\mathcal{N}_{eje}^t = \mathcal{N}_{ata}^t - \{k\} \quad (7)$$

Para el polítopo de la figura 11 se tiene $\mathcal{N}_{eje}^t = \{6, 5\}$. Es decir, el *eje ξ de coordenadas KKT* es la intersección de los hiperplanos de las restricciones 6 y 5.

Si ahora se ejecuta el procedimiento de factibilidad se obtendrá el índice $r = 3$ de la variable básica x_{Br} que saldrá de la base. Porque $r = 3$ es índice de la variable básica saliente, debe ser agregada al conjunto \mathcal{N}_{eje}^t , obteniéndose el conjunto de restricciones atadas al vértice \mathbf{x}^{t+1} , el cual se determina como sigue:

$$\mathcal{N}_{ata}^{t+1} = \mathcal{N}_{eje}^t \cup \{r\} \quad (8)$$

Para el polítopo de la figura 11 se tiene $\mathcal{N}_{ata}^{t+1} = \{3, 6, 5\}$. Es decir, en una iteración, el algoritmo simplex pasó del vértice \mathbf{x}^t al vértice \mathbf{x}^{t+1} a lo largo del eje de coordenadas.

La intersección del *eje ξ de coordenadas* con cada hiperplano del problema de programación lineal es una coordenada KKT. Estas intersecciones pueden ser calculadas con costo computacional adicional. Como se verá más adelante, el procedimiento de factibilidad puede ser adaptado para el cálculo explícito de las coordenadas KKT sin costo adicional para el algoritmo simplex.

El *eje de coordenadas* se aprecia claramente en la figura 11, sin embargo las *coordenadas KKT* son difíciles de visualizar geoméricamente aún en el espacio R^3 . En la siguiente sección se desarrollará el concepto de coordenadas KKT por medios geométricos en el espacio R^2 por facilidad de análisis.

5.2.2 Concepto de coordenadas KKT

En esta sección se desarrollará el concepto de coordenadas KKT por extensión del concepto de eje de coordenadas presentado anteriormente. Consideremos el ejemplo (2) cuyo polítopo \mathcal{P} , un *eje ξ de coordenadas KKT* y un eje ζ en la dirección del gradiente ∇f de la función objetivo se muestran en la figura 12. En la figura se usa \mathbf{x}^t como vértice de referencia cuya proyección ortogonal sobre ∇f es el punto \mathbf{q}^t . Una línea de referencia Q se hace pasar por los puntos \mathbf{x}^t y \mathbf{q}^t la cual es perpendicular a ∇f , como se muestra en la figura.

El cúmulo de restricciones para el ejemplo es: $\mathcal{C} = \{10, 9, 11, 8, 7, 1, 2, 6, 3, 4, 5\}$. Después de suprimir un conjunto propuesto de restricciones superfluas $\mathcal{F}_{ang} = \{3, 4, 5\}$ se obtiene la vecindad angular $\mathcal{N} = \{10, 9, 11, 8, 7, 1, 2, 6\}$. Las restricciones de \mathcal{N} aparecen enumeradas en la figura 12 y sus correspondientes *coordenadas KKT* están listadas en la tabla 3.

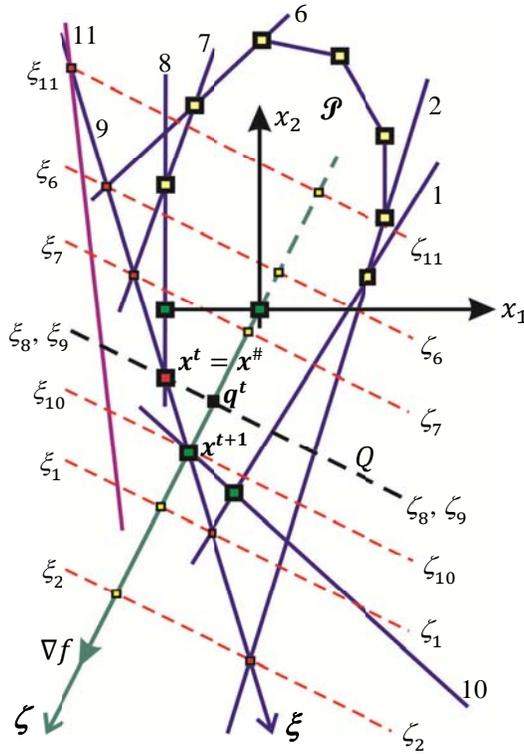


Fig. 4: Geometría de las coordenadas para $\mathcal{N}=\{10\ 9\ 11\ 8\ 7\ 1\ 2\ 6\}$

| Coordenadas KKT en el vértice x^t | |
|-------------------------------------|------------------------|
| Sobre el eje ξ | Sobre el eje ζ |
| $\xi_{11} = -1.2517$ | $\zeta_{11} = -6.8162$ |
| $\xi_6 = -0.7811$ | $\zeta_6 = -4.2538$ |
| $\xi_7 = -0.4162$ | $\zeta_7 = -2.2665$ |
| $\xi_8 = 0$ | $\zeta_8 = 0$ |
| $\xi_9 = 0$ | $\zeta_9 = 0$ |
| $\xi_{10} = 0.2841$ | $\zeta_{10} = 1.5471$ |
| $\xi_1 = 0.6203$ | $\zeta_1 = 3.3781$ |
| $\xi_2 = 1.1487$ | $\zeta_2 = 6.2554$ |

Tabla 1: Valores de *Coordenadas KKT* sobre los ejes ξ y ζ

Las *coordenadas KKT* se localizan sobre el eje ξ cuyo origen está en x^t . Se tiene una *coordenada KKT* por cada restricción del problema de programación lineal dado: $\xi_j, j = 1, \dots, |\mathcal{N}|$. Cada coordenada ξ_j es generada por el algoritmo simplex en cualquiera de sus versiones y es proporcional a la distancia desde el origen del eje ξ hasta la intersección del eje con el hiperplano de la j -ésima restricción. Como se puede observar en la figura 12 y en la tabla 3 se ha incluido un eje ζ (con origen posicionado en q^t) equivalente al eje ξ con el fin de confirmar el concepto de coordenadas KKT.

Cada coordenada ζ_j sobre el eje ζ se determinó de la forma siguiente: (i) se calculó el punto de cruce del eje ξ con cada hiperplano 11, 6, 7, 8, 9, 10, 1, 2, (ii) se calculó la proyección ortogonal del j -ésimo punto de cruce sobre el gradiente ∇f de la función objetivo, (iii) se calculó la coordenada ζ_j como la distancia desde el origen del eje ζ hasta el punto determinado por la $-$ ésima proyección ortogonal. Las coordenadas ζ_j se muestran en la figura 12 y se listan en la columna 2 de la tabla 3. Se observa que se mantiene la relación $\zeta_j/\xi_j = 5.4456$ para todo j por lo que ξ_j y ζ_j son equivalentes aun cuando fueron calculadas en formas distintas. Las coordenadas positivas ξ_j y ζ_j hacen crecer el valor de la función objetivo mientras que las negativas la hacen decrecer.

Reflexionemos en seguida acerca de las coordenadas sobre el *eje ξ de coordenadas KKT* de la figura 12. El vértice \mathbf{x}^t es la intersección de los hiperplanos 8 y 9, por lo que $\mathcal{N}_{ata}^t = \{8, 9\}$. El procedimiento de optimalidad determina el índice $k = 8$ de la variable no básica x_{Nk} que es candidata para entrar a la base, por lo que éste debe ser suprimido de \mathcal{N}_{ata}^t . Después de la supresión se tendrá $\mathcal{N}_{aje}^t = \{9\}$. Es decir, el hiperplano de la restricción 9 es el *eje ξ de coordenadas KKT*, el cual pasa por \mathbf{x}^t y por \mathbf{x}^{t+1} . Como se muestra en la figura 12, el origen del eje es el vértice \mathbf{x}^t , sus puntos positivos están en la dirección de \mathbf{x}^{t+1} y sus puntos negativos en dirección contraria. El procedimiento de factibilidad del algoritmo simplex tiene la función de seleccionar una variable básica x_{Br} para ser convertida en variable no básica. El procedimiento determina el índice r tal que $x_{Br} = 0$. Este índice es el de la coordenada ξ_r , la cual es estrictamente positiva y la más restrictiva. Analizando la columna 1 de la tabla 3, se tiene que $\xi_{10} = 0.2841$ es la estrictamente positiva más restrictiva, por lo que el índice buscado es $r = 10$ para el ejemplo de la figura 12.

El algoritmo simplex calcula las coordenadas $\xi_j, j = 1, \dots, |\mathcal{N}|$, como las de la tabla 3. De estas coordenadas el algoritmo determina el índice r de la coordenada ξ_r estrictamente positiva más restrictiva seleccionada de entre las coordenadas positivas. El resto de las coordenadas, sin embargo, son desechadas por el algoritmo. En este punto se pueden plantear las siguientes preguntas clave: ¿qué significado tienen las coordenadas positivas que de entre ellas se selecciona la estrictamente positiva más restrictiva?, y si las coordenadas positivas tienen un significado, ¿qué significado tienen las negativas?

5.2.3 Criterio de coordenadas para identificación de restricciones superfluas:

Después de haber suprimido del conjunto total de restricciones $\mathcal{M} = \mathcal{N} \cup \mathcal{F}_{ang}$ el conjunto de restricciones superfluas \mathcal{F}_{ang} , la vecindad angular \mathcal{N} permanece. Se supone que \mathcal{N} satisface la propiedad de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$ y que en \mathcal{N} permanecen todavía restricciones superfluas. El problema que se desea explorar es identificar las restricciones superfluas remanentes en \mathcal{N} . Es decir, se propone probar la hipótesis planteada anteriormente de que las coordenadas $\xi_j, j \in \mathcal{N}^t$ en cada iteración " t " poseen información de las restricciones superfluas $j \in \mathcal{F}_{rem}^t = \{\mathcal{N}^t - \mathcal{N}_{ata}^*\}$.

Consideremos el ejemplo en R^2 cuyo polígono \mathcal{P} se muestra en la figura 12. Sea $\mathcal{N}^t = \{10, 9, 11, 8, 7, 1, 2, 6\}$ la vecindad angular calculada en x^t , el cual corresponde al vértice frontera $x^\#$. Las coordenadas KKT $\xi_j, j \in \mathcal{N}^t$ se muestran en la tabla 3 y se grafican en la figura 13.

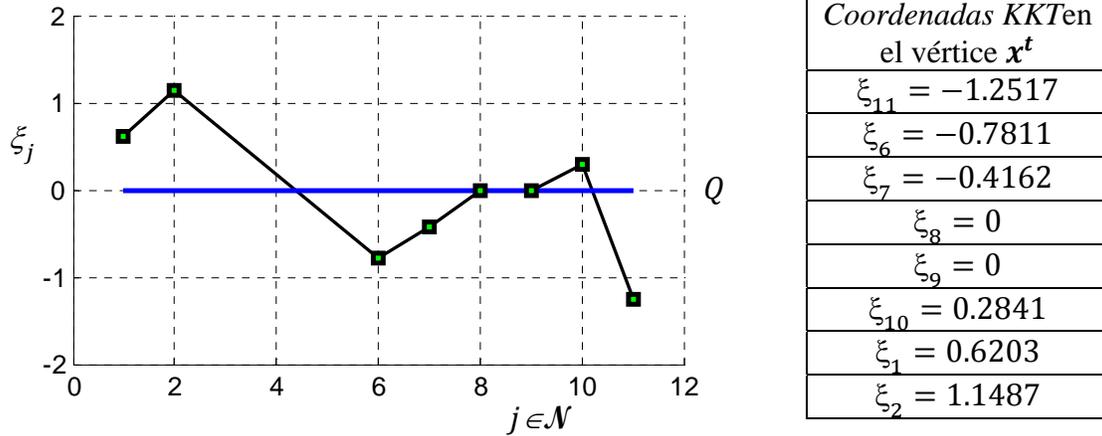


Fig. 5: Coordenadas KKT $\xi_j, j \in \mathcal{N}^t$

La línea Q mostrada en la figura 12 fue usada en el capítulo 4 como referencia para medidas angulares $\text{Cos}(\theta_j), j \in \mathcal{N}^t$. En las figuras 12 y 13 sin embargo, la línea Q se usa como referencia para las coordenadas $\xi_j, j \in \mathcal{N}^t$. Notamos por tanto que las coordenadas ξ_j están polarizadas respecto de la línea Q , y que: $\xi_j > 0$ hace crecer el valor de la función objetivo f en la dirección del vector ∇f , $\xi_j < 0$ hace decrecer el valor de la función objetivo f , y $\xi_j = 0$ deja la función objetivo f sin cambio para $j \in \mathcal{N}^t$.

La polarización expuesta en la figura 13 fuerza a las restricciones a agruparse en dos conjuntos por medio los signos de la coordenada KKT ξ_j :

$$\begin{aligned} \mathcal{N}_+^t &= \{j \in \mathcal{N}^t \mid \xi_j \geq 0\} \\ \mathcal{N}_-^t &= \{j \in \mathcal{N}^t \mid \xi_j < 0\} \end{aligned} \tag{9}$$

Para la gráfica de la figura 13 se tiene $\mathcal{N}_+^t = \{1, 2, 8, 9, 10\}$ y $\mathcal{N}_-^t = \{6, 7, 11\}$. Dado que $\mathcal{N}_{ata}^* = \{1, 10\}$ y que $\xi_8 = 0$ y $\xi_9 = 0$, solo $\xi_1 = 0.6203$, $\xi_2 = 1.1487$ y $\xi_{10} = 0.2841$ son candidatas a delimitar en cono KKT. Por tanto, la coordenada positiva más restrictiva es $\xi_{10} = 0.2841$, cuyo índice $r = 10 \in \mathcal{N}_+^t$.

Del análisis previo se puede decir que los signos de las coordenadas KKT tienen significados acerca de la pertenencia de las restricciones al cono $\text{KKT}\mathcal{K}^*$. Las expresiones dadas en (9) constituyen propiamente el criterio de coordenadas, el cual se expone verbalmente en la tabla 4 que sigue.

Consideremos la trayectoria de vértices: $\{\mathbf{x}^t | t = 1, \dots, T^*, \mathbf{x}^1 = \mathbf{x}^\#, \mathbf{x}^{T^*} = \mathbf{x}^*\}$
donde $\mathbf{x}^\#$ es el vértice frontera y \mathbf{x}^* es el vértice óptimo.

En el vértice corriente \mathbf{x}^t con su correspondiente vecindad angular \mathcal{N}^t , el
gradiente ∇h_j de la restricción $j \in \mathcal{N}^t$ es candidato a delimitar el cono KKT \mathcal{K}^*
siempre que su correspondiente *coordenada KKT* $\xi_j \geq 0, j \in \mathcal{N}_+^t$.

Además, dado que las restricciones candidatas a delimitar \mathcal{K}^* están en
correspondencia con las coordenadas positivas, las coordenadas negativas $\xi_j < 0,$
 $j \in \mathcal{N}_-^t$ deben estar en correspondencia con las restricciones superfluas.

Tabla 2: Criterio de coordenadas para identificación de restricciones superfluas

De acuerdo a este criterio, en cada vértice \mathbf{x}^t es posible identificar las restricciones
superfluas por medio de las *coordenadas KKT* estrictamente negativas. El criterio asegura
que existe cierta correspondencia entre las coordenadas negativas $\xi_j < 0$ con $j \in \mathcal{N}_-^t$ y las
restricciones superfluas que puedan encontrarse en la vecindad angular \mathcal{N}^t . Como se
mostrará más adelante, la fórmula para ξ_j involucra variables básicas, por lo que
renombraremos al conjunto de restricciones \mathcal{N}_-^t como \mathcal{F}_{var}^t :

$$\mathcal{F}_{var}^t = \mathcal{N}_-^t = \{j \in \mathcal{N}^t | \xi_j < 0\} \quad (10)$$

Se mostrará oportunamente que \mathcal{F}_{var}^t es subconjunto de \mathcal{F}_{rem}^t dado en (6). En las siguientes
secciones se derivará la fórmula para calcular las coordenadas $\xi_j, j \in \mathcal{N}^t$. Esta derivación
es parte del procedimiento de factibilidad del algoritmo simplex, el cual se revisa
enseguida.

5.2.4 Impacto de una variable entrante sobre el diccionario

Consideremos la función objetivo del diccionario (6) del capítulo 3:

$$f = \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{x}_N \quad (11)$$

En esta sección se desarrollará un procedimiento para incrementar el valor de la función
objetivo "f" por una modificación apropiada del valor de una variable no básica " x_{Nk} "

el problema (1) expresado en igualdades, lo cual se logra por la introducción del vector \mathbf{u}
de variables de holgura como se muestra enseguida:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximizar}} f(\mathbf{x}) &= \mathbf{c}_N^T \mathbf{x}_N + \mathbf{c}_B^T \mathbf{x}_B \\ \text{subjeto a: } N\mathbf{x}_N + B\mathbf{x}_B &= \mathbf{b} \end{aligned} \quad (11)$$

Donde: $\mathbf{c}_N = \mathbf{c}, \mathbf{x}_N = \mathbf{x}, \mathbf{c}_B = \mathbf{0}, \mathbf{x}_B = \mathbf{u}, N = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m]^T, B = I, m \times m$; con
acotamientos $L_N < \mathbf{x}_N < U_N$ y $L_B \leq \mathbf{x}_B < U_B$, donde $L_N = -\infty, L_B = 0$ y $U_N = U_B =$

$+\infty$ son valores iniciales. Las ecuaciones para la función objetivo f y para las variables básicas \mathbf{x}_B en función de \mathbf{x}_N se conocen como diccionario y se determinan de (11):

$$\begin{aligned} f &= \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{x}_N \\ \mathbf{x}_B &= B^{-1} \mathbf{b} - B^{-1} N \mathbf{x}_N \end{aligned} \quad (12)$$

Las variables y parámetros en (12) evolucionan en cada iteración del algoritmo simplex. Para el cálculo de las coordenadas nos interesa la evolución de \mathbf{x}_N y de \mathbf{x}_B . En esta sección se establecen las condiciones de factibilidad de las variables básicas $x_{Bj} \in \mathbf{x}_B$ y de la variable no básica x_{Nk} necesarias para el cálculo de las *coordenadas KKT* $\xi_j, j = 1, \dots, |\mathcal{N}^t|$.

Con el fin de mejorar el valor de la función objetivo, en cada iteración del algoritmo simplex (caracterizada por el vértice corriente \mathbf{x}^t y por la vecindad angular \mathcal{N}^t), una sola variable no básica x_{Nk} es cambiada a variable básica por medio de un número $\xi > 0$:

$$x_{Nk}(\xi) = x_{Nk} \pm \xi \quad (13)$$

El símbolo ξ representa un número ó representa el eje de coordenadas KKT dependiendo del contexto en donde se esté usando. El doble signo en el lado derecho de (13) tiene la función de considerar las variables libres del problema de programación lineal de la clase de problemas dados en (1). Es decir, la variable $x_{Nk}(\xi)$ es libre de adoptar valores positivos o negativos. Por definición $x_{Nk} = 0$ en el lado derecho de (13).

Se mostrará que la propiedad clave del número ξ es hacer que $x_{Nk}(\xi) = x_{Nk} \pm \xi$ se convierta en variable básica al mismo tiempo que una variable básica $x_{Br}(\xi) = x_{Br} \mp \xi d_{kr}$ se anule convirtiéndose por tanto en variable no básica. De la anulación $x_{Br} \mp \xi d_{kr} = 0$ se determinará el número $\xi > 0$.

Mientras el índice k de la variable no básica x_{Nk} es determinado por el procedimiento de optimalidad, el número ξ es calculado por el procedimiento de factibilidad. El número $\xi > 0$ debe satisfacer ciertas condiciones que garanticen la factibilidad de las variables básicas $x_{Bj}, j = 1, \dots, |\mathcal{N}^t|$, las cuales se analizan enseguida.

Se puede demostrar que el cambio escalar $x_{Nk} \pm \xi$ es equivalente al cambio vectorial $\mathbf{x}_N \pm \xi \mathbf{e}_k$, donde \mathbf{e}_k es el vector canónico con un 1 en su k -ésima coordenada y con ceros en las restantes. Sustituyendo \mathbf{x}_N por el cambio $\mathbf{x}_N \pm \xi \mathbf{e}_k$ en \mathbf{x}_B de (12) se tiene:

$$\begin{aligned} \mathbf{x}_B &= B^{-1} \mathbf{b} - B^{-1} N \mathbf{x}_N \\ &= B^{-1} \mathbf{b} - B^{-1} N [\mathbf{x}_N \pm \xi \mathbf{e}_k] \end{aligned}$$

Desarrollando se tiene:

$$\begin{aligned} \mathbf{x}_B(\xi) &= B^{-1} \mathbf{b} - B^{-1} N \mathbf{x}_N \mp \xi B^{-1} N \mathbf{e}_k \\ \mathbf{x}_B(\xi) &= \mathbf{x}_B \mp \xi B^{-1} N \mathbf{e}_k \end{aligned}$$

Notemos que \mathbf{x}_B del lado derecho es el vector de variables básicas antes del cambio $x_{Nk} \rightarrow x_{Nk} \pm \xi$, y que $\mathbf{N}_k = N\mathbf{e}_k$ es la k -ésima columna de la matriz no básica N . Por razones de simplicidad defínase:

$$\mathbf{d} = B^{-1}\mathbf{N}_k$$

para tener el siguiente efecto resultante del cambio $x_{Nk} \pm \xi$ sobre las variables básicas:

$$\mathbf{x}_B(\xi) = \mathbf{x}_B \mp \xi \mathbf{d} \quad (14)$$

La función clave del procedimiento de factibilidad del algoritmo simplex es encontrar el número $\xi > 0$ que satisfaga las siguientes condiciones de factibilidad:

$$L_{Nk} \leq x_{Nk} \pm \xi \leq U_{Nk} \& L_B \leq \mathbf{x}_B \mp \xi \mathbf{d} \leq U_B \quad (15)$$

Aunque el procedimiento de factibilidad tiene la función precisa de calcular el número ξ , en el intento de calcular éste número por medio de las desigualdades (15), se generarán las coordenadas KKT buscadas.

El cambio $x_N \pm \xi \mathbf{e}_k$ modifica también la ecuación de la función objetivo dada en (12), la cual se procesa como sigue:

$$\begin{aligned} f &= \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{x}_N \\ f(\xi) &= \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) [\mathbf{x}_N \pm \xi \mathbf{e}_k] \\ &= \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{x}_N \pm (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{e}_k \xi \\ &= \mathbf{c}_B^T B^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N) \mathbf{x}_N \pm (\mathbf{c}_N^T \mathbf{e}_k - \mathbf{c}_B^T B^{-1} N \mathbf{e}_k) \xi \\ &= f \pm (c_{Nk} - \mathbf{c}_B^T B^{-1} \mathbf{N}_k) \xi \end{aligned} \quad (16)$$

El crecimiento o decrecimiento de la variable no básica $x_{Nk}(\xi) = x_{Nk} \pm \xi$ se decide en el procedimiento de optimalidad, mientras que el cálculo del número $\xi > 0$ se realiza en el procedimiento de factibilidad. Los signos positivos y negativos en el cambio $x_{Nk} \rightarrow x_{Nk} \pm \xi$ serán considerados por separado en la generación de coordenadas. Es decir, las coordenadas se calcularán por medio de (15) solo para el crecimiento $x_{Nk} + \xi$ ó solo para el decrecimiento $x_{Nk} - \xi$, como se presentará en las dos secciones siguientes.

5.2.5 Cálculo de las coordenadas KKT por crecimiento de la variable no básica

Consideremos las condiciones sobre la variable no básica $x_{Nk}(\xi)$ dada en (13) y sobre el vector $\mathbf{x}_B(\xi)$ de variables básicas dado en (14) solo para el caso en que hay crecimiento de la variable no básica: $x_{Nk} + \xi$. Se analizará la desigualdad dada en (15), en primer lugar para la variable no básica $x_{Nk}(\xi) = x_{Nk} + \xi$, y en segundo lugar para las variables básicas $\mathbf{x}_B(\xi) = \mathbf{x}_B - \xi \mathbf{d}_k$.

Coordenada ξ que satisface las siguientes condiciones de la variable no básica:

$$L_{Nk} \leq x_{Nk} + \xi \leq U_{Nk}, k = 1, \dots, n \quad (17)$$

Dado que $x_{Nk} = 0, U_{Nk} = +\infty$, y

$$L_{Nk} = \begin{cases} 0 & \text{si } x_{Nk} \text{ es variable de holgura} \\ -\infty & \text{si } x_{Nk} \text{ es variable de decisión} \end{cases}$$

se tienen las condiciones sobre ξ correspondientes a $x_{Nk}(\xi) = x_{Nk} + \xi$, las cuales se resumen en la tabla 5:

$$\begin{aligned} \xi \in [0, +\infty) & \quad \text{si } x_{Nk} \text{ es variable de holgura } (L_{Nk} = 0) \\ \xi \in (-\infty, +\infty) & \quad \text{si } x_{Nk} \text{ es variable de decisión } (L_{Nk} = -\infty) \end{aligned}$$

| Condición | Intervalo |
|---|------------------------------|
| x_{Nk} es variable de holgura ($L_{Nk} = 0$) | $\xi \in [0, +\infty)$ |
| x_{Nk} es variable de decisión ($L_{Nk} = -\infty$) | $\xi \in (-\infty, +\infty)$ |

Tabla 3: Valores permisibles de ξ cuando $x_{Nk}(\xi)$ crece

Coordenada ξ que satisface las siguientes condiciones de la variable básica:

$$L_B \leq x_B - \xi d_k \leq U_B \quad (18)$$

Se desea determinar $\xi > 0$ que satisfaga estas desigualdades vectoriales. Expresando cada componente de los vectores se obtiene un valor ξ_j por cada componente de los cuales se podrá elegir el número ξ que satisfaga (18):

$$L_{Bj} \leq x_{Bj} - \xi_j d_{kj} \leq U_{Bj}, j = 1, \dots, m$$

$$\Rightarrow L_{Bj} - x_{Bj} \leq -\xi_j d_{kj} \leq U_{Bj} - x_{Bj}, j = 1, \dots, m$$

$$\Rightarrow x_{Bj} - L_{Bj} \geq \xi_j d_{kj} \geq x_{Bj} - U_{Bj}, j = 1, \dots, m \quad (19)$$

Dado que

$$U_{Bj} = \infty$$

$$L_{Bj} = 0 \text{ si } x_{Bj} \text{ es variable de holgura}$$

$$L_{Bj} = -\infty \text{ si } x_{Bj} \text{ es variable de decisión}$$

de las dos desigualdades en (19) se tienen para todo j las tres desigualdades siguientes:

$$\xi d_{kj} \leq (x_{Bj} - L_{Bj}) \Big|_{L_{Bj}=0}, \quad \xi d_{kj} \leq (x_{Bj} - L_{Bj}) \Big|_{L_{Bj}=-\infty} \quad \& \quad \xi d_{kj} \geq x_{Bj} - U_{Bj} \Big|_{U_{Bj}=\infty} \quad (20)$$

Considerando las tres clases de valores para d_{kj} : $d_{kj} = 0, d_{kj} > 0$, y $d_{kj} < 0$, las tres desigualdades dadas en (20) quedan como sigue:

$$\begin{aligned}
d_{kj} = 0: & \quad \xi_j \leq +\infty|_{L_{Bj}=0}, \xi_j \leq +\infty|_{L_{Bj}=-\infty} \& \xi_j \geq -\infty \Rightarrow \xi_j \in (-\infty, +\infty) \\
d_{kj} > 0: & \quad \xi_j \leq \frac{x_{Bj}}{d_{kj}} \Big|_{L_{Bj}=0}, \xi_j \leq +\infty|_{L_{Bj}=-\infty} \& \xi_j \geq -\infty \Rightarrow \xi_j \in (-\infty, \frac{x_{Bj}}{d_{kj}}] \\
d_{kj} < 0: & \quad \xi_j \geq \frac{-x_{Bj}}{|d_{kj}|} \Big|_{L_{Bj}=0}, \xi_j \geq -\infty|_{L_{Bj}=-\infty} \& \xi_j \leq +\infty \Rightarrow \xi_j \in [\frac{-x_{Bj}}{|d_{kj}|}, \infty)
\end{aligned} \tag{21}$$

Notamos de (21) que la cantidad d_{kj} selecciona un solo renglón. Notamos además que el hecho de que x_{Bj} sea una variable de holgura ó variable de decisión influye también sobre ξ_j . Sabremos que x_{Bj} es una variable de decisión cuando $L_{Bj} = -\infty$, y que es una variable de holgura cuando $L_{Bj} = 0$.

Definiremos por tanto las *coordenadas KKT* como sigue:

- Cuando se detecta que el límite inferior de x_{Bj} es $L_{Bj} = 0$, la coordenada ξ_j se elige como la cota finita de x_{Bj} siempre que $d_{kj} \neq 0$. Si $d_{kj} = 0$, ξ_j es libre.
- Cuando se detecta que el límite inferior de x_{Bj} es $L_{Bj} = -\infty$, se asigna $\xi_j = 0$. Esta asignación se explica como sigue: dado que las coordenadas cartesianas de un vértice \mathbf{x}^t son variables de decisión, y dado que el origen del eje ξ de coordenadas KKT se elige en éste vértice, las coordenadas KKT en este vértice son nulas.
- Notar que ξ_j puede también ser nula cuando $x_{Bj} = 0$.

La tabla 6 registra las fórmulas para las coordenadas ξ_j y sus intervalos de valores permisibles que satisfacen las condiciones (21) correspondientes al crecimiento de la variable no básica $x_{Nk} + \xi$.

| Condición | Intervalo permisible | Coordenadas $\xi_j, j = 1, 2, \dots, \mathcal{N}^t $ | |
|--------------|--|---|---|
| | | x_{Bj} es variable de holgura ($L_{Bj} = 0$) | x_{Bj} es variable de decisión ($L_{Bj} = -\infty$) |
| $d_{kj} = 0$ | $\xi_j \in (-\infty, +\infty)$ | ξ_j es libre | ξ_j es libre |
| $d_{kj} > 0$ | $\xi_j \in (-\infty, \frac{x_{Bj}}{d_{kj}}]$ | $\xi_j = \frac{x_{Bj}}{d_{kj}} \Big _{L_{Bj}=0}$ | $\xi_j = 0 _{L_{Bj}=-\infty}$ |
| $d_{kj} < 0$ | $\xi_j \in [\frac{-x_{Bj}}{ d_{kj} }, \infty)$ | $\xi_j = \frac{-x_{Bj}}{ d_{kj} } \Big _{L_{Bj}=0}$ | $\xi_j = 0 _{L_{Bj}=-\infty}$ |

Tabla 4: Fórmulas de las coordenadas $\xi_j, j = 1, \dots, |\mathcal{N}^t|$ para el caso $x_{Nk} + \xi$

Dado que d_{kj} puede adoptar un solo valor de los tres posibles: $d_{kj} = 0, d_{kj} > 0, d_{kj} < 0$, sucederá en cada j una sola coordenada de las tres que aparecen en los renglones 2, 3 y 4 de la tabla 6. Las coordenadas están registradas en las columnas 3 y 4.

Cuando $L_{Bj} = 0$, la variable básica x_{Bj} es variable de holgura por lo que $x_{Bj} \geq 0$. Por tanto, a partir de la tabla 6 se tendrá el siguiente resultado:

$$\begin{aligned} \text{Si } d_{kj} > 0 &\Rightarrow \xi_j \geq 0 \\ \text{Si } d_{kj} < 0 &\Rightarrow \xi_j \leq 0 \end{aligned} \quad (22)$$

Después de calcular el conjunto de coordenadas $\{\xi_j, j = 1 \dots |\mathcal{N}^t|\}$ a partir de la tabla 6 se determina la coordenada positiva más restrictiva en la forma siguiente:

$$\xi_r = \min\{\xi_j | j \in \mathcal{N}_{cand}^t\}, \quad \mathcal{N}_{cand}^t = \{j | \xi_j > 0\} \quad (23)$$

5.2.6 Fórmulas de las coordenadas KKT para decrecimiento de la variable no básica

Consideremos las condiciones sobre la variable no básica $x_{Nk}(\xi)$ dada en (13) y sobre el vector $\mathbf{x}_B(\xi)$ de variables básicas dado en (14) solo para el caso en que hay decrecimiento de la variable no básica: $x_{Nk} - \xi$. Se analizará la desigualdad (15), en primer lugar para la variable no básica $x_{Nk}(\xi) = x_{Nk} + \xi$, y en segundo lugar para $\mathbf{x}_B(\xi) = \mathbf{x}_B - \xi \mathbf{d}_k$.

Coordenada ξ que satisface las condiciones de la variable no básica:

$$L_{Nk} \leq x_{Nk} - \xi \leq U_{Nk} \quad (24)$$

Dado que

$$\begin{aligned} x_{Nk} &= 0 \\ U_{Nk} &= +\infty \\ L_{Nk} &= 0 \text{ si } x_{Nk} \text{ es variable de holgura} \\ L_{Nk} &= -\infty \text{ si } x_{Nk} \text{ es variable de decisión} \end{aligned}$$

se tienen las condiciones sobre ξ correspondientes a $x_{Nk}(\xi) = x_{Nk} - \xi$, las cuales se resumen en la tabla 7.

| Condición | Intervalo |
|---|------------------------------|
| x_{Nk} es variable de holgura ($L_{Nk} = 0$) | $\xi \in (-\infty, 0]$ |
| x_{Nk} es variable de decisión ($L_{Nk} = -\infty$) | $\xi \in (-\infty, +\infty)$ |

Tabla 5: Valores permisibles de ξ cuando $x_{Nk}(\xi)$ decrece

Notemos que el número $\xi > 0$ por definición. Por tanto, $\xi > 0$ es determinado de la tabla 7 solo cuando la variable no básica x_{Nk} es variable de decisión.

La variable no básica x_{Nk} puede ser variable de decisión solo cuando el algoritmo simplex KKT está generando la secuencia de vértices $\{0, \dots, \mathbf{x}^\#\}$. En cuanto se alcanza el vértice frontera $\mathbf{x}^\#$ y de allí en adelante, toda variable no básica x_{Nk} se convierte en variable de holgura (ver la sección 3.4 del capítulo 3). Se tienen por consiguiente las dos observaciones siguientes:

- Cuando se desea el decrecimiento $x_{Nk}(\xi) = x_{Nk} - \xi$, el número $\xi > 0$ se podrá calcular solo cuando x_{Nk} es variable de decisión, lo cual sucede cuando el algoritmo simplex KKT está generando la secuencia de vértices $\{0, \dots, \mathbf{x}^\#\}$.
- Solo durante la generación de vértices de $\{0, \dots, \mathbf{x}^\#\}$, del vector de variables no básicas \mathbf{x}_N sale una variable de decisión y recibe una variable de holgura. Cuando se alcanza el vértice frontera $\mathbf{x}^\#$ y de allí en adelante, toda variable no básica x_{Nk} es una variable de holgura.

Coordenada ξ que satisface las siguientes condiciones de la variable básica:

$$\mathbf{L}_B \leq \mathbf{x}_B + \xi \mathbf{d}_k \leq \mathbf{U}_B \quad (25)$$

Se desea determinar ξ que satisfaga estas desigualdades vectoriales. Expresando cada componente de los vectores se obtiene un valor ξ_j por cada componente de los cuales se podrá elegir el número ξ que satisfaga (25):

$$L_{Bj} \leq x_{Bj} + \xi_j d_{kj} \leq U_{Bj}, j = 1, \dots, m$$

$$\Rightarrow L_{Bj} - x_{Bj} \leq +\xi_j d_{kj} \leq U_{Bj} - x_{Bj}, j = 1, \dots, m \quad (26)$$

Dado que

$$U_{Bj} = \infty$$

$$L_{Bj} = 0 \text{ si } x_{Bj} \text{ es variable de holgura}$$

$$L_{Bj} = -\infty \text{ si } x_{Bj} \text{ es variable de decisión}$$

de las dos desigualdades en (26) se tienen para todo j las tres desigualdades siguientes:

$$\xi d_{kj} \geq (L_{Bj} - x_{Bj}) \Big|_{L_{Bj}=0}, \quad \xi d_{kj} \geq (L_{Bj} - x_{Bj}) \Big|_{L_{Bj}=-\infty} \quad \& \quad \xi d_{kj} \leq U_{Bj} - x_{Bj} \Big|_{U_{Bj}=\infty} \quad (27)$$

Considerando las tres clases de valores para d_{kj} : $d_{kj} = 0$, $d_{kj} > 0$, y $d_{kj} < 0$, las tres condiciones dadas en (27) quedan como sigue:

$$d_{kj} = 0: \quad \xi_j \geq -\infty \Big|_{L_{Bj}=0}, \quad \xi_j \geq -\infty \Big|_{L_{Bj}=-\infty} \quad \& \quad \xi_j \leq +\infty \quad \Rightarrow \xi_j \in (-\infty, +\infty)$$

$$d_{kj} > 0: \quad \xi_j \leq \frac{-x_{Bj}}{d_{kj}} \Big|_{L_{Bj}=0}, \quad \xi_j \geq -\infty \Big|_{L_{Bj}=-\infty} \quad \& \quad \xi_j \leq +\infty \Rightarrow \xi_j \in \left(-\infty, \frac{-x_{Bj}}{d_{kj}}\right] \quad (28)$$

$$d_{kj} < 0: \quad \xi_j \geq \frac{+x_{Bj}}{|d_{kj}|} \Big|_{L_{Bj}=0}, \quad \xi_j \leq +\infty \Big|_{L_{Bj}=-\infty} \quad \& \quad \xi_j \leq +\infty \Rightarrow \xi_j \in \left[\frac{+x_{Bj}}{|d_{kj}|}, \infty\right)$$

En la tabla 8 se definen las *coordenadas KKT* ξ_j para el decrecimiento de la variable no básica $x_{Nk} - \xi$ siguiendo los mismos argumentos usados en la definición de *coordenadas KKT* para el incremento $x_{Nk} + \xi$.

Dado que d_{kj} puede adoptar un solo valor de los tres posibles: $d_{kj} = 0$, $d_{kj} > 0$, $d_{kj} < 0$, sucederá en cada j una sola coordenada de las tres que aparecen en los renglones 2, 3 y 4.

Las coordenadas están registradas en las columnas 3 y 4, mientras que en la columna 2 se presentan los intervalos de sus valores permisibles.

| Condición | Intervalo permisible | Coordenadas $\xi_j, j = 1, 2, \dots, \mathcal{N}^t $ | |
|--------------|--|---|---|
| | | x_{Bj} es variable de holgura ($L_{Bj} = 0$) | x_{Bj} es variable de decisión ($L_{Bj} = -\infty$) |
| $d_{kj} = 0$ | $\xi_j \in (-\infty, +\infty)$ | ξ_j es libre | ξ_j es libre |
| $d_{kj} > 0$ | $\xi_j \in (-\infty, \frac{-x_{Bj}}{d_{kj}}]$ | $\xi_j = \frac{-x_{Bj}}{d_{kj}} \Big _{L_{Bj}=0}$ | $\xi_j = 0 \Big _{L_{Bj}=-\infty}$ |
| $d_{kj} < 0$ | $\xi_j \in [\frac{+x_{Bj}}{ d_{kj} }, \infty)$ | $\xi_j = \frac{+x_{Bj}}{ d_{kj} } \Big _{L_{Bj}=0}$ | $\xi_j = 0 \Big _{L_{Bj}=-\infty}$ |

Tabla 6: Fórmulas de las coordenadas $\xi_j, j = 1, \dots, |\mathcal{N}^t|$ para el caso $x_{Nk} - \xi$

Cuando $L_{Bj} = 0$, la variable básica x_{Bj} es variable de holgura por lo que $x_{Bj} \geq 0$. Por tanto, de acuerdo a la tabla 8 se tendrá el resultado dado en (22):

$$\text{Si } d_{kj} > 0 \Rightarrow \xi_j \geq 0$$

$$\text{Si } d_{kj} < 0 \Rightarrow \xi_j \leq 0$$

Después de calcular el conjunto de coordenadas $\{\xi_j, j = 1 \dots |\mathcal{N}^t|\}$ se determina la coordenada positiva más restrictiva, como en (23), la cual se reescribe enseguida:

$$\xi_r = \min\{\xi_j | j \in \mathcal{N}_{cand}^t\}, \quad \mathcal{N}_{cand}^t = \{j | \xi_j > 0\}$$

La elección de ξ_r satisface también la condición impuesta en la tabla 7: $\xi_r \in (-\infty, +\infty)$, la cual sucede solo cuando la variable no básica x_{Nk} es variable de decisión. Dado que se está analizando el decrecimiento $x_{Nk}(\xi_r) = x_{Nk} - \xi_r$, la coordenada ξ_r se podrá calcular solo cuando x_{Nk} es variable de decisión, lo cual sucede cuando el algoritmo simplex KKT está generando la secuencia de vértices $\{0, \dots, \mathbf{x}^\#\}$.

Se distinguen 3 casos [Chvatal (1983), p.122] en que el número $\xi_r > 0$ dado en (23) puede ser determinado como sigue:

- (i) El número $\xi_r > 0$ se determina por la cota superior de ξ impuesta por $\mathbf{L}_B \leq \mathbf{x}_B \mp \xi \mathbf{d}_k \leq \mathbf{U}_B$ que es más estricta que la cota superior (si la hay) impuesta por $L_{Nk} \leq x_{Nk} \pm \xi \leq U_{Nk}$. En este caso el valor de la función objetivo mejora.
- (ii) El número $\xi_r > 0$ se determina por la cota superior de ξ impuesta por $L_{Nk} \leq x_{Nk} \pm \xi \leq U_{Nk}$ que es al menos tan estricta como la cota superior (si la hay) impuesta por $\mathbf{L}_B \leq \mathbf{x}_B \mp \xi \mathbf{d}_k \leq \mathbf{U}_B$. En el Simplex KKT, la base permanecerá sin cambio solo si la variable entrante x_{Nk} permanece en su valor nulo, es decir: $\xi = 0$ en $x_{Nk} \pm \xi$ y en $\mathbf{x}_B \mp \xi \mathbf{d}_k$. De acuerdo a (28), $\xi_j = 0$ cuando $x_{Bj} = 0$ lo cual implica que el problema es degenerado. Este caso se ha evitado en el problema de programación lineal (1), el cual se ha supuesto no degenerado.

- (iii) El número $\xi_r > 0$ se determina cuando $L_{Nk} \leq x_{Nk} \pm \xi \leq U_{Nk}$ ni $L_B \leq x_B \mp \xi d_k \leq U_B$ imponen cotas superiores sobre ξ , es decir, $\xi = \infty$, lo cual implica que el problema es no acotado. Este caso también se ha evitado en el problema de programación lineal (1), donde se ha supuesto que el polítopo de soluciones factibles es acotado.

5.2.7 Mecanismo de generación de coordenadas KKT

En la tabla 9 se presenta un resumen de las fórmulas para el cálculo de las *coordenadas KKT*, las cuales se obtienen de las columnas 3 y 4 de las tablas 6 y 8.

| Condición | Crecimiento $x_{Nk} + \xi$ | Decrecimiento $x_{Nk} - \xi$ | x_{Bj} es variable de decisión |
|-----------------|--|---|-------------------------------------|
| $d_{kj} = 0$ | ξ_j es libre | ξ_j es libre | ξ_j es libre |
| $d_{kj} \neq 0$ | $\xi_j = \frac{x_{Bj}}{d_{kj}} \Big _{L_{Bj}=0}$ | $\xi_j = \frac{-x_{Bj}}{d_{kj}} \Big _{L_{Bj}=0}$ | $\xi_j = 0 \Big _{L_{Bj}=-\infty}$ |

Tabla 7: Resumen de fórmulas para el cálculo de las *coordenadas KKT*

Las fórmulas de la tabla 9 para el cálculo de las *coordenadas KKT* se codifican en el procedimiento de factibilidad del algoritmo simplex KKT que se presenta en la tabla 10. El procedimiento procesa variables libres porque considera el crecimiento y decrecimiento de la variable no básica $x_{Nk} \pm \xi$. El procedimiento puede ser representado por medio de la siguiente función computacional:

$$[k, r, \{\xi_j, j = 1, \dots, m\}, \xi_r] = \text{Factibilidad}(k^+, k^-, L_N, U_N, x_B, L_B, p_B, N, B^{-1}, m) \quad (29)$$

Los parámetros del lado derecho de la igualdad (25) son de entrada, los del lado izquierdo son de salida. Los parámetros k, r se usarán en el procedimiento de pivoteo. Las coordenadas KKT $\xi_j, j = 1, \dots, m$ se usarán en el procedimiento de identificación de restricciones superfluas. La coordenada positiva más restrictiva ξ_r se usará en el procedimiento pivoteo para actualizar el diccionario.

Notar de la Tabla 10 que $\xi_0 = -L_{Nk} > 0$ en el caso de decrecimiento $x_{Nk} - \xi$. Recordar que el decrecimiento sucede solo cuando x_{Nk} es variable de decisión, es decir cuando $L_{Nk} = -\infty$. Notar también que el cálculo de las coordenadas KKT requiere de los índices k^+, k^- de la variable entrante x_{Nk} . Si $k^+ \neq 0$ las coordenadas se calculan con la fórmula para crecimiento $x_{Nk} + \xi$. Si $k^- \neq 0$ las coordenadas se calculan con la fórmula para decrecimiento $x_{Nk} - \xi$. Si $k^+ = 0$ y $k^- = 0$, se entiende que no se encontró variable no básica x_{Nk} que mejore el valor de la función objetivo. La tabla 11 resume las combinaciones de los índices k^+, k^- y sus consecuencias en el procedimiento de factibilidad y en el procedimiento de optimalidad de la tabla 4 del capítulo 3.

Datos de entrada:

| | | |
|-----------------------|--|-----------------------------------|
| k^+ | : Subíndice de x_{Nk} para su crecimiento $x_{Nk} + \xi$ | |
| k^- | : Subíndice de x_{Nk} para su decrecimiento $x_{Nk} - \xi$ | |
| L_N | : Vector $n \times 1$ límite inferior de x_N , | $L_N = [L_{N1}, \dots, L_{Nn}]^T$ |
| U_N | : Vector $n \times 1$ límite superior de x_N , | $U_N = [U_{N1}, \dots, U_{Nn}]^T$ |
| x_B | : Vector de variables básicas $m \times 1$, | $x_B = [x_{B1}, \dots, x_{Bm}]^T$ |
| L_B | : Vector $m \times 1$ límite inferior de x_B , | $L_B = [L_{B1}, \dots, L_{Bm}]^T$ |
| N | : Matriz no básica, | $N = [N_1, \dots, N_n]$ |
| B^{-1} | : Matriz inversa de la matriz básica $m \times m$ | |
| $m = \mathcal{N}^t $ | : Cardinalidad de la vecindad angular \mathcal{N}^t | |

Cálculo de coordenadas KKT: $\{\xi_j, j = 1, \dots, m\}$ Caso de crecimiento $x_{Nk} + \xi$ Si ($k^+ \neq 0$):

$$k = k^+$$

$$\xi_0 = U_{Nk}$$

$$\mathbf{d} = B^{-1} \mathbf{N}_k = [d_1, \dots, d_m]^T$$

$$\xi_j = 0, j = 1, \dots, m$$

Para $j = 1, \dots, m$ Si ($d_j \neq 0$) y ($L_{Bj} = 0$):

$$\xi_j = \frac{x_{Bj}}{d_j}$$

De otra forma ($d_j = 0$):

$$\xi_j = \infty$$

Caso de decrecimiento $x_{Nk} - \xi$ Si ($k^- \neq 0$) y ($k^+ = 0$):

$$k = k^-$$

$$\xi_0 = -L_{Nk}$$

$$\mathbf{d} = B^{-1} \mathbf{N}_k = [d_1, \dots, d_m]^T$$

$$\xi_j = 0, j = 1, \dots, m$$

Para $j = 1, \dots, m$ Si ($d_j \neq 0$) y ($L_{Bj} = 0$):

$$\xi_j = \frac{-x_{Bj}}{d_j}$$

De otra forma ($d_j = 0$):

$$\xi_j = \infty$$

Cálculo de la coordenada KKT positiva más restrictiva:

$$\xi_r = \min(\{\xi_j | j \in \mathcal{K}_{cand}^t\} \cup \{\xi_0\}), \quad \mathcal{K}_{cand}^t = \{j | \xi_j > 0\}, \quad \xi_0 = +\infty$$

$$r = \arg(\xi_r)$$

Tabla 8: Procedimiento de factibilidad del algoritmo simplex KKT

| k^+ | k^- | Consecuencia |
|----------|----------|---|
| $\neq 0$ | $\neq 0$ | Implica $x_{Nk} + \xi$ para mejorar el valor de la función objetivo |
| $\neq 0$ | $= 0$ | Implica $x_{Nk} + \xi$ para mejorar el valor de la función objetivo |
| $= 0$ | $\neq 0$ | Implica $x_{Nk} - \xi$ para mejorar el valor de la función objetivo |
| $= 0$ | $= 0$ | Implica solución corriente óptima (Tabla 4 del capítulo 3). |

Tabla 9: Combinaciones de los índices k^+ y k^-

De la tabla 11 se deduce que el crecimiento $x_{Nk} + \xi$ tiene precedencia sobre el decrecimiento $x_{Nk} - \xi$, lo cual está codificado en el procedimiento de factibilidad de la tabla 10.

5.3 Mecanismo de identificación de restricciones superfluas por criterio de coordenadas

Suponer que se han generado las coordenadas KKT $\xi_j, j = 1, \dots, |\mathcal{N}^t|$ en la iteración t por medio del procedimiento de factibilidad de la tabla 10. Suponer también que se ha determinado el conjunto de coordenadas negativas $\mathcal{F}_{var}^t = \{j | \xi_j < 0\}$ definido en (10). Algunas restricciones $\mathcal{F}_{restr}^t \subseteq \mathcal{F}_{rem}^t$ superfluas remanentes en \mathcal{N}^t podrán ser calculadas a partir de la relación entre el conjunto $\mathcal{F}_{var}^t = \{j \in \mathcal{N}^t | \xi_j < 0\}$ y los parámetros y variables del diccionario de un problema de programación lineal.

5.3.1 Identificación de variables y parámetros superfluos por coordenadas negativas

El criterio de coordenadas de la tabla 4 asegura que existe cierta correspondencia entre las coordenadas estrictamente negativas $\xi_j < 0, j \in \mathcal{N}^t$ y las restricciones superfluas $\mathcal{F}_{restr}^t \subseteq \mathcal{F}_{rem}^t$ remanentes en la vecindad angular \mathcal{N}^t .

La correspondencia $(\xi_j < 0) \leftrightarrow j \in \mathcal{F}_{restr}^t$ no es tan obvia. Para descifrarla se necesitará determinar otros tipos de correspondencia:

$$(\xi_j < 0) \leftrightarrow (x_{Bj}) \leftrightarrow (B_j) \leftrightarrow j \in \mathcal{F}_{restr}^t \quad (30)$$

3.1.1 Correspondencia $(\xi_j < 0) \leftrightarrow (x_{Bj})$: De la tabla 9 se obtienen los valores posibles de la coordenada ξ_j :

$$\xi_j = \begin{cases} \frac{+x_{Bj}}{d_{kj}} \Big|_{L_{Bj}=0 \text{ y } d_{kj} \neq 0} & \text{para } x_{Nk} + \xi \\ \frac{-x_{Bj}}{d_{kj}} \Big|_{L_{Bj}=0 \text{ y } d_{kj} \neq 0} & \text{para } x_{Nk} - \xi \\ 0 & |_{L_{Bj} = -\infty} \end{cases} \quad (31)$$

De esta expresión notamos que el subíndice de ξ_j es el mismo que el de la variable básica x_{Bj} . Cuando $L_{Bj} = -\infty$, x_{Bj} es variable de decisión x_i y la coordenada ξ_j es nula. Pero cuando $L_{Bj} = 0$, x_{Bj} es variable de holgura u_j y la coordenada ξ_j puede ser positiva, negativa, o nula. Dado que los índices de ξ_j negativas pertenecen al conjunto:

$$\mathcal{F}_{var}^t = \{j \in \mathcal{N}^t | \xi_j < 0\}$$

se deduce que los índices $j \in \mathcal{F}_{var}^t$ son índices de variables básicas. Es decir, existe la correspondencia $(\xi_j < 0) \leftrightarrow (x_{Bj})$.

3.1.2 Correspondencia $(\xi_j < 0) \leftrightarrow (B_j)$: La j -ésima coordenada $\xi_j < 0$ está en correspondencia directa con la j -ésima variable básica x_{Bj} , y ésta con la j -ésima columna de la matriz básica B . Por tanto la columna B_j puede ser suprimida y las variables y parámetros básicos con índices $j \in \mathcal{F}_{var}^t$ pueden también ser suprimidos del problema de programación lineal.

Dado que $x_{Bj}, j \in \mathcal{F}_{var}^t$ es factor de vectores columna de B , pero no es factor de renglones de B , el índice j no señala necesariamente un renglón de B y por tanto, tampoco señala directamente a una restricción del sistema de ecuaciones $Nx_N + Bx_B = b$. Pero por el criterio de coordenadas de la tabla 4 debe existir cierta correspondencia del índice $j \in \mathcal{F}_{var}^t$ con los renglones de B , y por tanto con las restricciones superfluas \mathcal{F}_{restr}^t . Esta correspondencia se determina en seguida.

5.3.2 Identificación de restricciones superfluas por coordenadas negativas

En seguida se establece la correspondencia faltante: $(\xi_j < 0) \leftrightarrow j \in \mathcal{F}_{restr}^t$. Consideremos el conjunto de restricciones \mathcal{N}^t en la iteración "t", es decir el sistema de ecuaciones del diccionario (33), el cual se reescribe en seguida:

$$\begin{aligned} Nx_N + Bx_B &= b \\ L_N < x_N < U_N, L_B \leq x_B < U_B \end{aligned} \quad (32)$$

Donde: $x_N \in R^n$, $x_B, b \in R^m$, $N \in R^{m \times n}$, $B \in R^{m \times m}$, $m = |\mathcal{N}^t|$. Suponer que (32) es el sistema inicial, es decir, $L_N = [-\infty, \dots, -\infty]$, $U_N = [+ \infty, \dots, + \infty]$, y $L_B = [0, \dots, 0]$, $U_B = [+ \infty, \dots, + \infty]$, $B = [B_1, \dots, B_m] = [e_1, \dots, e_m]$, donde e_j es un vector canónico, el cual tiene un "1" en la j -ésima posición y ceros en las restantes: $e_j = [0, \dots, 0, 1, 0, \dots, 0]^T$. Cada restricción del problema es una ecuación del sistema (32). Dado que la matriz B es cuadrada, un índice que apunta a una de sus columnas está relacionado con el índice que apunta a uno de sus renglones, y por tanto a alguna restricción. Investiguemos en seguida esta relación.

Las columnas de las matrices $N = [N_1, \dots, N_n]$ y $B = [e_1, \dots, e_m]$, y sus cotas inferiores para el diccionario inicial, son referenciados como sigue:

$$\begin{array}{ccc} L_{N1}, \dots, L_{Nn} & L_{B1}, \dots, & L_{Bm} \\ N_1, \dots, N_n & e_1, \dots, & e_m \\ \uparrow, \dots, \uparrow & \uparrow, \dots, & \uparrow \\ 1, \dots, n & (n+1), \dots, & (n+m) \end{array}$$

Es decir, las columnas de N y B son apuntadas por los elementos de los vectores siguientes:

$$\begin{aligned} \mathbf{p}_N^0 &= [p_{N1}, \dots, p_{Nn}] = [1, \dots, n] \\ \mathbf{p}_B^0 &= [p_{B1}, \dots, p_{Bm}] = [(n+1), \dots, (m+n)] \end{aligned}$$

El vector canónico e_j está apuntado indirectamente por p_{Bj}^0 y directamente apuntado por $p_{Bj}^0 - n = j$. Las restricciones del diccionario inicial, y solo las del diccionario inicial, están directamente apuntadas también por $p_{Bj}^0 - n = j$, es decir, por los componentes del vector de índices a restricciones siguiente:

$$\mathbf{p}_R^0 = (\mathbf{p}_B^0 - [n, \dots, n]) = [1, \dots, m]$$

En cada iteración el algoritmo simplex, las columnas de N y B y los elementos de los vectores \mathbf{p}_N^0 y \mathbf{p}_B^0 están sujetos a intercambio. Por lo que, los elementos de \mathbf{p}_B^0 no solo cambiarán de posición sino que algunos se encontrarán en \mathbf{p}_N^0 . Por tanto, el vector de índices a restricciones $\mathbf{p}_R = [p_{R1}, \dots, p_{Rm}]$ en la iteración t no es tan simple de determinar como lo fue \mathbf{p}_R^0 . Sin embargo, debe notarse que $p_{Bj} \in \mathbf{p}_B = [p_{B1}, \dots, p_{Bm}]$ en la iteración t puede satisfacer: $p_{Bj} \in \mathbf{p}_B^0$ ó bien $p_{Bj} \in \mathbf{p}_N^0$. En el primer caso p_{Bj} apuntará al vector canónico e_j y a la j -ésima restricción. Notar que $p_{Bj} \in \mathbf{p}_B^0$ está asociado a $B_j = e_j$, y este a su vez está asociado a $L_{Bj} = 0$. Por tanto el índice a restricciones p_{Rj} se puede calcular como sigue:

$$p_{Rj} = \begin{cases} (p_{Bj} - n) & \text{si } L_{Bj} = 0 \\ p_{Bj} & \text{si } L_{Bj} = -\infty \end{cases}, j = 1, \dots, m$$

El elemento:

$$p_{Rj} = (p_{Bj} - n) \text{ si } L_{Bj} = 0 \quad (33)$$

apunta a la j -ésima renglón de B y apunta por tanto a la j -ésima restricción. Mientras que el elemento $p_{Rj} = p_{Bj}$ para $L_{Bj} = -\infty$ apunta a columnas de la matriz no básica N . En otras palabras, el elemento $p_{Rj} = (p_{Bj} - n)$ identifica el renglón de la matriz B localizado por medio de su columna B_j correspondientes a $L_{Bj} = 0$. Como se verá más adelante, éste renglón y ésta columna de B son los únicos que podrán ser suprimidos. Las restricciones superfluas \mathcal{F}_{restr}^t definidas en (52) están en correspondencia con las coordenadas negativas \mathcal{F}_{var}^t definidas en (10). Por tanto, el conjunto de restricciones superfluas \mathcal{F}_{restr}^t en la iteración " t " se precisa en términos de \mathcal{F}_{var}^t como sigue:

$$\mathcal{F}_{restr}^t = \{(p_{Bj} - n) \mid L_{Bj} = 0, j \in \mathcal{F}_{var}^t\} \quad (34)$$

Dado que $\mathcal{F}_{var}^t = \{j \in \mathcal{N}^t \mid \xi_j < 0\}$, la expresión (34) se establece la relación buscada: $(\xi_j < 0) \leftrightarrow j \in \mathcal{F}_{restr}^t$.

Los renglones de B , es decir, las restricciones del sistema (32), que son apuntados por los elementos de \mathcal{F}_{restr}^t calculados por (34) están sujetos a supresión. Las variables básicas y parámetros básicos con índices $j \in \mathcal{F}_{var}^t$ están también sujetos a supresión. Después de estas

dos clases de supresiones, la matriz básica B resultante continuará siendo cuadrada. Las expresiones para los conjuntos \mathcal{F}_{restr}^t y \mathcal{F}_{var}^t en la iteración "t" nos habilitarán para diseñar el mecanismo de identificación de restricciones superfluas remanentes en la vecindad angular \mathcal{N}^t por medio del criterio de coordenadas, siempre que \mathcal{N}^t satisfaga la condición de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$. Esta condición se garantiza por restitución de ciertas restricciones que fueron suprimidas en el conjunto de restricciones superfluas \mathcal{F}_{ang} .

5.3.3 Restitución de restricciones suprimidas por criterio angular

Es posible identificar las restricciones superfluas remanentes en la vecindad angular \mathcal{N} siempre y cuando se satisfaga la condición de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$. Esta condición aún no está garantizada porque el conjunto \mathcal{F}_{ang} puede contener algunas restricciones de \mathcal{N}_{ata}^* . Las restricciones $j \in \mathcal{F}_{ang}$ que tienen coordenadas KKT positivas $\xi_j > 0$ son candidatas a pertenecer en \mathcal{N}_{ata}^* porque $\xi_j > 0$ tiene esa propiedad. El conjunto de restricciones a ser restituidas a \mathcal{N} se denota como \mathcal{N}_{restt}^* y se calcula por tanto como sigue:

$$\mathcal{N}_{restt}^* = \{j | j \in \mathcal{F}_{ang} \& \xi_j > 0\} \quad (35)$$

Enseguida se revisa un ejemplo en el que no se satisface la condición de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$, el cual fue presentado en el capítulo 4 con el mismo propósito. En la figura 14 hay al menos una restricción cuyo gradiente, ∇h_1 en la figura, delimita el cono KKT y al mismo tiempo es suprimido por pertenecer a $\mathcal{F}_{ang} = \{j \in \mathcal{M} | \cos(\theta_j) < 0\}$.

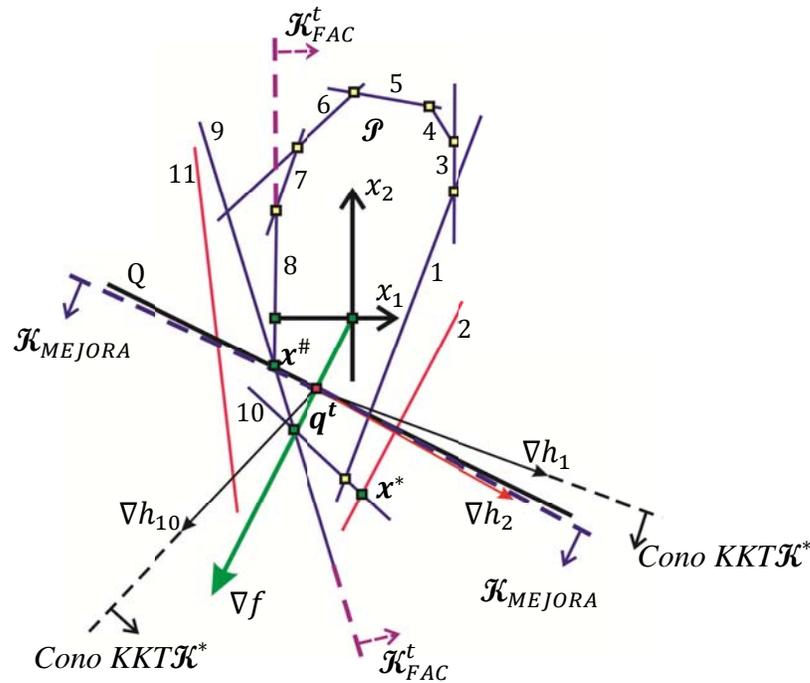


Fig. 6: Gradiente ∇h_1 excluido del Cono KKT \mathcal{K}^*

Los dos criterios de identificación de restricciones superfluas aplicados a las restricciones de la figura 14 en el vértice frontera $x^\#$ son los siguientes:

- (1) Criterio angular que selecciona las restricciones cuyos gradientes forman ángulo cuyo coseno es negativo respecto de la línea Q como se muestra en la figura 15 (a). Estas restricciones forman el conjunto $\mathcal{F}_{ang}^\# = \{1\ 3\ 4\ 5\ 6\}$
- (2) Criterio de coordenadas que selecciona los índices de las variables básicas superfluas cuyas coordenadas KKT son negativas con respecto a la línea Q como se muestra en la figura 15 (b): $\mathcal{F}_{var}^\# = \{4\ 5\ 6\ 7\ 11\}$

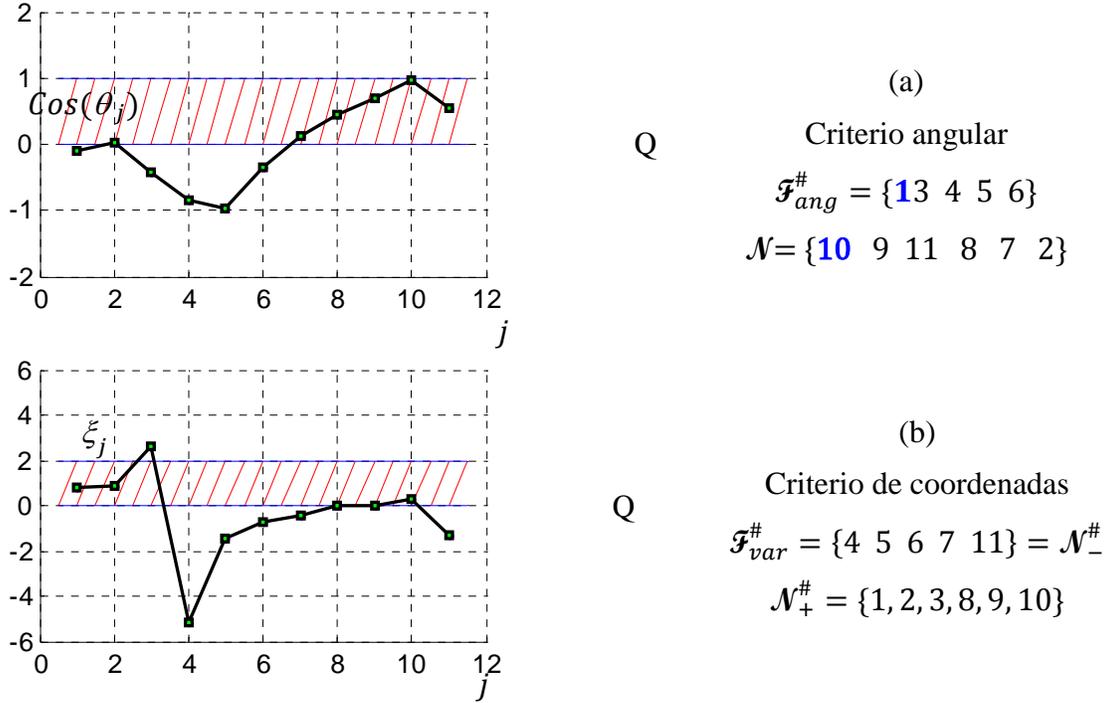


Fig. 7: Supresión de coordenadas superfluas

Dado que $\mathcal{N}_{ata}^* = \{10, 1\}$, notamos que $\mathcal{N} = \{10\ 9\ 11\ 8\ 7\ 2\}$ no satisface la condición de superfluidad $\mathcal{N}_{ata}^* \subset \mathcal{N}$. La restricción **1** faltante en \mathcal{N} se encuentra en $\mathcal{F}_{ang}^\# = \{1\ 3\ 4\ 5\ 6\}$. Sin embargo, el conjunto \mathcal{N}_{restt}^* determinado a partir de las dos gráficas de la figura 15 rescatará la restricción **1** que falta en \mathcal{N} :

$$\mathcal{N}_{restt}^* = \{j | j \in \mathcal{F}_{ang}^\# \& \xi_j > 0\} = \{1\ 3\}$$

Finalmente, la restitución de la restricción **1** a \mathcal{N} se realiza como sigue:

$$\begin{aligned} \mathcal{N} &= \mathcal{N} \cup \mathcal{N}_{restt}^* \\ \mathcal{N} &= \{10\ 9\ 11\ 8\ 7\ 2\} \cup \{1\ 3\} = \{10\ 9\ 11\ 8\ 7\ 2\ 1\ 3\} \end{aligned}$$

El resultado es que la vecindad angular \mathcal{N} ya satisface la condición de superfluidad $\mathcal{N}_{ata}^* \subset \mathcal{N}$.

Notamos que el conjunto útil de coordenadas KKT $\xi_j, j = 1, \dots, m$ está disponible a partir del vértice frontera $\mathbf{x}^\#$. Por tanto, el cálculo del conjunto \mathcal{N}_{restt}^* y su restitución a \mathcal{N} sólo se podrá realizar a partir de $\mathbf{x}^\#$, lo cual se puede lograr considerando las observaciones siguientes:

1. Los conjuntos $\mathcal{F}_{var}^\#$ y \mathcal{N} se pueden calcular antes de iniciar la ejecución del algoritmo simplex KKT por medio del criterio angular establecido en el capítulo 4.
2. El aseguramiento de la condición de superfluidad de \mathcal{N} por medio de \mathcal{N}_{restt}^* se podrá realizar sólo a partir del vértice frontera $\mathbf{x}^\#$.
3. La identificación de restricciones superfluas \mathcal{F}_{restr}^t y de variables y parámetros básicos superfluos \mathcal{F}_{var}^t se podrán realizara partir del vértice frontera $\mathbf{x}^\#$ en cada iteración "t" después de asegurar la condición de superfluidad de \mathcal{N} .

Suponiendo que se satisface la condición de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$ debido a la restitución de \mathcal{N}_{restt}^* . Lo que procede ahora es detectar las restricciones cuyas coordenadas ξ_j son negativas: $\mathcal{F}_{var}^t = \{j \in \mathcal{N}^t \mid \xi_j < 0\}$. Se ha encontrado que al suprimir todas las restricciones cuyas coordenadas son negativas se corre el riesgo de eliminar restricciones que delimitan el cono KKT. Es decir, la condición de superfluidad está todavía en riesgo de no cumplirse. Sin embargo, existe una forma de detectar restricciones candidatas a delimitar el cono KKT cuyas coordenadas ξ_j son negativas.

5.3.4 Preservación de la condición de superfluidad

En la sección anterior se asegura que una vez que el conjunto de restricciones $\mathcal{N}_{restt}^* = \{j \mid j \in \mathcal{F}_{ang} \& \xi_j > 0\}$ ha sido restituido a la vecindad angular \mathcal{N} , esta satisface la condición de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$. Se mostrará enseguida que esta restitución no es suficiente para garantizar la superfluidad de \mathcal{N} .

El criterio de coordenadas establece que toda restricción $j \in \mathcal{F}_{restr}^t$ puede ser suprimida, donde:

$$\mathcal{F}_{restr}^t = \{(p_{Bj} - n) \mid L_{Bj} = 0, j \in \mathcal{F}_{var}^t\}$$

Dado que $\mathcal{F}_{var}^t = \{j \in \mathcal{N}^t \mid \xi_j < 0\}$, las restricciones $j \in \mathcal{F}_{restr}^t$ son las de coordenadas negativas condicionadas a $L_{Bj} = 0$. Pero, como se mostrará en el siguiente ejemplo, no toda restricción con coordenada negativa puede ser suprimida.

Consideremos el problema de programación lineal con polítopo el de la figura 14 la cual se despliega en la figura 16 en forma simplificada para ilustrar la condición de superfluidad. En la figura se muestra el cambio de signo de la coordenada ξ_4 , de negativa a positiva, cuando sucede el cambio de vértices: $\mathbf{x}^2 \rightarrow \mathbf{x}^3$.

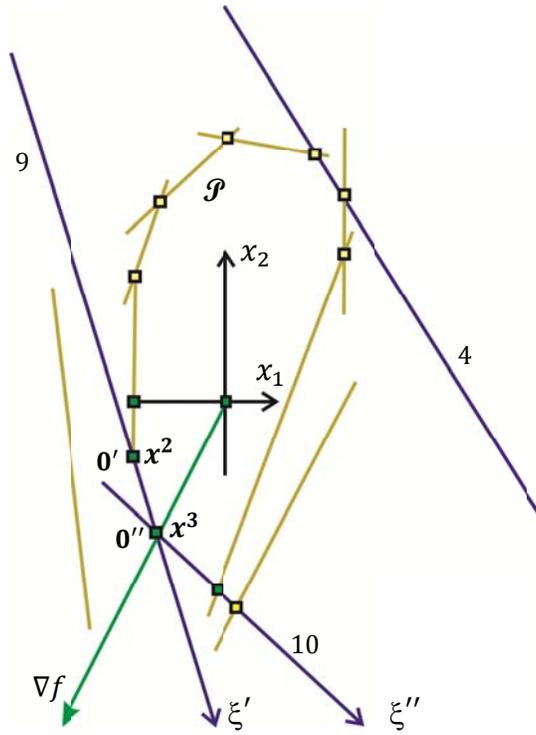


Fig. 8: Cambio de signo de la coordenada ξ_4 , de negativa a positiva

Para el vértice \mathbf{x}^2 se tiene el eje de coordenadas ξ' cuya intersección con el hiperplano 4 produce $\xi_4 < 0$. El signo de ξ_4 es negativo porque la intersección aparece en el negativo del eje ξ' . Al pasar al vértice \mathbf{x}^3 se tiene el eje de coordenadas ξ'' cuya intersección con el hiperplano 4 produce $\xi_4 > 0$ porque tal intersección sucede en el positivo del eje ξ'' .

Habiendo sucedido $\xi_4 < 0$ en el vértice \mathbf{x}^2 , la restricción 4 está sujeta a supresión en virtud del criterio de coordenadas. Esta supresión sucedería antes de arribar al vértice \mathbf{x}^3 . Pero en este vértice se tiene $\xi_4 > 0$ por lo que el gradiente de la restricción 4 califica como candidato a delimitar el cono KKT. Por tanto, la restricción 4 no debería ser suprimida solo porque su correspondiente coordenada adoptó un signo negativo en alguna iteración.

Se observa en la figura 16 que $\xi_4 < 0$ es de las coordenadas más negativas en el vértice \mathbf{x}^2 . Por esta razón se convierte en positiva al pasar al siguiente vértice. Por tanto, una regla simple de selección de restricciones a ser suprimidas por medio del criterio de coordenadas es que se supriman restricciones $j \in \mathcal{F}_{var}^t$ cuyas coordenadas sean las menos negativas más cercanas al origen, es decir que satisfagan la condición: $U_{coord} < \xi_j < 0$, es decir:

$$\mathcal{F}_{var}^t = \{j \in \mathcal{N}^t \mid U_{coord} < \xi_j < 0\} \quad (36)$$

Donde U_{coord} es un umbral negativo que evita que se supriman restricciones cuyos gradientes son candidatos a delimitar el cono KKT. Estas restricciones corresponden a coordenadas cambiantes en signo. El peor de los casos sucede cuando $U_{coord} \rightarrow 0$ porque

implica no suprimir restricciones. El mejor de los casos es cuando $U_{coord} \rightarrow -\infty$. Es decir, el umbral U_{coord} debe ser sintonizado de acuerdo al problema de aplicación. Un valor de U_{coord} razonable encontrado en problemas de sujeción por manos robóticas es el que preserva en \mathcal{F}_{var}^t las dos terceras partes del total de restricciones negativas más cercanas al origen. El mecanismo de identificación de restricciones superfluas \mathcal{F}_{restr}^t y de variables y parámetros básicos superfluos \mathcal{F}_{var}^t en la iteración "t" se muestra en la tabla 13. El mecanismo garantiza que se satisfaga la condición de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}$ por medio de la restitución de $\mathcal{N}_{rest}^* = \{j | j \in \mathcal{F}_{ang} \& \xi_j > 0\}$ a \mathcal{N} , y la condición de superfluidad $\mathcal{N}_{ata}^* \subseteq \mathcal{N}^t$ por medio de la selección apropiada de las restricciones de coordenadas negativas: $\mathcal{F}_{var}^t = \{j \in \mathcal{N}^t | U_{coord} < \xi_j < 0\}$.

Datos de entrada:

$\mathcal{N} \subseteq \mathcal{E}$: Vecindad angular calculada antes de iniciar el algoritmo

$\mathcal{F}_{ang} = \mathcal{E} - \mathcal{N}$: Conjunto de restricciones superfluas iniciales

$m = |\mathcal{E}|$: Cardinalidad de \mathcal{E}

$\xi_j, j = 1, \dots, m$: Las coordenadas KKT generadas por el mecanismo de la tabla 10

$t = 1, 2, \dots$: El número de iteraciones simplex

$t^\# = n$: El número de iteraciones en que se alcanza el vértice frontera $\mathbf{x}^\#$

$L_{Bj}, j = 1, \dots, m$: Límite inferior de la variable básica x_{Bj}

$p_{Bj}, j = 1, \dots, m$: Apuntador a la variable básica x_{Bj} .

Si $t \geq t^\#$ hacer lo que sigue:

Si $t = t^\#$ hacer lo que sigue:

Para cada $j \in \mathcal{F}_{ang}$ y $\xi_j > 0$ hacer restitución: $\mathcal{N} = \mathcal{N} \cup \{j\}$

Para $j = 1, \dots, |\mathcal{N}|$ hacer lo que sigue preservando superfluidad:

Si $(U_{coord} < \xi_j < 0)$ y $(L_{Bj} = 0)$ hacer:

$$\mathcal{F}_{var}^t(j) = j$$

$$\mathcal{F}_{restr}^t(j) = (p_{Bj} - n)$$

De otra forma ($t > t^\#$) hacer lo que sigue:

Para $j = 1, \dots, |\mathcal{N}^t|$ hacer lo que sigue preservando superfluidad:

Si $(U_{coord} < \xi_j < 0)$ y $(L_{Bj} = 0)$ hacer:

$$\mathcal{F}_{var}^t(j) = j$$

$$\mathcal{F}_{restr}^t(j) = (p_{Bj} - n)$$

Tabla 10: Mecanismo de identificación de restricciones superfluas \mathcal{F}_{restr}^t y de variables y parámetros básicos superfluos \mathcal{F}_{var}^t .

El mecanismo de la tabla 13 puede ser representado por la función computacional:

$$[\mathcal{F}_{restr}^t, \mathcal{F}_{var}^t] = \text{Identificación}(\mathcal{N}, \mathcal{F}_{ang}, \xi, t, t^\#, \mathbf{L}_B, \mathbf{p}_B) \quad (37)$$

5.4 Supresión de restricciones superfluas por criterio de coordenadas

Se desea reducir la cardinalidad de la vecindad angular \mathcal{N}^t en cada iteración t del algoritmo simplex KKT. Expresado de otra forma, se desea reducir la dimensión del problema de programación lineal sujeto a optimización en cada iteración t del algoritmo simplex KKT. La reducción se logra suprimiendo del problema el conjunto de restricciones superfluas \mathcal{F}_{coord}^t y el conjunto de variables y parámetros básicos \mathcal{F}_{basic}^t , los cuales fueron determinados por el mecanismo de la tabla 13. El problema está definido en el espacio R^n y está constituido por $m = |\mathcal{N}^t|$ restricciones funcionales, y se reescribe (32) en la forma siguiente:

$$\begin{aligned}
 \underset{x}{\text{maximizar}} \quad & f(x) = \mathbf{c}_N^T \mathbf{x}_N + \mathbf{c}_B^T \mathbf{x}_B \\
 \text{subjeta a:} \quad & N\mathbf{x}_N + B\mathbf{x}_B = \mathbf{b} \\
 & L_N < \mathbf{x}_N < U_N \\
 & L_B \leq \mathbf{x}_B < U_B
 \end{aligned} \tag{38}$$

Las variables y parámetros con sus valores de inicio se muestran enseguida:

$$\begin{aligned}
 \mathbf{x}_N &= [x_{N1}, x_{N2}, \dots, x_{Nn}]^T = [x_1, x_2, \dots, x_n]^T: \text{variables de decisión} \\
 \mathbf{x}_B &= [x_{B1}, x_{B2}, \dots, x_{Bm}]^T = [u_1, u_2, \dots, u_m]^T: \text{variables de holgura} \\
 \mathbf{c}_N &= [c_{N1}, c_{N2}, \dots, c_{Nn}]^T = [c_1, c_2, \dots, c_n]^T: \text{coeficientes de la función objetivo} \\
 \mathbf{c}_B &= [c_{B1}, c_{B2}, \dots, c_{Bm}]^T = [0, 0, \dots, 0]^T: \text{valor inicial} \\
 N &= [N_1, N_2, \dots, N_n] = G, \text{ } m \times n: \text{matriz de coeficientes} \\
 B &= [B_1, B_2, \dots, B_m] = I, \text{ } m \times m: \text{matriz inicial} \\
 \mathbf{b} &= [b_1, b_2, \dots, b_m]^T, \text{ } m \times 1 \\
 L_N &= [L_{N1}, L_{N2}, \dots, L_{Nn}] = [-\infty, -\infty, \dots, -\infty] \\
 U_N &= [U_{N1}, U_{N2}, \dots, U_{Nn}] = [+\infty, +\infty, \dots, +\infty] \\
 L_B &= [L_{B1}, L_{B2}, \dots, L_{Bm}] = [0, 0, \dots, 0] \\
 U_B &= [U_{B1}, U_{B2}, \dots, U_{Bm}] = [+\infty, +\infty, \dots, +\infty] \\
 \mathbf{p}_N &= [p_{N1}, \dots, p_{Nn}] = [1, \dots, n]: \text{apuntador inicial a elementos de } \mathbf{x}_N \\
 \mathbf{p}_B &= [p_{B1}, \dots, p_{Bm}] = [(n+1), \dots, (m+n)]: \text{apuntador inicial a elementos de } \mathbf{x}_B
 \end{aligned} \tag{39}$$

Después de la iteración $t^\#$ en que se alcanza el vértice frontera $\mathbf{x}^\#$, el modelo (38) se somete a supresiones dictadas por los conjuntos de restricciones superfluas \mathcal{F}_{coord}^t y de variables y parámetros básicos \mathcal{F}_{basic}^t , conjuntos que fueron calculados por el mecanismo de identificación de la tabla 13.

5.4.1 Supresión de restricciones superfluas:

Las restricciones funcionales del problema de programación lineal son las m ecuaciones del sistema $N\mathbf{x}_N + B\mathbf{x}_B = \mathbf{b}$. Por lo que solo basta suprimir los renglones correctos de los parámetros N , B y \mathbf{b} . Los renglones sujetos a supresión son los apuntados por los índices en \mathcal{F}_{restr}^t dado en (35) el cual se reescribe enseguida:

$$\mathcal{F}_{restr}^t = \{(p_{Bj} - n) | L_{Bj} = 0, j \in \mathcal{F}_{var}^t\}$$

Como se puede ver del código del mecanismo de identificación de la tabla 13, la condición $L_{Bj} = 0$ en la expresión de \mathcal{F}_{restr}^t está implícita en \mathcal{F}_{var}^t . La supresión de los renglones superfluos de N , B y \mathbf{b} apuntados por los elementos de \mathcal{F}_{restr}^t y la supresión de variables y parámetros superfluos apuntados por \mathcal{F}_{var}^t se codifican enseguida.

5.4.2 Supresión de variables y parámetros básicos superfluos:

Las variables básicas x_{Bj} sujetas a supresión son las apuntadas por los índices $j \in \mathcal{F}_{var}^t$. Dado que \mathbf{p}_B es apuntador a los elementos de \mathbf{x}_B , el índice $j \in \mathcal{F}_{var}^t$ es también apuntador a los elementos de \mathbf{c}_B , \mathbf{L}_B y \mathbf{U}_B como se puede ver de la estructura de (38). Por tanto, las cantidades básicas que están sujetas a supresión, son las siguientes:

$$\mathbf{x}_B, \mathbf{p}_B, B, \mathbf{c}_B, \mathbf{L}_B \text{ y } \mathbf{U}_B \quad (40)$$

En la matriz B se suprime su j -ésima columna, y en \mathbf{x}_B , \mathbf{p}_B , \mathbf{c}_B , \mathbf{L}_B y \mathbf{U}_B se suprimen sus j -ésimos elementos, donde $j \in \mathcal{F}_{var}^t$. Después de la supresión, la dimensión del problema de programación lineal dado en (11) habrá sido reducida en $|\mathcal{F}_{restr}^t|$ en la iteración t . Por tanto, el costo computacional de los cálculos realizados por el algoritmo simplex KKT será reducido en iteraciones subsecuentes. La supresión de restricciones superfluas y de variables y parámetros superfluos se codifica en el mecanismo de las tablas 14 y 15.

| | |
|---------------------------------|---|
| Datos de entrada: | |
| $t = 1, 2, \dots$ | : El número de iteraciones simplex |
| $t^\# = n$ | : El número de iteraciones en que se alcanza el vértice frontera $\mathbf{x}^\#$ |
| \mathbf{x}_B | : Vector de variables básicas $m \times 1$ |
| N | : Matriz no básica $m \times n$, $= [N_{r_1}, \dots, N_{r_n}]^T$ matriz en renglones |
| B | : Matriz básica $m \times m$, $= [B_{r_1}, \dots, B_{r_m}]^T$ matriz en renglones |
| $= [B_{c_1}, \dots, B_{c_m}]^T$ | matriz en columnas |
| \mathbf{b} | : Vector de límites de recursos $m \times 1$ |
| \mathbf{c}_B | : Vector coeficiente $m \times 1$ de variables básicas |
| \mathbf{L}_B | : Vector $m \times 1$ límite inferior de \mathbf{x}_B |
| \mathbf{U}_B | : Vector $m \times 1$ límite superior de \mathbf{x}_B |
| \mathbf{p}_B | : Vector $m \times 1$ puntador a elementos de \mathbf{x}_B |
| \mathcal{N}^t | : Vecindad angular $m \times 1$ en la iteración t , $\mathcal{N}^t = [\mathcal{N}_1^t, \dots, \mathcal{N}_m^t]$ |
| \mathcal{F}_{var}^t | : Conjunto de índices de coordenadas negativas |
| $L = \mathcal{F}_{restr}^t $ | : Cardinalidad de \mathcal{F}_{restr}^t |
| \mathcal{F}_{restr}^t | : Conjunto de restricciones superfluas $= [\mathcal{F}_{restr}^t(1), \dots, \mathcal{F}_{restr}^t(L)]$ |
| $m = \mathbf{b} $ | : Dimensión de \mathbf{b} |

Tabla 11: Datos de entrada para el mecanismo de supresión de la tabla 15

Si $(t > t^\#)$ y $(L > 0)$ hacer lo siguiente:

Para $j = L, (L - 1), \dots, 1$ suprimir renglones de B (restricciones superfluas):

$$L_1 = |\mathbf{b}|$$

$$L2 = \mathcal{F}_{restr}^t(j)$$

$$N1 = [N_{r1}, \dots, N_{r(L2-1)}]^T, N2 = [N_{r(L2+1)}, \dots, N_{rL_1}]^T, N = \begin{bmatrix} N1 \\ N2 \end{bmatrix}$$

$$B1 = [B_{r1}, \dots, B_{r(L2-1)}]^T, B2 = [B_{r(L2+1)}, \dots, B_{rL_1}]^T, B = \begin{bmatrix} B1 \\ B2 \end{bmatrix}$$

$$b1 = [b_1, \dots, b_{(L2-1)}]^T, b2 = [b_{(L2+1)}, \dots, b_{L_1}]^T, \mathbf{b} = \begin{bmatrix} b1 \\ b2 \end{bmatrix}$$

$$\mathcal{N}1 = [\mathcal{N}_1^t, \dots, \mathcal{N}_{L2-1}^t]^T, \mathcal{N}2 = [\mathcal{N}_{L2+1}^t, \dots, \mathcal{N}_{L_1}^t]^T, \mathcal{N}^t = \begin{bmatrix} \mathcal{N}1 \\ \mathcal{N}2 \end{bmatrix}$$

Para $j = L, (L - 1), \dots, 1$ suprimir columnas de B :

$$L3 = \mathcal{F}_{var}^t(j)$$

$$B1 = [B_{c1}, \dots, B_{c(L3-1)}] \quad , \quad B2 = [B_{c(L3-1)}, \dots, B_{cm}]^T, B = [B1 \quad B2]$$

m = número de columnas de B

Calcular B^{-1}

Para $j = L, (L - 1), \dots, 1$ suprimir variables y parámetros superfluos:

$$L_1 = |\mathbf{p}_B|$$

$$L_2 = \mathcal{F}_{var}^t(j)$$

$$\mathbf{X}1 = [x_{B1}, \dots, x_{B(L2-1)}]^T, \mathbf{X}2 = [x_{B(L2+1)}, \dots, x_{BL_1}]^T, \mathbf{x}_B = \begin{bmatrix} \mathbf{X}1 \\ \mathbf{X}2 \end{bmatrix}$$

$$\mathbf{c}1_B = [c_{B1}, \dots, c_{B(L2-1)}]^T, \mathbf{c}2_B = [c_{B(L2+1)}, \dots, c_{BL_1}]^T, \mathbf{c}_B = \begin{bmatrix} \mathbf{c}1_B \\ \mathbf{c}2_B \end{bmatrix}$$

$$\mathbf{L}1_B = [L_{B1}, \dots, L_{B(L2-1)}]^T, \mathbf{L}2_B = [L_{B(L2+1)}, \dots, L_{BL_1}]^T, \mathbf{L}_B = \begin{bmatrix} \mathbf{L}1_B \\ \mathbf{L}2_B \end{bmatrix}$$

$$\mathbf{U}1_B = [U_{B1}, \dots, U_{B(L2-1)}]^T, \mathbf{U}2_B = [U_{B(L2+1)}, \dots, U_{BL_1}]^T, \mathbf{U}_B = \begin{bmatrix} \mathbf{U}1_B \\ \mathbf{U}2_B \end{bmatrix}$$

$$\mathbf{p}1_B = [p_{B1}, \dots, p_{B(L2-1)}]^T, \mathbf{p}2_B = [p_{B(L2+1)}, \dots, p_{BL_1}]^T, \mathbf{p}_B = \begin{bmatrix} \mathbf{p}1_B \\ \mathbf{p}2_B \end{bmatrix}$$

Tabla 12: Mecanismo de supresión de restricciones superfluas y de variables y parámetros superfluos identificados por criterio de coordenadas

La función computacional que representa el mecanismo de supresión de restricciones superfluas y de variables y parámetros superfluos de las tablas 14 y 15 es la siguiente:

$$\begin{aligned} & [\mathcal{N}^t, \mathbf{x}_B, N, B, B^{-1}, \mathbf{b}, \mathbf{c}_B, \mathbf{L}_B, \mathbf{U}_B, \mathbf{p}_B] \\ = & \text{Supresión}(\mathcal{F}_{coord}^t, \mathcal{F}_{bas}^t, \mathcal{N}^t, t, t^\#, m, \mathbf{x}_B, N, B, \mathbf{b}, \mathbf{c}_B, \mathbf{L}_B, \mathbf{U}_B, \mathbf{p}_B) \end{aligned} \quad (41)$$

5.5 Procedimiento de pivoteo y actualización del algoritmo simplex KKT

En cada iteración del algoritmo simplex KKT sucede un intercambio entre los componentes de los parámetros y variables del problema, lo que se conoce como pivoteo. El pivoteo es el intercambio de papeles entre la k -ésima variable no básica $x_{Nk}(\xi_r)$ y la r -ésima variable básica $x_{Br}(\xi_r)$. Debido a que $x_{Nk} \in \mathbf{x}_N$ y que $x_{Br} \in \mathbf{x}_B$ y debido a la influencia de \mathbf{x}_N y \mathbf{x}_B en las ecuaciones del problema de programación lineal, el pivoteo de $x_{Nk}(\xi_r)$ y $x_{Br}(\xi_r)$ fuerza los seis pivoteos siguientes:

$$\mathbf{x}_N \leftrightarrow \mathbf{x}_B, \mathbf{c}_N \leftrightarrow \mathbf{c}_B, N \leftrightarrow B, \mathbf{L}_N \leftrightarrow \mathbf{L}_B, \mathbf{U}_N \leftrightarrow \mathbf{U}_B, \mathbf{p}_N \leftrightarrow \mathbf{p}_B \quad (42)$$

La actualización del diccionario se realiza por medio de las ecuaciones (13), (14) y (16):

$$\begin{aligned} x_{Nk}(\xi_r) &= x_{Nk} \pm \xi_r \\ \mathbf{x}_B(\xi_r) &= \mathbf{x}_B \mp \xi_r \mathbf{d}_k \\ f(\xi_r) &= f \pm (c_{Nk} - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}_k) \xi_r \end{aligned} \quad (43)$$

El signo superior que aparece en el doble signo se usa cuando $k^+ \neq 0$, mientras que el signo inferior se usa cuando $k^- \neq 0$. La actualización del diccionario y el pivoteo se presentan en las tablas 16 y 17.

| | |
|-------------------|--|
| Datos de entrada: | |
| $k^+ = 0$ | : Índice de x_{Nk} para su crecimiento $x_{Nk} + \xi$ |
| $k^- = 0$ | : Índice de x_{Nk} para su decrecimiento $x_{Nk} - \xi$ |
| k | : Índice a variable no básica x_{Nk} entrante |
| r | : Índice a variable básica x_{Br} saliente |
| f | : Valor corriente de la función objetivo |
| \mathbf{x}_N | : Vector de variables no básicas $n \times 1$ |
| \mathbf{x}_B | : Vector de variables básicas $m \times 1$ |
| \mathbf{c}_N | : Vector coeficiente de variables no básicas $n \times 1$ |
| \mathbf{c}_B | : Vector coeficiente de variables básicas $m \times 1$ |
| N | : Matriz no básica $m \times n$ |
| B | : Matriz básica $m \times m$ |
| \mathbf{L}_N | : Vector $n \times 1$ límite inferior de \mathbf{x}_N |
| \mathbf{L}_B | : Vector $m \times 1$ límite inferior de \mathbf{x}_B |
| \mathbf{U}_N | : Vector $n \times 1$ límite superior de \mathbf{x}_N |
| \mathbf{U}_B | : Vector $m \times 1$ límite superior de \mathbf{x}_B |
| \mathbf{u} | : $= [u_j, \dots, u_m]$, vector de variables de holgura ($u_j = b_j - \mathbf{a}_j^T \mathbf{x}$) |
| \mathbf{p}_N | : Vector $1 \times n$ apuntador a elementos de \mathbf{x}_N |
| \mathbf{p}_B | : Vector $1 \times m$ apuntador a elementos de \mathbf{x}_B |

Tabla 13: Datos de entrada para el procedimiento de actualización y pivoteo

Actualización del diccionario:

Si ($k^+ \neq 0$): caso de crecimiento $x_{Nk} + \xi$

$$x_{Nk} = x_{Nk} + \xi_r$$

$$\mathbf{x}_B(\xi) = \mathbf{x}_B - B^{-1} \mathbf{N}_k \xi_r$$

$$f = f + (\mathbf{c}_{Nk} - \mathbf{c}_B^T B^{-1} \mathbf{N}_k) \xi_r$$

Si ($k^- \neq 0$): caso de decrecimiento $x_{Nk} - \xi$

$$x_{Nk} = x_{Nk} - \xi_r$$

$$\mathbf{x}_B(\xi) = \mathbf{x}_B + B^{-1} \mathbf{N}_k \xi_r$$

$$f = f - (\mathbf{c}_{Nk} - \mathbf{c}_B^T B^{-1} \mathbf{N}_k) \xi_r$$

Hacer sucesivamente una asignación a la vez de w_N y de w_B :

$$w_N = \mathbf{x}_N, \mathbf{c}_N, N, \mathbf{L}_N, \mathbf{U}_N, \mathbf{p}_N$$

$$w_B = \mathbf{x}_B, \mathbf{c}_B, B, \mathbf{L}_B, \mathbf{U}_B, \mathbf{p}_B$$

Intercambiar el elemento w_{Nk} con el elemento w_{Br} :

$$aux = w_{Nk}$$

$$w_{Nk} = w_{Br}$$

$$w_{Br} = aux$$

Rescatar la solución $[\mathbf{x}, \mathbf{u}]$ a partir de \mathbf{x}_N y \mathbf{x}_B .

Tabla 14: Procedimiento de pivoteo y actualización del diccionario

El procedimiento de pivoteo y actualización del diccionario de las tablas 16 y 17 puede ser representado por medio de la siguiente función computacional:

$$\begin{aligned} & [\mathbf{x}_N, \mathbf{x}_B, N, B, \mathbf{c}_N, \mathbf{c}_B, \mathbf{L}_N, \mathbf{L}_B, \mathbf{U}_N, \mathbf{U}_B, \mathbf{p}_N, \mathbf{p}_B, f, \mathbf{x}, \mathbf{u}] \\ & = \text{Pivoteo}(k^+, k^-, k, r, \mathbf{x}_N, \mathbf{x}_B, N, B, \mathbf{c}_N, \mathbf{c}_B, \mathbf{L}_N, \mathbf{L}_B, \mathbf{U}_N, \mathbf{U}_B, \mathbf{p}_N, \mathbf{p}_B) \end{aligned} \quad (44)$$

5.6 Algoritmo simplex KKT

Los procedimientos del algoritmo simplex revisado [Chvatal, 1983] son esencialmente optimalidad, factibilidad y pivoteo. Sin embargo, el algoritmo simplex KKT que se presentará en las tablas 18 y 19 está formado por cinco procedimientos:

1. Procedimiento de optimalidad que incluye los mecanismos de variables libres.
2. Procedimiento de factibilidad, el cual se adapta para procesar las variables libres del problema. Incluye el mecanismo de generación de coordenadas.
3. Mecanismo de identificación de restricciones superfluas por criterio de coordenadas
4. Procedimiento de actualización y pivoteo.
5. Mecanismo de supresión de restricciones superfluas por criterio de coordenadas.

Los datos de entrada son los parámetros $\{G, \mathbf{b}, \mathbf{c}\}$ del problema de programación lineal: $\max_x f = \mathbf{c}^T \mathbf{x}$ sujeto a $G\mathbf{x} \leq \mathbf{b}$. Luego se calcula el cúmulo de restricciones \mathcal{C} y de aquí se determinan el conjunto de restricciones superfluas \mathcal{F}_{ang} y la vecindad angular \mathcal{N} .

Datos de entrada:

Modelo de programación lineal original en R^n :

- G : Matriz de coeficientes
 \mathbf{b} : Vector de recursos
 \mathbf{c} : Vector de costos
 \mathbf{e} : Cúmulo de restricciones

Modelo de programación lineal en R^n reducido en restricciones por \mathcal{F}_{ang} :

- \mathcal{N} : Vecindad angular determinada por el mecanismo de la tabla 2
 $m = |\mathcal{N}|$: Cardinalidad de \mathcal{N}
 $G_{\mathcal{N}}$: Matriz de coeficientes ajustada a \mathcal{N}
 $\mathbf{b}_{\mathcal{N}}$: Vector de recursos ajustado a \mathcal{N}
 \mathbf{c} : Vector de costos sin cambio

Parámetros y variables del diccionario: valores de inicio

- $\mathbf{c}_N = \mathbf{c}$: Vector coeficiente de variables no básicas $n \times 1$
 $\mathbf{c}_B = \mathbf{0}$: Vector coeficiente de variables básicas $m \times 1$
 $N = G_{\mathcal{N}}$: Matriz no básica $m \times n$
 $B = I$: Matriz básica $m \times m$
 $B^{-1} = I$: Matriz inversa de B $m \times m$
 $\mathbf{b} = \mathbf{b}_{\mathcal{N}}$: Vector de recursos $m \times 1$
 $L_N = -\infty$: Vector $n \times 1$ límite inferior de \mathbf{x}_N
 $U_N = +\infty$: Vector $n \times 1$ límite superior de \mathbf{x}_N
 $L_B = \mathbf{0}$: Vector $m \times 1$ límite inferior de \mathbf{x}_B
 $U_B = +\infty$: Vector $m \times 1$ límite superior de \mathbf{x}_B
 $\mathbf{p}_N = [1, \dots, n]$: Vector $1 \times n$ apuntador a elementos de \mathbf{x}_N
 $\mathbf{p}_B = [(n + 1), \dots, (n + m)]$: Vector $1 \times m$ apuntador a elementos de \mathbf{x}_B
 $\mathbf{x}_N = \mathbf{0}$: Vector de variables no básicas $n \times 1$
 $\mathbf{x}_B = \mathbf{b}$: Vector de variables básicas $m \times 1$
 $f = 0$: función objetivo escalar
 $\mathbf{x} = \mathbf{0}$: Vector de variables de decisión $1 \times n$
 $\mathbf{u} = \mathbf{b}^T$: Vector de variables de holgura $1 \times m$

Parámetros auxiliares: valores de inicio

- $Canon = [1, \dots, n]$: Vector $1 \times n$ de restricciones canónicas
 $Canon_cont = 0$: Contador de restricciones canónicas
 t_max : Máximo número de iteraciones propuesto
 $t^\# = n$: Iteración en que se alcanza el vértice frontera $\mathbf{x}^\#$
 $t = 0$: Iteración del algoritmo simplex KKT
 $Optimo = false$: Condición óptima de la solución $[\mathbf{x}, \mathbf{u}]$
 $\mathcal{N}^t = \mathcal{N}$: Vecindad angular en la iteración $t > t^\#$
 $k^+ = 0$: Índice de x_{Nk} para su crecimiento $x_{Nk} + \xi$
 $k^- = 0$: Índice de x_{Nk} para su decrecimiento $x_{Nk} - \xi$

Tabla 15: Datos de entrada al algoritmo simplex KKT

Mientras ($t < t_{max}$) ejecutar lo que sigue:

1.- Procedimiento de optimalidad de variables libres y vértice frontera (Tabla 4, Cap. 3):

$$[k^+, k^-] = \text{Optimalidad}(\mathbf{c}_N, \mathbf{c}_B, N, B^{-1}, \mathbf{x}_N, \mathbf{L}_N, \text{Canon}, t^\#, k^+, k^-)$$

Si ($k^+ = 0$) y ($k^- = 0$) terminar, la solución corriente $[\mathbf{x}, \mathbf{u}]$ es óptima

2.- Procedimiento de factibilidad (Tabla 10):

$$[k, r, \{\xi_j, j = 1, \dots, m\}, \xi_r] = \text{Factibilidad}(k^+, k^-, \mathbf{L}_N, \mathbf{U}_N, \mathbf{x}_B, \mathbf{L}_B, \mathbf{p}_B, N, B^{-1}, m)$$

3.- Mecanismo de identificación de restricciones superfluas (Tabla 13):

$$[\mathcal{F}_{restr}^t, \mathcal{F}_{var}^t] = \text{Identificación}(\mathcal{N}, \mathcal{F}_{ang}, \xi, t, t^\#, \mathbf{L}_B, \mathbf{p}_B)$$

4.- Procedimiento de pivoteo $k \leftrightarrow r$ y actualización (Tablas 16 y 17):

$$[\mathbf{x}_N, \mathbf{x}_B, N, B, \mathbf{c}_N, \mathbf{c}_B, \mathbf{L}_N, \mathbf{L}_B, \mathbf{U}_N, \mathbf{U}_B, \mathbf{p}_N, \mathbf{p}_B, f, \mathbf{x}, \mathbf{u}] \\ = \text{Pivoteo}(k, r, \mathbf{x}_N, \mathbf{x}_B, N, B, \mathbf{c}_N, \mathbf{c}_B, \mathbf{L}_N, \mathbf{L}_B, \mathbf{U}_N, \mathbf{U}_B, \mathbf{p}_N, \mathbf{p}_B)$$

5.- Mecanismo de supresión de restricciones superfluas y de variables y parámetros superfluos (Tablas 14 y 15):

$$[\mathcal{N}^t, \mathbf{x}_B, N, B, B^{-1}, \mathbf{b}, \mathbf{c}_B, \mathbf{L}_B, \mathbf{U}_B, \mathbf{p}_B] \\ = \text{Supresión}(\mathcal{F}_{restr}^t, \mathcal{F}_{var}^t, \mathcal{N}^t, t, t^\#, m, \mathbf{x}_B, N, B, \mathbf{b}, \mathbf{c}_B, \mathbf{L}_B, \mathbf{U}_B, \mathbf{p}_B)$$

$$t = t + 1$$

Tabla 16: Algoritmo simplex KKT

En el algoritmo simplex KKT desarrollado en este trabajo de tesis, las variables y parámetros del problema de programación lineal evolucionan en cada iteración. La evolución sucede por actualización, pivoteo, supresión de restricciones, y supresión de variables y parámetros superfluos. La supresión es el proceso que modifica la dimensión del problema de programación lineal, como se muestra en la tabla 20.

| | \mathbf{c}_N | \mathbf{c}_B | N | B | \mathbf{b} | \mathbf{L}_N | \mathbf{U}_N | \mathbf{L}_B | \mathbf{U}_B | \mathbf{p}_N | \mathbf{p}_B | \mathbf{x}_N | \mathbf{x}_B | \mathcal{N}^t |
|-----------|----------------|----------------|-----|-----|--------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|
| Pivoteo | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Supresión | | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | ✓ | | ✓ | ✓ |

Tabla 17: Parámetros y variables sujetos a pivoteo y supresión

En el segundo renglón de la tabla 20 se marcan las seis parejas de la expresión (42) de cantidades no básica \leftrightarrow básica cuyos componentes $k \leftrightarrow r$ son pivoteados. En el tercer renglón se marcan las variables y parámetros que están sujetos a supresión. Todos los parámetros y variables básicos están sujetos a supresión. La matriz N siendo matriz no básica, se le suprimen solo sus renglones porque éstos son parte de una restricción. Solamente a la matriz B se le suprimen columnas y renglones, columnas por ser matriz básica y renglones porque éstos son parte de una restricción. La vecindad angular \mathcal{N}^t está sujeta a supresión porque contabiliza las restricciones.

De la tabla 20 notamos que hay cantidades que están sujetas tanto a pivoteo como a supresión. Esta condición parece una dificultad porque el algoritmo simplex KKT de la tabla 19 primero realiza el pivoteo y después la supresión. La supresión de una columna "j" de B y el pivoteo $k \leftrightarrow r$, entre la k-ésima columna entrante de la matriz no básica N y la r-ésima columna saliente de la matriz básica B, no se traslapan. Solo se suprimen columnas de B pero no de N, por lo que el índice k no se toca en la supresión. Además, de B se suprime una j-ésima columna que corresponde a una coordenada $\xi_j < 0$, pero no se suprime una columna que corresponda a una coordenada $\xi_r > 0$, por lo que el índice r tampoco se toca en la supresión. Por tanto, el pivoteo y la supresión son operaciones que se afectan una a la otra.

5.7 Conclusiones del capítulo

En este capítulo se ha desarrollado un nuevo algoritmo para el cálculo eficiente del vértice óptimo x^* de un problema de programación lineal a partir de un punto interior del polítopo de soluciones factibles. El algoritmo ha sido denotado como *algoritmo simplex KKT* porque está fundamentado en el concepto de cono Karush-Kuhn-Tucker (KKT). El *cono KKT* es preservado en cada iteración del algoritmo mientras se suprimen las restricciones identificadas como superfluas.

El desarrollo del *algoritmo simplex KKT* ha estado motivado porque se ha observado una gran cantidad de restricciones no atadas en los modelos de programación lineal para representar la sujeción por manos robóticas. La supresión de una buena cantidad de restricciones no atadas reduce los tiempos de cálculo lo que es deseable en aplicaciones en tiempo real, como lo es la sujeción por manos robóticas. Además, los trabajos reportados a la fecha en identificación de restricciones no atadas se han limitado a proveer sólo ideas y conceptos fundamentales.

El *algoritmo simplex KKT* es resultado de la adaptación del *algoritmo simplex revisado* reportado por [Chvatal (1983)] al que se le han insertado procedimientos computacionales que reducen la complejidad de un problema de programación lineal, por lo que el algoritmo posee los atributos siguientes:

- (1) Contiene el algoritmo de vértice frontera el cual incluye los mecanismos de variables libres y de cálculo del vértice frontera. Este algoritmo posee las propiedades siguientes: (i) acepta variables libres en forma directa sin usar métodos de conversión a variables no negativas y (ii) calcula en "n" iteraciones un vértice en la frontera del polítopo de soluciones factibles a partir del origen de R^n en su interior, con la propiedad adicional de que el vértice calculado está angularmente cercano al vértice óptimo. Esta propiedad se explica porque el algoritmo usa la misma función objetivo del problema de programación que se está resolviendo. Otra propiedad interesante del algoritmo de vértice frontera es que es similar al algoritmo de punto interior. Ambos parten de un punto interior y generan trayectorias dirigidas por la función objetivo que conducen a la solución óptima. Un atractivo del algoritmo de vértice frontera es que se aproxima a la solución óptima en solo "n" iteraciones, mientras que el algoritmo de

punto interior resuelve problemas no lineales en cada iteración. Dadas sus propiedades, el algoritmo de vértice frontera es aplicable a la solución del problema de análisis de sujeción por manos robóticas, el cual se modela mediante un problema de programación lineal de variables libres con el origen de R^n en el interior del polítopo de soluciones factibles.

- (2) Contiene dos mecanismos de identificación de restricciones superfluas, es decir restricciones redundantes y restricciones no atadas a la solución óptima. La identificación está fundamentada en una clase de polarización de restricciones causada por las siguientes cantidades polarizadoras: el gradiente ∇f de la función objetivo y el cono Karush-Kuhn-Tucker. La polarización fuerza a las restricciones a agruparse en dos clases: (i) el conjunto de restricciones candidatas a delimitar el *cono KKT* y (ii) el conjunto de restricciones que contiene algunas restricciones superfluas. Las entidades sujetas a la polarización son todas las restricciones de un problema de programación lineal. El fenómeno de polarización sucede a través de cantidades derivadas de las restricciones que son sensibles a la polarización: (i) $\text{Cos}(\theta_j), j = 1, \dots, m$ y (ii) $\xi_j, j = 1, \dots, m$. Donde θ_j es el ángulo entre el gradiente ∇h_j de la j -ésima restricción y ∇f , y ξ_j es la coordenada KKT. La polarización de restricciones por medio de $\text{Cos}(\theta_j)$ dio como resultado el mecanismo de identificación de restricciones superfluas por criterio angular. La polarización por medio de la coordenada ξ_j nos condujo al mecanismo de identificación de restricciones superflua por criterio de coordenadas.
- (3) En ambos mecanismos de identificación de restricciones superfluas suceden proyecciones ortogonales de vectores en el espacio R^n sobre el vector gradiente de la función objetivo. En el mecanismo por criterio angular se proyectan las gradientes de cada restricción. La información clave de estas proyecciones es el coseno del ángulo entre el vector a proyectar y el gradiente de la función objetivo. En el mecanismo por criterio de coordenadas se usan las coordenadas generadas por el *algoritmo simplex KKT*, las cuales son equivalentes a ciertas coordenadas sobre un eje alineado con el gradiente de la función objetivo. Las coordenadas sobre este eje son las magnitudes de las proyecciones ortogonales de vectores correspondientes a los puntos de cruce del eje de coordenadas del algoritmo simplex con los hiperplanos de cada restricción. Aunque estas dos proyecciones son la idea clave de la polarización de restricciones, no son del todo confiables por si mismas en la identificación de restricciones superfluas, es decir, no garantizan la condición de superfluidad. Por esta razón cada uno de estos mecanismos tiene un procedimiento adicional para corregir sus deficiencias. El mecanismo por criterio angular tiene el procedimiento de restitución de restricciones atadas, y el mecanismo por criterio de coordenadas tiene un procedimiento de selección de las coordenadas negativas más cercanas al origen.
- (4) Una ventaja sobresaliente del algoritmo simplex KKT sobre el algoritmo simplex revisado es su eficiencia de cálculo. La eficiencia se debe a la supresión de restricciones identificadas como superfluas lo que repercute en la reducción de la dimensión del problema original de programación lineal. Es decir, la complejidad del problema es reducida drásticamente durante la ejecución del algoritmo.

- (5) El *algoritmo simplex KKT* es de aplicación inmediata a la solución del problema de sujeción por manos robóticas, el cual será abordado en el capítulo 6. En análisis de la sujeción por manos robóticas modelada por medio de un problema de programación lineal se ha observado la existencia de una gran cantidad de restricciones superfluas las cuales son causantes de los tiempos prolongados de cálculo en esta clase de problemas. Las restricciones superfluas encontradas en estos problemas son eminentemente no atadas y rara vez redundantes. Los mecanismos desarrollados en esta tesis solo polarizan todas las restricciones, identifica y suprime las redundantes por criterio angular o de coordenadas, es decir no distinguen entre redundantes y no atadas.

Capítulo 5

Resultados Experimentales

Una mano robótica, como la mostrada en la figura 1, es una cadena cinemática cerrada que involucra contactos múltiples con fricción entre objetos pasivos y estructuras mecánicas activamente coordinadas. La robustez, destreza y autonomía son atributos deseables en la mano robótica. Aunque esta tecnología ha sido investigada por tres décadas aproximadamente los atributos referidos están todavía en desarrollo en el laboratorio. No hay actualmente aplicaciones industriales aunque sus aplicaciones potenciales se encuentran en las áreas de teleoperación y telecirugía.



Fig. 1: The Shadow Dextrous Hand
© Copyright Shadow Robot Company Ltd.

Las manos robóticas humanoides se han desarrollado principalmente para imitar las funciones de la mano humana y el área que las estudia se conoce como manipulación diestra [Venkataraman, S. T (1994), Mason, M. (2001)]. El análisis de la sujeción es parte de la manipulación diestra y consiste en decidir si la sujeción definida por la geometría de los contactos entre los dedos y el objeto a sujetar es estable. El problema de la sujeción es fundamental porque tiene que resolverse antes de la manipulación. Una dificultad enfrentada por la tecnología de la mano robótica es la falta de algoritmos eficientes para la manipulación diestra en general, incluyendo el análisis de la sujeción.

En este capítulo se plantea la hipótesis de que el *algoritmo simplex KKT* puede resolver eficientemente el problema de análisis de la sujeción por manos robóticas cuando es modelado por un problema de programación lineal.

El problema de programación lineal que modela la sujeción ha sido desarrollado por Ding Dan (2002) y posee las siguientes características: (i) el problema está definido en el espacio R^6 , (ii) el origen de R^6 es punto interior del polígono de soluciones factibles lo cual implica que el problema es de variables libres, (iii) un número razonable de restricciones usado en el problema es del orden de 400, por lo que aparecerían 400-6 restricciones superfluas, es decir, que la solución del problema estaría determinada por sólo 6 restricciones y que las 394 restantes están de más.

Por otro lado, el *algoritmo simplex KKT* posee las cualidades siguientes: (i) resuelve problemas de programación lineal en el espacio R^n en los que el número de restricciones es mayor que " n ", (ii) maneja variables libres, (iii) el origen de R^n puede ser punto interior del polígono de soluciones factibles, (iv) migra desde el punto interior hasta un vértice en la frontera del polígono en solo " n " iteraciones, (v) suprime masivamente restricciones superfluas. El desarrollo del *algoritmo simplex KKT* estuvo motivado por el intento de resolver eficientemente el problema de sujeción por manos robóticas, por lo que el algoritmo está diseñado a la medida del problema de análisis de la sujeción.

El contenido de este capítulo se presenta como sigue. En la sección 1 se deriva un modelo matemático de la sujeción por manos robóticas. En la sección 2 se presentan tres criterios para el análisis de la sujeción, basados en modelos lineales, de los cuales se elige el de programación lineal en la sección 3. En la sección 4 se presentan los resultados numéricos para evaluar el desempeño del algoritmo simplex KKT en la solución del problema de programación lineal de sujeción por manos robóticas. En la sección 4 se presentan las conclusiones del capítulo.

5.1. Modelo de sujeción por manos robóticas

Considere la sujeción de un objeto regular por medio de cinco dedos cuyo esquema simplificado se muestra en la figura 2.

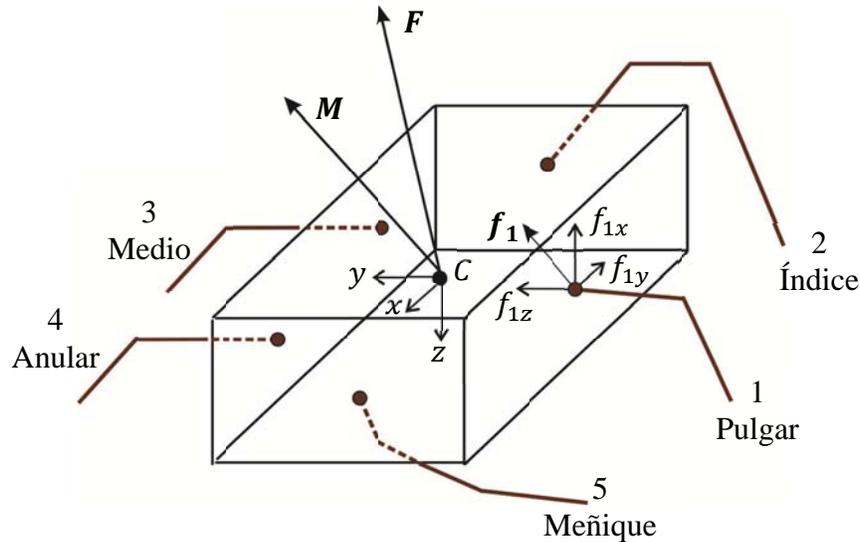


Fig. 2: Esquema simplificado de sujeción

Consideremos las suposiciones siguientes para la sujeción:

1. El objeto es rígido. Es decir, dos de sus partículas no cambian su posición relativa aún bajo la influencia de fuerzas o torques.
2. Los puntos de contacto entre los dedos y el objeto sujetado están fijos. Es decir, la fuerza $f_i = [f_{ix}, f_{iy}, f_{iz}]^T$ que aplica el i –ésimo dedo nunca se hace cero.
3. Se transmiten fuerzas únicamente desde cada dedo hacia el objeto y cada dedo no aplica torque durante la sujeción.
4. Se supone un coeficiente de fricción μ en los puntos de contacto.
5. El movimiento del objeto se supone nulo para lo operación de sujeción. Es decir, la fuerza resultante F y el momento resultante M son nulos.
6. Todas las cantidades vectoriales están referidas al centro geométrico C .

La fuerza f_i genera dos reacciones en el objeto: un movimiento translacional causado por la misma f_i , y un movimiento rotacional causado por el momento angular $r_i \times f_i$ relativo al centro geométrico C del objeto. La composición de estas dos reacciones causadas se conoce como wrench en el i –ésimo punto de contacto de dimensión 6×1 , el cual se denota como w_i y se expresa como sigue:

$$\mathbf{w}_i = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{r}_i \times \mathbf{f}_i \end{bmatrix} = \begin{bmatrix} I \mathbf{f}_i \\ S_i \mathbf{f}_i \end{bmatrix} = \begin{bmatrix} I \\ S_i \end{bmatrix} \mathbf{f}_i, i = 1, 2, \dots, 5 \quad (1)$$

Donde I es la matriz identidad de dimensión 3x3 y S_i es una matriz producto cruz asociada al vector de posición $\mathbf{r}_i = [r_{ix}, r_{iy}, r_{iz}]^T$ del i -ésimo punto de contacto:

$$S_i = \begin{bmatrix} 0 & -r_{iz} & r_{iy} \\ r_{iz} & 0 & -r_{ix} \\ -r_{iy} & r_{ix} & 0 \end{bmatrix}$$

El efecto resultante de los wrenches de los 5 dedos es la fuerza \mathbf{F} y el momento angular \mathbf{M} mostrados en la figura 2, el cual se expresa como sigue:

$$\mathbf{w}_1 + \mathbf{w}_2 + \dots + \mathbf{w}_5 = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \quad (2)$$

Asumiendo que la fuerza \mathbf{F} y el momento angular \mathbf{M} son nulos el objeto quedaría inmóvil si se garantizara que la sujeción es estable. La ecuación (2) se convierte entonces en la ecuación de sujeción con condición de estabilidad pendiente de ser verificada:

$$\mathbf{w}_1 + \mathbf{w}_2 + \dots + \mathbf{w}_5 = \mathbf{0} \quad (3)$$

La fuerza $\mathbf{f}_i = [f_{ix}, f_{iy}, f_{iz}]^T$ en cada wrench \mathbf{w}_i debe satisfacer la restricción impuesta por la fricción durante el contacto del dedo con el objeto, la cual se expresa como sigue:

$$\sqrt{f_{ix}^2 + f_{iy}^2} \leq \mu f_{iz} \text{ donde } \mu > 0 \text{ y } f_{iz} > 0, i = 1, 2, \dots, 5 \quad (4)$$

Los componentes f_{ix}, f_{iy} están dispuestos sobre la superficie de contacto mientras que f_{iz} es perpendicular a la superficie como se muestra en la figura 2.

5.1.1 Formulación del problema de sujeción

Dada la ecuación sujeción:

$$\mathbf{w}_1 + \mathbf{w}_2 + \dots + \mathbf{w}_5 = \mathbf{0} \quad (5)$$

sujeta a:

$$\sqrt{f_{ix}^2 + f_{iy}^2} \leq \mu f_{iz}, \mu > 0 \text{ y } f_{iz} > 0, i = 1, 2, \dots, 5$$

el problema de análisis de la sujeción consiste en determinar si la sujeción es estable o no es estable. La estabilidad de la sujeción dependerá de que se satisfaga o no la restricción no lineal impuesta por la fricción y del posicionamiento de los contactos manifestado en cada wrench \mathbf{w}_i . Notamos que el problema de sujeción (5) es no lineal.

Dado que $\mu > 0$ y $f_{iz} > 0$, la restricción dada en (5) define un cono en R^3 , llamado cono de fricción en el i -ésimo contacto, el cual se denota como $\mathcal{K}_f^{(i)}$ y se define como sigue:

$$\mathcal{K}_f^{(i)} = \{(f_{ix}, f_{iy}, f_{iz}) \mid \sqrt{f_{ix}^2 + f_{iy}^2} \leq \mu f_{iz}, \mu > 0, f_{iz} > 0\} \quad (6)$$

Este cono puede ser linealizado por medio de un número m_d de vectores $\mathbf{d}_j^{(i)}$ de discretización. El cono (6) linealizado se expresa entonces como la combinación lineal positiva siguiente:

$$\mathbf{f}_i = \sum_{j=1}^{m_d} \lambda_j^{(i)} \mathbf{d}_j^{(i)}; \lambda_j^{(i)} > 0, i = 1, \dots, 5 \quad (7)$$

5.1.2 Linealización del problema de sujeción

El número m_d de vectores de discretización puede ir desde 10 hasta 1000. Un número razonable es $m_d = 100$, el cual usaremos como ejemplo. Cuando la combinación lineal positiva (7) se sustituye en cada wrench \mathbf{w}_i de (5) usando $m_d = 100$, se tendrán 500 sumandos. Después de sustituir y reducir se obtiene una combinación lineal positiva con la estructura siguiente:

$$\ell_1 \mathbf{g}_1 + \ell_2 \mathbf{g}_2 + \dots + \ell_{500} \mathbf{g}_{500} = \mathbf{0} \quad (8)$$

Donde $\mathbf{g}_j, j = 1, \dots, 500$ son vectores unitarios en el espacio R^6 los cuales reciben el nombre wrenches primitivos de contacto, y $\ell_j > 0$ son coeficientes que satisfacen:

$$\sum_{j=1}^{500} \ell_j = 1, \ell_j > 0 \quad (9)$$

Notemos que (9) es una ecuación de convexidad por lo que (8) es una combinación lineal convexa. Notemos también que la combinación lineal convexa (8) con 500 sumandos es equivalente al problema no lineal dado en (5) de solo 5 sumandos. El problema no lineal se ha convertido en un problema lineal de dimensión expandida. La no linealidad ha sido eliminada en (5) por lo que el modelo linealizado (8) se conoce como modelo de sujeción de cono embebido.

Toda vez que los puntos de contacto han sido elegidos, los vectores $\mathbf{g}_j, j = 1, \dots, 500$ quedan determinados. Por tanto, la información de la estabilidad de la sujeción queda concentrada en los coeficientes $\ell_j, j = 1, \dots, 500$. Si los coeficientes satisfacen la condición de convexidad (9) la sujeción es estable. Si los coeficientes no satisfacen esta condición, la sujeción es no estable y habrá que reposicionar los puntos de contacto, lo cual implica generar otro conjunto de vectores \mathbf{g}_j para intentar satisfacer la condición. Notar que la verificación de la estabilidad de sujeción solo requiere que $\ell_j > 0$ para todo j , la expresión de convexidad (9) es solo una forma de explicar la estabilidad.

5.2 Criterios de análisis de la sujeción

5.2.1 Verificación de la estabilidad por positividad

Dado el sistema de ecuaciones lineales dado en (8) y que se reescribe enseguida:

$$\ell_1 \mathbf{g}_1 + \ell_2 \mathbf{g}_2 + \dots + \ell_{500} \mathbf{g}_{500} = \mathbf{0} \quad (10)$$

sujeto a $\ell_j > 0$

donde $\mathbf{g}_j, j = 1, \dots, 500$ son wrenches primitivos de contacto en el espacio R^6 y $\ell_j, j = 1, \dots, 500$ son coeficientes estrictamente positivos, el problema de análisis de la sujeción consiste en determinar si la sujeción es o no es estable por medio de la verificación de la condición de positividad de los coeficientes $\ell_j > 0$.

5.2.2 Verificación de la estabilidad por cascos convexos

Los wrenches primitivos de contacto $\mathbf{g}_j, j = 1, \dots, 500$ son vectores unitarios y definen un casco convexo $H(\mathcal{G}), \mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{500}\}$ en el espacio R^6 . Luego, decir que los coeficientes $\ell_j, j = 1, \dots, 500$ son estrictamente positivos es equivalente a decir que el origen de R^6 sea punto interior del casco convexo. Por tanto, otra forma de verificar la estabilidad de la sujeción es por verificación de que el origen de R^6 sea punto interior $H(\mathcal{G})$.

5.2.3 Verificación de la estabilidad por rayo emisor

El hecho de que el origen $\mathbf{0}$ de R^6 sea punto interior del casco convexo $H(\mathcal{V})$ puede ser verificado por medio del concepto de rayo emisor. Para ejemplificar estos conceptos consideremos un punto \mathbf{p} en el interior de $H(\mathcal{G}) \subset R^2$ como se muestra en la figura 3.

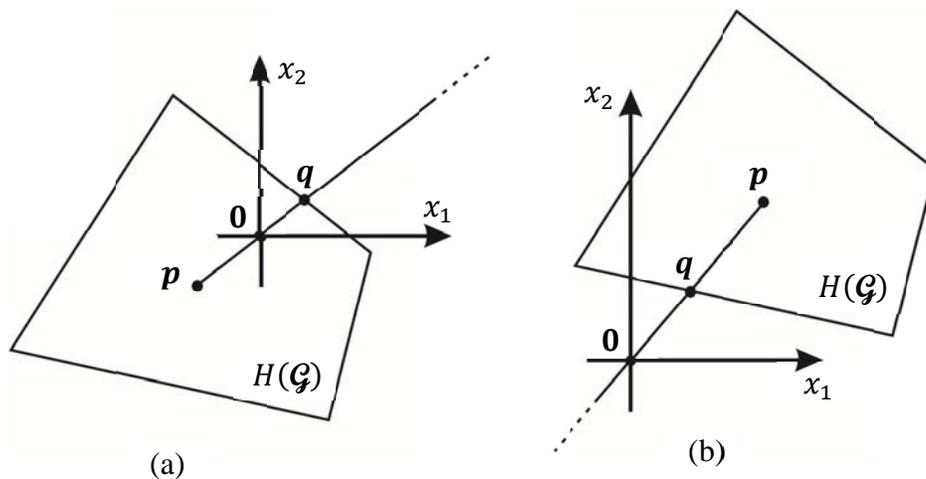


Fig. 3: Criterio de rayo emisor

Consideremos un rayo que parte de \mathbf{p} y se hace pasar por el origen de R^2 como se muestra en la figura 3. Dado que \mathbf{p} está en el interior del casco convexo $H(\mathcal{G})$, el rayo debe intersectar la frontera del casco en un único punto \mathbf{q} . Luego el criterio del rayo emisor consiste en verificar las relaciones siguientes:

$$\begin{aligned} \text{Si } \overline{\mathbf{pq}} > \overline{\mathbf{p0}} \text{ entonces } \mathbf{0} \in H(\mathcal{V}), \text{ lo que implica sujeción estable} \\ \text{Si } \overline{\mathbf{pq}} < \overline{\mathbf{p0}} \text{ entonces } \mathbf{0} \notin H(\mathcal{V}), \text{ lo que implica sujeción no estable} \end{aligned} \quad (11)$$

Donde $\overline{\mathbf{pq}}$ es el segmento desde el punto \mathbf{p} hasta el punto \mathbf{q} , y $\overline{\mathbf{p0}}$ es el segmento desde el punto \mathbf{p} hasta el origen de R^2 .

5.2.4 Verificación del criterio del rayo emisor

La comprobación del criterio del rayo emisor especificado en (11) solo necesita del conocimiento de los puntos \mathbf{p} y \mathbf{q} .

Notamos que el punto interior \mathbf{p} del casco convexo $H(\mathcal{G})$ se puede calcular por medio de la relación de convexidad de los vectores de $\mathcal{G}=\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{500}\}$:

$$\mathbf{p} = \sum_{j=1}^{500} a_j \mathbf{g}_j \quad (12)$$

Donde:

$$\sum_{j=1}^{500} a_j = 1, \quad a_j = \frac{1}{500} \quad (13)$$

Es decir, el cálculo del punto interior \mathbf{p} por medio de (12) es un simple promedio de los vectores de $\mathcal{G}=\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{500}\}$ que definen el casco convexo $H(\mathcal{G})$.

Sin embargo, el cálculo del punto \mathbf{q} de cruce del rayo emisor con la frontera del casco convexo $H(\mathcal{G})$ no es tan simple. Este punto puede ser calculado por medio de la solución de un problema de programación lineal, como será explicado más adelante. El problema de programación lineal necesitado será establecido a través de la dualidad entre cascos convexos y polítopos convexos.

5.3 Análisis de la sujeción por programación lineal

El problema de cálculo del punto q de cruce del rayo emisor con la frontera del casco convexo $H(\mathcal{G})$ puede ser transformado a un problema de programación lineal por medio de la dualidad entre cascos convexos y polítopos convexos.

5.3.1 Transformación de dualidad

Se obtiene un polítopo convexo $\mathcal{P}(\mathcal{G})$ a partir del casco convexo $H(\mathcal{G})$. Consideremos un wrench primitivo de contacto $\mathbf{g}_k \in \mathcal{G}$. El vector unitario $\mathbf{g}_k \in R^6$ define un hiperplano $h(\mathbf{g}_k)$ por medio de la siguiente transformación, conocida como transformación de dualidad:

$$h(\mathbf{g}_k) = \{\mathbf{x} \in R^6 \mid \mathbf{g}_k^T \mathbf{x} = 1\}, k \in [1, \dots, 500] \quad (14)$$

Todo vector \mathbf{g}_k es ortogonal a su hiperplano $h(\mathbf{g}_k)$. Asociado al hiperplano $h(\mathbf{g}_k)$ existe un semiespacio $E(\mathbf{g}_k)$ definido como:

$$E(\mathbf{g}_k) = \{\mathbf{x} \in R^6 \mid \mathbf{g}_k^T \mathbf{x} \leq 1\}, k \in [1, \dots, 500] \quad (15)$$

El polítopo $\mathcal{P}(\mathcal{G})$ se determina por tanto como la intersección de todos estos semiespacios:

$$\mathcal{P}(\mathcal{G}) = E_1 \cap E_2 \cap \dots \cap E_{500} \quad (16)$$

Sea \mathbf{v}_j el j -ésimo vértice del polítopo $\mathcal{P}(\mathcal{G})$ y sea $h(\mathbf{v}_j)$ su correspondiente hiperplano.

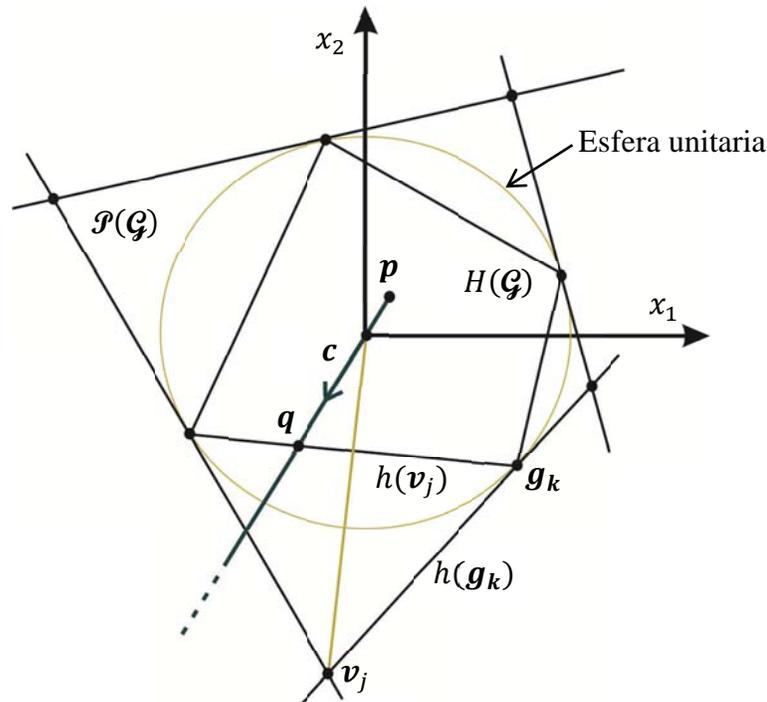


Fig. 4: Dualidad entre cascos convexos $H(\mathcal{G})$ y polítopos convexos $\mathcal{P}(\mathcal{G})$

5.3.2 Especificación de un hiperplano respecto de la esfera unitaria

Una propiedad de la transformación de dualidad (14) es que el vector unitario \mathbf{g} es mapeado al hiperplano $h(\mathbf{g})$ posicionado a una distancia $1/||\mathbf{g}||$ en las formas siguientes:

- (i) Si \mathbf{g} cae en la superficie de la esfera entonces $h(\mathbf{g})$ es tangente a la esfera.
- (ii) Si \mathbf{g} cae en el interior de la esfera entonces $h(\mathbf{g})$ cae fuera de la esfera.
- (iii) Si \mathbf{g} cae en el exterior de la esfera entonces $h(\mathbf{g})$ cae dentro de la esfera.

Por ejemplo, el punto \mathbf{g}_k de la figura 4 cae en la superficie de la esfera y su hiperplano $h(\mathbf{g}_k)$ es por tanto tangente a la esfera. Sin embargo, el punto \mathbf{v}_j que cae en el exterior de la esfera posiciona su hiperplano $h(\mathbf{v}_j)$ a una distancia $1/||\mathbf{v}_j||$ cortando por tanto el interior de la esfera como se muestra en la misma figura 4.

5.3.3 Propiedad de dualidad entre cascos y polítopos

Un vértice \mathbf{g}_k del casco convexo $H(\mathcal{C})$ define un hiperplano $h(\mathbf{g}_k)$ del polítopo $\mathcal{P}(\mathcal{C})$, o bien un hiperplano $h(\mathbf{g}_k)$ define un vértice \mathbf{g}_k de $H(\mathcal{C})$ como se muestra en la figura 4. Por el contrario, un vértice \mathbf{v}_j del polítopo $\mathcal{P}(\mathcal{C})$ define un hiperplano (cara) $h(\mathbf{v}_j)$ del casco convexo $H(\mathcal{C})$, o bien, un hiperplano $h(\mathbf{v}_j)$ de $H(\mathcal{C})$ define un vértice \mathbf{v}_j de $\mathcal{P}(\mathcal{C})$ como se muestra en la misma figura 4. Es decir, el polítopo $\mathcal{P}(\mathcal{C})$ es dual al casco convexo $H(\mathcal{C})$ por medio de la transformación de dualidad dada en (14).

5.3.4 Función objetivo para un polítopo

Dado un rayo emisor que parte de un punto \mathbf{p} que pasa por el origen de R^n , es posible definir un vector \mathbf{c} en la dirección del rayo como se muestra en la figura 4 para el caso de R^2 . El vector \mathbf{c} puede ser propuesto como el siguiente vector unitario:

$$\mathbf{c} = \frac{-\mathbf{p}}{||\mathbf{p}||} \quad (17)$$

De la figura se puede notar que el punto \mathbf{q} requerido para verificar el criterio de rayo emisor definido en (11) se puede calcular por la intersección del rayo con el hiperplano $h(\mathbf{v}_j)$, y que el plano $h(\mathbf{v}_j)$ puede ser determinado por el vértice \mathbf{v}_j del polítopo $\mathcal{P}(\mathcal{C})$. Por tanto, el problema pendiente de ser resuelto es calcular un vértice \mathbf{v}_j de $\mathcal{P}(\mathcal{C})$ por medio de un problema de programación lineal cuyo vector de costos es el vector unitario \mathbf{c} .

5.3.5 Problema de programación lineal

Dado el vector unitario \mathbf{c} y el polítopo $\mathcal{P}(\mathcal{C})$, un vértice \mathbf{v}_j del polítopo puede ser calculado por medio del siguiente problema de programación lineal:

$$\underset{x}{\text{maximizar}} f(x) = \mathbf{c}^T \mathbf{x}$$

$$\text{subjeta a: } \mathbf{g}_k^T \mathbf{x} \leq 1, k \in [1, \dots, 500] \quad (18)$$

Donde el vector de costos \mathbf{c} está dado en (17) y las restricciones están dadas por los semiespacios de $\mathcal{P}(\mathcal{G})$ definidos en (15). Dado que el lado derecho de las desigualdades es estrictamente positivo, el origen de R^6 es punto interior del casco convexo $H(\mathcal{G})$ y por tanto del polígono convexo $\mathcal{P}(\mathcal{G})$. Por consiguiente las variables de decisión del problema son variables libres.

5.3.6 Solución al problema de sujeción

El problema de programación lineal calcula el vértice \mathbf{v}_j del polígono $\mathcal{P}(\mathcal{G})$. Por la transformación de dualidad (14), el hiperplano correspondiente \mathbf{v}_j está dado por

$$h(\mathbf{v}_j) = \{\mathbf{x} \in R^6 \mid \mathbf{v}_j^T \mathbf{x} = 1\} \quad (19)$$

Como se aprecia en la figura 4, el rayo emisor está especificado por el vector de costos \mathbf{c} , es decir, el rayo y el vector \mathbf{c} están en la misma dirección. Por tanto el punto de intersección \mathbf{q} del rayo con el hiperplano $h(\mathbf{v}_j)$ se calcula por medio de la siguiente ecuación:

$$\mathbf{q} = \frac{1}{\mathbf{v}_j^T \mathbf{c}} \mathbf{c} \quad (20)$$

Comprobemos esta fórmula: dado que \mathbf{q} pertenece al hiperplano $h(\mathbf{v}_j)$ se satisface:

$$\mathbf{v}_j^T \mathbf{q} = 1 \quad (21)$$

Porque el vector \mathbf{q} está en la dirección del vector unitario \mathbf{c} se tiene:

$$\mathbf{q} = \|\mathbf{q}\| \mathbf{c} \quad (22)$$

Sustituyendo (22) en (21) se obtiene:

$$\mathbf{v}_j^T \|\mathbf{q}\| \mathbf{c} = 1 \quad (23)$$

De donde se tiene:

$$\|\mathbf{q}\| = \frac{1}{\mathbf{v}_j^T \mathbf{c}} \quad (24)$$

Sustituyendo (24) en (22), se tiene el resultado (20).

El problema de sujeción por manos robóticas se resuelve por medio del siguiente algoritmo, el cual incluye la solución del problema de programación lineal.

DATOS DE ENTRADA: El conjunto de wrenches primitivos de contacto:

$$\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{500}\}$$

CALCULOS: Ejecutar la siguiente secuencia de pasos:

1. Calcular el punto interior del casco convexo $H(\mathcal{V})$ por medio de (12):

$$\mathbf{p} = \frac{1}{500} \sum_{j=1}^{500} \mathbf{g}_j$$

2. Dado \mathbf{p} calcular el vector de costos \mathbf{c} por medio de (17):

$$\mathbf{c} = \frac{-\mathbf{p}}{\|\mathbf{p}\|}$$

3. Dados \mathbf{c} y el conjunto de wrenches primitivos de contacto \mathcal{G} , calcular el vértice \mathbf{v}_j del polígono $\mathcal{P}(\mathcal{G})$ por medio del problema de programación lineal (18):

$$\begin{aligned} & \underset{\mathbf{x}}{\text{maximizar}} f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ & \text{subjeta a: } \mathbf{g}_k^T \mathbf{x} \leq 1, k \in [1, \dots, 500] \end{aligned}$$

4. Dado el vértice \mathbf{v}_j especificar su correspondiente hiperplano $h(\mathbf{v}_j)$ para calcular el punto de intersección \mathbf{q} del rayo emisor con el hiperplano $h(\mathbf{v}_j)$ por medio de (20):

$$\mathbf{q} = \frac{1}{\mathbf{v}_j^T \mathbf{c}} \mathbf{c}$$

5. Dados los puntos \mathbf{p} y \mathbf{q} , el criterio del rayo emisor dado en (11) puede ser probado para determinar si la sujeción es o no es estable:

Si $\overline{\mathbf{p}\mathbf{q}} > \overline{\mathbf{p}\mathbf{0}}$ entonces $\mathbf{0} \in H(\mathcal{V})$, lo que implica sujeción estable

Si $\overline{\mathbf{p}\mathbf{q}} < \overline{\mathbf{p}\mathbf{0}}$ entonces $\mathbf{0} \notin H(\mathcal{V})$, lo que implica sujeción no estable

Tabla 1: Algoritmo para análisis de la sujeción por manos robóticas.

5.4 Resultados experimentales

Como se puede apreciar del algoritmo de la tabla 1 para análisis de la sujeción por manos robóticas, los pasos 1 y 2 son muy simples pero necesarios. En el paso 5 se resuelve el problema de análisis de la sujeción, el cual es también relativamente simple. Sin embargo, el paso 3 es el más costoso en tiempo de cálculo porque resuelve un problema de programación lineal. Por tanto, en este capítulo sólo se resolverá el problema de programación lineal que modela la sujeción por manos robóticas. En este capítulo se presentarán resultados acerca de la cualidad de supresión masiva de restricciones superfluas del algoritmo simplex KKT, así como la evolución de la reducción de la complejidad del problema de programación lineal mientras el algoritmo calcula vértices en la frontera del polígono de soluciones factibles.

5.4.1 Condiciones de experimentación

Los experimentos que se presentarán aquí se realizan con las siguientes suposiciones:

- (i) La sujeción es por medio de cuatro dedos.
- (ii) Se usa un coeficiente de fricción $\mu = 0.5$ en cada punto de contacto.
- (iii) Un cono de fricción en cada punto de contacto puede ser discretizado con 5 vectores y hasta 200 vectores de linealización.
- (iv) Los problemas de programación lineal de la clase (18) pueden tener por tanto desde 20 hasta 800 restricciones:

$$\begin{aligned} & \underset{x}{\text{maximizar}} \quad f(x) = \mathbf{c}^T x \\ & \text{subjeto a: } \mathbf{g}_k^T x \leq 1, k \in [1, \dots, 800] \end{aligned}$$

- (v) La combinación de las dos clases básicas de supresiones de restricciones superfluas producen una tercera. Las tres clases son las siguientes:
 - o Supresión por criterio angular
 - o Supresión por criterio de coordenadas
 - o Supresión por la acción conjunta del criterio angular y del criterio de coordenadas.
- (vi) Se eligen cuatro problemas de programación lineal con número de restricciones de 80, 200, 400 y 800 para evaluar el desempeño del algoritmo simplex KKT.

Comparación del desempeño del algoritmo simplex KKT con el algoritmo simplex clásico.

5.4.2 Supresión de restricciones superfluas por criterio angular

En la figura 5 se presenta la evolución de la dimensión " m " en cada iteración " t " de cada uno de los cuatro problemas de programación lineal causada por la supresión de restricciones superfluas basada en el criterio angular. Estos problemas son de la clase de problemas de programación lineal dados en (18) los cuales modelan la sujeción por manos robóticas. Los problemas tienen diferente número de restricciones, diferentes funciones objetivo y diferentes restricciones. Como se muestra en la figura, los números de restricciones con los que inicia cada problema son: 80, 200, 400 y 800.

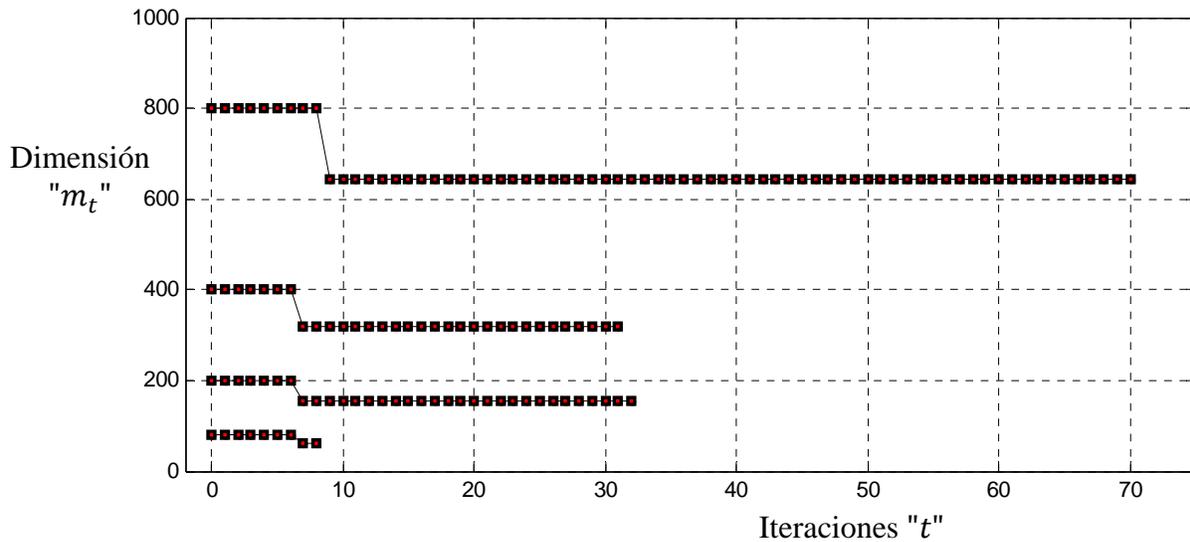


Fig. 5: Evolución de la dimensión de cuatro problemas de programación lineal debido a la supresión por criterio angular.

Como se observa en la figura, la supresión por criterio angular sucede una sola vez. La supresión sucede en la iteración $t = t^\# + 1$ con $t^\# = 6$ en los problemas con dimensión inicial de 80, 200 y 400. Sin embargo, en el problema con dimensión inicial de 800 la supresión sucede en $t = t^\# + 2$.

En la figura se observa una reducción en la dimensión de cada problema del 20% aproximadamente. Esto significa que en el problema con 800 restricciones se suprimieron 160 restricciones superfluas aproximadamente. Como se verá en las otras dos clases de supresión de restricciones, estos números son mejorables.

5.4.3 Supresión de restricciones superfluas por criterio de coordenadas

En la figura 6 se muestra la evolución de las dimensiones de los cuatro problemas de programación lineal. Como se observa en la figura, el cambio en la dimensión de cada problema es debido a la aplicación del criterio de coordenadas a partir de la iteración $t = t^\# + 1$ para los problemas con número de restricciones 80, 200 y 400, y $t = t^\# + 2$ para el problema con número de iteraciones de 800.

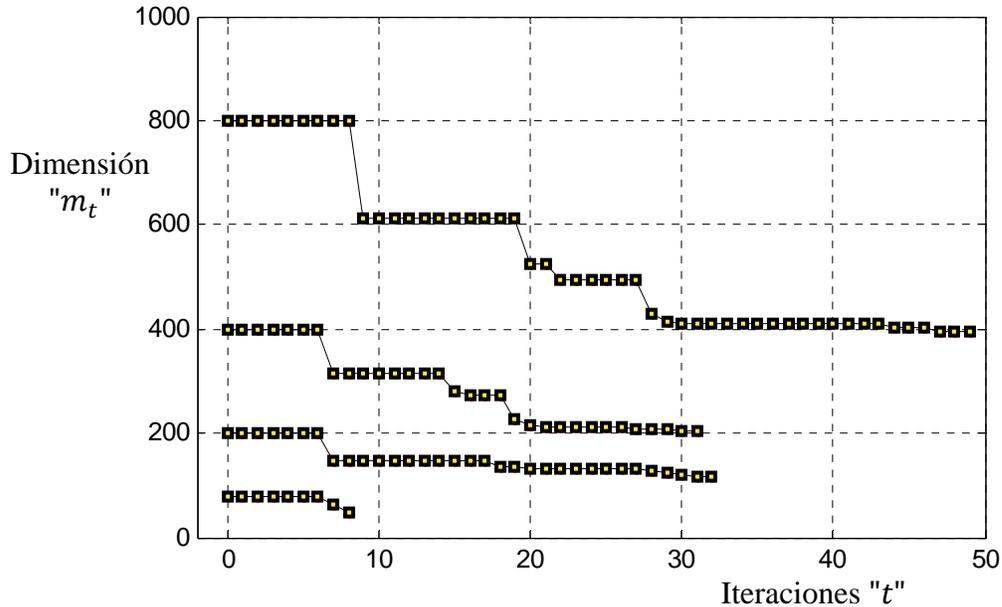


Fig. 6: Evolución de la dimensión de cuatro problemas de programación lineal debido a la supresión por criterio de coordenadas.

Los cambios de dimensión mostrados en la figura 6 no son uniformes. Esto significa que solo en algunos vértices del polítopo se suprimen restricciones. En la figura se observa una reducción global en la dimensión de cada problema del 50% aproximadamente. Esto significa que en el problema con 800 restricciones se suprimieron alrededor de 400 restricciones superfluas.

Se puede decir por tanto que el método de supresión de restricciones superfluas basado en el criterio de coordenadas suprime masivamente esta clase de restricciones.

5.4.4 Supresión de restricciones superfluas por criterios angular y de coordenadas

Analicemos ahora la acción conjunta del criterio angular y del criterio de coordenadas.

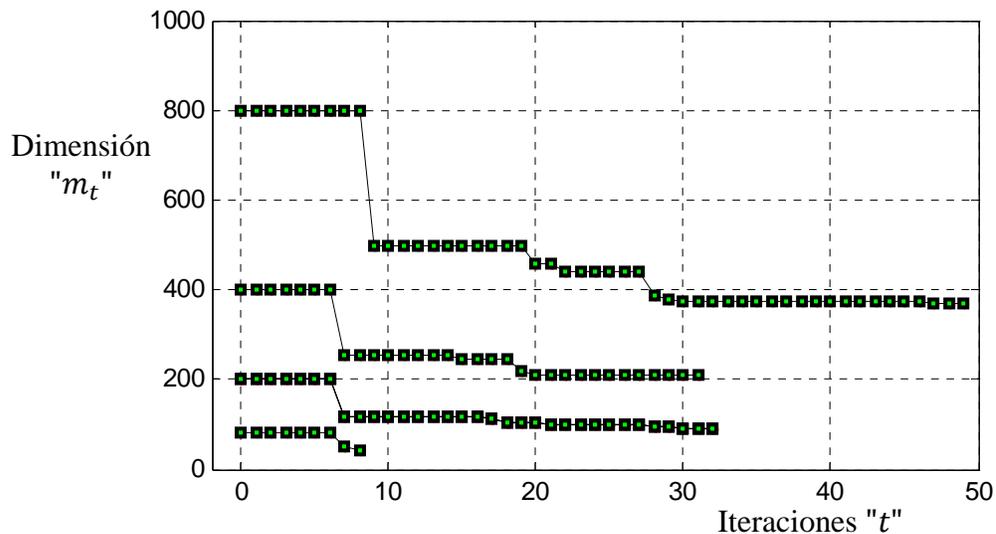


Fig. 7: Evolución de la dimensión de cuatro problemas de programación lineal debido a la acción conjunta del criterio angular y del criterio de coordenadas.

El comportamiento de las dimensiones al final de cada curva de la figura 7 es similar al comportamiento de las de la figura 6. Por cierto, el número de iteraciones en que se alcanza la solución óptima es exactamente el mismo en las dos figuras. Sin embargo, hay una diferencia muy importante entre ambas figuras: el cambio substancial en la dimensión de cada problema a partir de $t = t^\# + 1$. El problema de 800 restricciones iniciales observa un cambio de 800 a 600 aproximadamente en la figura 6, mientras que en la figura 7 el cambio es de 800 a 500 aproximadamente.

Estos cambios bruscos son importantes porque significa que el problema ha reducido su dimensión de tal forma que el algoritmo tendrá menos esfuerzo computacional a partir de $t = t^\# + 1$, como sucede en la figura 7.

5.4.5 Comparación de métodos de supresión con el problema de 400 restricciones

Se toma como ejemplo el problema de 400 restricciones para contrastar las cualidades de los tres métodos de supresión de restricciones superfluas como se muestra en la figura 8.

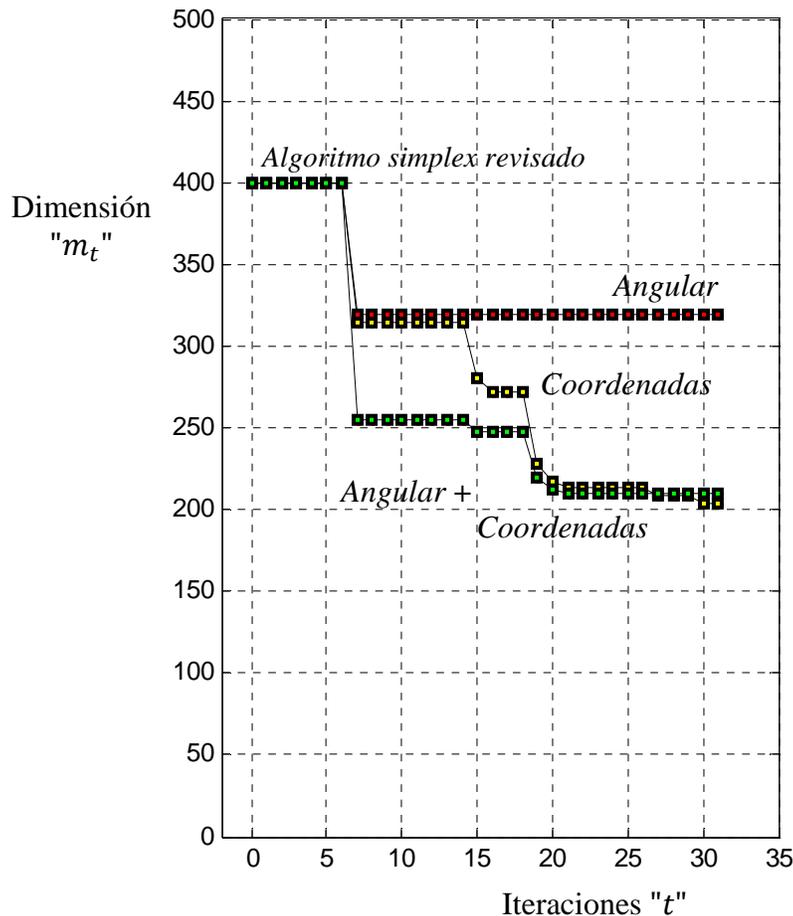


Fig. 8: Comparación de los cambios de dimensión por medio de los tres criterios de supresión de restricciones superfluas respecto del algoritmo simplex revisado

El método por criterio angular realiza un buen trabajo, sin embargo éste es mejorado por el método del criterio de coordenadas. El método de criterios combinados parece similar al método por criterio de coordenadas. El combinado sin embargo, es superior porque realiza una supresión masiva en la iteración $t = t^{\#} + 1$ lo que reduce la carga computacional del algoritmo a partir de esta iteración.

5.4.6 Análisis de la complejidad del problema de 400 restricciones

La evolución de las dimensiones de los cuatro problemas de programación analizados previamente, dan cuenta de las cualidades del algoritmo simplex KKT. Sin embargo, la mejor medida del desempeño del algoritmo es la evolución de complejidad. La medida normalizada de complejidad ρ_t del algoritmo por la supresión de restricciones se precisará más adelante en esta sección.

En la figura 9 se contrastan las complejidades ρ_t del algoritmo simplex KKT en la solución de un problema de programación lineal de 400 restricciones cuyas correspondientes curvas son distintas por causa de las tres clases básicas de supresión de restricciones superfluas: por criterio angular, por criterio de coordenadas y por acción conjunta de éstas.

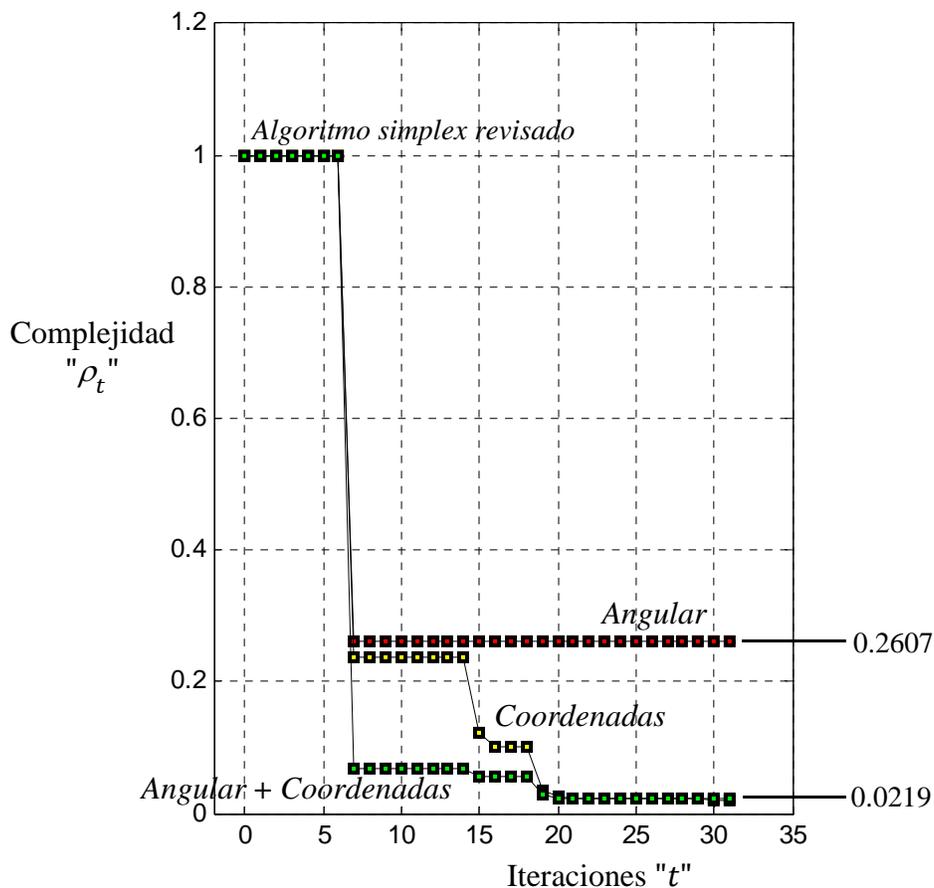


Fig. 9: Complejidades normalizadas de los tres criterios de supresión de restricciones superfluas comparada con la del algoritmo simplex revisado

Como se aprecia en la figura 9, el método de supresión por criterio angular es muy bueno por la reducción final de la complejidad al 26.07%. Sin embargo, el método basado en la acción conjunta de los criterios, angular y de coordenadas, es superior por el drástico decrecimiento de la complejidad al 2.19% debido a la correspondiente reducción de la dimensión del problema.

El *algoritmo simplex KKT* que usa el método de supresión angular y por coordenadas reduce drásticamente la complejidad del problema de programación bajo estudio. La figura 9 muestra la complejidad normalizada ρ_t producida por el *algoritmo simplex KKT* en comparación con la complejidad del algoritmo simplex revisado.

La complejidad ρ_t normalizada del *algoritmo simplex KKT* se definirá como el cociente de la complejidad C_t del problema de programación lineal determinada por el número " m_t " de restricciones que varía en cada iteración " t " y la complejidad C_m del problema de programación lineal determinada por el número fijo " m " de restricciones:

$$\rho_t = \frac{C_t}{C_m} \quad (25)$$

Donde las complejidades C_t y C_m están dadas por sus definiciones usuales [Hillier (2001)]:

$$\begin{aligned} C_t &= \binom{m_t + 6}{m_t} \\ C_m &= \binom{m + 6}{m} \end{aligned} \quad (26)$$

Para el problema de programación lineal de 400 restricciones que se está analizando se tiene $m = 400$ y $m_t < 400$. Los valores de m_t se muestran en las curvas de la figura 8.

5.5 Conclusiones del capítulo

En este capítulo se comprobó que el *algoritmo simplex KKT* resuelve eficientemente el problema de análisis de la sujeción por manos robóticas cuando esta es modelada por medio de un problema de programación lineal, lo cual se explica por el hecho de que su eficiencia computacional es superior a la del *simplex revisado*.

Se revisó el procedimiento de sujeción por manos robóticas presentado por Ding Dan (2002) para análisis de la sujeción, el cual incluye un problema de programación lineal en uno de sus pasos. Solo se expusieron resultados numéricos acerca de la solución del problema de programación lineal porque es el paso computacionalmente más costoso del procedimiento de sujeción.

El desempeño del *algoritmo simplex KKT* en cuanto a complejidad es superior al del algoritmo *simplex revisado*. Esta conclusión se apoya en dos resultados numéricos: (1) la reducción de la dimensión del problema de programación lineal debido a la supresión masiva de restricciones superfluas, y (2) por la reducción de la complejidad del problema de programación lineal causada por la reducción en dimensión. El desempeño del *algoritmo simplex KKT* se ve mejor reflejado por medio de la reducción en la complejidad que por la reducción en dimensión porque la complejidad es la que mide el esfuerzo de cómputo para procesar la matriz básica de dimensión variable.

El atractivo principal del *algoritmo simplex KKT* está basado en dos procedimientos: la identificación y la supresión de restricciones superfluas. La identificación por criterio de coordenadas no agrega costo de cómputo porque solo se revisan los signos de coordenadas que de por si son generadas por el algoritmo *simplex revisado*. La identificación por criterio angular agrega un costo de cómputo en la revisión de los signos de los componentes de un arreglo unidimensional, arreglo que se calcula antes de iniciar la ejecución del algoritmo. La supresión de restricciones sin embargo, tiene un costo computacional porque se suprimen renglones y/o columnas de matrices y vectores. Dado que la supresión mueve bloques de datos en matrices y vectores, los tiempos consumidos por la supresión podrían ser reducidos por medio de una tecnología apropiada de cómputo.

Los métodos de supresión desarrollados en esta tesis no distingue entre restricciones redundantes y no atadas, simplemente suprimen las restricciones superfluas que son identificadas. Los métodos de supresión presentados aquí contrastan con otros métodos como el del criterio del renglón que solo identifica restricciones redundantes.

Capítulo 6

Conclusiones Generales y Trabajos Futuros

6.1 Conclusiones generales

In this paper a methodology based on the identification of sets of superfluous constraints is proposed in order to improve the efficiency of the Simplex algorithm in solving the linear programming problem that models the grasping analysis with four fingers of a robotic hand in 3D space. The proposed methodology is summarized in the *KKT Simplex method* which was derived by aggregating to the *revised Simplex method* the following two methods:

En esta tesis se ha desarrollado una nueva metodología basada en la identificación de restricciones superfluas para mejorar la eficiencia del algoritmo Simplex en la solución de problemas de programación lineal que modelan la sujeción por cuatro dedos de una mano robótica en el espacio 3D.

Los dos métodos de identificación de restricciones superfluas desarrollados en esta tesis están fundamentados en el concepto clave de polarización de las restricciones. El fenómeno de polarización es establecido por el gradiente de la función objetivo y por el *cono Karush-Kuhn-Tucker*. El cono es preservado en cada acción de supresión de restricciones mientras las restricciones son organizadas a través de variables derivadas de ellas que son sensibles a la polarización.

El desempeño del algoritmo simplex KKT se manifestó en la solución de una clase de problemas de programación para la sujeción por manos robóticas. Como es reportado en los resultados numéricos, el algoritmo logra una reducción substancial en la complejidad del problema en virtud de la supresión de restricciones superfluas.

En los mecanismos de identificación de restricciones superfluas suceden proyecciones ortogonales de vectores en el espacio R^n sobre el vector gradiente de la función objetivo. En el mecanismo por criterio angular se proyectan los gradientes de cada restricción. La información clave de estas proyecciones es el coseno del ángulo entre el vector a proyectar y el gradiente de la función objetivo. En el mecanismo por criterio de coordenadas se usan las coordenadas generadas por el *algoritmo simplex KKT*, las cuales son equivalentes a ciertas coordenadas sobre un eje alineado con el gradiente de la función objetivo. Las coordenadas sobre este eje son las magnitudes de las proyecciones ortogonales de vectores correspondientes a los puntos de cruce del eje de coordenadas del algoritmo simplex con los hiperplanos de cada restricción. Aunque estas dos proyecciones son la idea clave de la polarización de restricciones, no son del todo confiables por si mismas en la identificación de restricciones superfluas, es decir, no garantizan la condición de superfluidad. Por esta

razón cada uno de estos mecanismos tiene un procedimiento adicional para corregir sus deficiencias. El mecanismo por criterio angular tiene el procedimiento de restitución de restricciones atadas, y el mecanismo por criterio de coordenadas tiene un procedimiento de selección de las coordenadas negativas más cercanas al origen.

Se asegura que la metodología propuesta es también robusta en la identificación de restricciones superfluas, y por tanto en la solución del problema de programación lineal que modela la sujeción por manos robóticas. Si la identificación estuviera basada solo en los signos negativos de las coordenadas, la reducción del tamaño del problema alcanzaría el 80%. Pero la preservación del *cono KKT* produce una reducción al 50%. Es decir, la eficiencia es sacrificada para asegurar robustez.

Una propiedad sobresaliente del *método Simplex KKT* es la preservación del *cono KKT* a pesar de la supresión de restricciones. Esta preservación sugiere nombrar el método como: *KKT Simplex method*. Ya que la identificación y supresión de restricciones superfluas están basadas en las coordenadas negativas, muchas restricciones superfluas persisten entre los conjuntos residuales de restricciones debido a que estas no pueden ser detectadas como superfluas como resultado de los signos positivos de sus coordenadas.

6.2 Trabajos futuros

Se encontró que el método de identificación de restricciones superfluas por criterio de coordenadas tiene que ser sintonizado para seleccionar el conjunto correcto de restricciones candidatas a ser suprimidas, aquellas cuyas coordenadas negativas más están cercanas al origen. Debido a la supresión de restricciones de coordenadas negativas en un vértice, habrá menos restricciones de coordenadas negativas que suprimir en los siguientes vértices. Por tanto, el conjunto de restricciones a ser suprimidas será menor, por lo que la sintonización debe ser adaptada convenientemente en cada iteración. Por tanto, como trabajo futuro se propone explorar un método de sintonización adaptativo que garantice un máximo número de restricciones identificadas como superfluas.

El algoritmo simplex KKT diseñado en esta tesis parte de un punto interior del polígono convexo de soluciones factibles para arribar a un vértice en la frontera del polígono en " n " iteraciones simplex. Una propiedad interesante del vértice frontera es que fue calculado en la dirección del gradiente de la función objetivo por el vértice fue seleccionado en cierta vecindad del vértice óptimo, como fue expuesto en el ejemplo del capítulo 3. A partir del vértice frontera comienza la identificación de las restricciones superfluas por lo que los cálculos se aceleran debido a la reducción de la dimensión del problema. Por tanto, el *algoritmo simplex KKT* podría ser comparado con el *método de punto interior*, el cual también parte del interior del polígono siguiendo una trayectoria por su interior hacia el óptimo. Se propone como trabajo futuro explorar esta comparación.

Se propone también explorar el comportamiento del polítopo de soluciones factibles durante la eliminación de restricciones. Es decir, existe la posibilidad de convertirse en un conjunto no acotado.

Se propone exportar el concepto de polarización de restricciones a los métodos heurísticos. Es decir, las restricciones clasificadas en candidatas a delimitar el cono Karush-Kuhn-Tucker y las candidatas a ser suprimidas podrían ser procesadas convenientemente por los algoritmos heurísticos.

Las proyecciones ortogonales de ciertas cantidades vectoriales sobre un eje alineado con el gradiente de la función objetivo resultaron útiles porque estas proyecciones facilitaron la polarización de restricciones en dos clases. Se propone explorar aún más la relación de estas proyecciones con los conos de factibilidad y de mejora que tienen que ver con la solución óptima.

Referencias

- Andersen, E. D. and Knud D. (1995).** *Andersen. Presolving in Linear Programming.* Mathematical Programming 71 (1995) 221-245.
- Avis, David, Konei Fukuda, and Stefano Picozzi (2002).** *On Canonical Representations of Convex Polyhedra.* Mathematical Software, ICMS 2002, Ed. A. Cohen, X-S Gao, N. Takayama, World Scientific, pp.350-360 (2002).
- Avis, David and Luc Devroye (1994).** *Estimating the Number of Vertices of a Polyhedron.* School of Computer Science, McGill University, Montréal, Canada. (1994).
- Bagnara, Roberto, Patricia M. Hill, and Enea Zaffanella (2005).** *Not Necessarily Closed Convex Polyhedra and the Double Description Method.* Department of Mathematics, University of Parma, Italy. (2005).
- Bazaraa, Mokhtar S., Hanif D. Sherali and C. M. Shety (2006).** *Nonlinear Programming,* John Wiley and Sons, 2006.
- Beraldi, Patrizia and Andrzej Ruszczyński (2002).** *The Probabilistic Set-Covering Problem.* Operations Research; Nov/Dec 2002; Vol. 50, No. 6.
- Boneh, Arnon (1983).** *A Probabilistic Algorithm Identifying Redundancy by a Random feasible Point Generator,* in Karwan, Mark H., Vahid Lotfi, Jan Telgen, Stanley Zionts. Redundancy in Mathematical Programming. Springer-Verlag, 1983.
- Boot, J. C. G. (1962).** “*On Trivial and Binding Constraints in Programming Problems*”, Management Sciences, 1962.
- Borgwardt, Karl Heinz (1987).** *The Simplex Method, A Probabilistic Analysis.* Springer-Verlag, 1987.
- Brearley, A. L., G. Mitra, and H. P. Williams (1975).** “*Analysis of Mathematical Programming Problems Prior to Applying the Simplex Algorithm*”, Mathematical Programming 8, 1975, 54-58.
- Bremner, David Dylan (1997).** *On the Complexity of Vertex and Facet Enumeration.* Thesis for Ph. D. Degree, 1997. School of Computer Science, McGill University, Montréal, Canada.
- Caron, Richard and Trim Taynor (2006).** *A General Framework for the Analysis of Sets of Constraints.* Department of Mathematics and Statistics, University of Windsor, Windsor, Ontario, Canada. December 2006.
- Caron, Richard J., Harvey J. Greenberg, and Allen G. Holder (2002).** *Analytic centers and repelling inequalities,* European Journal of Operational Research 143 (2002) 268–290.

- Chen, J. (1996).** *Computacional Geometry: Methods and Applications*. Computer Science Department, Texas A & M University, 1996.
- Chvátal, Vasek (1983).** *Linear Programming*. W. H. Freeman and Company, 1983.
- Dantzig, George B. (1963)** *Linear Programming and Extensions*. Princeton University, 1963.
- Dantzig, George B. and Mukund N. Thapa (2003).** *Linear Programming, 2: Theory and Extensions*. Springer, 2003.
- Ding Dan (2001).** *Grasp Synthesis of Multi-Fingered Robotic Hands*. A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Doctor in Philosophy in Mechanical and Automation Engineering. The Chinese University of Hong Kong, 2001.
- Dula, J. H. (1994).** “*Geometry of Optimal Value Functions with Applications to Redundancy in Linear Programming*”, *Journal of Optimization Theory and Applications*. Vol. 81, No. 1, April 1994.
- Dyer, M. E. and L. G. Proll. (1977)** *An Algorithm for determining all Extreme Points of a Convex Polytope*. *Mathematical Programming* 12 (1977).
- Enge, Andreas and Petra Huhn (1998)** “*Initialization of the Simplex Algorithm*”, *Society of Industrial and applied mathematics*, 1998.
- Feng, Jinghua (1999).** *Redundancy in nonlinear systems: A set covering approach*. A thesis for the Degree of Master of Science at the University of Windsor. Windsor, Ontario, Canada, 1999.
- Ferris, Michael C. (2007).** *Linear programming with Matlab*. Society of Industrial and applied mathematics and The Mathematical Programming Society, 2007.
- Floudas C. A. and Pardos M. Pardalos. (2009)** *Encyclopedia of Optimization*. Springer 2009.
- Fukuda, Komei. (2004).** *Frequently Asked Questions in Polyhedral Computation*. Swiss Federal Institute of Technology, Lausanne and Zurich, Switzerland, Version June 18, 2004.
- Fukuda, Komei and Alain Prodon. (1996).** *Double Description Method Revisited*. Institute for Operations Research, Zürich, Switzerland. (1996).
- Fukuda, Komei (2004).** “*Frequently Asked Questions in Polyhedral Computation*”, Swiss Federal Institute of Technology, Lausanne and Zurich, Switzerland, Version June 18, 2004.
- Gal, Tomas (1995).** *Postoptimal Analyses, Parametric Programming, and Related Topics*, Walter de Gruyter, 1995.
- Gale, David (1997).** *Linear Programming and the Simplex Method*. University of California, Berkely. (1997).

Gebert , Jürgen Ritche, and Gunter M. Ziegler (1997). Basic Properties of Convex Polytopes. CRC Prss, 1997.

GUANFENG, LIU (2003). *Synthesis of Dexterous Manipulation by Multifingered Robot Hand*. A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Doctor in Philosophy in Mechanical and Automation Engineering. The Chinese University of Hong Kong, 2003.

Hadwiger, H. (1998). *Lo Antiguo y lo Nuevo Acerca de los Conjuntos Convexos*. Sociedad Matemática Mexicana, 1998.

Halima M. El-Khatib (1997). “A probabilistic Method for Cleaning Contaminated Systems of Linear Inequalities”, A thesis for the Degree of Master of Science at the University of Windsor. Windsor, Ontario, Canada, 1997.

Hillier, Frederick S. and Gerald J. Liberman. *Investigación de Operaciones*. McGraw-Hill,2001

Howard, W. Stamps and Vijaj Kumar (1996). *On the Stability of Grasped Objects*. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 12, NO. 6, DECEMBER 1996.

Jones, C.N. and Manfred Morari (2008). “The Double Description Method for the Approximation of Explicit MPC Control Laws”, Proceedings of the 47th IEEE Conference on Decision and Control Cancun, Mexico, Dec. 9-11, 2008.

Khachiyan, Leonid, Endre Boros, Konrad Borys, Khaled Elbassioni, and Vladimir Gurvich (2006). *Generating all vertices of a polyhedron is hard*. SODA '06, January 22-26, Miami, FL, 2006 SIAM ISBN 0-89871-605-5/06/01.

Lassez, Jean-Louis and Michael J. Maher (1989). On Fourier's Algorithm for Linear Arithmetic Constraints. IBM Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, U.S.A. (1989).

Lassez, Jean-Louis (2006). *Parametric Queries, Linear Constraints and Variable Elimination*. IBM Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, U.S.A. (2006).

Luenberger, G. David (2008). *Linear and Nonlinear Programming*. Addison Wesley, 2008.

Liu Y-H. (1999). *Qualitative Test and Force Optimization of 3-D Frictional Form-Closure Grasps Using LP*. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 15, NO. 1, FEBRUARY 1999.

Karwan, Mark H., VahidLotfi, Jan Telgen and Stanley Zionts (1983). *Redundancy in Mathematical Programming*. Springer-Verlag, 1983.

- Mason, M. (2001).** *Mechanics of Robotic Manipulation*. The MIT Press, 2001
- Mason, M. and J. K. Salisbury, Jr. (1985).** *Robot hand and Mechanics of Manipulation*. The MIT Press, 1985
- Messina, V., S. Moody, and G. Mitra. (1992).** *Analysis and Reduction of Linear Programming Models*, TR/07/93, University of Milan, Italy, October 1992.
- Montana, David J. (1992).** *Contact Stability for Two-Fingered Grasps*. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 8, NO. 4, AUGUST 1992.
- Montana, David J. (1991).** *The Condition for Control Grasp Stability*. Proceeding of 1991 IEEE International Conference on Robotics and Automation, Sacramento, California. April 1991.
- Mulmuley, Ketan (1994).** *Computational Geometry*. Prentice Hall, 1994.
- Murray, Zexiang Li, S. Sh. Sastry (1990).** *Robotic Manipulation*. CRC Press, 1994.
- Murphy, Frederic H., Edward A. Stohr, and Pai-chun M.A. (1990).** “*Composition Rules for Building Linear Programming Models*”, School of Business, Temple University, Philadelphia, Pennsylvania, 1990.
- Nemirovski, Arkadi (200).** *Lectures on Modern Convex Optimization*. Department ISYE, Georgia Institute of Technology, 2005.
- Neuman, E. (1996).** *Linear Programming with Matlab*. Department of Mathematics Southern Illinois University at Carbondale, 1996.
- Nguyen, Van-Duc (1986).** *Constructing Force-Closure Grasp*. Artificial Intelligence Laboratory Massachusetts Institute Of Technology. Cambridge, MA 02139, (1986).
- Nguyen, Van-Duc (1988).** *Constructing Stable Force-Closure Grasp*. Artificial Intelligence Laboratory Massachusetts Institute Of Technology. Cambridge, MA 02139, (1988).
- Okamura, Allison M., Niels Smaby and Mark R. Cutkosky (2000).** *An Overview of Dexterous Manipulation*. IEEE International Conference on Robotics and Automation, p.255-262, 2000.
- Papadimitriou, Christos H. and Kenneth Steiglitz (1998).** *Combinatorial Optimization Algorithms and Complexity*, Dover Publications, 1998.
- Paulos, Eric (1993).** *Trends in Grasping*. Department of Electrical Engineering and Computer Science. University of California, 1993.

Pollard, Nancy S. and Tomás Lozano-Pérez (1990). *Grasp Stability and Feasibility*. IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, May 1990, pp.1581-1587.

Papadimitriou, Christos H. and Kenneth Steiglitz (1998). *Combinatorial Optimization Algorithms and Complexity*. Dover Publications, 1998.

Parker, Mark Richard (1995). *A Set Covering Approach to Infeasibility Analysis of Linear Programming Problems and Related Issues*. Thesis for Ph. D. Degree, 1995. University of Colorado at Denver.

Paul, R. P. (1981). *Robot Manipulators*. The MIT Press, 1981

Rosen, Kenneth H. (2004). *Handbook of Discrete and Computational Geometry*. CRC Press, 2004.

Sedra, A. S. y Kenneth C. Smith (2006). *Circuitos Microelectrónicos*. McGraw Hill, 2006

Taha, Hamdy A. (2003). *Investigación de Operaciones*. Prentice hall, 2003

Telgen, Jan (1986). “*Identifying Constraints and Implicit Equalities*”, Management Sciences, 1986.

Thompson, Gerald L., Fred M.Tonge and Stanley Zionts (1966). “*Techniques for Removing Nonbinding Constraints and Extraneous Variables from Linear Programming Problems*”, Management Science, Vol. 12, No. 7, Series A, Sciences (Mar., 1966), pp. 588-608.

Tomlin, J. A. and J. S. Welch. *Formal Optimization of some Reduced Linear Programming Problems*. Mathematical Programming, 27 (1983).

Trigos, Federico (2002). *Sobre la Reducción de Modelos de Programación Lineal*. Métodos Numéricos en Ingeniería y Ciencias Aplicadas, 2002.

Trinkle, Jeffry C. (1992). *A Quantitative Test For Form Closure Grasps*. Proceedings of the 1992 IEEEESJ International Conference on Intelligent Robots and Systems Raleigh, NC July 7-10, 1992

Trinkle, Jeffry C. (1992). *On the Stability and Instantaneous Velocity of Grasped Frictionless Objects*. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION. VOL 8. :-10. S. OCTOBER 1992.

Trinkle, Jeffry C. R. C. Ram, and O. A. Farahat (1992). *Dexterous Manipulation Planning and Execution of an Enveloped Slippery Workpiece*. Texas A & M University.

Vasko, P J Knolle and D. S. Spiegel (2005). *An Empirical Study of Hybrid Genetic Algorithms for the Set Covering Problem*. Journal of the Operational Research Society, 2005, 56.

Venkataraman, S. T. and T. Iberal (1990). *Dexterous Robot Hand*. Springer-Verlag, 1990.

Ye, Yinyu (1997). *Interior Point Algorithms: Theory and Analysis*. John Wiley and Sons, 1997.

Ziegler, Gunter M. (1997). *Basic Properties of Convex Polytopes*. CRC Press, 1997.