

UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA
DIVISIÓN DE ESTUDIOS DE POSGRADO

PROGRAMACIÓN CÓNCAVA CON
RESTRICCIONES LINEALES Y VARIABLES
ENTERAS

T E S I S

QUE PARA OBTENER EL GRADO DE
DOCTORA EN INGENIERÍA
(INVESTIGACIÓN DE OPERACIONES)

P R E S E N T A :

IDALIA FLORES DE LA MOTA

DIRECTOR: DR. HAMDY A. TAHA

MÉXICO D. F.

OCTUBRE DE 1998

TESIS CON
FALLA DE ORIGEN

01185

1
24

265848



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A la memoria de Graciela y Gonzalo

*He buscado en la poesía
respuestas a los arcanos de la vida.*

*He buscado en la ciencia
respuestas a la inteligencia humana*

*He encontrado en la poesía
la expresión del corazón humano.*

*He encontrado en la ciencia
la poesía de la naturaleza.*

AGRADECIMIENTOS

"Las criaturas tan pequeñas como nosotros sólo podemos soportar la inmensidad por medio del amor".

Carl Sagan

"Si la única oración que uno dijera en toda su vida fuera *gracias*, sería suficiente".

Meister Eckhart

Quiero agradecer a las siguientes personas e instituciones que hicieron posible que éste trabajo se llegara a buen término:

A mi director de tesis Dr. Hamdy Taha por darme todo su apoyo, comentarios, sugerencias, y sobre todo por abrirme las puertas de su casa y de su corazón.

A los miembros del jurado: Dr. Felipe Ochoa, Dr. Felipe Lara, Dr. Miguel Angel Gutierrez y Dr. Francisco Venegas, por sus valiosos comentarios.

A la Facultad de Ingeniería de la UNAM, mi alma mater, por darme las facilidades para poder desarrollar este trabajo.

A PADEP por darme el apoyo para poder viajar a la Universidad de Arkansas en los Estados Unidos y poder trabajar con mi director de tesis.

A CONACYT por darme su apoyo a través del programa de Cátedras Patrimoniales y porque siempre ha estado abierto a apoyar a profesores y estudiantes de Doctorado.

A la Universidad de Arkansas y el jefe de Ingeniería Industrial, Dr. Eric Malstrom, por darme todas las facilidades y la infraestructura necesaria para trabajar en el Departamento que él dirige.

Al Dr. Pablo Nuño, miembro de mi jurado por darme todas las facilidades para trabajar con el paquete CPLEX en la Universidad Anáhuac, así como todo su apoyo y amistad.

Al M. en I. Octavio García, coordinador de la Sección de Estructuras de la DEPMI por facilitarme el acceso a una estación de trabajo y poder trabajar con el paquete de cómputo BARON.

Al Departamento de Sistemas y en particular a mi tutor Dr. José Jesús Acosta por su apoyo y paciencia.

A mis profesores de inglés, en especial John Hertzberg, que me brindaron su amistad y su ayuda profesional.

A Miguel Mireles, amigo de toda la vida, por su apoyo y consejos en la programación del algoritmo.

A mis hermanos Ivone y Erick por el apoyo de captura en textos y dibujos, y por sus útiles sugerencias.

A Miguel Angel Villegas, por sus sugerencias profesionales y sobre todo por su apoyo espiritual.

A mi hermana Lilia Paz por ser una compañera incansable y que me dio su apoyo en los momentos mas difíciles.

A José Maclovio Sautto por su apoyo, confianza y cariño a lo largo de muchos años y, finalmente, a mis padres, a mis hijas y amigos que siempre confiaron en que podía lograr terminar este trabajo.

RESUMEN

Se desarrolla un algoritmo general para minimizar una función cóncava sobre un poliedro convexo, con la restricción adicional de variables enteras. Este algoritmo es una extensión del algoritmo de Taha para el problema sin restricción de variable entera. El algoritmo usa un procedimiento especial de planos de corte, para identificar el punto extremo que sea mínimo global en el poliedro convexo, pero si no existe tal punto que sea entero, el algoritmo usa un procedimiento de ramificación y acotamiento para buscar en el interior del poliedro. El método de planos de corte está basado en la teoría general de Glover para la construcción de cortes legítimos que ordenen los vértices en un poliedro dado. Se usa también una función lineal para aproximar inferiormente la función objetivo cóncava. La construcción de dicha función se hace con base en la fórmula de Taylor. Se presentan aplicaciones al problema de cargo fijo, un problema de producción transporte y casos de funciones cóncavas separables y no separables.

ABSTRACT

A general algorithm is developed for minimizing a concave function over a convex polyhedron, and with the additional constraint of integer variables. This algorithm is an extension of Taha's Algorithm for a problem without the integer variables constraint. The algorithm uses a special cutting plane procedure to identify the global minimum extreme point of the convex polyhedron, but if there is not any integer point in a vertex set, the algorithm uses a branch and bound procedure to look for it inside the polyhedron. The cutting plane method is based on Glover's general theory for constructing legitimate cuts to rank the vertex in a given convex polyhedron. It is important to say that a linear function is used that underestimates the constrained concave objective function. The construction of this function is based on the Taylor formula. Applications of the algorithm to the fixed-charge problem, production-transportation problem, concave separable functions as well as non separable concave functions are presented.

INDICE

1.	Introducción	4
2.	Capítulo 1: Introducción	8
	1.1 Descripción del Problema	8
	1.2 Aplicaciones: El Problema de Cargo Fijo	11
	1.3 Propiedades Generales del Problema de Cargo Fijo	17
	1.4 Calendarización de Proyectos	18
	1.5 Asignación de Armamento	20
	1.6 Evaluación de Cotizaciones	23
	1.7 Localización de un Almacén	24
	1.8 Problema de Producción-Transporte	26
	1.9 Concavidad Generalizada en Economía	29
3.	Capítulo 2: Estado del Arte	31
	2.1 Programación Entera	31
	2.2 Optimización Global	38
4.	Capítulo 3: Metodología	58
	3.1 El Algoritmo	58
	3.2 Sobre el Uso de Funciones Lineales	65
	3.3 El Problema Lineal Relajado	75
	3.4 El Uso del Método Simplex en el Problema Relajado	77
	3.5 El Arbol de Búsqueda	79
	3.6 Cortes Convexos	84
	3.7 Complejidad Computacional del Algoritmo	98
	3.8 Convergencia del Algoritmo	105
	3.9 Problema de Producción-Transporte	111
5.	Capítulo 4: Problemas de Prueba	116
	4.1 Problemas de Cargo Fijo	117
	4.2 Funciones Separables	126
	4.3 Funciones no Separables	142
	4.4 Problemas de Producción- Transporte	149
6.	Conclusiones	168
7.	Apéndice A: Implementación del Algoritmo	171
8.	Apéndice B: Funciones Cóncavas	180
9.	Apéndice C: La Estructura de los Poliedros	194
10.	Bibliografía	206

INTRODUCCION

"A vuestros ojos se ofrece este libro, por quedar ilustrado a tanto sol, digno de tanta deidad.

Divertiros sólo un rato es cuanto aspirar podrá, que fuera mucho emprender atrevésele a ocupar".

Sor Juana Inés de la Cruz.

Enigmas.

"Dios nos ha dado la razón para comprender lo que nos compete, pero no para entenderlo todo en el infinito y eterno universo".

Voltaire

La enorme necesidad de resolver problemas de programación entera aunada al rápido avance de la tecnología en computación, han permitido considerar problemas que hace algunos años hubieran sido computacionalmente intratables, en consecuencia hemos visto la creación de un número cada vez mayor de diversos algoritmos para resolver la gran variedad de problemas que se presentan en las diferentes áreas de aplicación. Después del aparente fracaso de los métodos exactos de los años setentas y el creciente desarrollo de los métodos heurísticos en los ochentas, vemos que para esta década se tienen algoritmos que combinan ambos métodos y el auge de nuevos algoritmos exactos.

El problema que aquí se aborda se considera dentro de un área denominada Optimización Global que contempla problemas que se pueden considerar dentro

de la programación no lineal pero cuyas técnicas no han tenido éxito en la solución de los mismos. Esta deficiencia se da principalmente debido a la formulación de los problemas donde se busca un mínimo global de una función real que posee diferentes mínimos locales. Sin embargo, en el caso de minimización de una función cóncava por la naturaleza de la misma se sabe que la solución óptima se alcanza en un punto extremo y debido a que agregamos la restricción de soluciones enteras, podemos ver que este problema y la naturaleza del algoritmo propuesto caen dentro de la programación entera, no obstante que no podemos afirmar que haya una división tajante entre las diferentes áreas de interés de la Investigación de Operaciones.

Es dentro de esta aparente línea divisoria donde se enmarca el problema que se plantea y resuelve aquí, es pues, el objetivo de este trabajo desarrollar un algoritmo basado en el método de los planos de corte convexos para resolver un problema lineal relajado, obtenido a partir de una función de aproximación basada en el desarrollo de Taylor para funciones.

Se analizan varios problemas y se considera el caso de soluciones degeneradas, mismo que se resuelve añadiendo un procedimiento de ramificación y acotamiento. La construcción de dicha función es importante en la medida que de su exactitud se desprende la rapidez del algoritmo para

encontrar la solución óptima. Los antecedentes para resolver este problema los podemos encontrar en el área de la optimización global, sin embargo debido a la consideración de variables enteras no podemos afirmar que haya un algoritmo que resuelva específicamente este caso. Las pruebas computacionales que se hacen son con base en la programación del algoritmo.

No podemos desarrollar un algoritmo de solución pensando en que " si la realidad no se ajusta a mi modelo, peor para ella", antes bien, se consideran varios casos donde el comportamiento del problema hace que se adapte el algoritmo para su solución. Así es el caso en que si la función objetivo es separable, se propone una función de aproximación basada en sus derivadas parciales, si la función es no separable se usa un procedimiento de cambio de variables y se procede como en el caso separable.

Si las variables son acotadas, o si hay discontinuidad en la función objetivo se hace un procedimiento de preprocesamiento para poder trabajar con el algoritmo. En el caso de que el problema sea de tipo red, se trabaja con el algoritmo producción -transporte desarrollado en el Capítulo 3. Todo esto no significa que se puedan resolver absolutamente cualquier tipo de problema que caiga en esta categoría, pero si se busca que sea lo suficientemente general para que se puedan resolver una amplia gama de problemas. Este trabajo se desarrolla de la siguiente manera:

En el capítulo 1 se expone el problema a resolver y varios ejemplos de casos de aplicación del mismo, en el capítulo 2 se hace una revisión de la literatura existente sobre el tema, así como los antecedentes que hay sobre algoritmos propuestos para resolver el problema pero sin considerar que se requieran variables enteras. En el capítulo 3 se desarrolla el algoritmo, se explica la complejidad computacional del mismo y la convergencia y se desarrolla un algoritmo que se especializa en el caso de producción-transporte con costos cóncavos.

En el capítulo 4 se resuelven problemas para probar la eficiencia de los algoritmos, en los casos en que las soluciones originales sean enteras se compara dicha solución con los métodos que se usaron para resolverlos originalmente, en los casos en que se reportaron solamente soluciones continuas, se expone el método usado y la solución entera obtenida por el algoritmo. Se anexan tres apéndices, en el primero se presenta el algoritmo en pseudocódigo así como las pruebas computacionales realizadas con los ejemplos del capítulo 4. El segundo es sobre convexidad y resultados de las funciones cóncavas, para quienes estén más interesados sobre la teoría de las mismas, y el tercero sobre teoría de poliedros, necesaria para explicar la convergencia en ramificación y acotamiento. Finalmente se presentan las conclusiones y la bibliografía.

CAPITULO 1

INTRODUCCION

“La imaginación es más importante que el conocimiento”.

Einstein

“El artista es el creador de cosas bellas. Revelar el arte y ocultar al artista es la finalidad del arte”.

Oscar Wilde

1.1 DESCRIPCION DEL PROBLEMA

En numerosos casos de la programación entera, se presenta el problema de optimizar una función que no es convexa sujeta a restricciones lineales, tales son los casos del problema de cargo fijo y el de localización de plantas, y numerosos casos de economía donde las funciones de costo pueden ser cóncavas.

El problema que se resuelve en este trabajo, es un problema de minimización de una función cóncava sujeta a restricciones lineales y variables enteras como se expresa a continuación:

$$\min_{x \in Q} f(x) \quad (P)$$

donde

$$\begin{aligned} x &= (x_1, x_2, \dots, x_n) \\ Q &= \{x \in E^n \mid Ax = b, x \geq 0, x \in Z\}, \end{aligned}$$

la función $f(x)$ se supone cóncava y bien definida sobre el poliedro convexo Q . Se supone también que el mínimo restringido de $f(x)$ es finito, es importante notar que la solución óptima de $f(x)$ se alcanza en un punto extremo del espacio de soluciones Q , siempre y cuando dicha solución sea entera.

Es importante notar también que, además del hecho de que un óptimo local no es necesariamente un óptimo global, el procedimiento de solución debe buscar también una solución óptima entera. En esencia el algoritmo desarrollado debe buscar un punto extremo de la región factible Q . En caso de que el punto extremo no sea entero, se busca en el interior de la región factible usando ramificación y acotamiento.

El procedimiento general para resolver el problema consiste en reemplazar la función objetivo original $f(x)$ por una función lineal $\ell(x)$ que aproxime inferiormente a la función original tal que $\ell(x) \leq f(x)$ para toda $x \in Q$, es decir $\ell(x)$ es una cota inferior de $f(x)$.

Para resolver el programa lineal relajado se hace uso de un algoritmo de rama y corte, donde el corte que se utiliza es un corte convexo, para así identificar un punto entero extremo que es un mínimo global en la región factible. Si no se encuentra ningún punto entero entre los puntos extremos entonces se procede con ramificación y acotamiento a partir de la mejor solución óptima no entera, hasta encontrar un punto entero que minimice la función objetivo. Si se encuentra un punto entero que sea extremo, entonces el algoritmo se detiene, dicha solución es óptima.

Para analizar cada punto extremo ordenado se considera al valor de la función objetivo obtenida usando la función de aproximación como una cota inferior, considerando primero aquella que sea una mejor aproximación. El corte convexo se usa para eliminar el punto extremo más recientemente ordenado.

En el campo de la Programación Entera, los métodos de planos de corte para resolver programas enteros han experimentado un renovado interés en años recientes, debido al éxito de las computadoras. A diferencia de los primeros métodos de Planos de Corte de Gomory (1958), que inicialmente parecían prometedores pero que eran computacionalmente ineficientes, recientes métodos de planos de corte han alcanzado éxito por el hecho de enfocarse a la generación de planos de corte más profundos, o el uso de los mismos

cortes como los de Gomory, usados dentro de algoritmos de Rama y Corte, Balas [5]. Por otro lado, en el campo de la Optimización Global en cuanto a los problemas de minimización cóncava, se cuenta con la experiencia de algoritmos como los que se describen en el capítulo dos, pero no encontramos un algoritmo que resuelva el problema de minimización cóncava con restricción de variables enteras, que es el algoritmo que se presenta en el capítulo 3.

A continuación se presentan aplicaciones de este problema, algunas de ellas se resolverán con el algoritmo propuesto en el capítulo 4.

1.2 APLICACIONES: EL PROBLEMA DE CARGO FIJO [41]

Suponga que se requiere decidir la ubicación de plantas de producción entre m lugares existentes. Todas las plantas producen productos homogéneos. La planta i tiene una capacidad de a_i unidades y necesita una inversión fija f_i . Los productos son enviados a n clientes, donde el j -ésimo cliente demanda b_j unidades. Si c_{ij} es el costo de producir una unidad en la planta i y enviarla al cliente j , es necesario determinar las capacidades de operación de las plantas de tal forma que los costos totales de producción e inversión se minimicen.

Sea x_{ij} la cantidad manufacturada en la planta i para el cliente j , entonces el modelo es

$$\min z = \sum_{i=1}^m \sum_{j=1}^n F_{ij}(x_{ij})$$

sujeto a

$$\sum_{i=1}^m x_{ij} \geq b_j \quad j = 1, 2, \dots, n$$

$$\sum_{j=1}^n x_{ij} \leq a_i \quad i = 1, 2, \dots,$$

donde

$$F_{ij} = \begin{cases} c_{ij}x_{ij} + f_i & \text{si } \sum_j x_{ij} > 0 \\ 0 & \text{en otro caso} \end{cases}$$

Este modelo es muy similar a un modelo de programación lineal ordinario, excepto por el hecho de que z es una función no lineal.

El modelo se puede convertir en un problema mixto (cero-uno) usando una sustitución conveniente.

$$y_i = \begin{cases} 1 & \text{si } \sum_j x_{ij} > 0 \\ 0 & \text{en otro caso} \end{cases}$$

de esta manera el modelo se convierte en

$$\min z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + f_i y_i$$

sujeto a

$$\sum_{i=1}^m x_{ij} \geq b_j \quad j = 1, 2, \dots, n$$

$$\sum_{j=1}^n x_{ij} \leq a_i y_i \quad i = 1, 2, \dots, m$$

$$x_{ij} \geq 0$$

$$y_i = (0, 1) \forall i$$

Si para una i dada, x_{ij} es mayor que cero, entonces y_i es necesariamente igual a uno, y la restricción correspondiente permanece sin cambio. Si por otro lado, x_{ij} es igual a cero para toda i , entonces y_i es igual a cero, ya que con la restricción de capacidad siendo redundante, el mínimo de z sólo puede alcanzarse con $y_i = 0$.

Note que a diferencia de los modelos enteros, las variables auxiliares y_i se introducen principalmente por conveniencia analítica y sólo arrojan información redundante acerca de la solución del problema. Aunque $y_i = 1$ ó 0 , se puede interpretar como la construcción (o no) de la planta i , la misma información se asegura observando cuando $\sum_j x_{ij}$ es positiva o cero.

El modelo anterior se puede generalizar incluyendo una función de costos de producción con discontinuidades; ésto significa que en lugar de suponer que

cada unidad se produce con un mismo costo, el costo por unidad supone valores marginales decrecientes con respecto al nivel de producción. Más aún, el cambio de un costo marginal a uno menor se alcanza a expensas de incurrir en un cargo fijo.

Para la planta i y el nivel de producción k defina:

a_{ik} = nivel de producción que representa el punto de ruptura.

w_{ik} = número de unidades producidas, $0 \leq w_{ik} \leq a_{ik} - a_{i,k-1}$

d_{ij} = costo de transporte por unidad de la planta i al cliente j .

Una función típica de costo de producción es la de la figura siguiente:

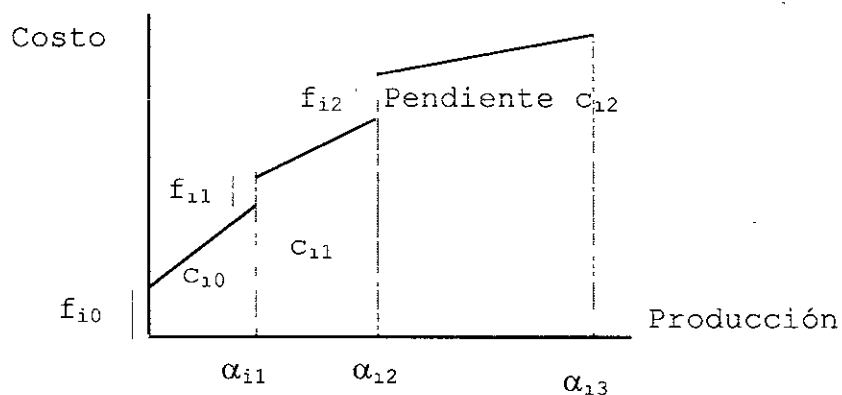


figura 1

El modelo se puede formular entonces como:

$$\min z = \sum_i^m \sum_j^n d_{ij} x_{ij} + \sum_k c_{ik} w_{ik} + \sum_k f_{ik} y_{ik}$$

sujeto a

$$\sum_{i=1}^m x_{ij} \geq b_j \quad j = 1, 2, \dots, n$$

$$\sum_{j=1}^n x_{ij} = \sum_k w_{ik} \quad i = 1, 2, \dots, m$$

$$\frac{w_{i,k-1}}{a_{ik} - a_{i,k-1}} \geq y_{ik} \geq \frac{w_{ik}}{a_{i,k+1} - a_{ik}}$$

$$x_{ij} w_{ik} \geq 0$$

$$y_{ik} = 0,$$

Se ha visto que si $w_{ik} > 0$, $y_{ik} = 1$ ya que debe satisfacer $0 < y_{ik} \leq 1$. Si $w_{ik} = 0$, entonces $0 \leq y_{ik} \leq 1$ y $y_{ik} = 0$, por la minimización de la función objetivo. Note que $w_{i,k-1} = a_{ik} - a_{i,k-1}$, si $w_{ik} > 0$, porque y_{ik} es una variable cero-uno, de otra manera $0 < y_{ik} < 1$ produciría una solución infactible.

Como se puede observar, el problema de cargo fijo se puede ver como un modelo mixto cero-uno. Aunque se pueden usar los algoritmos de programación entera, sus propiedades especiales permiten usar algoritmos especializados más eficientes. Tales algoritmos pueden ser tanto

aproximados como exactos, los aproximados no dan la solución exacta pero proporcionan buenas soluciones y requieren menos cálculos.

Los problemas de localización de plantas son una variante del modelo de cargo fijo. Hay una cierta cantidad de estaciones receptoras n y las demandas en esos destinos son satisfechas por m plantas potenciales o surtidoras. Usualmente n es considerablemente mayor que m . Se requiere decidir la ubicación y la capacidad de cada planta para poder satisfacer la demanda.

Hay un costo (fijo) asociado a cada instalación que representa el costo de construir cada planta y el objetivo es minimizar el costo total incluyendo los costos fijos y los costos de transportación entre orígenes y destinos.

El problema se reduce a un modelo ordinario de transporte con la excepción de que aparece el término de cargo fijo k_i en la función objetivo si para cualquier j , $x_{ij} > 0$ y no existe de otra manera. Donde x_{ij} es la cantidad transportada del origen i al destino j . Claramente para cargos fijos preestablecidos el problema es un problema ordinario de transporte.

Hay una amplia variedad de algoritmos y una rica literatura para problemas de localización de plantas. Ellos varían primordialmente en los detalles para

construir la función objetivo y las restricciones, pero la idea básica permanece estática.

1.3 PROPIEDADES GENERALES DEL PROBLEMA DE CARGO FIJO

El problema de cargo fijo involucra la minimización de una función objetivo cóncava $z(x_1, \dots, x_n)$ sobre un poliedro convexo $Q = \{x \mid \sum a_{ij}x_j \leq b_i, i = 1, \dots, m, x_j \geq 0, j = 1, \dots, n\}$. Se puede ver que $z(x_1, \dots, x_n)$ es cóncava, por el hecho de ser la suma de n funciones de variable real $f_j(x_j)$ y estas son cóncavas. Esta misma observación se aplica para la localización de plantas.

Una propiedad importante de este tipo de problemas es que la solución óptima debe alcanzarse en un punto extremo del espacio factible, es decir, debe estar asociado con una solución básica factible de Q . Esto significa que la búsqueda del óptimo global se puede restringir a considerar solamente los puntos extremos de Q . Este resultado es similar al usado en programación lineal. Sin embargo como la función objetivo es cóncava el algoritmo asociado es más complejo.

Una pregunta lógica que se desprende de esta propiedad es: ¿Cuándo el programa lineal obtenido al ignorar los términos $k_j y_j$ produce el mismo punto

óptimo extremo que el problema de cargo fijo correspondiente? La respuesta que dieron Hirsch y Dantzig (1954) demostró que una condición "suficiente" para que ésto ocurra es que todas las k_j sean iguales para toda j y que todos los puntos extremos de Q sean no degenerados. Se requiere la no degeneración ya que un punto extremo degenerado tiene pocas variables básicas positivas y puede ser óptimo para el problema de cargo fijo, pero no para el programa lineal.

1.4 CALENDARIZACION DE PROYECTOS

Un proyecto define una combinación de actividades interrelacionadas que deben ejecutarse en un cierto orden antes de que el trabajo completo pueda terminarse. Las actividades están interrelacionadas en una secuencia lógica en el sentido de que algunas de ellas no pueden comenzar hasta que otras hayan terminado.

Una actividad en un proyecto, generalmente se ve como un trabajo que requiere tiempo y recursos para su terminación. Se considera que cada actividad (i, j) tiene una duración d_{ij} que varía entre l_{ij} y u_{ij} , llamados tiempo de "quiebra" y el tiempo de mínimo costo respectivamente.

La relación costo-duración de la actividad se supone continua y cóncava. La suposición de concavidad se justifica en algunas situaciones donde, por ejemplo, el costo por rentar equipo ocurre a una tasa marginal decreciente a medida que la duración de las actividades crece. La formulación general del modelo está dada por:

$$\min f(d) = \sum_{(i,j) \in A} c_{ij}(d_{ij})$$

sujeto a

$$t_i + d_{ij} \leq t_j$$

$$l_{ij} \leq d_{ij} \leq u_{ij}$$

$$t_n \leq T_n$$

$$t_i, d_{ij} \geq 0 \quad \forall i, (i,j) \in A,$$

donde t_i es el tiempo en el que el evento i (definido al final de las actividades (i,j) para toda j) ocurre. El primer conjunto de restricciones representa las relaciones de precedencia entre las diferentes actividades del proyecto. La restricción $t_n \leq T_n$ indica que el último evento n del proyecto debe completarse en una fecha dada T_n .

La función de costos cóncava también se encuentra en procesos industriales, donde el problema consiste en determinar la inversión futura de capital y el tamaño óptimo de las plantas de producción.

El costo para la construcción de esas plantas es la función:

$$\text{costo de capital} = av^b$$

donde

a es el coeficiente de costo

v es la capacidad

b es un coeficiente que varía entre 0.5 y 0.8 para plantas químicas, así la función es cóncava para $b < 1$.

1.5 ASIGNACION DE ARMAMENTO [2]

El modelo de asignación de armamento consiste en asignar armas a objetivos para maximizar el valor esperado del daño en el objetivo. Hay restricciones en la disponibilidad de los distintos tipos de armas, así como en el número de tipos de armas a ser asignadas a varios objetivos. Las restricciones son lineales y la función objetivo es no lineal.

Sea x_{ij} = número de armas de tipo i asignadas al objetivo j ($i = 1, 2, \dots, p$,
 $j = 1, 2, \dots, q$).

El límite en el número de armas asignadas se especifica en términos de:

a_i = el número total de armas de tipo i disponibles.

b_j = el número mínimo de armas de todos tipos asignadas al objetivo j .

Las restricciones asociadas al número total de armas y el número mínimo de armas asignadas a los objetivos son:

$$\sum_{j=1}^q x_{ij} \leq a_i \quad i = 1, 2, \dots, p \quad (1)$$

$$\sum_{i=1}^p x_{ij} \geq b_j \quad j = 1, 2, \dots, q \quad (2)$$

La función objetivo se formula en términos de la probabilidad de daño de varios objetivos medidos por su valor militar. Sean

α_{ij} = la probabilidad de que el objetivo j no resulte dañado por un ataque usando una unidad de arma i .

u_j = el valor militar del objetivo j .

El daño esperado para el objetivo j por una asignación de x_{ij} armas de tipo i es $(1 - \alpha_{ij}^{x_{ij}})$, y el daño esperado para el objetivo j por la asignación de armas de todo tipo $\sum x_{ij}$ es:

$$1 - \prod_{i=1}^p \alpha_{ij}^{x_{ij}}$$

El valor total esperado de daño de un objetivo es la suma de los daños esperados a objetivos medidos por el valor militar de los objetivos:

$$\sum_{j=1}^q u_j \left[1 - \prod_{i=1}^p \alpha_{ij}^{x_{ij}} \right] \quad (3)$$

A partir de aquí se plantea el problema de programación no lineal como sigue: Seleccione x_{ij} 's para maximizar la función objetivo (3) sujeta a las restricciones lineales (1) y (2) y las condiciones de no negatividad. Una definición alternativa de α_{ij} es que se interprete como la fracción del objetivo j que no resultará dañado por un ataque usando una unidad de arma i . La función objetivo se interpreta entonces como el valor fraccionario total del daño (a ser maximizado).

Podemos interpretar la función objetivo como el número mínimo de armas de todo tipo asignadas al objetivo j , de esta manera la función objetivo queda definida como sigue:

$$\min z = \sum_{j=1}^n u_j \prod \alpha_{ij}^{x_{ij}}$$

sujeto a

$$\sum x_{ij} \leq a_i$$

$$\sum x_{ij} \geq b_j$$

$$x_{ij} \geq 0$$

donde la función z es cóncava.

1.6 EVALUACION DE COTIZACIONES [2]

Existen muchas variantes al problema de evaluación de cotizaciones. La siguiente descripción es suficientemente general e incluye la mayoría de los elementos que lo hacen difícil de resolver.

Una compañía quiere comprar un número específico de unidades de un artículo. Obtiene cotizaciones de n vendedores, cada uno de los cuales no puede suministrar la cantidad total. Los vendedores envían cotizaciones

indicando sus precios en función de cantidades compradas. Tales presupuestos usualmente reflejan costos fijos y unidades de costo decrecientes, dependiendo del tamaño de las órdenes, así como cantidades máximas y mínimas.

El problema consiste en escoger la cantidad a comprar a cada vendedor de tal manera que se minimice el costo total de compra de los artículos requeridos. Esta función de costos es una función cóncava.

1.7 LOCALIZACION DE UN ALMACEN BAJO ECONOMIAS DE ESCALA [2]

Un almacén es una instalación intermedia donde uno o varios artículos pueden procesarse, almacenarse, empacarse, etc. El artículo se transporta entonces desde el almacén hasta sus destinos finales.

Un aspecto esencial en la literatura de los problemas de localización es el problema de ignorar economías de escala en transporte y restricciones de capacidad con cota inferior, en esta aplicación podemos formular el problema como sigue:

$$\min \phi = \sum_{j \in I} f_j(P_j) + \sum_{j \in KK} \sum_{j \in JJ} t_{ij} x_{ij} + \sum_{j \in I} \sum_{k \in K} t_{jk} x_{jk}$$

sujeto a

$$\sum_{j \in JJ \cup K} x_j \leq Q_{u_i} \quad i \in KK$$

$$\sum_{i \in KK} x_j \leq Q_{u_i} \quad j \in JJ$$

$$\sum_{j \in I} x_j \geq S_k \quad k \in KK$$

$$x_{ij}, x_{jk} \geq 0 \quad \forall i, j, k$$

En este caso, tenemos $f_j(P_j) = V_j y_j + \lambda_j^* (P_j)^q$, donde y_j toma el valor de 1 si $P_j > 0$ y cero en otro caso, y q es un número obtenido del análisis estándar de inventarios. Geométricamente vemos en la figura siguiente, que la función es cóncava. Dicho problema se ha resuelto con anterioridad con un método iterativo que descansa en el problema de transporte, algunas variantes (como los retrasos) se han resuelto usando heurísticos.

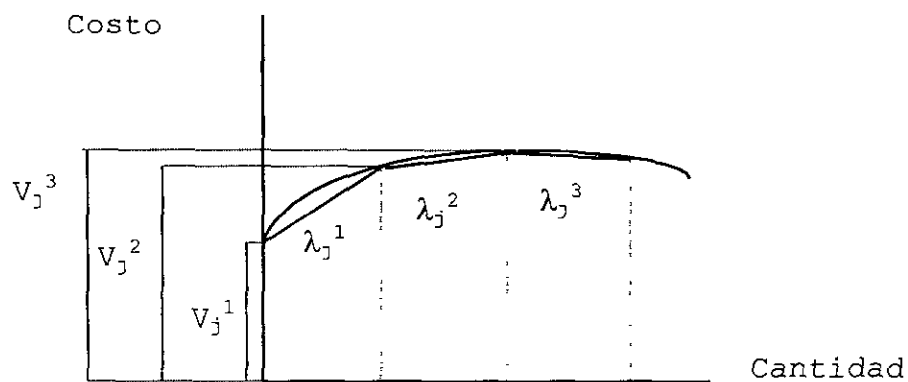


figura 2

1.8 PROBLEMA DE PRODUCCION-TRANSPORTE- CON COSTOS DE PRODUCCION CONCAVOS [85]

Una clase importante de problemas de decisión es el de diseño de localización óptima y sistemas de distribución. En esos problemas el objetivo es minimizar el costo total, que consta de: costos de producción (incluidos costos de construcción, mantenimiento y operación de instalaciones) los que usualmente exhiben economías de escala y que pueden describirse a través de funciones cóncavas, y costos de distribución-transporte, que a menudo se suponen lineales.

Tal composición de costos es también típica de muchos otros modelos de la vida real en diferentes áreas, tales como administración de producción-inventario, tamaño de lote económico, redes de telecomunicación, etc.

Aunque este tipo de problemas son difíciles, la mayoría de ellos tiene una estructura especial que a menudo puede explotarse para propósitos de métodos de solución más eficientes. Una estructura común a estos problemas es la de monotonicidad y poca densidad, lo que significa que la función objetivo es no decreciente a lo largo de ciertas direcciones y aunque la dimensión completa del problema puede ser muy grande, la no convexidad está concentrada en una pequeña proporción de variables.

CAPITULO 2

ESTADO DEL ARTE

"La eternidad está enamorada de las creaciones del tiempo".

William Blake. *Proverbios del infierno.*

"Todo arte es, a la vez, superficie y símbolo.

Los que buscan bajo la superficie, lo hacen a su propio riesgo.

Los que intentan descifrar el símbolo, lo hacen también a su propio riesgo.

Es al espectador, y no a la vida, a quien refleja realmente el arte".

Oscar Wilde

2.1 PROGRAMACION ENTERA

Muchos problemas de la vida real, tales como enrutamiento de vehículos, localización de plantas, diseño de redes de telecomunicaciones, selección de portafolio, calendarización de maquinaria o de tripulación aérea, se pueden formular como problemas de variables enteras que deben satisfacer un conjunto finito de restricciones lineales y una función objetivo, que puede ser lineal o no, a ser optimizada. Estos problemas pueden ser mixtos, si algunas variables son enteras y el resto continuas, o enteros puros si todas las variables tienen que ser enteras. Por muchos años la programación entera no ha tenido muchas expectativas debido a que las formulaciones

para problemas de gran escala, que son los que a menudo surgen de problemas reales, fueron difíciles de manejar, debido principalmente a la falta de un adecuado poder computacional y a la complejidad de los algoritmos disponibles en ese tiempo. Sin embargo, esto ha cambiado, aunque la programación entera es teóricamente "dura" (los problemas generales enteros y enteros mixtos pertenecen a la clase NP-hard), nuevos desarrollos en la tecnología tanto algorítmica como computacional hacen posible resolver las aplicaciones de la vida real que sólo hace algunos años se consideraban intratables.

Casi todos los métodos para resolver problemas enteros mixtos comienzan relajando el problema a uno menos restringido, lo más usual es hacerlo a través de la eliminación momentánea de la restricción de variable entera y resolviendo el problema de programación lineal. Desgraciadamente es raro que este problema relajado tenga una solución entera, por lo cual se toma su solución sólo como el paso inicial para un algoritmo más sofisticado.

A inicios de los sesentas se desarrollaron dos clases de métodos para tratar estos problemas. El primero de ellos es el de planos de corte. Los métodos de planos de corte son en realidad las primeras técnicas sistemáticas que se desarrollaron para el problema entero (lineal). Los primeros trabajos de Dantzig et.al. en 1954 y de Markowitz y Manne en 1957 dirigieron la

atención de los investigadores hacia la importancia de resolver programas lineales en enteros, pero Dantzig fue el primero en proponer un corte para resolver tales problemas. Su idea es resolver primero el programa lineal ignorando las condiciones de variable entera. Si la solución resultante no es entera, entonces se debe asegurar un nuevo conjunto de valores para las variables no básicas. Esto se logra haciendo que la suma de las variables no básicas debe ser al menos igual a uno.

Aumentando este corte al tableau actual, la factibilidad se puede reestablecer aplicando el método simplex dual. Aunque este corte representa una condición necesaria para una solución entera, y se ha aplicado a algunos problemas con éxito, no hay garantía de que sucesivas aplicaciones del corte produzcan una solución entera en un número finito de iteraciones.

El primer algoritmo de cortes finito lo desarrolló Gomory en 1958, para un problema entero puro. En el algoritmo de planos de corte, la relajación de programación lineal se ve enriquecida por nuevas desigualdades o planos de corte. Estos planos de corte se pueden construir a partir de argumentos modulares o de disjunción, de tal forma que ninguna solución entera quede fuera del corte. Se puede pensar en los planos de corte como un procedimiento para reescribir o reformular el problema original, de tal forma

que el conjunto de soluciones enteras se preserva, pero la relajación obtenida al eliminar las condiciones de variable entera es más rígida. Gomory demostró que se pueden construir los cortes a partir de un tableau simplex y, aunque el algoritmo converge en un número finito de iteraciones, los errores de redondeo representan una dificultad importante. Esto condujo a que Gomory construyera un nuevo algoritmo en 1960, que mejora al anterior en cuanto a los errores de redondeo. Gomory también extendió su teoría para resolver problemas mixtos enteros.

Los algoritmos de Gomory generaron un entusiasmo general entre la comunidad, ya que la programación entera se reducía a resolver una sucesión de problemas lineales. Desafortunadamente, los resultados computacionales no fueron tan buenos como la teoría. Era claro que los primeros experimentos que provenían de problemas reales eran más difíciles que lo esperado o, en otras palabras, que la tasa de convergencia era muy lenta y el algoritmo numéricamente inestable. Aunque la investigación teórica en planos de corte continuó por muchos años, los experimentos computacionales no dieron indicios de que los planos de corte se podrían usar en forma práctica. Otros tipos de corte que se introdujeron fueron los de Glover [39], conocido como un método de escalación de cota, y por Young [87], Balas [3], Balas et.al. [4] y Glover [40], con los cortes de intersección o convexos.

Una característica común de los cortes anteriores es que los algoritmos asociados son todos del tipo dual, esto es: que la solución del problema no está disponible hasta que el algoritmo termina. Esta es una gran desventaja si se pretende detener los cálculos prematuramente. Un algoritmo primal rudimentario fue hecho por Ben-Israel y Charnes, pero Young [87] fue el primero en desarrollar un algoritmo primal finito.

En los últimos años ha habido un progreso considerable usando técnicas combinatorias para generar planos de corte, éstos se derivan de propiedades combinatorias de la programación entera. El éxito de estos planos de corte ha sido impresionante, especialmente en problemas con una estructura específica como el Problema del Agente Viajero. Lo más importante es que el problema tenga una estructura combinatoria para poder generar los planos. Más aún, ya que los planos de corte dependen de dicha estructura, cambios mínimos en el problema pueden hacer inválido el plano de corte. Otra dificultad es que no son fáciles de obtener, razón por la cual algunos investigadores como Padberg y Rinaldi [64] desarrollaron otra técnica llamada Ramificación y Corte, en el contexto de resolver el problema del agente viajero.

Otros elementos que han hecho eficiente el código de rama y corte en problemas enteros mixtos incluyen preprocesamiento y heurísticos. Sin

embargo, la línea de investigación que considera sólo planos de corte sigue desarrollándose y obteniendo buenos resultados, razón por la cual los métodos de planos de corte para resolver programas enteros han experimentado un renovado interés en años recientes debido al avance de la computación. A diferencia del Método de Planos de Corte de Gomory que inicialmente pareció prometedor, pero que eventualmente se probó computacionalmente ineficiente, recientes métodos de planos de corte han alcanzado éxito por el hecho de centrarse en la generación de planos de corte profundos. Dependiendo de la naturaleza del problema se construyen diferentes tipos de planos de corte, aunque el objetivo es el de construir un algoritmo general que sirva para resolver problemas enteros en general. Así encontramos en Boyd [18] un algoritmo general para resolver programas enteros generales a través de lo que denomina los Cortes de Fenchel, donde explota la dualidad entre separación y optimización. Otros trabajos de Boyd se refieren a la solución de problemas como el del Poliedro para la Mochila, Boyd [16], y para problemas enteros binarios, Boyd [17].

Otros planos de corte son los desarrollados por Balas et.al. [5], llamados cortes de elevación y proyección, pero dentro de un ambiente de rama y corte. Los algoritmos de Rama y Corte, bautizados así por Padberg y Rinaldi [64], usan en forma combinada planos de corte y ramificación y

acotamiento y la idea consiste en usar cortes cuando así sea necesario en los nodos del árbol de búsqueda.

Es en este contexto que los planos de corte de Balas [4], incluidos los de Gomory, tienen un desarrollo más eficiente, haciendo que el tiempo de ejecución del algoritmo sea menor. Dentro de este ambiente de Rama y corte podemos encontrar también algoritmos para programación entera que resuelven el problema lineal relajado con algún algoritmo de punto interior como en Mitchell [11] comprendido en estudio completo de Beasley [11], donde se presenta un desarrollo alternativo de los algoritmos de la Investigación de Operaciones pero usando como base algoritmos de punto interior.

En programación entera existen algoritmos para resolver problemas como el de cargo fijo o de inversión de capital considerando que la función objetivo es convexa. Si la función objetivo no es convexa se puede hacer uso de estos mismos algoritmos usando algún procedimiento de convexificación, el uso de una envoltura convexa como en Falk y Soland [30] o de una función lineal que aproxime a la función cóncava. La investigación desarrollada durante las últimas dos décadas ha producido una amplia variedad de métodos para encontrar soluciones a problemas de optimización no lineal, que han sido de interés dentro de optimización global. Aunque muchos

algoritmos de programación no lineal obtendrán una solución local a un problema no lineal, en optimización global no hay en general un criterio local para decidir cuándo una solución local dada es global.

En ciertas clases de problemas no lineales la solución local siempre es global. Por ejemplo, en problemas de minimización con una función objetivo convexa (o cuasi convexa) sujeta a restricciones convexas el mínimo local es la solución global, Avriel [2], Horst [49], Zang [88]. Para funciones no convexas puede haber muchos mínimos locales para los cuales los criterios locales no darán información acerca del mínimo global. Por ejemplo, una función cóncava en un politopo puede tener un mínimo local en cada vértice del politopo.

2.2 OPTIMIZACION GLOBAL

En la clase general de problemas de optimización global vamos a considerar una subclase que parece más tratable que las otras. Esta es la clase de problemas de minimización global cóncava restringida. El problema general se puede formular como:

$$\begin{aligned} & \min \text{ global } f(x) \\ & \text{sujeta a:} \\ & x \in D \subseteq R \end{aligned} \tag{P}$$

donde $f(x)$ es una función real cóncava y D es un conjunto compacto convexo.

Una amplia variedad de aplicaciones prácticas importantes conducen a la formulación de problemas como (P), incluyendo problemas con economías de escala y cargo fijo. Además, varios problemas importantes de optimización pueden formularse como problemas de minimización cóncava. Debido a Raghavachari [67], se sabe que el problema entero binario o cero-uno es equivalente a un problema de minimización cóncava (cuadrática) sujeto a restricciones lineales. En general, un programa entero no convexo y no lineal puede reducirse a un programa cóncavo real (suponiendo que la función objetivo no convexa es acotada y satisface la condición de Lipschitz). En forma semejante, el problema de asignación cuadrática se puede formular como un problema (P). Frieze [35] ha reducido el problema de asignación de tres dimensiones a un problema de programación bilineal y así a un problema de programación cóncava especial.

En general, los problemas de programación bilineal son equivalentes al problema de minimización cóncava. Como puede observarse el problema de minimización global cóncava (P) incluye un amplio rango de problemas de interés.

Desde el punto de vista de la complejidad computacional el problema (P) pertenece al tipo de problemas NP-duros, aún para casos especiales como el de minimizar una función cóncava cuadrática sobre un hipercubo unitario; ver por ejemplo Mangasarian [61], Garey [37] y Hammer [43]. En contraste, el problema convexo (cuadrático) correspondiente puede resolverse por algoritmos cuyo tiempo de ejecución es polinomial, Koslov [55].

Los métodos mas generales para optimización global se pueden dividir en dos clases: determinísticos y estocásticos, en este trabajo se considera sólo el caso determinístico.

Los métodos determinísticos más importantes para problemas de programación cóncava global son: el uso de técnicas enumerativas, métodos de planos de corte, ramificación y acotamiento, solución de subproblemas aproximativos, métodos de programación bilineal o combinaciones diferentes de esas técnicas. Existen algunos métodos específicos para problemas donde la función objetivo tiene una estructura especial (cuadrática, separable, factorizable, etc) o el conjunto factible tiene una geometría simplificada (hipercubo unitario, restricciones de red, etc).

1. **Métodos Enumerativos que Ordenan los Puntos Extremos:** Una propiedad importante de las funciones cóncavas es que cada solución local

y global se alcanza en algún punto extremo del dominio factible (ver anexo A). Esta propiedad hace al problema más tratable, ya que la búsqueda de una solución óptima se puede restringir al conjunto de puntos extremos, aún cuando este conjunto en general puede ser muy grande para manejarlo.

Una forma obvia de resolver el problema de programación cóncava en el caso en que el dominio factible es un conjunto poliédrico, es la enumeración completa de los puntos extremos. Aunque la mayoría de los algoritmos en el peor caso degenerarán en una inspección completa de todos los vértices del poliedro, este método será computacionalmente infactible para problemas grandes.

Cabot y Francis [22] presentan el siguiente procedimiento para resolver el problema de programación cuadrática cóncava: Se resuelve primero un programa lineal donde la función objetivo lineal es una aproximación de la función objetivo original, y se aplica a esta función el criterio de Murty dado a conocer en 1969, para ordenar los puntos extremos. Taha [76] también usó la idea de ordenar los puntos extremos basándose en funciones de aproximación lineal inferior.

Técnicas generales para enumeración total de vértices se presentan en Balinski [7], Benson [12], Burdet [21], Dyer y Proll [28], Manas y Nedoma

[60] y Rossler [71]. Un estudio y comparación de los métodos para encontrar todos los vértices de un conjunto poliédrico está dado por Dyer [28]. La efectividad computacional de los algoritmos de ordenamiento de puntos extremos se discuten en McKeown [59].

2. Métodos de Planos de Corte y Tecnicas de Partición del Dominio

Factible: H. Tuy [80] fue de los primeros autores en considerar el problema de programación global cóncava. Su método está basado en el uso de cortes (cortes de Tuy) para excluir partes del dominio factible, y un procedimiento de separación de un cono. Se puede pensar al conjunto factible como si estuviera contenido en un cono generado por las orillas coincidentes en un vértice. El método resuelve una sucesion de subproblemas generados con base en este cono inicial.

Zwart [89] probó que el algoritmo de separación del cono de Tuy no es convergente. En el mismo artículo Zwart da un contraejemplo a un método desarrollado por Ritter [69]. En el método de Ritter, el dominio factible se reduce por el uso de planos de corte en forma secuencial. Zwart da un ejemplo de cuándo es necesaria una sucesión infinita de planos de corte.

Bali [6] y Zwart [89] proponen algunas modificaciones al método original de Tuy. El algoritmo de Zwart es computacionalmente finito y está diseñado

para converger rápidamente en problemas donde hay pocos óptimos locales o el óptimo global es significativamente mejor que la mayoría de los otros óptimos.

Jacobsen [51] dio una prueba de un algoritmo de minimización general cóncava. También Thoai y Tuy [78] propusieron una clase de algoritmos que están basados en una combinación de técnicas de ramificación y acotamiento con el método original de Tuy.

El corte de Tuy fue generalizado por Glover [39], dando origen a los cortes convexos. Además, Glover describe un nuevo método para la generación de cortes (búsqueda de cortes). Otros algoritmos basados en el método de Tuy son el de Krynski [56], Bullatov [20]. Y una discusión en programación cóncava y programación entera se puede encontrar en Glover [30].

Rosen [70] también trabaja sobre la minimización global de una función cóncava a través de particionar el dominio factible, esto se hace particionando el dominio en un dominio rectangular, el cual se puede excluir de mayor consideración, y posteriormente se toman $r \leq 2n$ subdominios, en donde al menos en alguno de ellos se encuentra el mínimo global. Se puede aplicar un algoritmo secuencial o paralelo para buscar en esos r subdominios el óptimo global.

Bolintineanu [15] trata sobre el problema de programación no convexa de minimización de una función cuasi-cóncava sobre un conjunto eficiente (o débilmente eficiente) de un problema de programación lineal multiobjetivo. Se resuelve usando algún algoritmo de planos de corte que encuentra una solución óptima aproximada en un número finito de pasos.

Finalmente, cabe mencionar que los Planos de Corte se pueden considerar como parte fundamental de un algoritmo para resolver el problema del Agente Viajero a Gran Escala como en Grotschel y Holland [42], donde la parte central del algoritmo la constituye un método de Planos de Corte.

3. Métodos de Ramificación y Acotamiento que usan aproximaciones de la función objetivo: Falk Y Soland [30] consideraron el problema general de minimización global (no convexa) separable. El algoritmo hace uso de envolturas convexas. Dichas envolturas fueron propuestas por primera vez por Kleinbohm en 1967. La idea general consiste en construir una envoltura convexa a la función objetivo original; como la función es separable, se construye una función envolvente para cada variable. El algoritmo propuesto aquí es del tipo ramificación y acotamiento, y resuelve una sucesión de subproblemas en cada uno de los cuales la función objetivo es lineal o es convexa. Estos problemas corresponden a sucesivas particiones del dominio factible. También se presentan dos reglas que permiten la convergencia del

algoritmo dependiendo de los requerimientos de las funciones en cada caso. Este mismo algoritmo fue extendido más tarde por el mismo Soland [75] para trabajar con restricciones no convexas separables.

Falk [30] propuso un algoritmo para el problema no convexo separable, mismo que Grotte codificó. Este algoritmo proporciona soluciones aproximadas reemplazando cada una de las funciones originales por su envoltura convexa lineal a trozos. Un procedimiento de ramificación y acotamiento resuelve primero este problema aproximado para dar estimaciones del valor óptimo de la función de aproximación, y entonces establece problemas si las estimaciones no alcanzan una solución global. Este método se usa también para resolver el problema lineal complementario.

En general la construcción de la envoltura convexa de una función continua es un problema muy difícil computacionalmente, y las reglas de refinamiento dadas por el algoritmo de Falk y Soland son tales que ninguno de los teoremas de convergencia se cumple si se usan simplemente subfunciones en lugar de cubiertas convexas. Horst [47] propuso un método diferente para el problema no convexo separable. En lugar de considerar las envolturas convexas y minimizar varios subproblemas convexas, sólo minimiza cada $f_j(x_j)$ sobre subintervalos convenientes. Da también un método heurístico para obtener el mínimo global en una variable.

En un artículo anterior, Horst considera el siguiente problema general no convexo:

$$\min \{f(x) : x \in D \subseteq \mathbb{R}^n\}$$

D compacto y $f(x)$ una función continua.

Donde propone un algoritmo del tipo ramificación y acotamiento, el cual resuelve una sucesión de subproblemas en cada uno de los cuales tiene una función objetivo convexa o incluso lineal. La diferencia entre este método y los anteriores, por ejemplo Falk [30], es el uso de particiones generales compactas en lugar de rectangulares y una regla de refinamiento diferente tal que el algoritmo no descansa en el concepto de cubierta convexa y trabaja con funciones no separables.

Se demuestra que al menos un punto de acumulación de la sucesión $\{z_k\}$ generada por el algoritmo, resuelve el problema. En un artículo posterior del mismo autor, Horst [48] demuestra que cada punto de acumulación de la sucesión $\{z_k\}$ resuelve el problema.

Benson [12] presenta una nueva propiedad de convergencia para cada uno de los dos algoritmos de ramificación y acotamiento para problemas de programación no convexa desarrollados por Falk-Soland y Horst. Para cada algoritmo, se ha demostrado previamente que bajo ciertas condiciones, siempre que el algoritmo genere una sucesión infinita de puntos, al menos un punto de acumulación de la sucesión es un mínimo global. El autor de este artículo demuestra que bajo esas condiciones todo punto de acumulación de tal sucesión es un mínimo global. En el mismo artículo se presenta un algoritmo prototipo de ramificación y acotamiento que además usa subproblemas de aproximación inferior. Los algoritmos de Falk-Soland y Horst son casos especiales de este algoritmo prototipo. Véase también en Benson [13]. Un algoritmo similar a éste fue propuesto por Reeves [68].

Algunos métodos para resolver problemas no convexos separables a través de métodos primales duales se discuten en Everett [29] y Bertsekas [14]. En este último artículo se reportan resultados computacionales.

Un método nuevo diseñado para minimizar globalmente una función cóncava sobre un politopo lo describen Falk y Hoffman [32]. Este método está basado en la tesis de Hoffman [44]. Es un algoritmo finito que no involucra cortes de la region factible, que genera una sucesión de subproblemas lineales y que garantiza que termina en una solución global. Este método

usa efectivamente funciones lineales de aproximación inferior. La desventaja del algoritmo es que los subproblemas lineales crecen en tamaño de una iteración a otra. Sin embargo, el número máximo de renglones involucrado en cualquier subproblema está limitado por el número de restricciones del problema original y el crecimiento toma lugar fundamentalmente al agregar nuevas columnas. Pero nuevamente el número total de subproblemas que se tienen que generar está limitado por el número de restricciones del problema original. Resultados de cómputo también se discuten en este artículo. Una extensión del algoritmo para minimizar funciones cóncavas sobre un conjunto convexo se da en Hoffman [45].

Hoffman [45] genera programas lineales cuyas soluciones minimizan la envoltura convexa de la función original sobre politopos sucesivos que encierran cada vez más a la región factible. Este método es una extensión del algoritmo de Falk-Hoffman [32], ambos algoritmos encierran a la región factible en un poliedro y generan una cota inferior para la función objetivo desarrollando minimizaciones sobre ese poliedro. Este proceso continúa hasta que la región que encierra el poliedro es suficientemente cercana a la región factible, para así exhibir una solución global al problema original. Desgraciadamente, el algoritmo requiere mucho almacenamiento y tiempo de computadora.

Beale y Forrest [8], proponen un método diferente para optimización global. De hecho, describen una nueva metodología que permite una aproximación arbitrariamente cercana a la función objetivo. Una forma de hacerlo es usando "conjuntos especialmente ordenados". Introducidos por Beale y Tomlin [10] y revisados por Beale y Forrest [8]. Estos se aplican a sumas de funciones no lineales de una variable, pero se pueden adaptar a productos usando logaritmos o alguna otra transformación. Una extensión de los "conjuntos especialmente ordenados", los "conjuntos ligados ordenados", permiten el uso de productos en forma directa.

En Beale [9] los métodos de ramificación y acotamiento se usan para encontrar un óptimo global de una función definida como suma de productos de funciones de argumentos únicos. Estos métodos se puede incorporar como extensiones a programación entera en programación matemática general. Aquí la implantación usa conjuntos ordenados especiales y ligados, los resultados computacionales que se reportan son usando el sistema de programación matemática SCICONIC

Mc. Cormick [57] considera el problema de minimización global para problemas de programación no lineal que son factorizables. (los detalles de las funciones factorizables se pueden encontrar en Mc. Cormick [58]). En este artículo se describe un método para obtener programas depurados de

aproximación inferior convexa que se usan para excluir de consideración las regiones donde se sabe que el valor de la función excede el mínimo global. Es importante en este método el uso de la cubierta convexa de una función en una sola variable. La eficiencia de los métodos depende de la bondad de los problemas de aproximación inferior.

Una vision general del problema se encuentra en Mc. Keown [59], este artículo se refiere a problemas de minimización de una función cóncava sobre un poliedro convexo pero donde las soluciones no se restringen a ser enteras.

Parker y Sahinidis [46] resuelven el problema de minimización cóncava separable usando un algoritmo de ramificación y acotamiento combinado con técnicas de reducción del dominio para acelerar la convergencia del algoritmo. Para enfatizar la importancia de las técnicas de reducción del dominio, algunas veces se refieren a este algoritmo como de Rama y Reducción. El algoritmo determina cotas de cada subproblema cóncavo, resolviendo el problema lineal relajado. La función lineal de aproximación no es otra que la envoltura convexa de la función original, tomada en cada variable como en Falk y Soland. Este algoritmo se implantó usando el paquete BARON, desarrollado por el propio grupo de optimización de la Universidad de Illinois y dirigido por Nikolaos Sahinidis. Los resultados

obtenidos son muy prometedores, ya que se puede resolver un problema típico de expansión de un complejo químico industrial para maximizar el valor presente neto. Dicho problema, que tradicionalmente se ha planteado como un problema entero mixto multiperiodico, se reformula como un problema de cargo fijo cóncavo, que resuelto con este algoritmo requiere de un tercio del tiempo necesario para resolver un problema entero mixto usando OSL.

4. **Algunos Algoritmos de Aproximación General:** En un artículo de Tuy [81] se considera el problema general cuando la región factible es cualquier subconjunto convexo cerrado de R^n . Los métodos discutidos involucran ideas que tienen como base el método de planos de corte de Kelly [52] para resolver programas convexos, y pueden llamarse "métodos de aproximación outer". Básicamente, estos métodos consisten en aproximar el conjunto de restricciones dado por un conjunto poliédrico convexo que lo contiene, cuyos vértices son conocidos o se pueden calcular prácticamente. Entonces el mínimo de la función objetivo sobre este poliedro da una solución aproximada al problema original. Si la exactitud de la solución alcanzada no es aún satisfactoria, la aproximación se refina aún mas, y el proceso completo de aproximaciones sucesivas se puede arreglar de tal forma que converja a una solución óptima del programa original. Tales procedimientos

de aproximación se han combinado con otras técnicas de aproximación inferior en los algoritmos de Falk-Hoffman [32] y Hoffman [45].

Tuy, et.al. [85], resuelve un problema de transporte-producción con costos de producción cóncavos y donde la función objetivo no es separable. El algoritmo que ellos proponen se basa en un método paramétrico que toma ventaja de la estructura especial del problema: la monotonicidad de la función objetivo a lo largo de ciertas direcciones, una pequeña proporción de variables no lineales y propiedades combinatorias dadas por las restricciones de transporte.

El problema original se convierte en un problema de transporte lineal que depende de un parámetro k -dimensional. Este problema de transporte paramétrico también se puede obtener a través de fijar temporalmente las variables no lineales y omitiendo en la función objetivo la función asociada a ellas. Como se sabe de la teoría de programación lineal paramétrica, el valor óptimo de dicha función es una función afín convexa y lineal a trazos, sin embargo debido a la concavidad de la función original, se puede buscar la solución óptima sólo en los vértices de todas las piezas lineales de la función parametrizada. Exploando la estructura de red, se puede observar que el número de piezas lineales es polinomial. En este artículo resuelven un

problema para $k = 4$, donde k representa el número de fábricas desde las cuales se enviarán artículos hacia 10 almacenes.

En un artículo previo, Tuy et.al. [82] resuelven el mismo problema pero con $k=3$. Cabe aclarar que el algoritmo es diferente en este caso, pero ellos mismos reportan que en el caso de $k>5$ el problema se hace muy difícil y que es muy eficiente para k pequeñas. Para k grandes sugieren que la función objetivo sea separable, y así poder hacer uso de algoritmos de descomposición.

Un procedimiento diferente de aproximación es el desarrollado por Mukhamediev [62]. Aquí se propone un nuevo algoritmo de aproximación, que permite obtener una solución óptima aproximada con cierta exactitud. La idea del método propuesto es aproximarse desde el interior (aproximación interna) al conjunto de soluciones factibles a través de politopos. Bajo algunas suposiciones adicionales en la función objetivo, se prueba que después de un número finito de pasos el algoritmo genera una solución aproximada al global óptimo.

5. **Métodos de Programación Bilineal:** El problema de programación bilineal en su forma general se puede establecer como sigue:

$$\min \text{ global } \{f(x,y):x \in X, y \in Y\}, \quad (\text{BLP})$$

donde X, Y son poliedros en R^n y R^m respectivamente y $f(x,y)$ es una función bilineal. (BLP) pertenece a la clase de problemas de programación matemática que tienen varios mínimos locales. Un ejemplo importante de una función bilineal es $f(x,y) = c^T x + d^T y + x^T Q y$. Thieu [78] prueba la equivalencia de la programación bilineal y la minimización cóncava bajo restricciones lineales.

Una amplia cantidad de autores reconocen una cercana relación entre los algoritmos de programación bilineal y los algoritmos existentes para minimización cóncava. Entonces los algoritmos especializados (BLP) se pueden adaptar para programación cóncava cuadrática. Konno [53] considera el problema de minimización cóncava cuadrática sujeto a restricciones lineales. Primero da la equivalencia entre este problema con el problema bilineal asociado. De hecho prueba que los dos problemas

$$\max 2c^T x + x^T Qx, \quad (1)$$

sujeto a

$$Ax = b, x \geq 0$$

donde $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, $Q \in \mathbb{R}^{n \times n}$ es una matriz simétrica positiva definida y

$$\max f(x_1, x_2) = c^T x_1 + c^T x_2 + x_1^T Qx_2 \quad (2)$$

sujeto a

$$Ax_1 = b,$$

$$Ax_2 = b, x_1, x_2 \geq 0,$$

son equivalentes en el siguiente sentido: Si x' es una solución óptima del problema (1), entonces (x, x') es una solución óptima del problema (2). Inversamente, si (x_1', x_2') es óptima para el problema (2) entonces ambos x_1' x_2' son óptimos para (1). Para resolver el problema de programación cóncava se usa el método de planos de corte para el (BLP) desarrollado por Konno. Problemas y métodos relacionados con esto se pueden encontrar en Konno [54].

Czochralska [25] considera un método similar usando programación bilineal. El primer artículo trata con el problema general (BLP) donde el conjunto

factible no es necesariamente acotado. Se presentan las propiedades más importantes del (BLP) así como los fundamentos teóricos. El método busca una solución óptima entre aquellas soluciones básicas que son puntos de equilibrio del problema. El algoritmo se ilustra a través de ejemplos numéricos. Este algoritmo de programación bilineal para programación cuadrática está ampliamente simplificado, como consecuencia de la estructura del problema y por el hecho de la verificación de las condiciones de necesidad y suficiencia para la existencia de una solución óptima se reduce a la solución de un programa de programación lineal.

Algunos otros artículos de programación bilineal son Vaish [90], Altman [1] y Gallo [36]. De hecho el algoritmo propuesto por Gallo y Ulkulu en el último artículo es similar al método de Tuy para programación cóncava.

Cabe aclarar que aunque se han propuesto muchos métodos para el problema de minimización global cóncava, sólo unos cuantos han sugerido algoritmos que se han programado y probado. Los resultados computacionales reportados en la literatura antes de 1980 están limitados al caso en que el número de variables es menor o igual que 25. Problemas más grandes se han resuelto en el caso de programación cóncava cuadrática a gran escala. Los pocos métodos que se han implantado son para funciones con estructura especial, tales como funciones cóncavas separables, Parker y

Sahinidis [66], o cuadráticas, emplean técnicas de ramificación y acotamiento junto con problemas de aproximación lineal (a trozos) inferior.

Con base en estos antecedentes, podemos ubicar entonces nuestro problema dentro de la optimización global sin restringir si la función objetivo es separable o no y con la restricción adicional de variables enteras y proponer un algoritmo que use cortes convexos, ramificación y acotamiento, y una función lineal de aproximación para su solución.

CAPITULO 3

METODOLOGIA

"Dejo a los varios porvenires (no a todos) mi jardín de senderos que se bifurcan. Casi en el acto comprendí; el jardín de senderos que se bifurcan era la novela caótica, la frase varios porvenires (no a todos) me sugirió la imagen de la bifurcación en el tiempo, no en el espacio. La relectura general de la obra confirmó esa teoría. En todas las ficciones, cada vez que el hombre se enfrenta con diversas alternativas, opta por una y elimina a las otras; en la del casi inextricable Ts'ui Pên opta -simultáneamente- por todas. Crea, así, diversos porvenires, diversos tiempos que también proliferan y se bifurcan".

Jorge Luis Borges

3.1 EL ALGORITMO

Ya se planteó en el Capítulo 1 el problema que se resuelve en este trabajo, en el presente capítulo se expone la metodología que se requiere para implantar el algoritmo, así como su complejidad computacional. También se presenta un algoritmo derivado del primero que resuelve el problema de producción-transporte con costos cóncavos.

A continuación se presenta una versión muy general del algoritmo sólo para establecer las ideas que están detrás del mismo, posteriormente se describe en forma más detallada. Se exponen también los conceptos teóricos más importantes a considerar de la teoría de convexidad, sobre funciones

lineales y sus propiedades en cuanto a aproximación se refiere; los cortes convexos y la forma en que se generan, así como los criterios de selección de la variable a ramificar y la estrategia de búsqueda en el árbol de ramificación y acotamiento; la manera en que se decida cuándo detener la ramificación y, sobre todo, la comparación de las diferentes cotas.

Ramificación y Acotamiento es uno de los métodos mas usados en programación entera. Aquí, la ramificación se refiere a sucesivos particionamientos del poliedro convexo, una vez que no se encontró la solución óptima entera a través de los planos de corte. El acotamiento se refiere al cálculo de las cotas inferior $f(x)$ y superior $\bar{f}(x)$ respectivamente para encontrar el óptimo global.

ALGORITMO DE RAMA Y CORTE PARA EL PROBLEMA DE MINIMIZACION CONCAVA

PASO 1 (Establecimiento de una función lineal de aproximación inferior).
Establezca una función objetivo lineal $l(x)$ que aproxime inferiormente a la función objetivo original, tal que:

$$l(x) \leq f(x) \quad x \in Q$$

PASO 2 (Solución del Programa Lineal Relajado)

Resuelva el programa lineal

$$\min_x \{ \ell(x) \mid x \in Q \}$$

Sea x^0 ($i = 0$) donde $i = 0, 1, 2, \dots, n$ el punto extremo óptimo.

Defina $\underline{f}(x^0) = \ell(x^0)$ como una cota inferior del valor óptimo de la función objetivo original. Sea $x^* = x^0$. Entonces $f(x^*) = f(x^0)$ valuada en la función original es una cota superior.

Si la solución es entera el procedimiento termina es la óptima. Si no, construya un árbol de búsqueda comenzando con el primer nodo que representará la solución del programa lineal relajado y con una cota inferior dada por $\underline{f}(x^0)$. Haga $i = 0$ y vaya al paso 3.

PASO 3 (Ordenamiento de los Puntos Extremos)

Las actuales cotas superior e inferior están dadas por $f(x^*)$ y $\underline{f}(x^i) = \ell(x^i)$, respectivamente. Sea x^{i+1} el siguiente punto extremo ordenado de Q , y establezca la nueva cota inferior $\underline{f}(x^{i+1}) = \ell(x^{i+1})$ y vaya al Paso 4.

PASO 4 (Comparación de Cotas Inferiores y Superiores)

- a) Si $\ell(x^{i+1}) \geq f(x^*)$ y x^* es entera, pare, x^* es el óptimo,
- b) Si $\ell(x^{i+1}) \geq \bar{f}(x^*)$ pero x^* no es entera vaya al paso 6, de otra forma:
- c) Si $\ell(x^{i+1}) < \bar{f}(x^*)$, vaya al paso 5.

PASO 5 (Adición de Cortes Convexos)

Verifique si $f(x^{i+1}) < \bar{f}(x^*)$ y haga $x^* = x^{i+1}$, $\bar{f}(x^*) = f(x^{i+1})$, de otra forma la cota superior permanece sin cambios, haga $i = i + 1$.

Agregue el corte:

$$\sum_{j \in N} \frac{t_j}{t_j^*} \geq 1$$

N = variables no básicas

$$\text{donde : } t_j^* = \begin{cases} \min_{i \in M} \frac{b_{i0}}{b_{ij}}, & b_{ij} > 0 \\ \infty, & b_{ij} \leq 0 \end{cases}$$

M = variables básicas

y vaya al paso 2.

Si la solución encontrada es entera, compare el valor con las cotas que se establecieron en el paso anterior, si cumple con la condición de optimalidad pare, se ha encontrado el óptimo, si no es entera y no cumple con esta condición, agregue otro corte para buscar otro punto extremo y vaya al paso 2.

PASO 6 (Ramificación de un nodo)

A partir de la mejor función objetivo, entendida como la mínima cota superior dada por el paso 4, seleccione una variable x^{l+1} para ramificar.

Selección del Nodo a Ramificar [5]

Mejor Cota. Se selecciona el nodo con el menor valor de la función objetivo. Siguiendo esta estrategia es posible evitar enumerar incesantemente porciones del árbol.

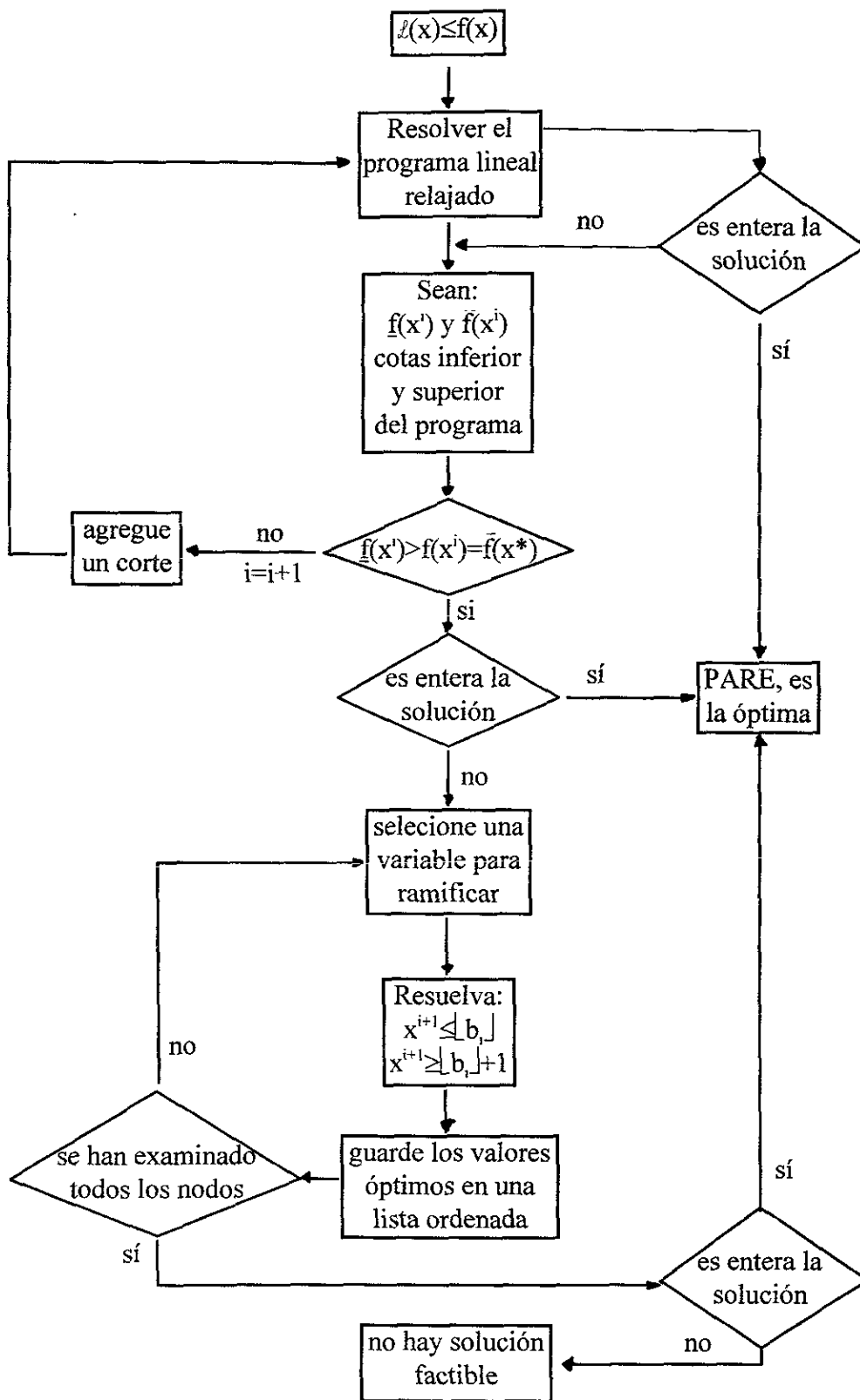
Selección de la variable.

Más Fraccional: la variable que se elige es aquella con valor fraccional más cercano a $1/2$. Se implanta esta regla fácilmente y se define como un conjunto de variables candidatas a ramificar como sigue:

$$l = \{ i \in (1, \dots, p) \mid |f_i - \frac{1}{2}| \leq \epsilon, \epsilon > 0, \text{ y suficientemente pequeña} \}$$

Se selecciona un variable de l , aquella con el mayor coeficiente (en valor absoluto) de la función objetivo. Las ramas están dadas por $\lfloor x_i \rfloor$ y $\lfloor x_i \rfloor + 1$. Continúe este proceso hasta encontrar una solución entera o que el problema no tiene soluciones factibles.

DIAGRAMA DE FLUJO DEL ALGORITMO



3.2 SOBRE EL USO DE FUNCIONES LINEALES

En esta sección se muestra cómo una función lineal $\ell(x)$ se puede desarrollar para aproximar inferiormente a la función cóncava original $f(x)$ en general. Ya que la eficiencia del algoritmo propuesto depende de la selección de dicha función lineal, los ejemplos de prueba muestran cómo se pueden desarrollar aproximaciones inferiores más estrictas para una clase importante de problemas de minimización cóncava. Esto incluye el problema de cargo fijo, el problema de programación separable, el problema cuadrático y el problema de producción-transporte.

De las propiedades de las funciones cóncavas, un hiperplano tangente para $f(x)$ en x [suponga $x \in Q$, donde Q es el poliedro convexo definido en (1)] sobreestima $f(x)$. Consecuentemente, parece plausible que podamos usar un hiperplano tangente para $g(x) = -f(x)$ para aproximar inferiormente $f(x)$. Sea $tg(x)$ un hiperplano tangente a $g(x)$ en un punto¹ dado. Claramente para cualquier x ,

$$tg(x) \leq g(x)$$

¹Mas aún, suponemos que el hiperplano tangente que está determinado en x satisface $\nabla g(x) \neq 0$ donde $\nabla g(x)$ es el vector gradiente de $g(x)$ en x . Esto asegurará que el estimador inferior resultante no es trivial.

Se puede hacer una transición de $g(x)$ a $f(x)$. Si $g(x) \leq f(x)$. Desafortunadamente, ésto no es cierto en general. Sin embargo, si los valores de x se restringen a aquellos en Q , entonces la transición se puede obtener como sigue:

Proposición 1: Sea $M \geq 0$ un número real, entonces existe un número M tal que $M < \infty$ y que además cumple con:

$$-M + tg(x) \leq f(x) \quad x \in Q$$

en este caso $l(x) = -M + tg(x)$

Demostración:

Necesitamos demostrar sólo que $-M + tg(x) \leq f(x) \quad \forall x \in Q$. El mínimo valor de $f(x)$ ocurre en un punto extremo de Q . Si $\min_{x \in Q} f(x) \geq 0$, entonces $\max_{x \in Q} g(x) \leq 0$, y obviamente el resultado deseado se cumple para $M = 0$.

Ahora suponga que $\min_{x \in Q} f(x) < 0$, entonces $\max_{x \in Q} g(x) > 0$.

Por la suposición de que $f(x)$ posee un mínimo finito sobre Q . Así M se puede seleccionar tal que:

$$M \geq | \min_{x \in Q} f(x) |$$

ya que por simetría:

$$| \min_{x \in Q} f(x) | = \max_{x \in Q} g(x)$$

se sigue que:

$$-M + g(x) \leq -M + \max_{x \in Q} g(x) \leq 0 \leq M + \min_{x \in Q} f(x) \leq M + f(x) \quad x \in Q$$

ya que:

$$M \geq | \min_{x \in Q} f(x) |$$

se puede tomar arbitrariamente grande, haciendo:

$$M \geq 2 | \min_{x \in Q} f(x) |$$

la conclusión se sigue inmediatamente.

La anterior proposición implica que una aproximación inferior lineal para $f(x)$, $x \in Q$ se puede tomar como:

$$l(x) = \sum_{j=1}^n m_j x_j - M$$

donde m_j son constantes positivas y:

$$M \geq \left| \min_{x \in Q} f(x) \right|$$

Si $\min f(x) \geq 0$ entonces M se puede tomar igual a cero.

Note que, como la cota inferior en M obviamente no es conocida a priori, debemos tener confianza en alguna estimación práctica para determinar un valor numérico para M . Aunque se pueden utilizar valores arbitrarios para $M_j > 0$, se verá mas adelante que la función que se escoja para aproximar a la función $f(x)$ dependerá del tipo de problema a resolver.

Proposición 2: (Mangasarian (1969); Avriel (1976))

a) Sea f una función diferenciable sobre un conjunto convexo abierto $C \subset \mathbb{R}^n$.

Entonces f es cóncava si y sólo si:

$$x^1 \in C, x^2 \in C \Rightarrow f(x^2) \leq f(x^1) + (x^2 - x^1)^T \nabla f(x^1) \quad (1)$$

b) Es estrictamente cóncava si y sólo si la desigualdad en (1) es estricta para $x^1 \neq x^2$. Por a) para una función (estrictamente) convexa, el sentido de la desigualdad (estrictamente) en (1) se invierte.

Demostración del inciso a.

Suponga que $f(x)$ es cóncava en S , $x_1, x_2 \in S$ y que $0 < \lambda \leq 1$. Entonces:

$$f(x_1 + \lambda(x_2 - x_1)) = f(\lambda x_2 + [1-\lambda]x_1) \geq \lambda f(x_2) + [1-\lambda]f(x_1),$$

tal que:

$$\left[\frac{f(x_1) + \lambda(x_2 - x_1)}{\lambda} - f(x_1) \right] \geq f(x_2) - f(x_1)$$

La expresión en el lado izquierdo de la anterior desigualdad aproxima $\|x_2 - x_1\|$ veces la derivada direccional de $f(x)$ en x_1 , en la dirección $x_2 - x_1$ a medida que λ se aproxima a cero. Haciendo $\lambda \rightarrow 0$, obtenemos:

$$\nabla f(x_1) (x_2 - x_1) \geq f(x_2) - f(x_1),$$

entonces:

$$f(x_1) + \nabla f(x_1) (x_2 - x_1) \geq f(x_2) \quad \forall x_1, x_2 \in S$$

Inversamente, suponga que:

$$f(x_1) + \nabla f(x_1) (x_2 - x_1) \geq f(x_2) \quad \forall x_1, x_2 \in S \quad (2)$$

Sean $u, v \in S$, y $0 < \lambda < 1$. Si $w = \lambda u + [1-\lambda]v$, entonces w está en S y:

$$v = w - \frac{\lambda}{1-\lambda}u = w - \frac{\lambda}{1-\lambda}(u - v)$$

tal que:
$$v-w = -\frac{\lambda}{1-\lambda} (u-w)$$

En consecuencia, si aplicamos (2) a las parejas w, u y w, v , obtenemos:

$$f(w) + \nabla f(w) (u-w) \geq f(u),$$

$$f(w) + \nabla f(w) (u-w) - \frac{\lambda}{1-\lambda} \geq f(v)$$

Multiplicando la primera de las desigualdades anteriores por λ , la segunda por $[1-\lambda]$, y sumando los resultados se obtiene:

$$f(\lambda u + [1-\lambda]v) = f(w) \geq \lambda f(u) + [1-\lambda]f(v)$$

De lo que se concluye que $f(x)$ es cóncava en S .

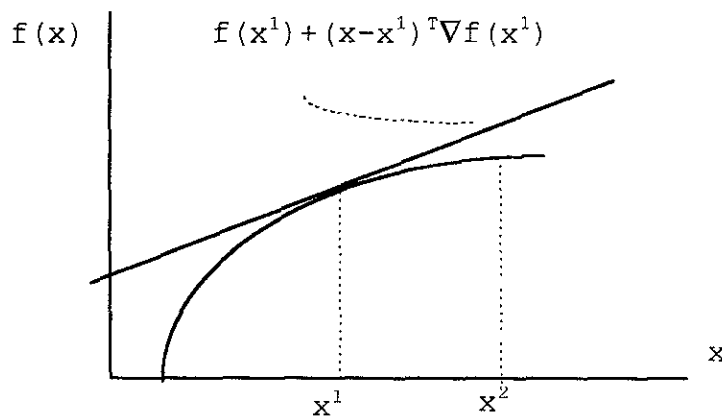


figura 1

La figura 1 ilustra la idea detrás de la Proposición 2: tomando cualquier punto x^1 en el dominio de f y construyendo una aproximación lineal (de Taylor de primer orden) de f en x^1 obtenemos una función lineal que es tangente a f en x^1 y en cualquier otro punto en el dominio de f que está sobre f o coincide con ella.

Proposición 3.

Sea $f: E^n \rightarrow E^1$ una función diferenciable. La aproximación lineal de f en el punto x_0 está dada por:

$$l(x) = f(x_0) + \nabla f(x_0) (x - x_0),$$

Esta función $l(x)$ se usa tal como se presenta para el caso del problema de producción-transporte expuesto en el capítulo 1. Para el caso de cargo fijo, se hace como sigue:

i) PROBLEMA DE CARGO FIJO.

En el problema de cargo fijo, $f(x)$ se define por:

$$f(x) = \sum_{j \in N} c_j x_j + \sum_{j \in N} k_j \delta(x_j) \quad N = (1, \dots, n)$$

donde $\delta(x_j) = 0$ si $x_j = 0$, y $\delta(x_j) = 1$ si $x_j > 0$. Los coeficientes c_j y k_j son números reales con $k_j > 0$ para toda j . Se puede probar que $f(x)$ es una función cóncava continua en todos los lados excepto en $x = 0$. Ya que supusimos $k_j > 0$, se sigue que:

$$\sum_{j \in N} c_j x_j \leq \sum_{j \in N} c_j x_j + \sum_{j \in N} k_j \delta(x_j) \quad x_j \geq 0$$

Esto demuestra que la aproximación lineal se puede tomar como:

$$l(x) = \sum_{j \in N} c_j x_j$$

En los siguientes casos, $l(x)$ se propone como la suma de las primeras derivadas parciales más una constante M apropiada.

ii) PROBLEMA DE PROGRAMACION SEPARABLE.

Sea:

$$f(x) = \sum f_j(x_j),$$

donde $f_j(x_j)$ es una función cóncava bien definida.

Sea:

$$l_j(x_j) = f_j'(x_j) + M$$

Como $f_j(x_j)$ es cóncava, por definición:

$$l_j(x_j) \leq f_j(x_j), \quad x_j \in Q$$

Entonces:

$$l(\mathbf{x}) = \sum l_j(x_j).$$

iii) FUNCIONES NO SEPARABLES.

En el caso de funciones no lineales de segundo grado y que tengan términos con productos cruzados como x_1x_2 antes de proponer una función lineal, se hace un cambio de variables para expresar esta función no separable en términos de una que si lo sea

Dada la función x_1x_2 :

se aumentan las restricciones lineales

$$y_1 = \frac{x_1 + x_2}{2}$$

$$y_2 = \frac{x_1 - x_2}{2}$$

se reemplaza cada término x_1x_2 por $y_1^2 - y_2^2$ como sigue:

$$x_1x_2 = y_1^2 - y_2^2.$$

Esto se justifica si notamos que:

$$y_1^2 - y_2^2 = \frac{x_1^2 + 2x_1x_2 + x_2^2}{4} - \frac{x_1^2 - 2x_1x_2 + x_2^2}{4} = \frac{4x_1x_2}{4} = x_1x_2 .$$

Lo anterior sugiere que cada que se tenga un producto de dos variables, se pueden agregar dos nuevas variables al modelo y reemplazar el término con el producto por una suma de dos variables al cuadrado. Si se tienen n variables originalmente, se pueden tener $n(n-1)/2$ términos con productos cruzados. Lo que sugiere que se podrían tener cuando mucho $n(n-1)$ nuevas variables para sustituir los productos cruzados. De hecho, bajo ciertas condiciones las ideas anteriores se pueden generalizar (usando la factorización de Cholesky) a que se agreguen solamente n nuevas variables.

iv) PROBLEMA CUADRÁTICO.

Para el problema cuadrático se usa la función anterior, una vez que se ha probado que existen todas las primeras derivadas parciales de $f(x)$ en el conjunto Q .

Sea: $f(x) = z(x) + D(x),$

donde $z(x)$ es lineal y $D(x)$ es una función cuadrática negativa (semi)definida.

La aproximación lineal en este caso es:

$$l(x) = z(x_0) + D(x_0) + \nabla D(x_0) (x-x_0).$$

3.3 EL PROBLEMA LINEAL RELAJADO

Casi todos los métodos para resolver problemas enteros comienzan relajando el problema a uno menos restringido, lo más usual es hacerlo a través de la eliminación momentánea de la restricción de variable entera y resolviendo el problema de programación lineal. Desgraciadamente, es raro que este problema relajado tenga una solución entera; por lo cual se toma su solución sólo como el paso inicial para un algoritmo más sofisticado.

Algunos conceptos en que se basa este procedimiento son:

Definición 1

Sean P_1 un problema de programación lineal entera y P_2 un problema de programación lineal sobre n variables con la misma función objetivo. Si la

región factible de P_1 , llamada $F(P_1)$ es un subconjunto de la región factible de P_2 , llamada $F(P_2)$, entonces P_2 se llama una relajación de P_1 .

Teorema 3

Si P_2 es una relajación de P_1 , entonces el valor de la función objetivo en la solución óptima factible para el problema P_2 es menor o igual que (o mayor o igual que, para problemas de maximización) el valor de la función objetivo óptima para P_1 .

Demostración:

La prueba de este teorema consiste en que al maximizar la función objetivo de P_2 , los argumentos para el rango de la función objetivo son sobre $F(P_2) \supseteq F(P_1)$, entonces P_2 es una relajación de P_1 .

Corolario 1

Sea P_2 una relajación de P_1 , si la solución óptima factible para P_2 está contenida también en $F(P_1)$ entonces también optimiza a P_1 .

Hemos visto un punto importante para la demostración del teorema 3. Si la región factible es acotada (es decir, cada variable satisface $x_i \leq u_i$, con u_i constante) y el problema es entero, entonces la función objetivo alcanzará su mínimo, ya que sólo hay un número finito de soluciones factibles.

Corolario 2

Sea un programa lineal formado de un programa entero al ignorar las restricciones de variable entera, si la solución óptima al programa lineal es entera, entonces también es óptima para el programa entero.

3.4 EL USO DEL METODO SIMPLEX EN EL PROBLEMA RELAJADO

El uso del método simplex en este algoritmo se fundamenta en lo siguiente:

1. Hay muchas aplicaciones industriales en que "un buen comienzo" desde una primer solución en una vecindad ligeramente perturbada de un problema de programación lineal proporciona una ventaja considerable. En esos casos, los algoritmos de punto interior tienen una ejecución pobre, en contraste con el simplex. Un ejemplo de tal aplicación lo constituye el uso de sucesivos problemas de programación lineal para resolver problemas de programación

no lineal en la industria del petróleo. La solución de subproblemas de programación lineal en programación entera y análisis postóptimo son otros ejemplos de la importancia de "un buen comienzo".

2. Los precios sombra (costos) juegan un papel importante en el análisis descriptivo de muchas aplicaciones de la programación lineal en la planeación económica y los negocios. En estas situaciones, la teoría establecida que relaciona los multiplicadores básicos óptimos del simplex y los valores de la solución dual tienen una amplia aceptación. Esto requiere que el método simplex se use para interpretar económicamente dichos costos.

El método simplex ha tenido un desarrollo sustancial en las últimas dos décadas. Este desarrollo ha sido fundamentalmente computacional y tiene el propósito de hacer al método simplex más eficiente y confiable con respecto a la solución de problemas prácticos de rango amplio. Se puede decir que otra ventaja de usar el método simplex es que, para problemas de gran escala, se puede usar preprocesamiento y, si la estructura del problema contiene matrices esparcidas, se explota esta condición y se modifica el simplex, dando como resultado un algoritmo que se conoce como método simplex para matrices esparcidas.

Una desventaja del simplex se presenta cuando se tienen problemas de degeneración. Una base se llama degenerada si al menos una de las variables básicas está en su cota superior o inferior. En tal caso el requerimiento de mantener factibilidad de las variables básicas puede forzar a cero a la variable que entra, esto mantiene el valor de la función objetivo sin cambio; teóricamente, conduce a ciclado a menos que se prevea esta situación y se evite.

La gran ventaja del método simplex es que, en el caso promedio, el algoritmo tiene un tiempo de ejecución polinomial.

En este algoritmo se hace uso del método simplex, para resolver el problema relajado, y del dual simplex, cuando se agregan los planos de corte. La tabla que se usa es una tabla del método simplex revisado para evitar que crezca mucho al agregar cortes.

3.5 EL ARBOL DE BUSQUEDA

Como este algoritmo es de rama y corte, se tiene entonces que construir un árbol de búsqueda para lo cual se considera qué política se debe seguir en cuanto a la selección del nodo a ramificar y la estrategia de búsqueda de soluciones en el árbol.

SELECCION DEL NODO A RAMIFICAR (MEJOR COTA)

En la literatura, existen varias reglas para seleccionar el nodo a ramificar. De hecho, debido a que un nodo se selecciona en cada paso, se puede ver la selección de nodos como una forma de ordenarlos en una lista de nodos no explorados. Las reglas usadas para ordenar y asignar los nodos son:

Búsqueda a primera profundidad (DFS).

Una regla usada ampliamente que se considera (esencialmente) *a priori* es la búsqueda a primer profundidad más un proceso de retroceso, también conocido como LIFO (último en entrar primero en salir). En la búsqueda a primer profundidad, si el nodo actual no se descarta, el siguiente nodo a considerar se selecciona entre sus dos hijos. Retroceso significa que cuando un nodo se descarta, regresamos por la trayectoria desde ese nodo hacia la raíz hasta que encontramos el primer nodo, si hay alguno, que tiene un hijo que no ha sido considerado.

La búsqueda de primer profundidad más retroceso es una regla completamente *a priori* si fijamos una regla para escoger las variables a ramificar y específicamente que el hijo izquierdo se considere antes que el hijo derecho.

La regla de primer profundidad tiene dos ventajas:

1. La relajación de programación lineal para un hijo se obtiene de la relajación de programación lineal de su padre al aumentar una restricción de cota superior o inferior. Entonces, dada una solución óptima para el nodo padre, podemos reoptimizar directamente usando el dual simplex sin una reinversión de base o una transferencia de los datos.
2. La experiencia parece indicar que las soluciones factibles se encuentran con mayor facilidad profundamente en el árbol que en los nodos cerca de la raíz. El éxito de un algoritmo de ramificación depende mucho de tener una buena cota inferior, para "podar" el árbol.

La principal dificultad es que la búsqueda puede hacerse en partes del árbol donde no hay buenas soluciones factibles, con lo que se incrementa el tiempo total.

Mejor Cota (BB).

Se selecciona un nodo que da el menor valor de la función objetivo. Esta regla se usó originalmente en el algoritmo de Land y Doig y fue retomada por Dakin [27]. El árbol se explora y hay más posibilidades de encontrar

buenas soluciones factibles. Sin embargo, la lista de nodos no explorados puede ser muy larga, aumentando con ésto los requerimientos de memoria. El razonamiento que se sigue de esta estrategia es que podemos evitar enumerar porciones innecesarias del árbol a través del análisis de las cotas.

Infactibilidad Entera (II).

Si x es una solución factible, la infactibilidad entera de x , denotada por $I(x)$, es:

$$I(\bar{x}) = \sum_{i=1}^p \min(|1 - \bar{x}_i|, |\bar{x}_i|).$$

En esta regla, el siguiente nodo que se selecciona es el que tiene menor infactibilidad entera.

SELECCION DE LA VARIABLE A RAMIFICAR

Si el nodo actual no se puede sondear, se tiene que seleccionar una variable tal que los dos nodos descendientes puedan generarse. La experiencia con ramificación y acotamiento indica que el tamaño de la enumeración final del árbol depende mucho de esta selección. Las siguientes son algunas de las reglas más utilizadas para seleccionar la variable a ramificar.

Variable Más Fraccional.

La variable seleccionada es aquella con un valor en su expansión decimal muy cercano a $1/2$. Los empates se rompen arbitrariamente. Esta regla se puede implantar fácilmente, pero su mayor desventaja es que no captura información de la función objetivo.

Penalizaciones.

Para cada variable fraccional se calcula el deterioro en la función objetivo haciendo la variable igual a cero o uno. Estos valores se llaman penalizaciones. Una posibilidad es tomar una variable para ramificar como aquella que tenga la mayor penalización, se ponen los descendientes correspondientes a la mayor penalización en la lista de los nodos no explorados y se resuelve otro descendiente inmediatamente. La principal desventaja de las penalizaciones es que son costosas de calcular y su relevancia decrece con el tamaño del problema.

Pseudo-costos.

Es similar a las penalizaciones, los pseudo-costos son aproximaciones a los valores de deterioro en la función objetivo al hacer la variable igual a 0 y 1.

ESTRATEGIA DE ENUMERACION

En todos los problemas de prueba se usa la regla de mejor cota (BB) para seleccionar el nodo. Esto se hace a través de enlistar los nodos en una cola en orden creciente del valor de la función objetivo y tomando siempre el primer nodo.

Si el nodo actual no puede descartarse, se selecciona un variable para ramificar. Para seleccionarla usamos el siguiente criterio simple: seleccionamos la variable más fraccional, aquella con expansión decimal cercana a $\frac{1}{2}$, y definimos el conjunto de variables para ramificar como:

$$I = \{y \in (1, \dots, 0) \mid |f_i - 1/2| \leq \varepsilon, \varepsilon \text{ suficientemente pequeña}\},$$

entonces seleccionamos a la variable en I con el mayor coeficiente (en valor absoluto) de la función objetivo.

3.6 CORTES CONVEXOS

Como ya se mencionó en el capítulo 1, el punto óptimo, si es entero, es un punto extremo. Esta afirmación se hace con base en el siguiente teorema:

Teorema 1

Sea $f: E_n \rightarrow E_1$ una función cóncava, y sea S un poliedro compacto no vacío en E_n y considere el problema de minimizar $f(x)$ sujeto a $x \in S$. Entonces existe una solución x' al problema y x' es un punto extremo de S .

Demostración:

Como f es continua y S es compacto, f alcanza un mínimo y un máximo en S . Supondremos que el mínimo se alcanza en x' y demostraremos que x' es un punto extremo de S . Si x' es un punto extremo, no tenemos nada que demostrar. Asumimos simplemente que x' es un punto de S , entonces como S es un poliedro convexo podemos suponer que $x' = \sum_{j=1}^k \lambda_j x_j$, donde $\sum_{j=1}^k \lambda_j = 1$, $\lambda_j \geq 0$ y x_j son los puntos extremos de S . Por la concavidad de f tenemos:

$$f(x') = f(\sum_{j=1}^k \lambda_j x_j) \geq \sum_{j=1}^k \lambda_j f(x_j).$$

Siendo que $f(x') \leq f(x_j)$ para $j = 1, \dots, k$.

Para poder ordenar los puntos extremos se hace uso de cortes convexos. Los cuales irán descartando cada vez que se agregan a un punto extremo, si ese punto es entero, se valúa en la función objetivo original y se compara con la función lineal de aproximación para ver si se cumple la condición de

optimalidad, si no se cumple se continúan agregando cortes hasta encontrar otro punto entero o hasta que la función objetivo ya no mejore.

El corte convexo es una forma general de corte entero legítimo que fue establecido inicialmente con el nombre de corte "hipercilíndrico" por Young [87] para resolver una clase especial de problemas mixtos cero-uno en que las variables cero-uno x_j están relacionadas por restricciones del tipo:

$$\sum_{j=1}^n x_j = k \quad y \quad 0 \leq x_j \leq 1, \quad \text{donde } 0 < k < n$$

La propiedad hipercilíndrica consiste en convertir las restricciones especiales en:

$$\sum_{j=1}^n x_j^2 = k$$

Independientemente del trabajo de Young, Balas [4] desarrolló lo que él llamó el corte "intersección" para resolver problemas lineales enteros (mixtos o puros). Su idea es utilizar la propiedad local de los puntos enteros que encierran o circundan la solución óptima local continua para desarrollar un corte legítimo entero. Actualmente, este corte está determinado para pasar a través de los puntos de intersección de una hiperesfera especial definida. La esfera pasa a través de todos los vértices del hipercubo unitario que encierra el punto extremo continuo.

Más tarde, Glover [39] desarrolló las ideas de Young y Balas a una clase más general de programas matemáticos incluyendo, por supuesto, programación entera. Su idea es que la hiperesfera (o hipercilindro) se puede reemplazar por cualquier conjunto convexo proporcionado de tal forma que ninguno de los puntos factibles esté en su interior. En este caso, la intersección de los segmentos de recta que emanan del vértice contínuo con el conjunto convexo define lo que Glover llama el corte "convexo", mismo que se define como:

$$\sum_{j \in NB} w_j / w_j^* \geq 1 \quad (\text{c-corte})$$

Se pueden hacer varias observaciones con respecto a la fuerza y convergencia del algoritmo de Balas:

1. Balas observa que el i -ésimo corte puede ser más fuerte en dos formas: La primera consiste en reemplazar el i -ésimo corte por su equivalente con enteros a través del empleo de la teoría de Gomory para desarrollar el λ -corte. La condición de variables enteras será atractiva sólo si produce un corte más fuerte, en el sentido de hacer los coeficientes apropiados menores. Uno puede notar también que no se requiere que el elemento pivote sea igual a -1, a menos que el tableau actual sea (cercanamente)

todo entero. Esto se sigue porque esta condición generalmente debilita el corte.

2. La segunda forma consiste en reemplazar la hiperesfera por un hipercubo dual unitario, que está definido como un cubo "rotado" en el que el vértice

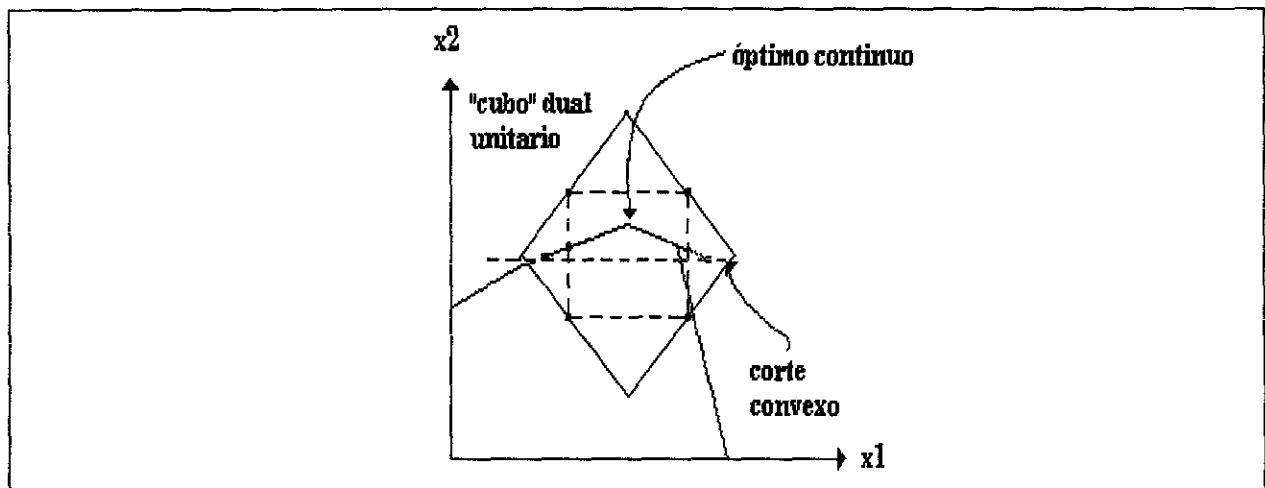


figura 2

de cubo unitario $[x^*] \leq x \leq [x^*] + e$, toca el punto medio de una cara del cubo dual. Un ejemplo gráfico de esta situación se muestra en la figura 2 para un problema de dos dimensiones que indica que el nuevo corte es más fuerte.

Un desarrollo general de Balas et. al. [4] prueba este resultado. De hecho, debe ser obvio que el conjunto R , como se definió en la teoría de Glover, se puede construir de varias formas para producir un corte mas fuerte [ver Balas (1971a)]. Sin embargo, la dificultad aqui estriba en que puede involucrar muchos cálculos.

Balas también observa que cuando $x = x^*$ es degenerada, el cubo unitario $[x^*] \leq x \leq [x^*] + e$ no es único. En este caso la selección específica de un cubo puede afectar la fuerza del corte. Balas muestra cómo puede seleccionarse el mejor cubo, desafortunadamente, a expensas de cálculos adicionales.

Finalmente, la cuestión de convergencia está dada. Balas desarrolla un argumento que asegura la finitud del algoritmo, basado fundamentalmente en: 1) El desarrollo de reglas que aseguran que el determinante de la base actual permanece finito cuando el número de pivotes tiende a infinito, 2) la función objetivo está acotada inferiormente, 3) los cortes se aumentan en forma entera, y 4) se usa el método I-dual para resolver el problema.

Como la teoría de Glover comprende las ideas de Young y Balas (de hecho, Glover muestra que el m-corte de Gomory se puede interpretar como un corte convexo) esta sección comienza discutiendo los resultados de Glover y se especializa posteriormente en casos específicos. Para efecto de apreciar la idea del nuevo corte, se usa un ejemplo gráfico, ver figura 4, como ilustración.

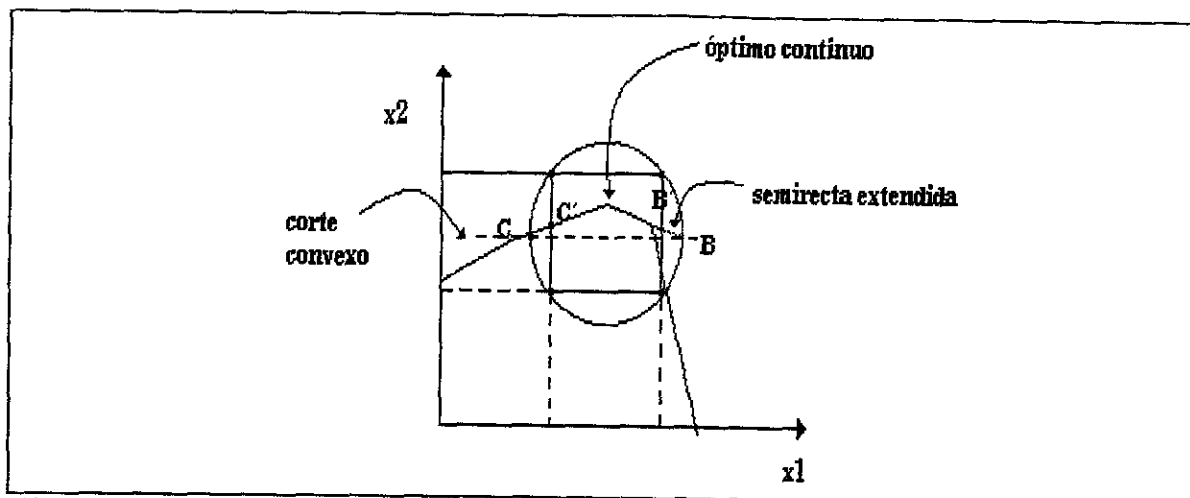


figura 3

Se suponen x_1, x_2 enteras. El vértice óptimo continuo actual es A. El cuadrado que encierra a A reemplaza al hiper cubo de Balas referido a la intersección del corte y el círculo que representa al corte convexo. Note que ningún punto entero factible está en el interior del círculo. Se puede ver que un conjunto convexo legítimo se puede construir en una amplia variedad de formas pero la selección específica del conjunto convexo está determinada por la fortaleza o debilidad del corte resultante.

Por ejemplo, el círculo se puede reemplazar por el cuadrado que encierra; en este caso el corte pasa por B' y C' en lugar de B y C, que es un corte más débil.

Es obvio, sin embargo, que un corte es más fuerte si un conjunto convexo que se ha escogido es tal que las semirectas (extendidas) que emanan del punto continuo son las más largas. En la figura A1: $AB > AB'$ $AC > AC'$, y así el corte BC es más fuerte que $B'C'$. Esta observación es la que hace a la teoría de Glover interesante, ya que permite flexibilidad al seleccionar el mejor conjunto que explota la estructura del problema. Los resultados generales de Glover se explican en seguida.

Sea el problema de programación matemática tal que el punto solución actual está definido por las restricciones:

$$x = x^* - \sum_{j \in NB} \alpha_j w_j$$

que se expresan en forma matricial con x , x^* y α_j vectores columna. En términos de un problema de programación lineal equivalente, x representa la variable básica, mientras w_j , $j \in NB$ son las variables no básicas asociadas iguales a cero.

El corte de Glover se describe en el siguiente lema:

LEMA DEL CORTE CONVEXO DE GLOVER

Sea S un conjunto de puntos factibles dados. Si R es un conjunto convexo cuyo interior no contiene puntos en S y si $x = x^*$ (posiblemente un conjunto acotado de R) tiene una vecindad agujerada factible que está en el interior de R , entonces para constantes cualesquiera $w_j^* > 0, j \in NB$ tal que los puntos:

$$\xi_j^* = (x^* - \alpha_j w_j^*) \in R \quad \forall j \in NB$$

el corte convexo:

$$\sum_{j \in NB} \frac{w_j}{w_j^*} \geq 1 \quad (\text{c-corte})$$

excluye el punto $x = x^*$, pero nunca cualquier punto en S .

Aunque el corte convexo desarrollado aquí se hace en el contexto de programación entera, la misma idea es aplicable a otros problemas de programación matemática. Notable entre ellos es la minimización cóncava sobre un poliedro convexo en que la solución óptima está caracterizada porque ocurre en un punto extremo, lo que mostrará cómo el problema de

cargo fijo, que es un caso especial de minimización cóncava sobre un poliedro, se puede resolver usando el corte convexo.

El uso del corte convexo con problemas enteros lineales se presenta ahora, también una interpretación del m-corte de Gomory en términos de la teoría de Glover.

INTERSECCION DEL CORTE DE BALAS

Se supone como usual que el método l-dual se use para obtener la solución, tal que las ecuaciones restricción asociadas con el tableau continuo actual se pueden escribir como:

$$x = x^* - \sum_{j \in NB} \alpha_j w_j \quad S = S^* - \sum_{j \in NB} \delta_j w_j$$

donde x es un vector entero y S el vector de holgura (continua). Note que, a diferencia del algoritmo fraccional, S no se restringe a tomar valores enteros. Sin embargo, si S o un subconjunto de él es entero, es necesario tener cuidado en el análisis subsecuente al incorporar los elementos apropiados de δ_j como parte de α_j , $j \in NB$.

Para propósitos de ilustrar el método, asumimos que el problema es del tipo entero puro.

El desarrollo de Balas se interpreta en términos de la teoría de Glover como sigue:

Sea S el conjunto de puntos enteros factibles definidos por los vértices factibles del hipercubo unitario, cuyos semiespacios cerrados $2n$ están dados por:

$$[x^*] \leq x \leq [x^*] + e$$

donde cada elemento del vector e es 1. Suponga temporalmente que x^* no tiene elementos enteros, entonces el hipercubo unitario está únicamente definido. La definición de S muestra que lo que concierne inmediatamente es concentrarse en los puntos enteros locales alrededor de x^* .

Como se mencionó previamente, ya que el hipercubo unitario se puede usar directamente para representar el conjunto convexo R , se puede obtener un corte más fuerte si este hipercubo unitario se reemplaza por una hiperesfera (circundante) cuyo centro es $[x^*] + e/2$ y cuyo radio es $n^{1/2}/2$.

La ecuación de esta hiperesfera está dada por:

$$x^T(e + 2[x^*] - x) = [x^*]^T (e + [x^*]).$$

Claramente, el nuevo conjunto R (hiperesfera) no contiene puntos enteros interiores factibles.

De acuerdo a la teoría de Glover, los valores de w_j^* que definen el corte son determinados desde la intersección de las semirectas.

Otra vez, suponga que $x = x^*$ es no degenerada, por lo que hay exactamente n semirectas distintas asociadas con $x = x^*$ (el caso de degeneración se tratará posteriormente). Entonces los puntos de intersección definidos por w_j^* están determinados haciendo $x = \xi_j$ en la ecuación de la esfera y resolviendo para w_j , esto es:

$$(x^* - w_j^* \alpha_j)^T (e + 2[x^*] - x^* + w_j^* \alpha_j) = [x^*]^T (e + [x^*])$$

después de algunas manipulaciones algebraicas, ésto da:

$$w_j^* = 1/p_j \{ h_j + [(h_j^2 + f^T(e - f) p_j)]^{1/2} \} \quad j \in NB$$

donde:

$$f = x^* - [x^*]$$

$$p_j = \alpha_j^T \alpha_j \quad (\text{producto interior de } \alpha_j)$$

$$h_j = (f - 1/2 e)^T \alpha_j,$$

El valor absoluto de la raíz cuadrada es consistente con el requerimiento de que $w_j^* > 0$. Se sigue que el corte de intersección de Balas es:

$$\sum_{j \in NB} w_j / w_j^* \geq \quad \text{Corte } i\text{-ésimo de Balas}$$

La expresión para w_j^* siempre está definida.

La base para el desarrollo anterior es que existen n segmentos de recta distintos. Sin embargo, si $x = x^*$ es degenerada, esto significa que el actual punto extremo está "sobredeterminado"; esto es, el vértice es la intersección de $n+k$ hiperplanos $k > 0$ mientras en realidad sólo n hiperplanos son necesarios para determinar el punto.

Balas se sobrepone a esta dificultad eliminando cada restricción en que la variable básica asociada está al nivel cero. Estas son las restricciones que

crean la situación degenerada. Entonces prueba que ésto debe estar en un conjunto de n segmentos de recta distintos que emanan de x^* . La prueba depende primariamente del hecho de que los vectores columna α_j son, por construcción, independientes. A pesar de que este método resuelve el problema de degeneración, el corte i -ésimo resultante puede ser más débil dependiendo del conjunto de restricciones eliminadas.

CORTE M DE GOMORY VISTO COMO CORTE CONVEXO

Sea x_k una variable entera cuyo valor x_k^* es actualmente no entero. El conjunto $[x_k^*] \leq x_k \leq [x_k^*] + 1$, que es obviamente convexo, puede usarse para reemplazar R en el lema de Glover. Los planos de acotación $x_k = [x_k^*]$ y $x_k = [x_k^*] + 1$ deben incluir todos los vértices del cubo unitario:

$$[x^*] \leq x \leq [x^*] + e$$

y la nueva R no tiene punto interior factible.

Cada una de las semirectas desde x^* intersectará, a lo más, uno de los dos planos $x_k = [x_k^*]$ y $x_k = [x_k^*] + 1$, ya que por definición:

$$[x_k^*] < x_k^* < [x_k^*] + 1$$

El plano de intersección se determina fácilmente como el que produce $w_j^* > 0$. Así, las semirectas asociadas con x_k^* son $x_k^* - \alpha_{kj} w_j$ y sus puntos de intersección están determinados por:

$$x_k^* - \alpha_{kj} w_j^* = [x_k^*]$$

ó:

$$[x_k^*] + 1$$

que produce:

$$w_j^* = \begin{cases} \frac{f_k}{\alpha_{kj}} & \text{si } \alpha_{kj} > 0 \\ \frac{f_k - 1}{\alpha_{kj}} & \text{si } \alpha_{kj} < 0 \\ \infty & \text{si } \alpha_{kj} = 0 \end{cases}$$

donde $f_k = x_k^* - [x_k^*]$. Sustituyendo para w_j^* en el c-corte, se obtiene directamente el m-corte.

3.7 COMPLEJIDAD COMPUTACIONAL DEL ALGORITMO.

Por muchos años, la programación entera no ha tenido muchas expectativas debido a que las formulaciones para problemas de gran escala, que son las que a menudo surgen de problemas reales, fueron difíciles de manejar,

principalmente por la falta de un adecuado poder computacional, y por la complejidad de los algoritmos disponibles en ese tiempo. Sin embargo, esto ha cambiado; aunque la programación entera es teóricamente "dura" (los problemas generales enteros y enteros mixtos pertenecen a la clase NP-dura), nuevos desarrollos en la tecnología algorítmica y computacional hacen posible resolver las aplicaciones de la vida real que sólo hace algunos años se consideraban intratables.

Definición 3

PROBLEMAS P : Es el conjunto de todos los problemas que pueden resolverse por algoritmos determinísticos en tiempo polinomial. Por determinístico entendemos que en cualquier tiempo, sin importar qué está haciendo el algoritmo, sólo existe una actividad subsecuente. Por no determinístico entendemos que un algoritmo se enfrenta a la selección de varias opciones y que tiene el poder de "adivinar" la opción correcta.

Definición 4

PROBLEMAS NP : (Donde NP se entiende por polinomial no determinístico). Es el conjunto de todos los problemas que se pueden resolver por algoritmos no determinísticos en tiempo polinomial.

Cualquier problema P pertenece a NP. Para demostrar que un problema está en NP sólo necesitamos encontrar un algoritmo de tiempo polinomial para verificar que dada una solución (la que se adivinó) es válida.

Definición 5

PROBLEMAS NP-Completos: Todos los problemas NP-Completos parecen requerir, como parte de su naturaleza, que probemos con combinaciones parciales, asignaciones verdaderas parciales, arreglos parciales, o coloraciones parciales, y extenderlos continuamente esperando alcanzar una solución final completa. Cuando una solución parcial no puede extenderse, aparentemente tenemos que regresar sobre nuestros pasos, es decir, deshacer cosas que ya estaban hechas, para poder probar una alternativa. Si este proceso se lleva a cabo con cuidado no se pierde ninguna solución posible, pero en el peor de los casos se requiere de un tiempo exponencial.

Entonces, dado un conjunto de datos iniciales para un problema NP-Completo, es extremadamente difícil contestar como respuesta "sí" o "no" a una pregunta. Sin embargo, es interesante que en todos estos problemas, si la respuesta es "sí", existe una forma sencilla de convencer a alguien de ello. Hay un llamado *certificado*, que contiene evidencia conclusiva de que en efecto la respuesta es "sí". Más aún, este certificado siempre puede

hacerse suficientemente corto, su tamaño siempre puede acotarse por un polinomio en N (de hecho, a menudo es lineal en N).

Por ejemplo, sabemos que es difícil decir cuándo una gráfica contiene una trayectoria Hamiltoniana, o cuando contiene una cuya longitud no es mayor que K . Por otro lado, si tal trayectoria existe, se puede exhibir y checar fácilmente que es del tipo deseado, lo que sirve como una excelente prueba de que la respuesta es "sí". En forma similar, aunque es difícil encontrar una asignación verdadera que satisfaga a una proposición en cálculo de proposiciones, es fácil certificar que la proposición se satisface, simplemente exhibiendo una asignación que la cumpla. Más aún, es fácil verificar en tiempo polinomial que esta asignación particular verdadera se cumple.

Es difícil darse cuenta cuándo un problema NP-C dice "sí" a un conjunto de datos, pero es fácil certificar que es cierto cuando sí lo es.

Técnicamente, decimos que tales algoritmos son no determinísticos, ya que ellos siempre "adivinan" cuál de las opciones disponibles es mejor, en lugar de tener que emplear algún procedimiento determinístico para resolverlos. De alguna manera, ellos siempre hacen la selección correcta. Si los algoritmos tuvieran la posibilidad de explotar tal no determinismo mágico,

podríamos mejorar las soluciones para ciertos problemas, ya que así se evitaría el trabajo asociado a éstos. Para los problemas NP-C, este mejoramiento no es del todo marginal: cada problema NP-C tiene un algoritmo no determinístico de tiempo polinomial. Este hecho se puede probar, mostrando que la certificación corta que se mencionó anteriormente corresponde directamente a ejecuciones "mágicas" de tiempo polinomial.

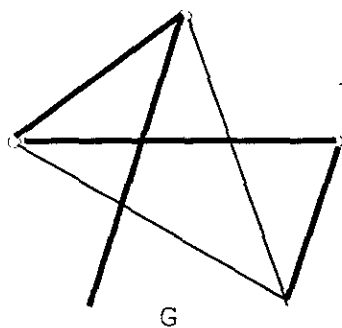
Así, los problemas NP-C son aparentemente intratables, pero se convierten en tratables usando no determinismo mágico. Esto explica parte del acrónimo NP-C (tiempo polinomial no determinístico) con NP para problemas que admiten certificaciones cortas, ahora veamos lo que se refiere a la parte de Completo.

El término completo se usa para denotar la siguiente propiedad: todo los problemas NP-C son tratables o ninguno de ellos. Más explícitamente, si alguien encontrara un algoritmo de tiempo polinomial para un problema NP-completo, habría inmediatamente algoritmos de tiempo polinomial para todos ellos, esto implica el siguiente hecho dual: si alguien probara una cota inferior de tiempo exponencial para cualquier problema NP-completo, estableciendo con ello que no puede resolverse en tiempo polinomial, se concluiría inmediatamente que no hay un problema NP-completo que pueda resolverse en tiempo exponencial.

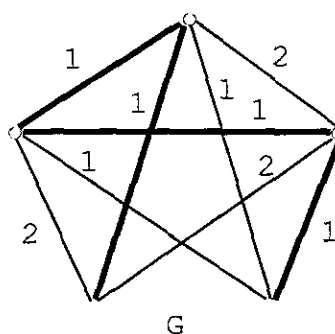
El concepto que se usa para establecer esta propiedad de estar o caer todos juntos es el de *reducción a tiempo polinomial*. Dados dos problemas NP-completos P_1 y P_2 , una reducción a tiempo polinomial es un algoritmo que corre en tiempo polinomial, y reduce un problema en otro.

Es fácil reducir por ejemplo el problema de trayectoria Hamiltoniana al problema del agente viajero. Dada una gráfica G con N nodos, construya una red de agente viajero G' como en la figura 4.

Los nodos de G' son los N nodos de G , y los arcos se dibujan entre cada par de nodos, asignando el costo 1 a un arco si estaba presente en G , y 2 si no lo estaba.



Trayectoria Hamiltoniana enfatizada



Recorrido del agente viajero de longitud 6, enfatizado.

figura 4

La figura ilustra la transformación. No es difícil ver que G' tiene un recorrido no mayor que $N + 1$ donde N es el número de nodos en G .

La respuesta a la primer pregunta en G es "sí", precisamente cuando la respuesta a la segunda en G' es "sí". Cabe resaltar que la transformación sólo toma una cantidad de tiempo polinomial.

Este hecho es interesante porque demuestra en términos de tratabilidad que el problema de una trayectoria Hamiltoniana no es peor que el problema del agente viajero. Si el último tiene una solución razonable entonces también el primero.

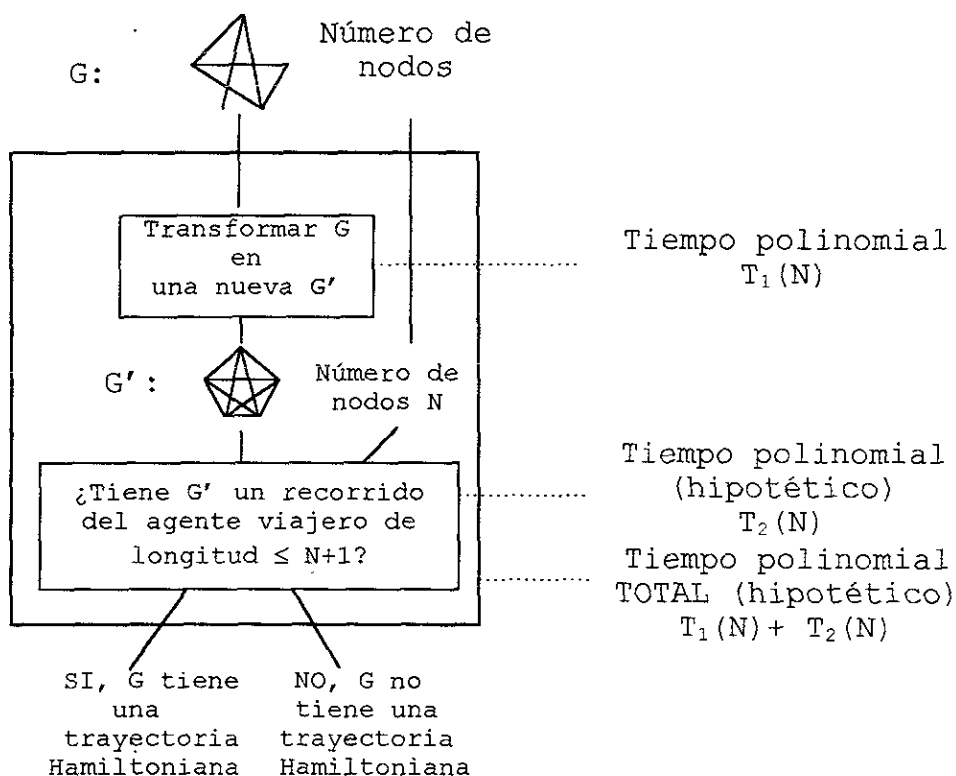


figura 5

La figura 5 ilustra cómo la reducción se usa para obtener un algoritmo razonable para una trayectoria Hamiltoniana desde un (hipotético) algoritmo razonable de agente viajero.

Ahora planteamos el hecho que establece el fenómeno de destino común de los problemas NP-completos: todo problema NP-completo es polinomialmente reducible a otro.

Consecuentemente la tratabilidad de uno implica tratabilidad de todos, y la intratabilidad de uno implica intratabilidad de todos. Esta afirmación se basa en el Teorema de Cook, establecido en 1971.

3.8 CONVERGENCIA DEL ALGORITMO

El análisis de la convergencia se hace con base en la Convergencia de los Planos de Corte Convexos y la convergencia de Ramificación y Acotamiento.

Convergencia de Cortes Convexos.

Los cortes convexos se usan para ordenar los puntos extremos. En el peor de los casos se necesitarán tantos cortes como puntos extremos haya en el

poliedro. Si hay n puntos extremos se requerirán n cortes convexos. Pero si hay un punto extremo entero, el proceso termina. Si no se encuentra tal punto, se usan las cotas superior e inferior para verificar si el vértice actual es mejor que el último, si no, el proceso termina. En el peor de los casos será necesario verificar n puntos extremos, por lo cual el algoritmo tiene un tiempo de ejecución de $O(n!)$.

Convergencia de Ramificación y Acotamiento.

Sea A una matriz racional de $m \times m$, y sea $b \in Q^m$, $c \in Q^n$. Se quiere resolver el siguiente problema de programación lineal entera:

$$\begin{array}{ll}
 \min & cx \\
 \text{s.a:} & \\
 & Ax \leq b \\
 & x \text{ entera}
 \end{array} \tag{1}$$

El método consta de un cierto número de iteraciones. En la primera etapa tenemos $\Pi_1 := \{P\}$, donde $P := \{x \mid Ax \leq b\}$. Suponga que en la etapa k tenemos la colección $\Pi_k = \{P_1, \dots, P_k\}$ tal que:

i) P_1, \dots, P_k son poliedros disjuntos por pares en \mathbb{R}^n (cada uno dado por un sistema de desigualdades lineales); (2)

ii) Todos los vectores enteros en P están contenidos en $P_1 \cup \dots \cup P_k$.

Entonces, determine:

$$\mu_j = \min \{cx \mid x \in P_j\} \quad \text{para } j = 1, \dots, k. \quad (3)$$

Estas μ_j se pueden encontrar con un método de programación lineal.

Seleccione j^* tal que:

$$\mu_{j^*} = \min \{\mu_j \mid j = 1, \dots, k\}, \quad (4)$$

y sea $x^* = (\xi_1^*, \dots, \xi_n^*)^T$ que alcanza el mínimo (3) para $j = j^*$.

Caso 1. x^* no es entera.

Seleccione una componente no entera, $\xi_{i^*}^*$, de x^* y defina:

$$\begin{aligned} Q_1 &:= \{x = (\xi_1, \dots, \xi_n)^T \in P_{j^*} \mid \xi_i \leq \lfloor \xi_{i^*}^* \rfloor\}, \\ Q_2 &:= \{x = (\xi_1, \dots, \xi_n)^T \in P_{j^*} \mid \xi_i \geq \lfloor \xi_{i^*}^* \rfloor + 1\}. \end{aligned} \quad (5)$$

Sea

$$\Pi_{k+1} := \{P_1, \dots, P_{j^*-1}, Q_1, Q_2, P_{j^*+1}, \dots, P_k\}.$$

Esta colección satisface (2), y podemos empezar en la etapa $k + 1$.

Note que en la siguiente iteración se necesita determinar (3) sólo para los dos nuevos poliedros Q_1 and Q_2 -para los otros poliedros, uno conoce μ_j de la anterior iteración. Si determinamos (3) con el método simplex, podemos usar el último tableau usado para encontrar μ_j^* , agregar la restricción $\xi_i \leq \lfloor \xi_y^* \rfloor$ y aplicar el método dual simplex para encontrar $\min \{cx \mid x \in Q_1\}$. En forma semejante, lo hacemos para Q_2 .

Caso 2. x^* es entera.

Entonces x^* alcanza el mínimo (1).

Si cada P_j es vacía, el problema de programación entera (1) no es factible.

Teorema 2.

Si P es acotado, el método anterior termina.

Demostración:

Si P es acotado, entonces existe un número T tal que:

$$P \subseteq \{x = (\xi_1, \dots, \xi_n)^T \in \mathbb{R}^n \mid -T \leq \xi_i \leq T, \text{ para } i = 1, \dots, n\} \quad (6)$$

Podemos tomar $T := 2^{4n^2\phi}$, donde ϕ es el máximo tamaño de los renglones de la matriz $[A \mid b]$. Así, en cada iteración cada poliedro P_j en $\Pi_k = \{P_1, \dots, P_k\}$ está definido por el sistema $Ax \leq b$, aumentado por algunas desigualdades tomadas de:

$$\begin{aligned} \xi_i &\leq \tau \quad (i = 1, \dots, n; \tau = -T, \dots, T-1) \\ \xi_i &\geq \tau \quad (i = 1, \dots, n; \tau = -T+1, \dots, T) \end{aligned} \quad (7)$$

donde $x = (\xi_1, \dots, \xi_n)^T$. Ahora, si $j' \neq j''$, los poliedros $P_{j'}$ y $P_{j''}$ se definen por diferentes subconjuntos de las $4nT$ desigualdades en (7), como se puede ver por inducción en k . Por lo tanto, $k \leq 2^{4nT}$, y así, a lo más en 2^{4nT} iteraciones, el algoritmo termina.

Si no sabemos que $P = \{x \mid Ax \leq b\}$ está acotada, podemos proceder de la siguiente forma:

Sea ϕ el tamaño máximo de las desigualdades en $Ax \leq b$. Si P contiene un vector entero, contiene un vector de tamaño, a lo más, $6n^3\phi$. Asimismo, si el mínimo (1) es finito, tiene una solución óptima, a lo más, de tamaño $6n^3\phi$. Defina:

$$P_0 := P \cap \{x = (\xi_1, \dots, \xi_n)^T \in \mathbb{R}^n \mid -2^{6n^3\phi} \leq x \leq 2^{6n^3\phi}, \text{ para } i = 1, \dots, n\},$$

y aplique el algoritmo arriba expuesto para encontrar:

$$\min \{cx \mid x \in P_0; x \text{ entera}\}. \quad (8)$$

Si (8) es infactible, (1) también lo es. Si (8) tiene una solución óptima x^* , entonces verifique si la relajación del problema de programación lineal de (1), i.e. $\min \{cx \mid Ax \leq b\}$, es finita o acotada (por ejemplo, con el método simplex). Si es finita, (1) también lo es y x^* es una solución óptima. Si no es acotada, (1) tampoco lo es.

El tiempo de ejecución para el método de ramificación y acotamiento no está polinomialmente acotado por el tamaño de los datos. Al respecto, considere los siguientes problemas de programación entera ILP_t , para $t = 1, 2, \dots$:

$$ILP_t : \max\{n \mid 2^t x = (2^t + 1)n; 0 \leq x \leq 2^t; x, n \text{ enteras}\}$$

La única solución factible para estos problemas es el origen $(x,n)=(0,0)$. El método de ramificación y acotamiento necesita al menos 2^t iteraciones.

Como el tamaño de ILP_t es lineal en t , no hay una cota superior polinomial en el tiempo de ejecución.

3.9 PROBLEMA DE PRODUCCIÓN-TRANSPORTE CON UN NÚMERO FIJO DE VARIABLES NO LINEALES.

Tuy [85] plantea un problema de producción-transporte que involucra un número fijo de fábricas con costos de producción cóncavos. La función objetivo es monótona, cóncava y las restricciones son lineales, el problema general se plantea de la siguiente manera:

$$\min_{x,y} g(y_i) + \sum_{i=1}^k \sum_{j=1}^m c_{ij} x_{ij} \quad (PT)$$

sujeto a:

$$\sum_{j=1}^m x_{ij} = y_i, \quad i = 1, 2, \dots, k$$

$$\sum_{i=1}^k x_{ij} = b_j, \quad j = 1, 2, \dots, m$$

$$y_i \leq s_i, \quad i = 1, 2, \dots, k$$

$$y_i x_{ij} \geq 0, \quad \forall \quad i = 1, 2, \dots, k \quad j = 1, 2, \dots, m$$

Donde k =número de fábricas, m =número de almacenes, s_i =máxima capacidad de la fábrica F_i , b_j =demanda del almacén W_j , $b_j > 0$, c_{ij} =costo de enviar una unidad de la fábrica F_i al almacén W_j , $c_{ij} \geq 0$; $g(y)$ =costo de producir y_i unidades en cada fábrica F_i , $i = 1, 2, \dots, k$, mismo que se supone representado por una función cóncava de $y = (y_1, y_2, \dots, y_k)$. El problema consiste en determinar el nivel de producción y_i de cada fábrica F_i , $i = 1, 2, \dots, k$, y las cantidades x_{ij} de bienes a ser enviados desde la fábrica F_i , $i = 1, 2, \dots, k$ al almacén W_j , $j = 1, 2, \dots, m$ de tal manera que se satisfagan todas las demandas con un costo mínimo de producción-transporte.

El algoritmo usado por Tuy et. al. [85] para este problema convierte el problema original en un problema de transporte lineal que depende de un parámetro k . Este problema de transporte lineal también se puede obtener fijando temporalmente a "y" y omitiendo el término $g(y)$ en la función objetivo del problema original. Como se puede saber de la teoría de la programación lineal paramétrica, el valor óptimo $\phi(y)$ de este problema de transporte es una función afín convexa lineal a trazos. Sin embargo, debido a la concavidad de $g(y)$, la búsqueda de una solución óptima para el problema (PT) se puede restringir sólo al conjunto de vértices de todas las piezas lineales de $\phi(y)$. Más aún, explotando la estructura de red, se demuestra que el número de piezas lineales es polinomial en m y no depende de los costos lineales, un hecho que es crucial para obtener un algoritmo fuertemente polinomial para (PT).

Aunque el algoritmo es eficiente para k pequeño ($k \leq 5$) ya que el tiempo de ejecución es $k-1$, el algoritmo es exponencial en k , como es de esperarse pues se trata de un problema NP-duro.

Así, para valores grandes de k , el algoritmo es más de interés teórico que práctico, lo que nos conduce a buscar mejores algoritmos de tiempo polinomial para el problema (PT). Para el ejemplo 1 resulta muy laborioso calcular los vértices de $\binom{10}{3} = 120$ piezas lineales, por lo que proponemos otro método de solución basado en el algoritmo de rama y corte.

En el caso de los problemas de producción-transporte presentados por Tuy [85], Tuy [84] y Tuy [83], se tiene que las características del problema son: una estructura de red de transporte y soluciones enteras a los puntos extremos; de esta manera, el algoritmo anterior no nos es útil para resolverlos, sólomente usamos el hecho de construir una función lineal que aproxime a la función objetivo que es monótona y cóncava.

En este caso se consideran los puntos dados por $y^1 = (s, 0, \dots, 0)$, $y^2 = (0, s, \dots, 0)$, ..., $y^k = (0, 0, \dots, s)$ con $i=1, \dots, k$. Se evalúa entonces la función $f(y_0) + \nabla f(y_0)(y - y_0)$ en cada uno de estos puntos, que cuando mucho son k . De esta manera, el algoritmo se plantea como sigue:

ALGORITMO PRODUCCIÓN-TRANSPORTE.

PASO 1 (Establecimiento de una función lineal de aproximación inferior).
Establezca una función objetivo lineal $d(x)$ que aproxime inferiormente a la función objetivo original, tal que:

$$d(y^i) = f(y^0) + \nabla f(y^i)(y - y^i)$$

en el punto extremo y^i .

PASO 2 (Solución del programa lineal relajado)

Resuelva el programa lineal:

$$\min_x \{d(x) \mid x \in Q\}$$

Sea x^0 ($i=0$), donde $y=0, 1, 2, \dots, n$ el punto extremo óptimo.

Defina $\underline{f}(x^0) = d(x^0)$ como una cota inferior del valor óptimo de la función objetivo original. Sea $x^* = x^0$, entonces $\bar{f}(x^*) = f(x^0)$ valuada en la función original es una cota superior. Guarde ambas cotas en una lista, haga $i=i+1$ y vaya al paso 1. Continúe con este proceso hasta que $i=k$ y entonces vaya al paso 3.

PASO 3 (Comparación de soluciones)

Compare cuál de las soluciones dadas en la lista para la función objetivo original es la que arroja el mínimo valor, ésta es la óptima.

En conclusión, en este capítulo se han sentado las bases para aplicar los dos algoritmos a problemas de prueba, mismos que se han seleccionado de la literatura existente. No se analizó la convergencia del segundo algoritmo por considerarse un caso particular del primero, por lo que analizando el algoritmo Rama y Corte implícitamente se analiza el algoritmo Producción-Transporte.

CAPITULO 4

PROBLEMAS DE PRUEBA

“El llegar a la conclusión de un problema de investigación normal es lograr lo esperado de una manera nueva y eso requiere la resolución de toda clase de complejos enigmas instrumentales, conceptuales y matemáticos”.

T. S. Kuhn. *La estructura de las revoluciones científicas.*

En este capítulo se presentan varios problemas donde se prueba la eficacia de los dos algoritmos expuestos en el capítulo anterior. La manera en que se exponen estos problemas es la siguiente:

- 1.- Problemas de cargo fijo
- 2.- Funciones separables.
- 3.- Funciones no separables.
- 4.- Problemas de producción-transporte.

Los primeros tres casos se resuelven con el algoritmo de rama y corte y, el cuarto caso, con el algoritmo de producción-transporte.

1.- PROBLEMAS DE CARGO FIJO

Ejemplo 1 (Taha, 1973)

En este ejemplo y el siguiente, se busca evitar cortes excesivos al tener soluciones degeneradas.

Consideramos aquí dos ejemplos dados por Taha (1973), en el segundo se aplica el algoritmo en su totalidad.

$$\min f(x) = \phi_1(x_1) + \phi_2(x_2)$$

sujeto a:

$$2x_1 + x_2 \leq 4$$

$$x_1 + x_2 \leq 3$$

$$0 \leq x_2 \leq 5/2$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in \mathbb{I}$$

donde:

$$\phi_1(x_1) = \begin{cases} 0 & x_1 = 0 \\ -4x_1 + 1 & x_1 > 0 \end{cases}$$

$$\phi_2(x_2) = \begin{cases} 0 & x_2 = 0 \\ -3x_2 + \frac{1}{2} & x_2 > 0 \end{cases}$$

Gráficamente se puede ver en la figura 4.1

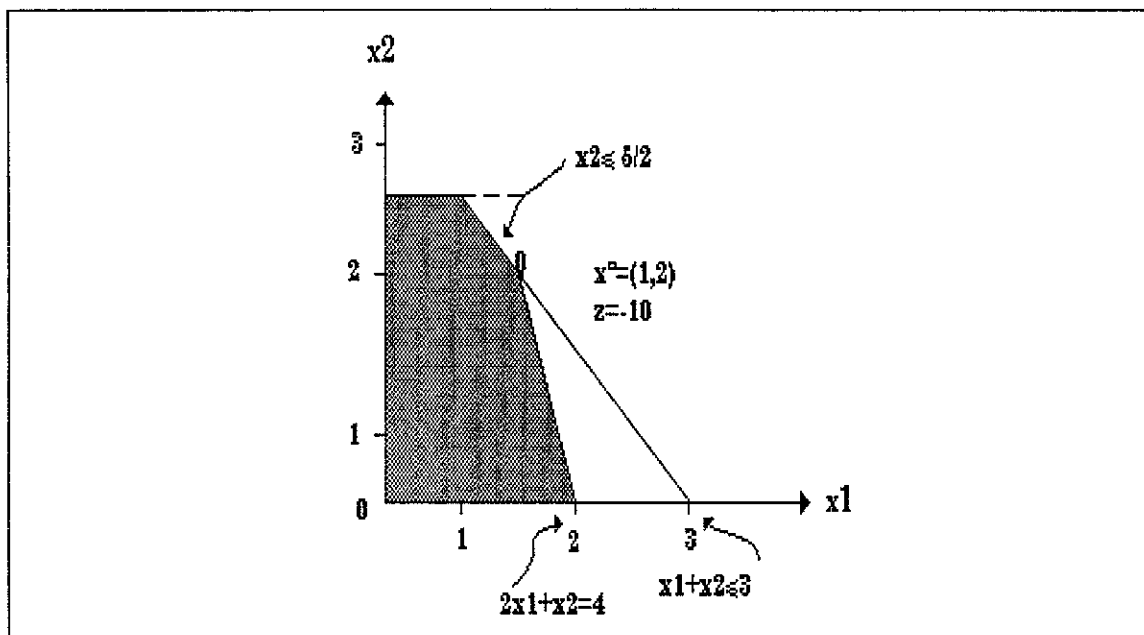


figura 4.1

Solución:

PASO 1

Proponemos $l(x) = -4x_1 - 3x_2$.

PASO 2

La tabla siguiente nos muestra la solución para el problema continuo; es decir, sin considerar que las variables sean enteras.

TABLA 4.1

		S_1	S_2	
z	-10	-1	-2	$x^0=(1,2)$ el punto 0
x_1	1	1	1	$f(x^0)=-10$
x_2	2	1	2	$f(x^0)=-10+(1+\frac{1}{2})=-8\frac{1}{2}$
S_2	1/2	1	2	$\bar{f}(x^*)=f(x^0)=-8\frac{1}{2}$

Como la solución es entera, esta solución es la óptima. Con $\bar{f}(x^*)=-2$ y $x^*=(1,2)$.

Ejemplo 2 (Taha, 1973)

El siguiente problema propuesto por Taha tiene la particularidad de no tener soluciones enteras en los vértices, por lo cual el uso de los planos de corte convexos nos conduce a una solución continua y ello nos lleva a usar ramificación y acotamiento.

Sea: $\min f(x) = -4x_1 - 3x_2 + 3/2$

Sujeto a: $x_1 + 3.5x_2 \leq 22.75$

$$2x_1 + 3x_2 \leq 23.5$$

$$3.5x_1 + 2x_2 \leq 29.75$$

$$x_1, x_2 \geq 0 \quad x_1, x_2 \in \mathbb{I}.$$

Podemos ver en la figura 4.2 la región factible de este problema:

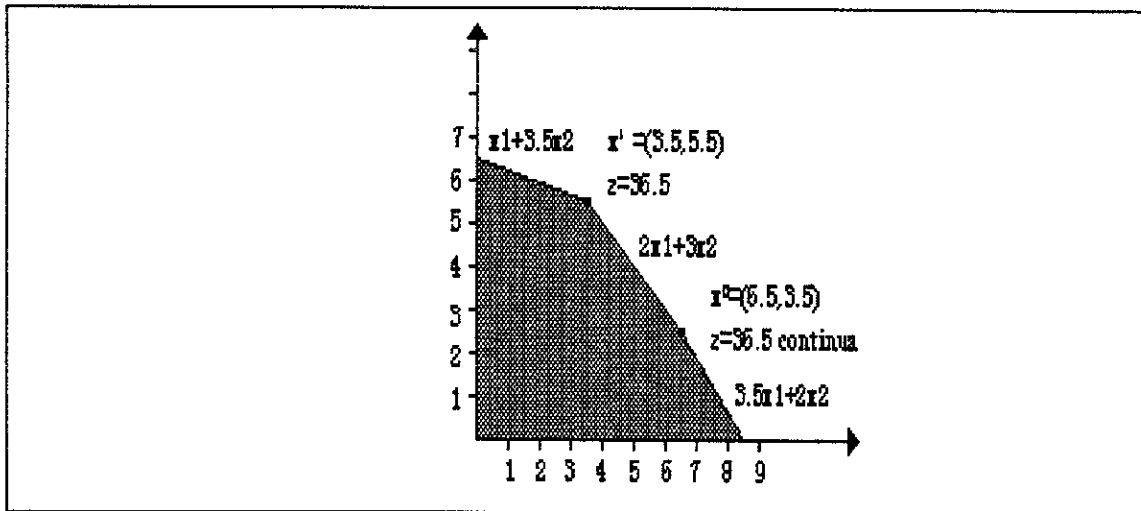


figura 4.2

Multiplicamos por escalares apropiados en las restricciones para evitar errores de redondeo con los decimales y se obtiene el siguiente conjunto de restricciones equivalentes:

$$4 x_1 + 14 x_2 \leq 91$$

$$4 x_1 + 6 x_2 \leq 47$$

$$14 x_1 + 8 x_2 \leq 119$$

PASO 1

Se propone $d(x) = -4x_1 - 3x_2$.

PASO 2

Resolviendo el programa lineal, tenemos la tabla 4.2:

TABLA 4.2

	S_2	S_3	LD
z	-0.38	-0.92	-36.5
S_1	-1.58	0.62	4
x_2	0.54	-0.31	3.5
x_1	-0.31	0.46	6.5

PASO 3

$$x^0 = (6.5, 3.5),$$

$$f(x^0) = d(x^0) = -36.5$$

$$f(x^0) = -36.5 + 3/2 = -35,$$

$$\bar{f}(x^*) = f(x^0) = -35.$$

Haga $x^* = x^{l+1} = x^l$ y vaya al paso 5.

Paso 5

Se desarrolla el primer corte:

$$S_2^* = \min \{ \infty, 3.5/0.54, \infty \} = 6.5,$$

$$S_3^* = \min \{ 4/0.62, \infty, 6.5/0.46 \} = 6.5,$$

Y el corte está dado por:

$$\frac{S_2}{6.5} + \frac{S_3}{6.5} \geq 1$$

Expresado en terminos de x_1 y x_2 , el corte es:

$$5.5 x_1 + 5 x_2 \leq 46.75; \quad \text{Corte 1}$$

PASO 2

Denotamos la holgura como S_4 y, resolviendo el problema de PL con esta nueva restricción, vemos los resultados en la tabla 4.3:

Tabla 4.3

	S_3	S_4	LD
z	0.538	0.385	-34
S_1	2.192	-1.577	14.25
S_2	1	-1	6.5
x_1	0.769	-0.308	8.5
x_2	-0.846	0.538	0

La tabla 4.3 da como resultado x^1 al aumentar el corte 1 en la tabla 4.2.

Como x^1 no es entero, vaya al paso 3.

PASO 3

Las actuales cotas superior e inferior están dadas por $f(x^*) = f(x^0)$, y $\underline{f}(x^0)$, sea x^1 el siguiente punto extremo de Q. Haga la nueva cota inferior $\underline{f}(x^1) = \ell(x^1)$, y vaya al paso 4.

PASO 4

De la tabla anterior, tenemos:

$$x^1 = (8.5, 0),$$

$$\underline{f}(x^1) = \ell(x^1) = -34,$$

$$f(x^1) = -34 + 3/2 = -32.5,$$

$$b) \quad \ell(x^1) = \underline{f}(x^1) = -34 > f(x^*) = f(x^0) = -35,$$

Dado que $\ell(x^1) = \underline{f}(x^1) > f(x^0)$, el proceso termina, y vaya al paso 6.

PASO 6

En este caso tenemos la mejor cota inferior en el punto $x^0 = (3.5, 6.5)$, y la solución $f(x^0) = -36.5$. Las dos variables tienen un valor fraccional de 0.5, por lo tanto:

$$I = \{1, 2\}$$

La variable seleccionada de I es x_1 , la cual tiene el mayor (en valor absoluto) coeficiente de la función objetivo, igual a -4.

Ramificando desde esta variable con $x_1 \leq 6$ y $x_1 \geq 7$ tenemos las siguientes soluciones:

$$x^1 = (6, 3.83), \ell(x^1) = -35.5,$$

$$x^2 = (7, 2.62), \ell(x^2) = -35.87.$$

Si construimos una lista de soluciones en orden creciente, tendremos:

$$l(x^0) = -36.5,$$

$$l(x^2) = -35.87,$$

$$l(x^1) = -35.5,$$

Continuando con este proceso tenemos el siguiente árbol de búsqueda y una lista completa de cotas inferiores.

$$l(x^0) = -36.5,$$

$$l(x^2) = -35.8,$$

$$l(x^1) = -35.5,$$

$$l(x^3) = -35.4,$$

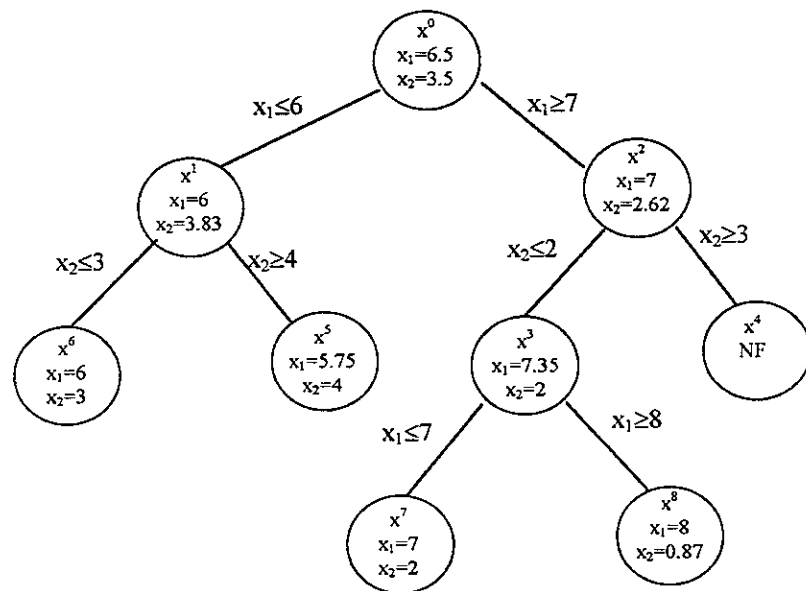
$$l(x^4) = \text{NF},$$

$$l(x^5) = -35,$$

$$l(x^8) = -34.62,$$

$$l(x^7) = -34,$$

$$l(x^6) = -33,$$



La solución óptima entera está en el nodo 7, con $f(x^7) = -34$, $x^7 = (7, 2)$.

2.- FUNCIONES SEPARABLES

Ejemplo 3 (Horst y Tuy, 1995).

Sea la función f una función separable tal que:

$$\min f(x) = -(x_1 - 1.2)^2 - (x_2 - 0.6)^2$$

sujeto a:

$$-2x_1 + x_2 \leq 1,$$

$$x_2 \leq 2,$$

$$x_1 + x_2 \leq 4,$$

$$x_1 \leq 3,$$

$$0.5x_1 - x_2 \leq 1,$$

$$x_1, x_2 \geq 0, \text{ y enteras.}$$

Solución

Este ejemplo supone que la función objetivo es cóncava, y fue resuelto usando un algoritmo de cortes de caras del poliedro convexo que representa la región factible. Dicho algoritmo establece que el máximo

número de cortes que se pueden construir no debe exceder el número de caras del poliedro. Se resuelve en dos ciclos de 1 y 3 iteraciones, respectivamente.

La forma en que aquí lo resolvemos es usando el algoritmo rama y corte.

PASO 1

Claramente, $f(x)$ es diferenciable. Entonces construimos $\ell(x)$ con base en sus derivadas, sumándolas obtenemos:

$$\ell(x) = -2x_1 - 2x_2 + 3.6$$

PASO 2

Resolviendo el programa lineal obtenemos:

$$\underline{f}(x^0) = \ell(x^0) = -13.6 \quad \text{con } x^0 = (3, 1)$$

Sea $x^* = x^0$, entonces $\bar{f}(x^*) = f(x^0) = -3.4$

Esta solución es entera, por lo que es óptima. La región factible está dada en la figura 4.3.

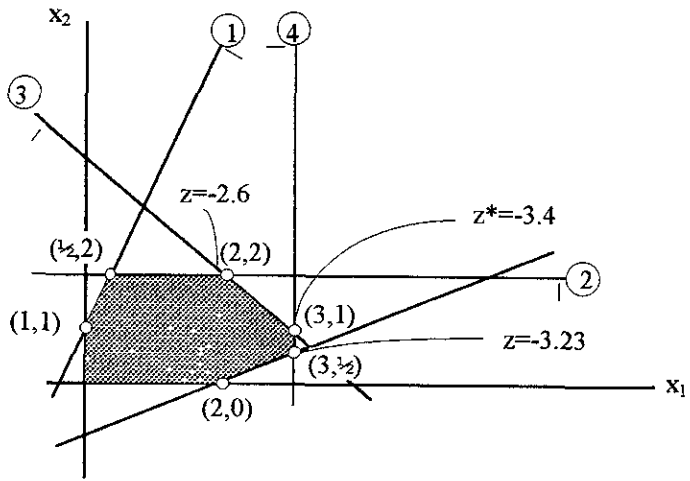


figura 4.3

La función original se puede observar en la figura 4.4

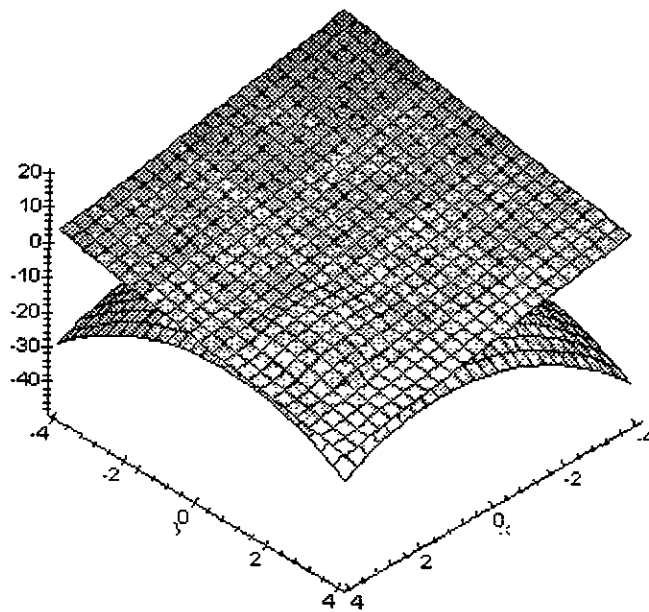


figura 4.4

Ejemplo 4 (Horst y Tuy, 1995).

Sea f una función separable y

$$\min f(\mathbf{x}) = -(x_1 - 4.2)^2 - (x_2 - 1.9)^2$$

$$\text{sujeto a: } -x_1 + x_2 \leq 3,$$

$$x_1 + x_2 \leq 11,$$

$$2x_1 - x_2 \leq 16,$$

$$-x_1 - x_2 \leq -1,$$

$$x_2 \leq 5,$$

$$x_1, x_2 \geq 0 \text{ y enteras.}$$

Solución

Este ejemplo fue resuelto con un algoritmo de anexamiento poliédrico, que trabaja en dos fases. Antes de cada iteración de regreso a la fase I se agrega un corte cóncavo. El anexamiento poliédrico es una técnica desarrollada originalmente para resolver un problema especial que se puede formular de la siguiente manera:

(DG) Dado un politopo D contenido en un cono $K_0 \subset \mathbb{R}^n$ y un conjunto compacto convexo G con $0 \in \text{int } G$, encuentre un punto $y \in D \setminus G$, o establezca que $D \subset G$.

El problema (DG) es importante para una gran clase de problemas de optimización global, incluyendo problemas de minimización cóncava, estos se pueden trasladar a un problema (DG) a través de la construcción de un cono. Por otro lado, el método de anexamiento poliédrico consiste en reemplazar a G por la intersección de ésta con el cono y construir un simplex, generándose una sucesión de politopos que se expanden y se aproximan a dicha intersección desde el interior.

El procedimiento anterior se desarrolla en el algoritmo de dos fases, resolviendo el problema en 4 iteraciones.

Usando el algoritmo rama y corte:

PASO 1

Construimos la función:

$$f(x) = -2x_1 - 2x_2 + 12.2,$$

considerando que sólo una variable es acotada, se supone diferenciable la función objetivo y se construye $\ell(x)$ a partir de sus derivadas parciales.

PASO 2

Resolviendo el problema lineal obtenemos la solución:

$$\underline{f}(x^0) = \ell(x^0) = -22 \quad \text{con } x^0 = (9, 2),$$

como no hay discontinuidades ni soluciones degeneradas, ésta es la solución óptima.

Valuando en la función objetivo original: $f(x^*) = -23.03$. La gráfica de la región factible se muestra en la figura 4.5.

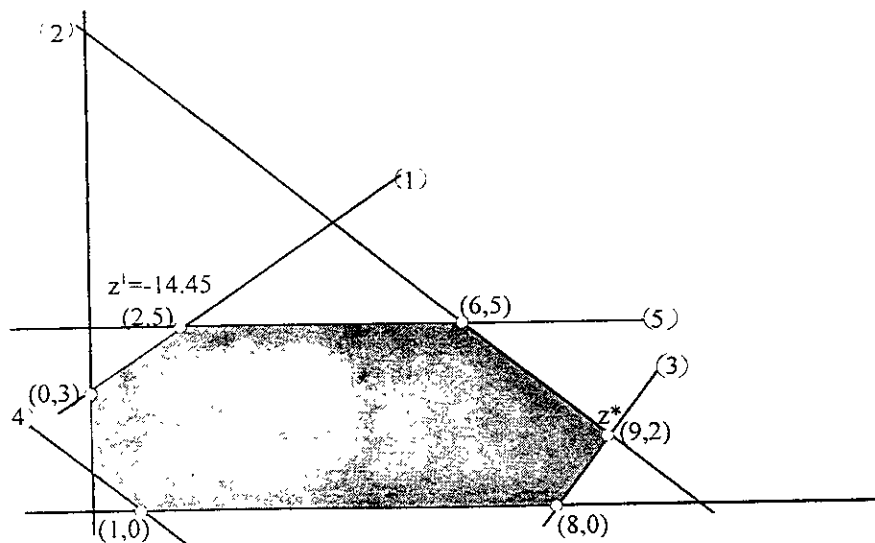


figura 4.5

La función objetivo original puede verse en la figura 4.6

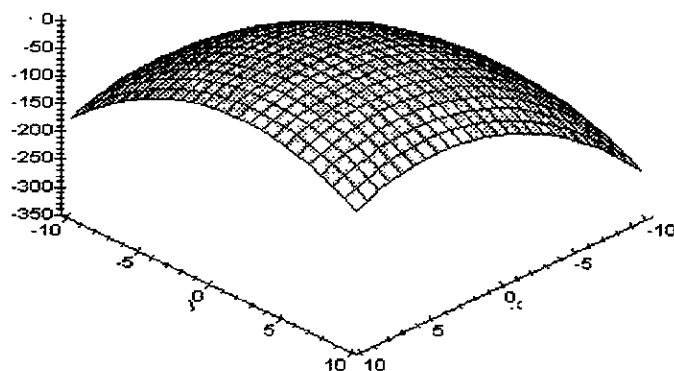


figura 4.6

FUNCIONES SEPARABLES DISCONTINUAS

Ejemplo 5 (Falk y Soland 1969)

Sean las funciones f_1 y f_2 cóncavas y con una discontinuidad en cero:

$$f_1(x_1) = \begin{cases} 0 & x_1 = 0 \\ -x_1^2 + 4x_1 + 2 & 0 < x_1 \leq 6 \end{cases}$$

$$f_2(x_2) = \begin{cases} 2 & x_2 = 0 \\ -x_2^2 + 2x_2 + 4 & 0 < x_2 \leq 5 \end{cases}$$

Se trata de minimizar $Z = f_1 + f_2$.

Las restricciones están representadas por los conjuntos G y C, donde la región de factibilidad es $G \cap C$, G es un polígono convexo y C es un rectángulo, definidos de la siguiente manera:

$$-x_1 + 3x_2 \leq 5,$$

$$2x_1 - x_2 \leq 5,$$

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

$$x_1 - 3x_2 \leq 0,$$

$$0 \leq x_1 \leq 6,$$

$$0 \leq x_2 \leq 5.$$

Gráficamente se tiene la siguiente región factible:

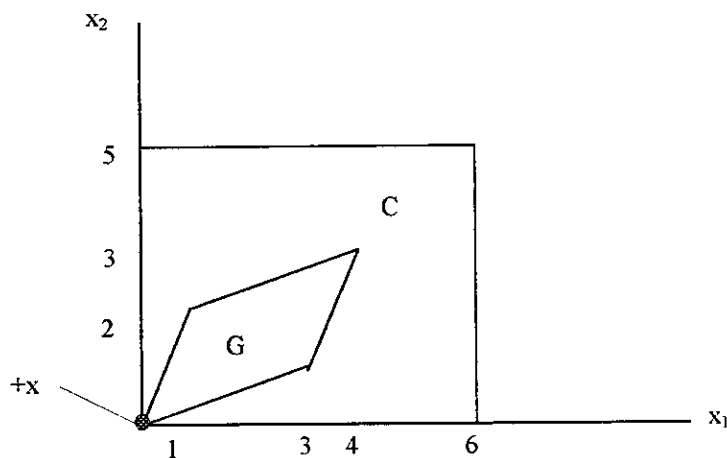


figura 4.7

Solución

Este ejemplo fue resuelto originalmente con un algoritmo de ramificación y acotamiento, y resuelve una sucesión de problemas en

cada uno de los cuales la función objetivo es convexa. Esos problemas corresponden a particiones sucesivas del dominio factible. Se usan también dos reglas de refinamiento de las particiones para asegurar la convergencia del algoritmo.

El concepto de envoltura convexa es crucial en este algoritmo ya que los subproblemas a resolver tendrán como función objetivo a la envoltura convexa de la función original, pero en pedazos, y las soluciones serán una cota inferior de la función original. Para resolver este problema se da la siguiente tabla:

$$\{x^k\} \text{ y } \{\mu^k\}$$

Tabla 4.4

k	x_1^k	x_2^k	μ^k
1	4	3	-12.47
2	4	3	-3.8
3	4	3	-3.8
4	1	2	1.83
5	0	0	2

Note que se encuentra una solución local al inicio.

Procediendo a resolver este problema usando el algoritmo rama y corte y considerando que la función es separable, podemos simplemente obtener sus derivadas parciales.

PASO 1

Se propone $\ell(\mathbf{x}) = -2x_1 - 2x_2 + 6$

PASO 2

Resolviendo con $\ell(\mathbf{x}) = -2x_1 - 2x_2 + 6$ sujeta a las mismas restricciones, obtenemos la siguiente solución:

$$\underline{f}(\mathbf{x}^0) = \ell(\mathbf{x}^0) = -14 + 6 = -8, \quad \text{con } \mathbf{x}^0 = (4, 3)$$

y valuando en la función original tenemos $\bar{f}(\mathbf{x}^*) = 3$. Podríamos decir que al haber encontrado esta solución entera ya terminamos; sin embargo, como la función objetivo original no es continua y de hecho la discontinuidad se da en el cero, es posible que haya mejores soluciones enteras que ésta, por lo cual agregamos un corte. Del tableau óptimo se puede ver:

TABLA 4.5 (Tableau óptimo)

	S_1	S_2	LD
Z	1.2	1.6	14
S_4	1	0	5
x_2	0.4	0.2	3
S_3	0	1	5
x_1	0.2	0.6	4
S_5	-0.2	-0.6	2
S_6	-0.4	-0.2	2

PASO 5

De aquí se obtiene el corte haciendo:

$$S_1^* = \min\left\{\frac{5}{1}, \frac{3}{0.4}, \frac{4}{0.2}\right\} = 5$$

$$S_2^* = \min\left\{\frac{3}{0.2}, \frac{5}{1}, \frac{4}{0.6}\right\} = 5$$

Por lo tanto se tiene:

$$\frac{S_1}{5} + \frac{S_2}{5} \geq 1$$

que en términos de las variables originales x_1 y x_2 , está dado por:

$$x_1 + 2x_2 \leq 5,$$

Primer corte

con la solución:

$$\underline{f}(x^1) = d(x^1) = -8 + 6 = -2 \quad \text{con } x^1 = (3,1)$$

PASO 3

Se tiene $f(x^0) = 3$, $d(x^0) = -8$ y ahora $d(x^1) = \underline{f}(x^1) = -2$ y $f(x^1) = 10$

PASO 4

$$d(x^0) = -8 < f(x^1) = 10$$

Ambas soluciones son enteras y la mejor hasta ahora es $x^0 = (4,3)$,
agregamos otro corte de la tabla siguiente:

TABLA 4.6 (Tableau óptimo)

	S_4	S_7	LD
Z	0.4	1.2	8
S_1	1	-1	5
x_2	-0.2	0.4	1
S_3	1	0	5
x_1	0.4	0.2	3
S_5	-0.4	-0.2	3
S_6	0.2	-0.4	4
S_4	-1	1	0

PASO 5

Comparamos $f(x^0) = 3$ con $f(x^1) = 10$.

$f(x^0) < f(x^1)$, por lo tanto $\bar{f}(x^*) = f(x^0) = 3$

Tomando:

$$S_4^* = \min\left\{\frac{5}{1}, \frac{3}{0.4}, \frac{5}{1}, \frac{4}{0.2}\right\} = 5$$

$$S_7^* = \min\left\{\frac{5}{1}, \frac{3}{0.6}, \frac{1}{0.2}\right\} = 5$$

De aquí se tiene:

$$\frac{S_4}{5} + \frac{S_7}{5} \geq 1$$

que en términos de las variables originales queda:

$$-2x_1 + x_2 \geq 0$$

Segundo Corte.

PASO 2

Lo agregamos al problema original y obtenemos la solución:

$$\underline{f}(x^2) = d(x^2) = -6 + 6 = 0, \quad \text{con } x^3 = (1, 2)$$

TABLA 4.7

	S_7	S_8	LD
Z	1.2	0.4	6
S_1	-1	1	0
x_2	0.4	-0.2	2
S_3	0	1	0
x_1	0.2	0.4	1
S_5	-0.2	-0.4	5
S_6	-0.4	0.2	3
S_4	1	-1	5
S_2	0	-1	5

PASO 3

$$f(x^2) = 0 \text{ y } f(x^2) = 9.$$

PASO 4

$$f(x^1) = -2 < f(x^2) = 9 \text{ ir al paso 5.}$$

PASO 5

De aqui se obtiene el corte:

$$S_7^* = \min\left\{\frac{1}{0.2}, \frac{2}{0.4}, \frac{5}{1}\right\} = 5$$

$$S_8^* = \min\left\{\frac{1}{0.4}, \frac{3}{0.2}\right\} = 2.5$$

Y se tiene:

$$\frac{S_7}{5} + \frac{S_8}{2.5} \geq 1$$

que en términos de las variables originales queda:

$$-5x_1 \geq 0$$

Tercer corte.

PASO 2

Obtenemos la solución:

$$f(x^3) = d(x^3) = 0, \quad \text{con } x^3 = (0,0)$$

Esta solución es óptima en la función objetivo original $f(x^*) = 2$ que es la mínima encontrada, además ya no podemos evaluar mas puntos, porque ya llegamos al mínimo valor que pueden tomar las variables.

Resumiendo, tenemos la siguiente tabla:

TABLA 4.8

Punto	Valor de z aproximada	z^*
(4,3)	-8	3
(3, 1)	-2	10
(1,2)	0	9
(0,0)	6	2

La función objetivo original puede verse en la figura 4.8

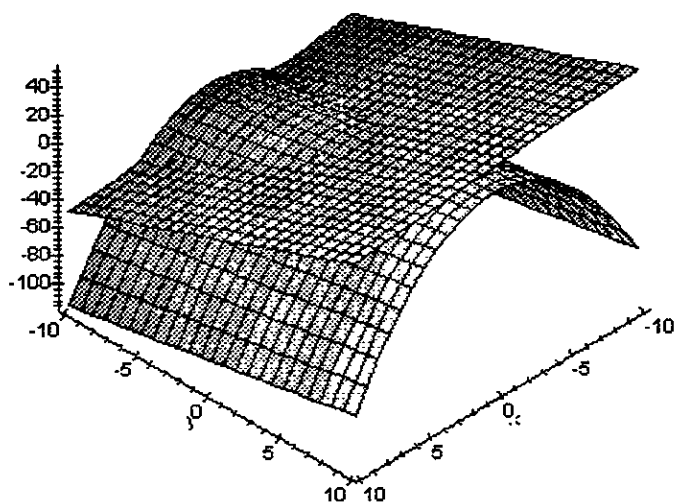


figura 4.8

3.- FUNCIONES NO SEPARABLES

Ejemplo 6 (Horst y Tuy, 1995).

El siguiente es un problema de programación cóncava cuadrática y tiene una especial importancia en los problemas de minimización cóncava. Consiste en encontrar el mínimo global de la función objetivo sobre un poliedro, pero aquí le hemos agregado la restricción de variable entera.

La función f es una función no separable.

Sea:

$$\min f(x) = 2x_1 + 3x_2 - 2x_1^2 + 2x_1x_2 - 2x_2^2$$

Sujeto a:

$$-x_1 + x_2 \leq 1,$$

$$x_1 - x_2 \leq 1,$$

$$-x_1 + 2x_2 \leq 3,$$

$$2x_1 - x_2 \leq 3,$$

$$x_1, x_2 \geq 0 \text{ y enteras.}$$



Solución

Este ejemplo se resolvió usando el Método de Konno de Planos de Corte [53] para la solución de un problema de programación cóncava cuadrática. Se reportan resultados similares obtenidos por Balas y Burdet [4] usando la teoría de Polares exteriores generalizados.

El algoritmo de Konno comienza construyendo un corte cóncavo y se considera un mínimo local. Se obtiene la mayor raíz de una ecuación cuadrática, y se procede a resolver un programa lineal, considerando también el problema dual, dicho problema se resuelve en 3 iteraciones. Como en este caso se tiene una función no separable, se usa el procedimiento de cambio de variables dado en el capítulo 3.

Tenemos entonces que:

$$x_1 \cdot x_2 = y_1^2 - y_2^2,$$

y las restricciones:

$$y_1 = \frac{x_1 + x_2}{2},$$
$$y_2 = \frac{x_1 - x_2}{2},$$

lo que implica $2x_1x_2 = 2y_1^2 - 2y_2^2$ con las mismas restricciones.

Reescribiendo el problema tenemos:

$$\min f(\mathbf{x}) = 2x_1 + 3x_2 - 2x_1^2 + 2y_1^2 - 2y_2^2 - 2x_2^2,$$

Sujeto a:

$$-x_1 + x_2 \leq 1,$$

$$x_1 - x_2 \leq 1,$$

$$-x_1 + 2x_2 \leq 3,$$

$$2x_1 - x_2 \leq 3,$$

$$2y_1 - x_1 - x_2 = 0,$$

$$2y_2 - x_1 + x_2 = 0.$$

PASO 1

Definimos $\ell(\mathbf{x})$ con base en la suma de sus derivadas parciales:

$$\ell(\mathbf{x}) = -4x_1 - 4x_2 + 4y_1 - 4y_2 + 5.$$

PASO 2

Se resuelve el programa lineal y obtenemos la solución:

$$\underline{f}(\mathbf{x}^0) = \ell(\mathbf{x}^0) = -12 + 5 = -7 \quad \text{con } \mathbf{x}^0 = (3,3),$$

valuando en la función original $f(\mathbf{x}^*) = -3$.

Como la solución es entera, es la óptima y el algoritmo termina.

La región factible se puede ver en la figura 4.8:

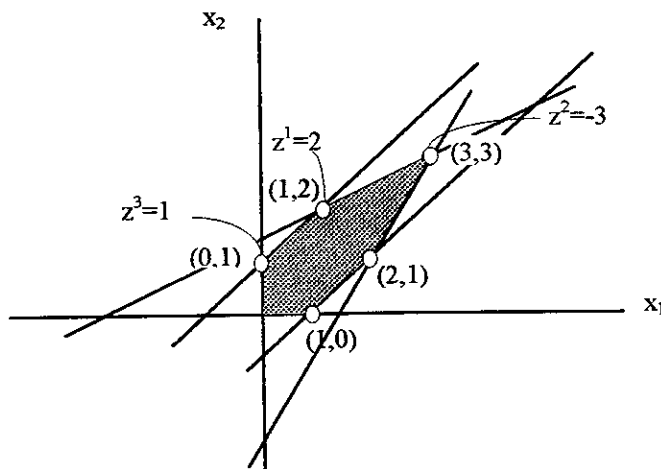


figura 4.8

Ejemplo 7 (Ryoo y Sahinidis, 1997)

Sea f una función no separable tal que:

$$f(\mathbf{x}) = -x_1 + x_1x_2 - x_2$$

sujeta a:

$$-6x_1 + 8x_2 \leq 3,$$

$$3x_1 - x_2 \leq 3,$$

$$x_1 \leq 5,$$

$$x_2 \leq 5,$$

$$x_i \geq 0 \text{ y enteras.}$$

Solución

Este ejemplo se resolvió originalmente para el caso continuo, así que comparar los algoritmos no tendría mucho sentido ya que las soluciones son diferentes, una es continua y la otra es entera.

Cabe mencionar, sin embargo, que el algoritmo usado es de tipo Ramificación y Reducción donde se usa una técnica de reducción de rango para mejorar el procedimiento de acotamiento en cada nodo. Reportan ahí que si se usa un enfoque clásico de ramificación y acotamiento se requieren 51 iteraciones para comprobar que la solución encontrada es la del nodo raíz y, siguiendo su procedimiento, es inmediata; la solución continua es $f(x)^* = -1.08333$, con $x = (1.1667, 0.5)^t$.

En este caso, como la función es no separable usamos el procedimiento de separación de variables descrito en el capítulo 3.

Sea:
$$x_1 \cdot x_2 = y_1^2 - y_2^2,$$

y las restricciones:

$$y_1 = \frac{x_1 + x_2}{2},$$
$$y_2 = \frac{x_1 - x_2}{2},$$

Reescribiendo el problema tenemos:

$$\min f(x) = -x_1 + y_1^2 - y_2^2 - x_2$$

Sujeta a:

$$-6x_1 + 8x_2 \leq 3$$

$$3x_1 - x_2 \leq 3$$

$$x_1 \leq 5$$

$$x_2 \leq 5$$

$$2y_1 - x_1 - x_2 = 0$$

$$2y_2 - x_1 + x_2 = 0$$

$$y_1 \leq 5$$

$$y_2 \leq 0$$

PASO 1

Sea:

$$d(x) = -x_1 + 2y_1 - 2y_2 - x_2 = 0$$

PASO 2

La solución de programación lineal está dada por:

$$f(x^0) = d(x^0) = 0, \quad \text{con } x^0 = (1, 0) \text{ y } f(x^0) = -1$$

como es entera, es la óptima y el proceso termina.

La región factible se muestra en la figura 4.9.

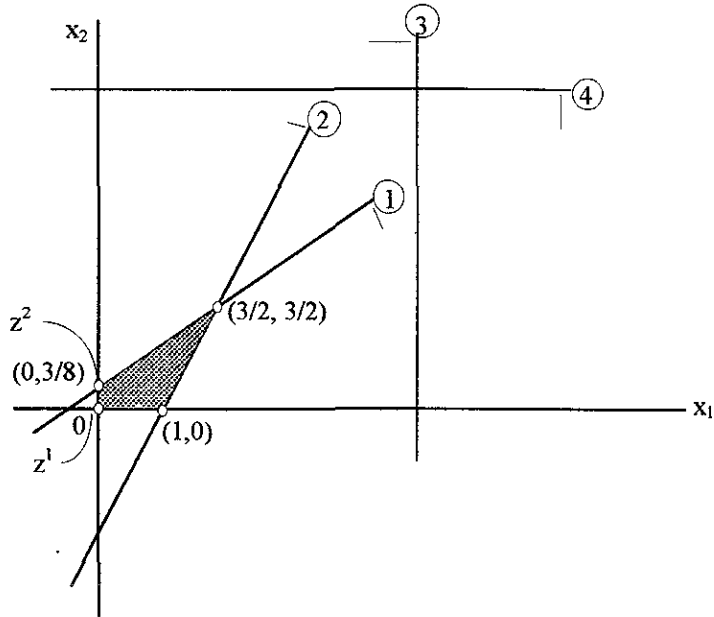
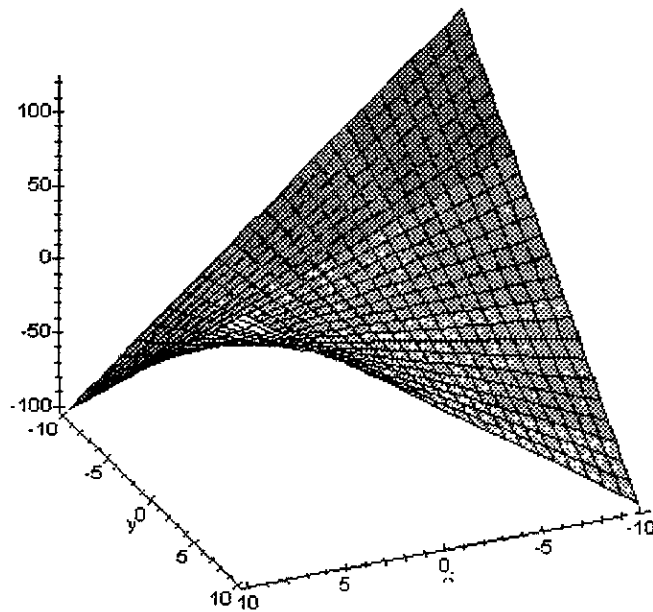


figura 4.9

La función objetivo original puede verse en la figura 4.10:



4.- PROBLEMAS DE PRODUCCION-TRANSPORTE

Ejemplo 8 (Tuy, 1996)

Si consideramos cuatro fábricas (F_1, \dots, F_4) que producen un cierto producto final a ser enviado a diez almacenes (W_1, \dots, W_{10}). Suponga que las demandas de los almacenes están dadas por:

$$b^T = \{b_i\} = [23, 12, 24, 13, 12, 53, 87, 15, 38, 29]$$

Además, cada fábrica tiene una capacidad superior a la demanda total. Para producir una unidad de el producto final en cada fábrica se requiere una cierta cantidad de materia prima que debe comprarse de un proveedor. Las cantidades de materia prima requerida para una unidad del producto final en diferentes fábricas y están dadas por:

$$\alpha = \{\alpha_i\} = (3, 5, 2, 4)$$

El costo de comprar z unidades de un proveedor es una función cóncava:

$$h(z) = z^{1/3}$$

Si se agregan los costos de transporte entre el proveedor y las fábricas y éstos son lineales o cóncavos, se sigue teniendo una función de costo cóncava $h(z)$. El costo de producir y_i unidades del producto final en la fábrica F_i es una función cóncava:

$$p_i(y_i) = y_i^{1/2}, \quad i = 1, \dots, 4.$$

Entonces el costo total de producción es igual a:

$$g(y) = g(y_1, \dots, y_4) = y_1^{1/2} + y_2^{1/2} + y_3^{1/2} + y_4^{1/2} + (3y_1 + 5y_2 + 2y_3 + 4y_4)^{1/3}$$

que es una función cóncava no separable.

Los costos de transporte lineales entre las fábricas y los almacenes están dados por:

$$c = \{c_{ij}\} = \begin{bmatrix} 1 & 12 & 13 & 4 & 6 & 7 & 8 & 6 & 65 & 19 \\ 1 & 32 & 5 & 7 & 1 & 2 & 7 & 8 & 98 & 10 \\ 3 & 23 & 4 & 5 & 6 & 8 & 9 & 2 & 45 & 17 \\ 8 & 45 & 3 & 1 & 4 & 6 & 1 & 9 & 24 & 28 \end{bmatrix}$$

Solución

La demanda total está dada por: $s = \sum_{i=1}^4 b_i = 306$, así Ω es el tetraedro:

$$\Omega = \{y \in R^4 + \sum_{i=1}^4 y_i = 306\}$$

Tenemos entonces que la región factible la podemos graficar de acuerdo con la figura 4.11.

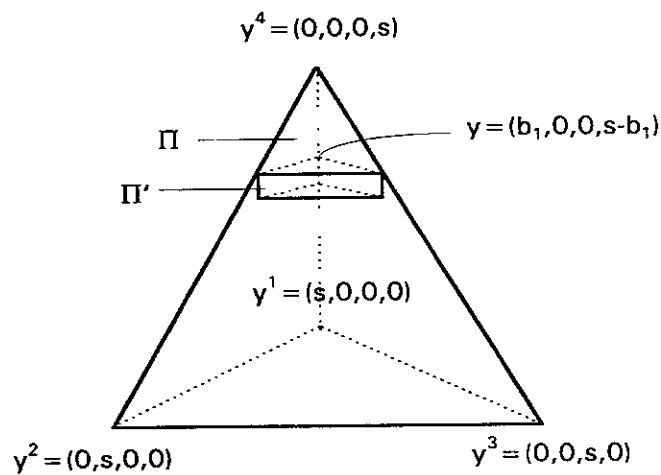


figura 4.11

Usando el algoritmo producción-transporte:

PASO 1

Se propone una aproximación lineal de Taylor que considera a la función lineal: $f(y^0) + \nabla f(y^0)(y - y^0)$, en un punto y^0 , a la que llamaremos $d(y)$. En este caso consideramos a los puntos extremos de la región factible, para de ahí evaluar la función objetivo y considerar este valor como una cota inferior para el valor de la función objetivo original.

Sea la función $d(y^1)$ evaluada en el punto $y^1 = (306, 0, 0, 0)$:

$$d(y^1) = 0.040y_1 + 0.0176y_2 + 0.0023y_3 + 0.0047y_4 + \sum_{i=1}^4 \sum_{j=1}^{10} c_{ij} x_{ij} + 14.97$$

sujeta a:

$$\sum_{j=1}^{10} x_{1j} - y_1 = 0,$$

$$\sum_{j=1}^{10} x_{2j} - y_2 = 0,$$

$$\sum_{j=1}^{10} x_{3j} - y_3 = 0,$$

$$\sum_{j=1}^{10} x_{4j} - y_4 = 0 ,$$

$$\sum_{i=1}^4 x_{i1} = 23 ,$$

$$\sum_{i=1}^4 x_{i2} = 12 ,$$

$$\sum_{i=1}^4 x_{i3} = 24 ,$$

$$\sum_{i=1}^4 x_{i4} = 1 ,$$

$$\sum_{i=1}^4 x_{i5} = 12 ,$$

$$\sum_{i=1}^4 x_{i6} = 53 ,$$

$$\sum_{i=1}^4 x_{i7} = 87 ,$$

$$\sum_{i=1}^4 x_{i8} = 15 ,$$

$$\sum_{i=1}^4 x_{i9} = 3 ,$$

$$\sum_{i=1}^4 x_{i10} = 29 ,$$

$$\sum_{i=1}^4 y_i = 306 ,$$

$$x_{ij}, y_i \geq 0$$

PASO 2

La solución que se obtiene usando programación lineal es:

$$f(y) = \ell(y^1) = 1692.335 + 14.97 = 1707.305$$

con $y^* = (12, 117, 15, 162)$, y

$$x^* = \begin{bmatrix} 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 23 & 0 & 0 & 0 & 12 & 53 & 0 & 0 & 0 & 29 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 24 & 13 & 0 & 0 & 87 & 0 & 38 & 0 \end{bmatrix}$$

Evaluando esta solución en la función objetivo original se obtiene:

$$f(y^1) = 1730.7228$$

Como se puede observar, podemos considerar a $\ell(y^1)$ como una buena aproximación lineal.

PASO 1

Procediendo entonces a evaluar en $y^2 = (0, 306, 0, 0)$ obtenemos:

$$d(y^2) = 0.007y_1 + 0.0411y_2 + 0.0050y_3 + 0.0100y_4 + \sum_{i=1}^4 \sum_{j=1}^{10} c_{ij} x_{ij} + 16.43,$$

sujeto a las mismas restricciones.

PASO 2.

Obtenemos la siguiente solución:

$$f(y^2) = d(y^2) = 1694.832 + 16.43 = 1711.262$$

con $y^* = (35, 94, 15, 162)$, y:

$$x^* = \begin{bmatrix} 23 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12 & 53 & 0 & 0 & 0 & 29 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 24 & 13 & 0 & 0 & 87 & 0 & 38 & 0 \end{bmatrix}$$

Evaluando esta solución en la función objetivo original se tiene:

$$f(y^2) = 1731.9931,$$

PASO 3

Comparando $f(y^2)$ con la función en y^1 , podemos observar que es mejor (y^1) y, de hecho, es la solución óptima pues al evaluar en los puntos $y^3 = (0,0,306,0)$ y $y^4 = (0,0,0,306)$ se obtiene la misma solución.

Otro ejemplo del mismo tipo pero más pequeño es el siguiente: Tuy, [83].

Ejemplo 9 (Tuy, 1993)

Considere tres fábricas (F_1, F_2, F_3) y tres almacenes (W_1, W_2, W_3) y los costos dados por:

$$c = \{c_{ij}\} = \begin{bmatrix} 5 & 3 & 4 \\ 10 & 8 & 4 \\ 5 & 3 & 7 \end{bmatrix} \quad b = \{b_j\} = \begin{bmatrix} 4 \\ 3 \\ 8 \end{bmatrix}$$

$$y \ g(y) = 4y_1^{1/2} + 1y_2^{1/2} + 3y_3^{1/2} + (y_1 + \frac{1}{2} y_2 + \frac{2}{3} y_3)^{2/3},$$

con una demanda total $s = 15$, por lo cual se obtiene el triángulo:

$$\Omega = \{y = (y_1, y_2, y_3) : y_1 + y_2 + y_3 = s (= 15), y_i \geq 0\}.$$

Ver la figura 4.12 para más detalles.

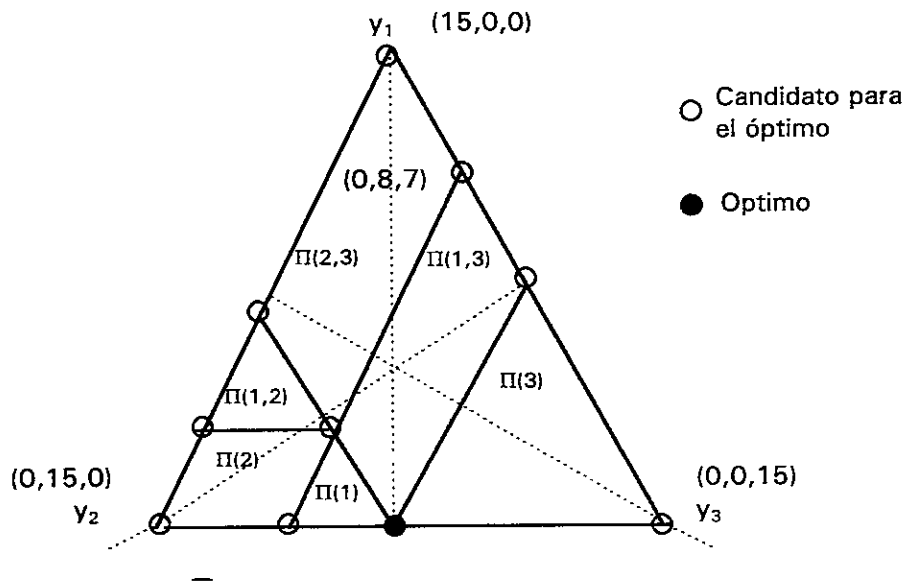


figura 4.12

Solución

De aquí podemos observar que los puntos extremos son $y^1 = (15, 0, 0)$, $y^2 = (0, 15, 0)$, $y^3 = (0, 0, 15)$, nuevamente calculamos, con base en estos puntos, tres funciones lineales de Taylor de aproximación, considerándolas como cotas inferiores al valor de la función objetivo original.

PASO 1

Sea la función $d(y^1)$ obtenida a través de evaluar el gradiente y la función en el punto $y^1 = (15, 0, 0)$.

$$\begin{aligned} \min d(y^1) = & 0.7867y_1 + 0.1351y_2 + 0.1802y_3 + 5x_{11} + 3x_{12} + 4x_{13} + \\ & 10x_{21} + 8x_{22} + 4x_{23} + 5x_{31} + 3x_{32} + 7x_{33} + 9.774 \end{aligned}$$

sujeto a:

$$x_{11} + x_{12} + x_{13} - y_1 = 0,$$

$$x_{21} + x_{22} + x_{23} - y_2 = 0,$$

$$x_{31} + x_{32} + x_{33} - y_3 = 0,$$

$$x_{11} + x_{21} + x_{31} = 4,$$

$$x_{12} + x_{22} + x_{32} = 3,$$

$$x_{13} + x_{23} + x_{33} = 8,$$

$$y_1 + y_2 + y_3 = 15.$$

PASO 2

Obtenemos la solución $f(y^1) = d(y^1) = 63.3422 + 9.774 = 73.1165$,

con $y^* = (0, 8, 7)$, y

$$x^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 8 \\ 4 & 3 & 0 \end{bmatrix}$$

evaluando esta solución en la función objetivo original se obtiene:

$$f(y^1) = 75.985$$

PASO 1

Al hacer el mismo procedimiento con $y^2 = (0, 15, 0)$ obtenemos la misma solución que con y^1 . Para el caso de $y^3 = (0, 0, 15)$ se tiene la siguiente función:

$$\begin{aligned} \min d(y^3) = & 0.30944y_1 + 0.15472y_2 + 0.5936y_3 + 5x_{11} + 3x_{12} + 4x_{13} + \\ & + 10x_{21} + 8x_{22} + 4x_{23} + 5x_{31} + 3x_{32} + 7x_{33} + 7.3565 \end{aligned}$$

sujeta a las mismas restricciones.

PASO 2

Obtenemos la solución:

$$\underline{f}(y^3) = d(y^3) = 64.40384 + 7.3565 = 71.76034$$

con $y^* = (7, 8, 0)$, y:

$$x^* = \begin{bmatrix} 4 & 3 & 0 \\ 0 & 0 & 8 \\ 0 & 0 & 0 \end{bmatrix}$$

evaluando esta solución en la función objetivo original se obtiene:

$$f(y^3) = 79.3575.$$

PASO 3

La función anterior es mayor que $f(y^1) = f(y^3) = 75.985$, por lo cual no es óptima y consideramos como solución óptima la primera, no habiendo más puntos que evaluar.

Ejemplo 10 (Una extensión al ejemplo 8)

Consideramos ahora el caso del ejemplo 1, pero aumentando una fábrica, es decir, tenemos F_i con $i=1, \dots, 5$, y por supuesto aumentando un renglón de costos a la matriz c_{ij} , y un costo fijo y^5 , procediendo a

resolver de la misma manera que en los otros dos casos, pero ahora buscamos en cinco puntos extremos.

Si consideramos cinco fábricas (F_1, \dots, F_5) que producen un cierto producto final a ser enviado a diez almacenes (W_1, \dots, W_{10}). Suponga que las demandas de los almacenes están dadas por:

$$b^T = \{b_j\} = [23, 12, 24, 13, 12, 53, 87, 15, 38, 29].$$

Las cantidades de materia prima requerida para una unidad del producto final en diferentes fábricas están dadas por:

$$\alpha = \{\alpha_i\} = (3, 5, 2, 4, 6).$$

El costo total de producción es igual a:

$$g(y) = g(y_1, \dots, y_4) = y_1^{1/2} + y_2^{1/2} + y_3^{1/2} + y_4^{1/2} + y_5^{1/2} + (3y_1 + 5y_2 + 2y_3 + 4y_4 + 6y_5)^{1/3}$$

que es una función cóncava no separable.

Los costos de transporte lineales entre las fábricas y los almacenes están dados por:

$$c = \{c_{ij}\} = \begin{bmatrix} 1 & 12 & 13 & 4 & 6 & 7 & 8 & 6 & 65 & 19 \\ 1 & 32 & 5 & 7 & 1 & 2 & 7 & 8 & 98 & 10 \\ 3 & 23 & 4 & 5 & 6 & 8 & 9 & 2 & 45 & 17 \\ 8 & 45 & 3 & 1 & 4 & 6 & 1 & 9 & 24 & 28 \\ 2 & 12 & 5 & 3 & 5 & 1 & 9 & 5 & 30 & 21 \end{bmatrix}$$

Solución

PASO 1

Sea la función $d(y^1)$ evaluada en el punto $y^1 = (306, 0, 0, 0, 0)$.

$$d(y^1) = 0.040y_1 + 0.0176y_2 + 0.0023y_3 + 0.0047y_4 + 0.021174y_5 + \\ + \sum_{i=1}^{10} \sum_{j=1}^5 c_{ij}x_{ij} + 14.97$$

sujeta a

$$\sum_{j=1}^{10} x_{1j} - y_1 = 0 ,$$

$$\sum_{j=1}^{10} x_{2j} - y_2 = 0 ,$$

PASO 1

Procediendo entonces a evaluar en $y^2 = (0, 306, 0, 0, 0)$, obtenemos:

$$\begin{aligned} \ell(y^2) &= 0.007y_1 + 0.0411y_2 + 0.0050y_3 + 0.0100y_4 + 0.015063y_5 + \\ &+ \sum_{i=1}^{10} \sum_{j=1}^5 c_{ij}x_{ij} + 16.43 \end{aligned}$$

sujeto a las mismas restricciones.

PASO 2

Obtenemos la siguiente solución:

$$\underline{f}(y^2) = \ell(y^2) = 1640.451 + 16.43 = 1656.8767$$

con $y^* = (35, 41, 15, 162, 53)$, y:

$$x^* = \begin{bmatrix} 23 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12 & 0 & 0 & 0 & 0 & 29 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 24 & 13 & 0 & 0 & 87 & 0 & 38 & 0 \\ 0 & 0 & 0 & 0 & 0 & 53 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Evaluando esta solución en la función objetivo original se tiene:

$$f(y^2) = 1683.1306$$

Al evaluar en los puntos $y^3 = (0, 0, 306, 0, 0)$, $y^5 = (0, 0, 0, 0, 306)$ se obtiene la misma solución.

PASOS 1 Y 2

En $y^4 = (0, 0, 0, 306, 0)$, la solución que se obtiene es :

$$\underline{f}(y^4) = d(y^4) = 1682.4026$$

PASO 3

Podemos observar que la solución óptima se obtiene en y^1 , y podemos resumir el trabajo sobre estos tres problemas en las siguientes tablas:

TABLA 4.9 (K = 3)

Punto Extremo	Valor de $\underline{f}(y^i)$	Valor en la f.o. original
(15,0,0)	73.11650	75.9850
(0,15,0)	68.18900	75.9850
(0,0,15)	71.76034	79.3575

TABLA 4.10 (k = 4)

Punto Extremo	Valor de $l(y)$	Valor en la f.o. original
(306,0,0,0)	1707.3050	1730.7228
(0,306,0,0)	1711.2620	1731.7228
(0,0,306,0)	1706.9330	1731.9931
(0,0,0,306)	1709.5208	1731.9931

TABLA 4.11 (K = 5)

Punto Extremo	Valor de $l(y)$	Valor en la f.o. original
(306,0,0,0,0)	1654.2690	1679.8200
(0,306,0,0,0)	1656.8767	1683.1306
(0,0,306,0,0)	1656.8740	1683.1306
(0,0,0,306,0)	1659.6070	1682.4026
(0,0,0,0,306)	1635.6292	1683.1306

CONCLUSIONES

"Pero de la estupidez nadie está descartado: los intelectuales la llevamos dentro como una enfermedad profesional, es para nosotros como la silicosis para los mineros...Por eso toda vigilancia es poca y cada cual debe hacerse chequeos periódicos a sí mismo para descubrir a tiempo la incubación de la estupidez. Los síntomas más frecuentes: espíritu de seriedad, sentirse poseído de una alta misión, miedo a los otros acompañado de loco afán de gustar a todos, impaciencia ante la realidad, mayor respeto a los títulos académicos que a la sensatez o fuerza racional de los argumentos expuestos, olvido de los límites y tendencia al vértigo intoxicador, etcétera".

Fernando Savater. *Diccionario filosófico*.

Con base en lo expuesto en este documento, se puede concluir que, aún queda mucho camino por recorrer en este campo, y que el algoritmo que aquí se propone, puede mejorarse en varios aspectos:

En cuanto a la manera de formular la función de aproximación lineal, se puede seguir investigando en el sentido de ampliar el rango de funciones y tratar de establecer un criterio general aplicable a una mayor cantidad de funciones.

Sobre la parte de ramificación y acotamiento, se puede verificar si métodos como el de Balas [5] se pueden implementar en este caso y cuanto mejora la eficiencia del algoritmo. Cabe recalcar aquí que este algoritmo aunque es del

No pretendo dar aquí todas las respuestas, este trabajo solo busca contribuir en este campo con el uso de técnicas como Rama y Corte que han demostrado ser eficientes. Y que ahora con el avance de la computación han mostrado que es posible seguir desarrollando algoritmos que proporcionen soluciones exactas. Es posible también auxiliarse de métodos heurísticos que aceleren la parte de Ramificación. Por todo esto puedo afirmar como decía en un inicio, que esta área está abierta a muchos desarrollos paralelos, así como al uso de problemas prácticos que permitan seguir perfeccionando estas técnicas.

También se puede seguir la línea de investigación de los modelos de producción-transporte, así como la elaboración de un programa para dicho algoritmo. En cuanto a la programación del algoritmo, también detecté la necesidad de programarlo con el paquete CPLEX, pero debido a la falta de tiempo y de recursos para poder conseguirlo no fue posible hacerlo en este trabajo, queda pues, abierta la puerta para seguir desarrollándolo con esta herramienta.

Tipo Rama y Corte, no lo es en el sentido tradicional de los que se mencionan en el Capítulo 2, es decir, los cortes no cumplen la misma función que los cortes del tipo de Gomory, y aquí también hay mucho que hacer en cuanto a los criterios de cuándo se debe cortar y cuándo se debe ramificar.

Me gustaría concluir diciendo que la manera en que está estructurado el trabajo, incluyendo las citas que se presentan al inicio de cada capítulo, se sustenta en los trabajos desarrollados con anterioridad y se presenta la aportación del mismo evitando caer en la grandielocuencia. Como dice Fernando Savater (¡prometo que ésta es la última cita!): "¿Por qué citar? Hay dos razones: la modestia y el orgullo. Se cita por modestia, reconociendo que el acierto que se comparte tiene origen ajeno y que uno llegó después. Se cita por orgullo, ya que es más digno y más cortés, según dijo Borges, enorgullecerse de las páginas que uno ha leído que de las que ha escrito".

APENDICE A

IMPLEMENTACION DEL ALGORITMO

El punto fundamental en el desarrollo del algoritmo, es tener control sobre la información interna en el proceso de los algoritmos Simplex y Rama y Corte, los cuales son utilizados como herramientas base en el desarrollo.

Requerimientos:

- Rutina que permita examinar los valores del Tableau.
- Rutina de Ramificación y acotamiento con control en los parámetros de decisión.

Opciones de desarrollo:

Se revisaron posibles bibliotecas de funciones para los lenguajes Fortran, C y Visual Basic, no encontrando ninguna que cumpla con los requisitos antes descritos.

Al no contar con dichas rutinas y de aquí su obligada implementación, se consideran otros factores en la elección de la herramienta, reduciéndose a contar fundamentalmente con una rutina que permita la resolución del programa lineal con facilidad en la revisión de resultados parciales.

Desarrollo:

Se elige con dicho fin el lenguaje de programación Visual Basic adjunto a la Hoja de Cálculo Excel, ya que éste permite revisar de forma dinámica los resultados parciales, así como la utilización de funciones (macros) incluidas en el mismo.

Se desarrolla el método Simplex, con la finalidad de agregar cortes al modelo. Para la parte de Rama y Corte se utiliza un módulo desarrollado en Visual Basic. Dicho módulo, comprende un conjunto de macros para la solución del problema lineal, lo cual permite desarrollar la versión de Rama y Corte del algoritmo, teniendo control sobre los parámetros de decisión.

El Algoritmo al ser trasladado a un lenguaje de programación se transforma en un conjunto de rutinas:

- a) *Branch_Cut*. Rutina principal que refleja la iteración mayor al ir agregando los cortes y verificando condiciones
- b) *Resuelve*: Implementación del Simplex. Versión propia que permite la revisión del Tableau y adición de cortes.
- c) *Ramificación*: Implementación de Ramificación y Acotamiento que permite la selección de parámetros para la elección de la rama.
- d) *Solve*. Rutina para la solución del problema lineal. Utilizada en Ramificación y Corte.

Observaciones:

El utilizar la plataforma Excel para el desarrollo del Algoritmo, abre la posibilidad de conjuntar un macro que pueda ser utilizado como herramienta común en una hoja de cálculo, siendo éste el software más común utilizado para análisis y administración.

Algoritmo: Branch and Cut

Step 1. Definición de la función L estimación de $F(x)$

	B	C	D	E	F	G
					Branch Cut	
Parametros Iniciales						
	-4	-3				
1		3.5	22.75			
2		3	23.5			
	3.5	2	29.75			Botón de Ejecución

Sub Branch_Cut

Step 2. Resolución del problema lineal.

2.1 Call Inicia(Res)

```
If Res = 1 Then
    Call Marca_Res
    Entero y óptimo. Exit Sub
End-If
```

2.2. $i = 0$

Repetir. Do

Step 3. Actualización de Puntos extremos

$X[i + 1] = \text{Puntos extremos}$

$F \text{ inf } (x[i + 1]) = L (x[i + 1])$

Step 4. Comparación de Cotas

If $i > 0$ Then

Si $L(x [i + 1]) \geq F \text{ sup } (x^*)$ entonces

Si x^* es entero entonces

Call Marca_Res

Entero y óptimo. **Exit Sub**

en caso contrario

Ejecutar 6 (Ramificación). **Call Ramifica**

Exit Sub

En caso contrario

Si $F \text{ sup } (x[i+1]) < F \text{ sup } (x^*)$ y $x^* = x[i+1]$ entonces

$F \text{ sup } (x^*) = F \text{ sup } (x[i+1])$

Step 5. Agregar Plano de Corte Convexo.

5.1. $i = i + 1$

5.2. Agregar Plano. **Call Agrega**

5.3. Resolver. **Call Resuelve**

5.4. Si la solución es entera entonces

Call Marca_Res

Entero y óptimo. **Exit Sub**

Loop

End Sub

Sub Ramifica

Step 6. Ramificación.

Repetir. Do

- Resolver. **Call Solve**
- Seleccionar Variable
- Si solución entera entonces
 Call Marca_Res
 Entero y óptimo. **Exit Sub**
- Si la solución no es factible entonces
 MsgBox "Sin Solución"
 sin solución. **Exit Sub**
- Generar rama

Loop

End Sub

Sub Inicia(Res as Integer)

- $F \text{ inf } (x[0]) = L (x[0])$
- $x^* = x[0]$
- $F \text{ sup } (x^*) = F \text{ sup } (x[0])$
- Resolver. **Call Resuelve**
- Si solución entera entonces

Res = 1

En caso contrario

Res = 0

Exit Sub

Ejemplo 1 (Taha, 1973)

Iteration 1						
4	3					
2	1		4			
1	1		3			
0	1		2.5			
1	0		1		0	1
0	1		1	2	0	2
0	0.00E+00		1	2	1	0.5
0	0		1	2	0	10
				1	0.5	

Iteration 2						
4	3					
1	3.5	22.75				
2	3	23.5				
3.5	2	29.75				
0	1.4921E-07		1	1.57580191	0.51613195	4.00435882
0	1		0	0.53799599	0.308	3.49799991
1	0.00E+00		0	0.30719798	0.48186801	6.50264212
0	0		0	0.38620803	0.92347203	36.5045682
				6.50199986	6.49900007	

APENDICE B

FUNCIONES CONCAVAS

NOTACION

Se denotarán los vectores como puntos del espacio Euclídeano n-dimensional R^n . Los subíndices se referirán a componentes de los vectores y los superíndices a vectores particulares.

Por ejemplo:

Para $n = 3$, los puntos 1 y 2 son

$$x^1 = (x^1_1, x^1_2, x^1_3)^T$$

$$x^2 = (x^2_1, x^2_2, x^2_3)^T$$

todos los vectores se considerarán como vectores columnas y T denotará transposición.

Si $x, y \in R^n$ entonces

$$x^T y = \sum_{i=1}^n x_i y_i \quad \text{es su producto interior}$$

Si f es una función real diferenciable de n variables entonces $\nabla f(x)$ es el gradiente de f , que es un vector de tamaño n consistente en las primeras derivadas parciales de f en x , $\delta f(x) / \delta x_i$, $i = 1, \dots, n$.

FUNCIONES CONCAVAS Y FUNCIONES ESTRICTAMENTE CONCAVAS.

Definición 1: Un subconjunto C de \mathbb{R}^n se dice un conjunto convexo si

$$x^1 \in C, x^2 \in C, 0 \leq \lambda \leq 1 \Rightarrow (\lambda x^1 + (1-\lambda)x^2) \in C \quad (1)$$

en otras palabras, dados dos puntos cualesquiera en C , el segmento de recta que los une también pertenece a C .

En la figura 1 se dan ejemplos gráficos de este tipo de conjuntos en \mathbb{R}^2 : a) el área circunscrita por una elipse, b) el poliedro formado por la intersección de cuatro semiespacios, y c) el subconjunto de \mathbb{R}^2 que está sobre la curva $x_2 = 1/x_1$. Los conjuntos convexos no necesitan ser cerrados, abiertos o compactos. Un punto aislado o el espacio completo \mathbb{R}^n son ejemplos triviales de conjuntos convexos. Los conjuntos convexos son importantes para el estudio

de las funciones cóncavas o convexas y algunas de sus generalizaciones ya que el dominio de dichas funciones siempre es un subconjunto convexo de \mathbb{R}^n .

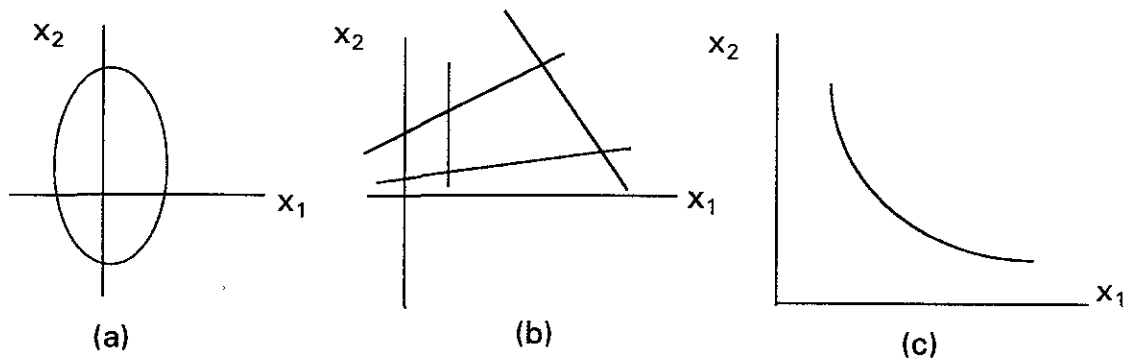


figura 1

Definición 2: a) Una función real definida en un conjunto convexo $C \subset \mathbb{R}^n$ se dice cóncava si

$$x^1, x^2 \in C, 0 \leq \lambda \leq 1 \Rightarrow f(\lambda x^1 + (1-\lambda)x^2) \geq \lambda f(x^1) + (1-\lambda)f(x^2) \quad (2)$$

b) Se dice que una función es estrictamente cóncava si

$$x^1, x^2 \in C, x^1 \neq x^2, 0 \leq \lambda \leq 1 \Rightarrow f(\lambda x^1 + (1-\lambda)x^2) > \lambda f(x^1) + (1-\lambda)f(x^2) \quad (3)$$

c) Si f es (estrictamente) cóncava, entonces $g \equiv -f$ es (estrictamente) convexa.

Es fácil ver que todas las funciones estrictamente cóncavas (convexas) son cóncavas (convexas) pero no inversamente. Más aún, encontramos que la convexidad en C se requiere para poder definir f en términos de $\lambda x^1 + (1-\lambda)x^2$.

La figura 2 muestra:

- a) una función estrictamente cóncava $f(x) = \log(x+1)$ definida sobre un intervalo abierto $C = (-1, \infty)$,
- b) una función estrictamente convexa $g(x) = x^2$ definida en la recta real \mathbb{R} completa,
- c) una función lineal $h(x) = 2 - \frac{1}{2}x$ definida en el intervalo semicerrado $C = (-\infty, 2]$, ésta función es tanto cóncava como convexa pero no estrictamente cóncava ni estrictamente convexa.

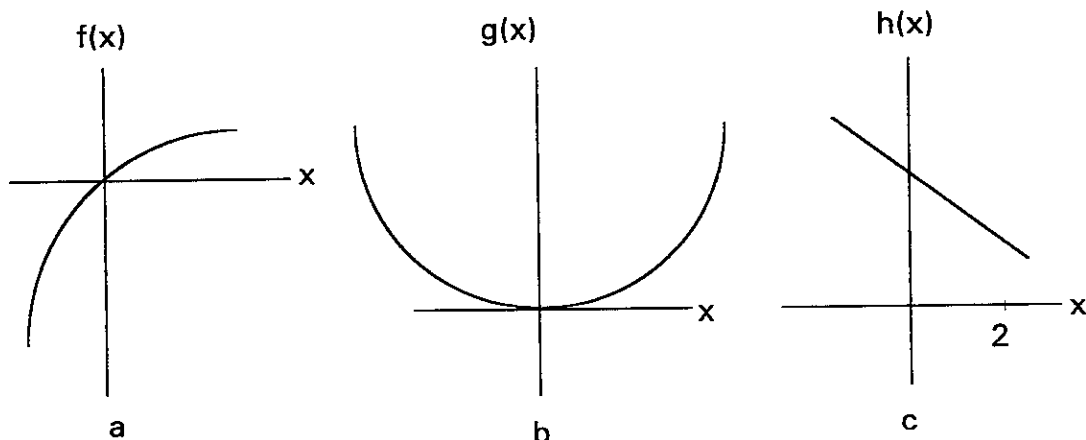


figura 2

Las funciones cóncavas, convexas, estrictamente cóncavas y estrictamente convexas se encuentran en forma natural en varias circunstancias. En administración, las funciones de costo convexas se encuentran en modelos de planeación de la producción. En este caso los costos marginales se incrementan y por consecuencia se incrementan los resultados. En algunas aplicaciones estos se decrementan en lugar de incrementar por lo que la función de costos es cóncava. Funciones que describen retornos totales a menudo son cóncavas, como por ejemplo:

$$R(x) = \sum_{i=1}^n p_i(x_i)x_i$$

donde $p_i(x_i)$ es la función de precio-demanda que se supone decreciente y lineal. La función cóncava $R(x)$ es estrictamente cóncava para $x_i \geq 0$ a menos para algún bien cuyo precio p_i no dependa de la demanda x_i . Combinando una función de retorno y una función de costo convexa, se obtiene una función de ganancia cóncava. En los modelos de minimización de riesgo en la teoría de portafolio o programación estocástica, el riesgo total es a menudo una función estrictamente convexa en las variables de decisión.

En la teoría de utilidad, la definición de concavidad es precisamente equivalente a la noción de que un individuo nunca preferiría una apuesta justa actuarial y

entonces esto es aversión al riesgo. En otras palabras, si lo que está establecido es que $w^0 = \lambda w^1 + (1-\lambda)w^2$ y apuestas 1 y 2 resultan en una ganancia final de w^1 y w^2 con probabilidades λ y $(1-\lambda)$ respectivamente, entonces $u(w^0) \geq \lambda u(w^1) + (1-\lambda)u(w^2)$. Si lo que ya está establecido se prefiere a una apuesta justa entonces f es estrictamente cóncava como en (3).

En economía las funciones de producción a menudo se suponen cóncavas, particularmente cuando el productor está maximizando sus ganancias. Un ejemplo es la función de Cobb-Douglas que tiene la forma:

$$f(x_1, \dots, x_n) \equiv \alpha_0 \prod_{i=1}^n x_i^{\alpha_i}$$

donde $x_i \geq 0$, es la cantidad de ingreso i que se usa y $\alpha_i > 0$ son los parámetros de tecnología. La función f es cóncava en el ortante no negativo si

$$\sum_{i=1}^n \alpha_i \leq 1$$

Las funciones cóncavas también se pueden encontrar en problemas de minimización de costos de producción, en el problema de minimización de precios para consumidores, en teoría del índice de costo de la vida y en el análisis de costo beneficio.

Las funciones estrictamente cóncavas también se encuentran frecuentemente en economía. Para obtener una solución única para la maximización de ganancia del productor, se supone usualmente que la función de producción es estrictamente cóncava.

Las funciones cóncavas son importantes en optimización, ya que cada máximo local de dicha función es global y si la función es estrictamente cóncava, entonces dicho máximo no se puede alcanzar en más de un punto. Las condiciones de optimalidad y las relaciones de dualidad para programas matemáticos que involucren funciones cóncavas o convexas fueron los primeros en motivar la investigación en funciones cóncavas (convexas) generalizadas.

Las funciones mostradas en la figura 2 son continuas en sus respectivos dominios y son también diferenciables en muchos conjuntos abiertos contenidos en sus dominios. Para ver que esas propiedades no son necesarias, considere la función de la figura 2c. Suponga que $C = (-\infty, 2]$, pero ahora defina h por:

$$h = \begin{cases} 2 - \frac{1}{2}x & \text{si } -\infty < x < 2 \\ 0 & \text{si } x = 2 \end{cases} \quad (4)$$

Esta función es discontinua en la frontera de su dominio pero satisface (2), entonces es una función cóncava, ya no es lineal por lo cual no puede ser cóncava y convexa al mismo tiempo. En general las funciones que son cóncavas y convexas son continuas en el interior de sus dominios, pero muchas tienen discontinuidades en sus puntos frontera.

Es fácil construir funciones cóncavas o convexas no diferenciables, por ejemplo:

$$f(x) = \begin{cases} x & \text{si } 0 < x \leq 1 \\ \frac{1}{2}x + \frac{1}{2} & \text{si } x > 1 \end{cases} \quad (5)$$

es cóncava aunque no es diferenciable en $x = 1$.

Una propiedad importante de las funciones cóncavas es que cada máximo local es global sobre su dominio de definición. Para una función estrictamente cóncava si ésta tiene un máximo, éste siempre es un máximo estricto (único).

Proposición 1 (Katzner (1970), Avriel (1976), Bazaraa and Shetty (1976))

a) Sea f una función real diferenciable en un conjunto abierto convexo $C \subset \mathbb{R}^n$. Entonces f es cóncava si y sólo si

$$x^1 \in C, x^2 \in C \Rightarrow (x^2 - x^1)^T [\nabla f(x^2) - \nabla f(x^1)] \leq 0 \quad (6)$$

- b) Es estrictamente cóncava si y sólo si la desigualdad en (6) es estricta para $x^1 \neq x^2$.
- c) Para una función (estrictamente) convexa el sentido de la desigualdad (estricta) en (6) se invierte.

Note que ambas caracterizaciones de funciones cóncavas diferenciables involucran condiciones en dos puntos. Si la función bajo consideración es doblemente diferenciable, existe una caracterización para las segundas derivadas en un solo punto.

Proposición 2 (Fenchel (1953)):

- a) Sea f una función real continua doblemente diferenciable en un conjunto convexo abierto $C \in \mathbb{R}^n$. Entonces f es cóncava si y sólo si

$$x \in C \Rightarrow \nabla^2 f(x) \text{ es negativa semidefinida} \quad (7)$$

- b) En forma semejante, f es convexa si y sólo si $\nabla^2 f(x)$ es semidefinida positiva para toda $x \in C$.

En otras palabras, los valores propios de la matriz Hessiana de una función cóncava (convexa) son no positivos (no negativos). La caracterización dada por

esta Proposición 2 no se puede extender completamente a funciones estrictamente cóncavas, al requerir que la matriz Hessiana $\nabla^2 f$ sea negativa definida. Para ilustrar este argumento considere la función estrictamente cóncava $f(x) = -x^4$ en \mathbb{R} . Claramente, la segunda derivada de f se desvanece en el origen, por lo que es no negativa en todo punto de su dominio. Sin embargo tenemos:

Proposición 3 (Fenchel (1953)):

a) Sea f una función real doblemente diferenciable definida en un conjunto abierto convexo $C \subset \mathbb{R}$. Si

$$x \in C \Rightarrow \nabla^2 f(x) \text{ es negativa definida} \quad (8)$$

entonces f es estrictamente cóncava.

b) En forma similar, si $\nabla^2 f(x)$ es positiva definida para toda $x \in C$, entonces f es estrictamente convexa.

Por ejemplo, la función $f(x) = \log x$, definida en $C = \{x: x \in \mathbb{R}, x > 0\}$ es estrictamente cóncava ya que la segunda derivada es $f''(x) = -1/x^2 < 0$ para

$x > 0$. En forma semejante. $f(x) = -\frac{1}{2}x^T x$ definida en \mathbb{R}^n es estrictamente cóncava, ya que la matriz Hessiana que consta de los elementos $\delta^2 f / \delta x_j^2 = -1$, $j = 1, \dots, n$, $\delta^2 f / \delta x_i \delta x_j = 0$, $i = 1, \dots, n$, $j = 1, \dots, n$; $i \neq j$ es negativa definida.

Una propiedad importante de las funciones (estrictamente) cóncavas y (estrictamente) convexas es que son cerradas bajo la suma y el producto por un escalar positivo. Mas aún, la suma de una función estrictamente cóncava y una función cóncava es también estrictamente cóncava, y en forma semejante en el caso convexo. Esta última propiedad ha probablemente motivado las definiciones de funciones fuertemente cóncavas y funciones fuertemente convexas.

FUNCIONES FUERTEMENTE CONCAVAS Y FUERTEMENTE CONVEXAS.

Definición 3 (Poljak (1966), Rockafellar (1976), Diewert, Avriel y Zang (1981)):

- a) Una función real f , definida sobre un conjunto convexo $C \subseteq \mathbb{R}^n$ se dice fuertemente cóncava si $f(x) = \phi(x) - \frac{1}{2}\alpha x^T x$, donde ϕ es una función cóncava definida en C , y α es un número positivo.
- b) En forma semejante, una función g se dice fuertemente convexa si $g(x) = X(x) + \frac{1}{2}\alpha x^T x$, donde X es convexa y $\alpha > 0$

Las funciones fuertemente cóncavas se usan en economía en el sentido de que si la función de producción es fuertemente cóncava, continua y doblemente diferenciable podemos entonces deducir que el sistema de demanda y oferta asociado es continuo y diferenciable una vez. Este sistema se obtiene al resolver el problema de maximización de beneficio del productor. Bajo estas condiciones las pendientes de las funciones de oferta son mayores y las de demanda son menores (con respecto a las de sus precios). La función de producción de Cobb-Douglas

$$f(x_1, \dots, x_n) \equiv \alpha_0 \prod_{i=1}^n x_i^{\alpha_i}$$

donde los parámetros positivos α_i satisfacen $\sum \alpha_i < 1$ es una función fuertemente cóncava sobre cualquier subconjunto compacto convexo del ortante positivo.

APENDICE C

LA ESTRUCTURA DE LOS POLIEDROS

CONOS

Un conjunto no vacío de puntos C es un espacio euclideo. Es un cono convexo si $\lambda x + \mu y \in C$ siempre que $x, y \in C$ y $\lambda, \mu \geq 0$. Un cono C es un *poliedro* si

$$C = \{x \mid Ax \leq 0\} \quad (1)$$

para alguna matrix A ; es decir, si C es la intersección de un número finito de semiespacios. Entendemos como un semiespacio lineal el conjunto de la forma $\{x \mid ax \leq 0\}$ para algún vector renglón a . El cono generado por los vectores x_1, \dots, x_m es el conjunto

$$\text{cono } \{x_1, \dots, x_m\} := \{ \lambda_1 x_1 + \dots + \lambda_m x_m \mid \lambda_1, \dots, \lambda_m \geq 0 \} \quad (2)$$

es el menor cono convexo que contiene a x_1, \dots, x_m . Un cono con estas características es un *cono generado en forma finita*.

Corolario 1 (Del Teorema de Farkas-Minkowski).

Un cono convexo es un poliedro si y solo si es un cono generado en forma finita.

Corolario 2 (Teorema de Descomposición para poliedros)

Un conjunto P de vectores en un espacio euclideo es un poliedro, si y solo si $P = Q + C$ para algún politopo Q y algún cono poliédrico.

CONOS CARACTERISTICOS.

El cono característico de P , denotado por cono $\text{car.}P$, es el cono poliédrico

$$\text{cono car.}P = \{y \mid x+y \in P, \forall x \in P\} = \{y \mid Ay \leq 0\}. \quad (3)$$

Observación: La última igualdad es sencilla si partimos de el hecho de que $P \neq \emptyset$, algunos autores llaman al cono característico, *cono de recesión* de P .

De aqui podemos afirmar que:

- i) $y \in \text{cono car.}P$ si y sólo si hay un x en P tal que $x + \lambda y \in P$ para toda $\lambda \geq 0$. (4)
- ii) $P + \text{cono car.}P = P$;
- iii) P es acotada si y sólo si $\text{cono car.}P = \{0\}$
- iv) Si $P = Q + C$ con Q politope y C un cono poliédrico C entonces $C = \text{cono car.}P$

Los vectores diferentes de cero en el cono $\text{car.}P$ se llaman las *direcciones infinitas* de P . El *espacio lineal* de P denotado por el espacio $\text{lin.}P$, es el espacio lineal.

$$\text{espacio lin.}P = \text{cono car.}P \cap -\text{cono car.}P = \{y \mid Ay = 0\}. \quad (5)$$

Si el espacio lineal tiene dimensión cero se dice que P es *apuntado*.

Cada poliedro no vacío P tiene una representación única $P = H + Q$, donde H es un espacio lineal, y Q es un poliedro apuntado no vacío con $Q \subseteq H^\perp$ (la única posibilidad es $H = \text{espacio lin. } P$, y $Q = H^\perp \cap P$).

CARAS

Si C es un vector diferente de cero, y $d = \max \{cx \mid Ax \leq b\}$ el hiperplano afín $\{x \mid cx = d\}$ es un *hiperplano de soporte* de P . Un subconjunto F de P es una *cara* de P si $F = P$ o si F es la intersección de P con un hiperplano de soporte de P . Directamente tenemos:

F es una cara de P si y sólo si hay un vector C para el cual F es el conjunto de vectores que alcanzan $\max\{cx \mid x \in P\}$ dado que el máximo es finito (posiblemente $c=0$). (6)

Alternativamente, F es una cara de P si y sólo si F es no vacío y

$$F = \{x \in P \mid A^1x = b^1\}, \quad (7)$$

para algún subsistema $A^1x \leq b^1$ de $Ax \leq b$.

De aquí se puede afirmar que:

- i) P tiene un número finito de caras; (8)
- ii) cada cara es un poliedro no vacío;
- iii) si F es una cara de P y $F' \subseteq F$, entonces F' es una cara de P si y sólo si F' es una cara de F.

CARAS MINIMAS

Una *cara mínima* de P es una cara que no contiene a otra cara cualquiera es decir:

Una cara F de P es una cara mínima si y sólo si F es un subespacio afín.

Teorema 1(Hoffman y Kruskal, 1956)

Un conjunto F es una cara mínima de P, si y sólo si $\phi \neq F \subseteq P$ y

$$F = \{x \mid A'x = b'\} \quad (9)$$

para algún subsistema $A'x \leq b'$ de $Ax \leq b$.

Más aún, cada cara mínima de P es una traslación del espacio lineal de P .

FACETAS

Una faceta de P es una cara mínima distinta de P .

Cada cara de P excepto para P mismo, es la intersección de facetas de P .

La dimensión de cualquier faceta de P es igual a la dimensión de P menos uno.

DESCOMPOSICION DE UN POLIEDRO.

Sean F_1, \dots, F_r las caras mínimas del poliedro $P = \{x \mid Ax \leq b\}$ y seleccione un elemento x_i de F_i para $i = 1, \dots, r$. Entonces

$$P = \text{cubierta conv.}\{x_1, \dots, x_r\} + \text{cono car.}P, \quad (10)$$

asi cada poliedro tiene una representación única.

$$P = \text{cubierta conv.}\{x_1, \dots, x_r\} + \text{cono}\{y_1, \dots, y_t\} + \text{espacio lin.}P, \quad (11)$$

donde $x_1, \dots, x_r, y_1, \dots, y_l$ son ortogonales al espacio lineal de P , y donde las y_i tienen un producto único por un escalar positivo.

La representación (11) es en cierta forma una representación interna de un poliedro.

Sea F una cara de P , ya que las caras mínimas de F son también caras mínimas de P , y ya que las caras mínimas propias de cono $\text{car.}F$ son también caras mínimas propias de cono $\text{car.} P$, entonces F tiene una representación como:

$$F = \text{cubierta conv.}(X') + \text{cono}(Y') + \text{espacio lin.}P, \quad (12)$$

con $X' \subseteq \{x_1, \dots, x_r\}$ y $Y' \subseteq \{y_1, \dots, y_l\}$.

Si P es apuntado entonces espacio lin. $P = \{0\}$, y las x_i son exactamente los vértices de P , y las y_i son representaciones no negativas de los rayos extremos del cono característico de P .

Si P es un politopo, $P = \text{cubierta conv.}\{x_1, \dots, x_r\}$, entonces todos los vértices de P están entre x_1, \dots, x_r . Así si $\{x_1, \dots, x_r\}$ es mínima, es el conjunto de vértices de P .

Resumiendo, tenemos el siguiente teorema:

TEOREMA 2.

Sea $P = \{x \mid Ax \leq b\}$ un poliedro no vacío

- i) Seleccione para cada cara mínima F de P , un vector arbitrario $X_F \in F$ (que es una solución arbitraria de $A'x \leq b'$ de $Ax \leq b$),
- ii) Seleccione para cada cara mínima propia F de $\text{cono car.}P$, un vector arbitrario y_F en $F \setminus \text{espacio lin.}P$ (que es una solución arbitraria de $A'x = 0$; $ax < 0$ para alguna submatriz renglón $[A' \ a]$ de A),
- iii) Seleccione una colección arbitraria z_1, \dots, z_t que genere el espacio $\text{lin.}P$ (que puede ser un sistema arbitrario fundamental de soluciones de $Ax = 0$). Entonces,

$$P = \text{cubierta conv.} \{X_F \mid F \text{ es una cara mínima de } P\} + \text{cono} \{y_F \mid F \text{ es una cara mínima propia de } \text{cono car.}P\} + \text{cubierta lin.} \{z_1, \dots, z_t\}. \quad (13)$$

VERTICES Y COMPLEJIDAD DE FACETAS

Sea $P \subseteq \mathbb{R}^n$ un poliedro racional. Definimos la *complejidad de una faceta* de P como el menor número ϕ tal que $\phi \geq n$ y existe un sistema $Ax \leq b$ de desigualdades racionales lineales que definen a P , donde cada desigualdad

en $Ax \leq b$ tiene un tamaño a lo mas de ϕ . La *complejidad del vértice* de P es el menor número v , tal que $v \geq n$ y existen vectores racionales $x_1, \dots, x_k, y_1, \dots, y_t$ con:

$$P = \text{cubierta conv.} \{x_1, \dots, x_k\} + \text{cono} \{y_1, \dots, y_t\}, \quad (14)$$

donde cada $x_1, \dots, x_k, y_1, \dots, y_t$ tiene tamaño a lo mas de n . (Note que la complejidad de facetas y vértices también se definen aunque P no tenga facetas o vértices). El siguiente teorema muestra que la complejidad de facetas y vértices es polinomial.

Teorema 3.

Sea P un poliedro racional en R^n de complejidad de facetas ϕ y complejidad de vértices v . Entonces $v \leq 4n^2\phi$ y $\phi \leq 4n^2v$.

Demostración.

Primero demostramos que $v \leq 4n^2\phi$. Sea $P = \{x \mid Ax \leq b\}$, donde cada desigualdad en $Ax \leq b$ tiene a lo mas un tamaño ϕ . Por el teorema 2 es suficiente demostrar que:

- i) Cada cara mínima de P contiene un vector de tamaño a lo mas de $4n^2\phi$, esto se sigue del hecho de que cada cara mínima es de la forma $\{x \mid A'x = b'\}$, con $A'x \leq b'$ siendo un subsistema de $Ax \leq b$;
- ii) El espacio lineal de P es generado por vectores de tamaño a lo más $4n^2\phi$, esto se sigue de que el espacio lineal es $\{x \mid Ax = 0\}$ y del corolario siguiente:

Corolario

Sea A una matriz racional de $m \times n$ y sea b un vector columna racional tal que cada renglón de la matriz $[A \mid b]$ tiene un tamaño de a lo mas ϕ . Si $Ax = b$ tiene solución, entonces

$$\{x \mid Ax = b\} = \{x_0 + \lambda_1 x_1 + \dots + \lambda_t x_t \mid \lambda_1, \dots, \lambda_t \in \mathbb{R}\}$$

para vectores racionales x_0, \dots, x_t de tamaño a lo más de $4n^2\phi$.

- iii) Cada cara mínima propia F de cono $\text{car.}P$ contiene un vector y que no esta en el espacio lin P de tamaño a lo mas de $4n^2\phi$, que se sigue también del corolario anterior (como $F = \{x \mid A'x = 0, ax \leq 0\}$ para alguna submatriz A' de A y algún renglón a de A , donde a no está en la cubierta lineal de los renglones de A').

Ahora demostraremos $\varphi \leq 4n^2v$. El caso $n=1$ es fácil. Suponemos $n \geq 2$. Sea $P =$ cubierta conv. $X +$ cono Y , donde X e Y son conjuntos de vectores racionales cada uno de tamaño de a lo más v . Primero suponga que P es de dimensión completa. Entonces cada faceta se determina por una ecuación lineal de la forma:

$$\begin{vmatrix} 1 & 1 & \dots & 1 & 0 & \dots & 0 \\ x & x_1 & \dots & x_k & y_1 & \dots & y_{n-k} \end{vmatrix} = 0 \quad (15)$$

donde $x_1, \dots, x_k \in X$ y $y_1, \dots, y_{n-k} \in Y$. Expandiendo este determinante por su primer columna obtenemos (Con $x = (\xi_1, \dots, \xi_n)^T$):

$$\sum_{i=1}^n (-1)^i (\det D_i) \xi_i = \det D_0, \quad (16)$$

donde D_i denota el $(y+1, 1)$ menor de la matriz en (15), y $\det D_0$ tiene un tamaño menor que $2nv$, para $i \neq 0$, $\det D_i$ tiene tamaño menor que $2n(v+1)$. Entonces la ecuación y la correspondiente desigualdad para la faceta, tiene a lo más un tamaño de $4n^2v$.

Si P no tiene dimensión completa, podemos encontrar desigualdades de tamaño a lo más de $4n^2v$, que definen la cubierta afín de P . Más aún, existen $n - \dim(P)$ coordenadas de tal forma que al suprimirlas hace a P de

dimensión completa. Este poliedro proyectado Q tiene una complejidad de vértices a lo más de v , y entonces, por lo anterior, puede definirse por desigualdades lineales de tamaño máximo de $4(n-\dim(P))^2v$. Aumentando coeficientes cero en las coordenadas eliminadas, podemos extender esas desigualdades a desigualdades válidas para P . Junto con las desigualdades que definen la cubierta afín de P ellas definen P .

BIBLIOGRAFIA

1. Altman, M. (1968), "Bilinear Programming", *Bull. Acad. Polon. Sci. Ser. Sci. Math. Astronom. Phys* 16, 741-745.
2. Avriel, M. (1976), *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs. NJ.
3. Balas, E. (1971), "Intersection Cuts- A New Type of Cutting Plane for Integer Programming", *Operations Research* 19, 19-39.
4. Balas, E., Bowmas V. J., Glover, F. and Sommer, D. (1971), "An Intersection Cut from the Dual of the Unit Hypercube" *Operations Research* 19, 40-44.
5. Balas, E., Ceria, S., and Cornuejols, G. (1996), "Mixed 0-1 Programming by Lift-and-Project in a Branch-and-Cut Framework", *Management Science* 42, No. 9, 1229-1246.
6. Bali, S. (1973), "Minimization of a Concave Function on a Bounded Convex Polyhedron", Ph.D dissertation. *University of California, Los Angeles*.

7. Balinski, M.L. (1961), "An Algorithm for Finding All Vertices of Polyhedral Sets", *J.Soc.Indust. Appl. Math* 9, 72-89.
8. Beale, E., Forrest, J., (1976), "Global Optimization Using Special Ordered Sets", *Mathematical Programming* 10, 52-69.
9. Beale, E. (1978), "Global Optimization as an Extension of Integer Programming", *Towards Global Optimization* 2, 131-149.
10. Beale, E., Tomlin, J. (1970), "Special Facilities in a General Mathematical Programming System for Nonconvex Problems Using Ordered Sets of Variables", *Proc. Fifth International Conference on Operations Research*, 447-454.
11. Beasley, J.E. (1996), *Advances in Linear and Integer Programming*, Clarendon Press Oxford.
12. Benson, Harold P. (1982), " On the Convergence of Two Branch and Bound Algorithms for Nonconvex Programming Problems", *JOTA* 36, 129-134.

13. Benson, Harold P., Sayin, S. (1994), " A Finite Concave Minimization Algorithm Using Branch and Bound and Neighbor Generation", *Journal of Global Optimization* 5, 1-14.
14. Bertsekas, D.P. (1979), "Convexification Procedures and Decomposition Methods for Nonconvex Optimization Problems", *JOTA* 29, 169-197.
15. Bolintineanu, S.(1993), "Minimization of a Quasi-Concave Function over an Efficient Set", *Mathematical Programming* 61, 89-110.
16. Boyd, A. (1993), "Generating Fenchel Cutting Planes for Knapsack Polyhedra", *Siam J. Optimization* 3-4, 734-750.
17. Boyd, A. (1994), "Solving 0/1 Integer Programs with Enumeration Cutting Planes" *Annals of Operations Research* 50, 61-72
18. Boyd, A. (1994), "Fenchel Cutting Planes for Integer Programs", *Operations Research* 42, 53-64.
19. Bracken, J. and Mc. Cormick, Garth P. (1968), *Selected Applications of Nonlinear Programming*, John Wiley and Sons, Inc.

20. Bullatov, V.P., Kasinskaya, L.I. (1982), "Some Methods of Concave Minimization on a Convex Polyhedron and their Applications", *Methods of Optimization and Their Applications*, Nauka Sibirsk.Otdel. Novosibirsk, 71-80.
21. Burdet, C.A. (1974), "Generating all the Faces of a Polyhedron", *SIAM J. Appl. Math.* 26, 72-89.
22. Cabot, A., Francis R. (1970), "Solving Certain Nonconvex Quadratic Minimization Problems by Ranking the Extreme Points", *Opns. Res.* 18, 82-86.
23. Carrol, C., Kimball, M. (1996), "On the Concavity of the Consumption Function", *Econométrica* 64, No. 4, 981-992.
24. Ceria, S. (1996), *Algoritmos para Programación Entera*, apuntes de la 3a. Escuela Latinoamericana de Verano en Investigación Operativa (8 al 13 de Enero, Bariloche, Argentina).
25. Czochralska, I. (1982), *Bilinear Programming*, *Zastosow. Mat.*, 17, 495-514.
26. Cook, W., Cunningham, W., Pulleyblank, W. and Shrijever, A. (1995), *Combinatorial Optimization*, Apuntes de la 3a ELAVIO, Bariloche Argentina.

27. Dakin, R.J. (1965), "A tree-search algorithm for mixed integer programming problems", *The Computer Journal* 8, 250-255.
28. Dyer, M.E., Proll, L.G. (1977), "An Algorithm for Determining all Extreme Points of A Convex Polytope", *Operations Research*, 81-96.
29. Everett, H.III. (1963), "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", *Operations Research* 11, 399-417.
30. Falk, J.E., Soland R.M.(1969), "An Algorithm for Separable Nonconvex Programming Problems", *Management Science* 15, No 9, 550-569.
31. Falk, J.E., Horowitz, J.L.(1972), "Critical Path Problems with Concave Cost-Time Curves", *Management Science*, Vol 19, No 4, Part I.
32. Falk, J.E. and Hoffman, K.R. (1976), "A Successive Underestimation Method for Concave Minimization Problems", *Mathematics of Operations Research*, Vol. 1, No.3, 251-258.
33. Floudas, C., Pardalos, P.M. (1992), *Recent Advances in Global Optimization*, Princeton University Press.

34. Forgó, F.(1973), "Cutting Plane Methods for Solving Nonconvex Programming Problems", *Acta Cybernetica* 1, 171-192.
35. Frieze, A.M. (1974), "A Bilinear Programming Formulation of the 3-dimensional Assignment Problem", *Mathematical Programming* 7, 376-379.
36. Gallo, G., Ulculu, A. (1977), "Bilinear Programming: An Exact Algorithm", *Mathematical Programming* 12, 173-174.
37. Garey, M.R., Johnson, D.S. and Stockmeyer, L. (1976), "Some Simplified NP-Complete Problems", *Theoret.Comp.Sci.* 1, 237-268.
38. Gendron, B. and Crainic, T.(1994), "Parallel Branch and Bound Algorithms: Survey and Synthesis" *Operations Research Vol 42, No.6*, 1042-1066
39. Glover, F. (1972), "Convexity Cuts and Cut Search" *Operations Research*, 123-134.
40. Glover, F., Klingman, D.(1973), "Concave Programming Applied to a Special Class of 0-1 Integer Programs", *Operations Research* 21, 135-140.

41. Garfinkel, R.S., Nemhauser, G.L. (1972), *Integer Programming*, John Wiley and Sons.
42. Grotchel, M. and Holland, O. (1991), "Solution of Large Scale Symmetric Travelling Salesman Problems", *Mathematical Programming* 51, 141-202.
43. Hammer, P.L. (1965), "Some Network Flow Problems Solved with Pseudo-Boolean Programming", *Operations Research* 13, 388-399.
44. Hoffman, Karla L. (1975), "A Successive Underestimating Method for Concave Minimization Problem", *Ph. D. Thesis, George Washington University DC*.
45. Hoffman, Karla L. (1981), "A Method for Globally Minimizing Concave Functions over Convex Sets", *Mathematical Programming* 20, 22-32.
46. Hoffman, K., Padberg, M. (1993), "Solving Airline Crew Scheduling Problems by Branch and Cut", *Management Science* 39-6, 657-682.
47. Horst, R. (1978), "A New Approach for Separable Nonconvex Minimization Problems Including a Method for Finding the Global Minimum of a Function of a Single Variable", *Proceedings in Operations Research* 7, Sixth Annual Meeting, 39-47.

48. Horst, R. (1980), "A Note on the Convergence of an Algorithm for Nonconvex Programming Problems", *Mathematical Programming* 19, 37-38.
49. Horst, R. (1982), "A Note on Functions, whose Local Minimum are Global", *JOTA* 36, 457-463.
50. Horst, R., Tuy, H. (1996), *Global Optimization*, Springer. 3rd. Edition.
51. Jacobsen, S.E. (1981), "Convergence of a Tuy-type algorithm for Concave Minimization Subject to Linear Inequality Constraints", *Appl. Math. Opt.*, 71-79.
52. Kelly, J.E. Jr. (1960), "The Cutting Plane Method for Solving Convex Programs", *J. Soc. Indust. Appl. Math.* 8, 703-712.
53. Konno, H. (1976), "A Cutting Plane Algorithm for Solving Bilinear Programs", *Math. Progr.* 11, 171-189.
54. Konno, H. (1981), "An Algorithm for Solving Bilinear Knapsack Problems", *J. Oper. Res. Soc. Japan* 24, 360-373.
55. Koslov, M.K., Tarasov, S.P. and Khachian, L.G. (1979), "Polynomial Solvability of Convex Quadratic Programming", *Soviet Math. Dokl.* 20, 1108-1111.

56. Krynski, S. L. (1979), "Minimization of a Concave Function under Linear Constraints (Modification of Tuy's Method)", *Survey of Mathematical Programming 1*, North-Holland Amsterdam, 479-493.
57. McCormick, G.P. (1976), "Computability of Global Solutions to Factorable Nonconvex Programs: Part 1 -Convex Underestimating Problems", *Mathematical Programming 10*, 147-175.
58. McCormick, G.P. (1983), *Nonlinear Programming: Theory, Algorithms and Applications*, John Wiley, New York.
59. Mc. Keown, P. G. (1976), "Extreme Point Ranking Algorithms: A Computational Survey" *College of Business Administration. University of Georgia*.
60. Manas, M., Nedoma J. (1974), "Finding all Vertices of a Convex polyhedral Set", *Numer. Math. 9*, 35-39.
61. Mangasarian, O.L., Shiau, T.H. (1984), "A Variable Complexity Norm Maximization Problem", *Computer Science Department, Univ. Wisconsin Tech. Report 568*, Univ. Wisconsin, Madison.

62. Mukhamediev, B.M. (1982), "Approximate Methods of Solving Concave Programming Problems", *Zh. Vychisl. Mat. Fiz.* 22, 727-732.
63. Nemhauser, G., Wolsey, L. (1988), *Integer and Combinatorial Optimization*, John Wiley & Sons.
64. Padberg, M., Rinaldi, G. (1991), "A Branch and Cut Algorithm for the Resolution of Large Scale Symmetric Traveling Salesman", *Siam Review* 33, No.1, 60-100.
65. Pardalos, P.M. and Rosen J.B. (1986), "Methods for Global Concave Minimization: A Bibliographic Survey", *Siam Review* 3, 367-376.
66. Parker Shectman, J. y Sahinidis, Nikolaos (1997), "A Finite Algorithm for Global Minimization of Separable Concave Programs", *Journal of Global Optimization* 00:1-36.
67. Raghavachari, M. (1969), "On Connections between Zero-One Integer Programming and Concave Programming under Linear Constraints", *Operations Research* 17, 680-684.

68. Reeves, G. R. (1975), " Global Minimization in Non-Convex-all-Quadratic Programming", *Management Science* 22, 76-86.
69. Ritter, K. (1966), "A Method for Solving Maximum Problems with a Nonconcave Quadratic Objective Function", *Z. Wahrsch. Verw. Geb.* 4, 340-351.
70. Rosen, J.B. (1983), "Global Minimization of a Linearly Constrained Concave Function by Partion of Feasible Domain", *Math. Oper. Res.* 8, 215-230.
71. Rossler, M. (1973), "A Method to Calculate an Optimal Production Plan with a Concave Objective Function", *Unternehmensforschung* 20, 373-382.
72. Ryoo,Hong S., Sahinidis, Nikolaos V.(1996), "A Branch and Reduce Approach to Global Optimization", *Journal of Global Minimization* 8-2, 107-139.
73. Schirjver, A. (1986), *Theory of Linear and Integer Programming*, John Wiley & Sons.
74. Schaible, S. and Ziemba, W.T. (1981), "Generalized Concavity in Optimization and Economics", *Academic Press*.

75. Soland, R. M. (1971), "An Algorithm For Separable Nonconvex Programming Problems II: Nonconvex Constraints", *Management Science Vol 17-11*.
76. Taha, H.A. (1973), "Concave Minimization over a Convex Polyhedron" *Office of Naval Research Vol 20-3*.
77. Taha, H.A. (1975), *Integer Programming, Theory, Applications, and Computations*, Academic Press.
78. Thoai, N.V. and Tuy, H. (1980), "Convergent Algorithms for Minimizing a Concave Function", *Math. Oper. Res. 5*, 556-566.
79. Thieu, T.V. (1980), "Relationship Between Bilinear Programming and Concave Minimization under Linear Constraints", *Acta Math. Vietnam 5*, 106-113.
80. Tuy, H. (1964), "Concave Programming under Linear constraints", *Dokl. Akad. Nauk. SSSR*, 159 k, 32-35, *Soviet Math Dokl*, 5, 1437-1440.
81. Tuy, H. (1983), "On outer Approximation Methods for Solving Concave Minimization Problems", *Report No. 108*, West Germany.

82. Tuy, H., Dinh Dan, N. y Ghannadan, S.(1993), "Strongly Polynomial Time Algorithms for Certain Concave Minimization Problems on Networks", *Operations Research Letters* 14, 99-109.
83. Tuy, H., Ghannadan, S., Migdalas, A., y Varbrand, P. (1993), " Strongly Polynomial Algorithm for a Production-Transportation Problem with Concave Production Cost", *Optimization* 27, 205-227.
84. Tuy, H., Ghannadan, S., Migdalas, A.y Varbrand, P. (1995), " Strongly Polynomial Algorithm for Two Special Minimum Concave Cost Network Flow Problems" *Optimization* 32, 23-43.
85. Tuy, H., Ghannadan, S., Migdalas, A.y Varbrand, P. (1996), " A Strongly Polynomial Algorithm for a Concave Production-Transportation Problem with a Fixed Number of Nonlinear Variables" *Mathematical Programming* 72, 229-258.
86. Walukiewicz, S. (1991), *Integer Programming*, Kluwer Academic Publishers.
87. Young, R.D. (1972), "Hypercylindrically Deduced Cuts in Zero-One Integer Programs", *Operations Research*, 1393-1405.
88. Zang, I. and Avriel, M. (1976), "A Note on Functions whose local Minima are Global", *JOTA*, 556-559.

89. Zwart, P.B. (1974), "Global Maximization of a Convex Function with Linear Inequality Constraints", *Oper. Res* 22, 602-609.

90. Vaish, H. (1977), "A Cutting Plane Algorithm for the Bilinear Programming Problem", *Naval Res. Logist. Quart.* 24, 83-94.