



01168

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DIVISION DE ESTUDIOS DE POSGRADO

RAMIFICACION Y ACOTAMIENTO DE UN PROBLEMA
NO COMBINATORIO

T E S I S

P R E S E N T A D A P O R :

HERNANDEZ BARAJAS RICARDO

PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERIA

(INVESTIGACION DE OPERACIONES)

DIRIGIDA POR: M.I. IDALIA FLORES DE LA MOTA

CIUDAD UNIVERSITARIA,

MARZO DE 1997

TESIS CON
FALLA DE ORIGEN



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.

Sé lo utópico que resulta querer agradecer a todos los que de alguna forma, contribuyeron a la consecución de esta meta; así, agradeciendo de antemano a cada uno de ellos, quiero mencionar de manera especial a:

DIOS, "GRACIAS".

Mis padres, por lograr sembrar en mí el valor de la confianza en mis aptitudes y el de la seguridad de mis limitaciones. Gracias "ogros".

Maribell Rojas, ya que estando a su lado siempre tuve fuerzas y ánimo para seguir adelante. Y.T.I.G.Y.A.L.A.TI.T.

Luisa Pérez, Norma Peralta, Rosa García, Mario Delgadillo y Pablo Ornelas, pues, a pesar de que nuestros caminos se han ido apartando, han querido conservar nuestra amistad. Gracias amigos.

Dr. Hugo Rincón, por quien aprendí a razonar los problemas matemáticos. Hugo: gracias por tu forma de dar clase.

Dr. Ricardo Aceves, por aceptar ser mi tutor en la Maestría y, siéndolo, siempre dejó que yo eligiera mi camino. Gracias "tocayo".

M.I. Idalia Flores, por su increíble confianza en mí, la cual me impulsó a dar mi mejor esfuerzo en el presente trabajo. Gracias Idalia.

Finalmente, quiero expresar mi especial reconocimiento a Norma e Ita por su contribución para hacer posible la impresión del texto y gráficos contenidos en el presente trabajo.

RICK

INDICE

INTRODUCCIÓN	1
CAPÍTULO 1.	
RAMIFICACIÓN Y ACOTAMIENTO	7
1.1 PROBLEMAS COMBINATORIOS	7
1.1.1 El Problema de la Mochila	8
1.1.2 El Problema del Agente Viajero	10
1.2 ALGORITMOS DE RAMIFICACIÓN Y ACOTAMIENTO	14
1.2.1 Definiciones	14
1.2.2 Estructura General de los Algoritmos tipo Ramificación y Acotamiento	17
1.2.3 Características de los Algoritmos R-A	17
CAPÍTULO 2.	
DOS ALGORITMOS TIPO R-A	19
2.1 RESOLVIENDO LA MOCHILA	19
2.1.1 El Algoritmo	20
2.1.2 Un Ejemplo Numérico	21
2.2 RESOLVIENDO EL AGENTE VIAJERO	27
2.2.1 Obtención de la Matriz Reducida y la Cota Asociada a un Problema	28
2.2.2 El Algoritmo	29
2.2.3 El Ejemplo Numérico	30
CAPÍTULO 3.	
CONJUNTOS Y FUNCIONES CONVEXAS	37
3.1 CONCEPTOS Y RESULTADOS	37
3.2 CUBIERTAS CONVEXAS	42
CAPÍTULO 4.	
R-A EN UN PROBLEMA NO COMBINATORIO	49
4.1 EL ALGORITMO	50
4.2 TEOREMAS DE CONVERGENCIA	57

4.3 EL EJEMPLO	66
----------------------	----

CAPÍTULO 5.

<u>CONCLUSIONES</u>	85
----------------------------------	-----------

<u>APÉNDICE</u>	88
------------------------------	-----------

<u>BIBLIOGRAFÍA</u>	93
----------------------------------	-----------

INTRODUCCIÓN

En todo ser vivo se halla de manera natural la necesidad de optimizar, por ejemplo, las abejas construyen su panal de forma tal que la cantidad de cera empleada sea mínima y tenga gran resistencia; otro ejemplo es el de las aves migratorias que, en su viaje hacia los santuarios de reproducción, siguen el recorrido más corto tomando en cuenta los elementos favorables y peligrosos en su camino.

El hombre no escapa a esta regla de necesidad, desde tiempo inmemorial optimiza, aún antes de existir los números o las matemáticas. Por ejemplo, cuando requiere de agua, sigue el recorrido más corto desde el punto donde está, hacia el lugar donde se encuentra el agua; sin embargo, pueden existir restricciones para obtenerla, como el hecho de que el camino más corto sea intransitable por la presencia de animales salvajes u otros peligros, y se tenga que seleccionar una ruta más larga pero más segura y que, al final, resultará la ruta más corta bajo las condiciones de seguridad que él considera convenientes.

Problemas como los anteriores son estudiados por diversas disciplinas científicas y tecnológicas, las cuales recurren a las matemáticas para su representación y resolución. Es aquí donde entra la Investigación de Operaciones, la cual empieza a trabajarse de manera coherente y profesional, durante los primeros años de la Segunda Guerra Mundial y, a partir de entonces, ha sido objeto de estudio para la ciencia. Aunque en un principio fue concebida para responder a una necesidad militar, su campo de actividades se ha ido extendiendo durante los casi 60 años de existencia.

Originalmente, los científicos colaboraban con oficiales para resolver problemas de operaciones militares y; aunque muchas veces los análisis científicos

eran ignorados y vistos sólo como instrumento de consulta, otras tantas fueron de gran utilidad, tal como se puede observar en la siguiente nota escrita por Sir Hugh Dowding a la Sección de Investigación de Operaciones en Gran Bretaña: "Gracias. Esta guerra será ganada por la ciencia aplicada a las necesidades en operaciones. H.D."

A pesar de tener sus inicios en la guerra, la Investigación de Operaciones ha ido abarcando campos como la Ingeniería, los negocios, las comunicaciones y la industria. Este crecimiento del campo de uso de la Investigación de Operaciones, ha traído consigo el desarrollo y diferenciación de técnicas para la solución de problemas; entre otras, se pueden identificar a la Teoría de Juegos, Teoría de Colas, Simulación Digital, Procesos Estocásticos y Optimización.

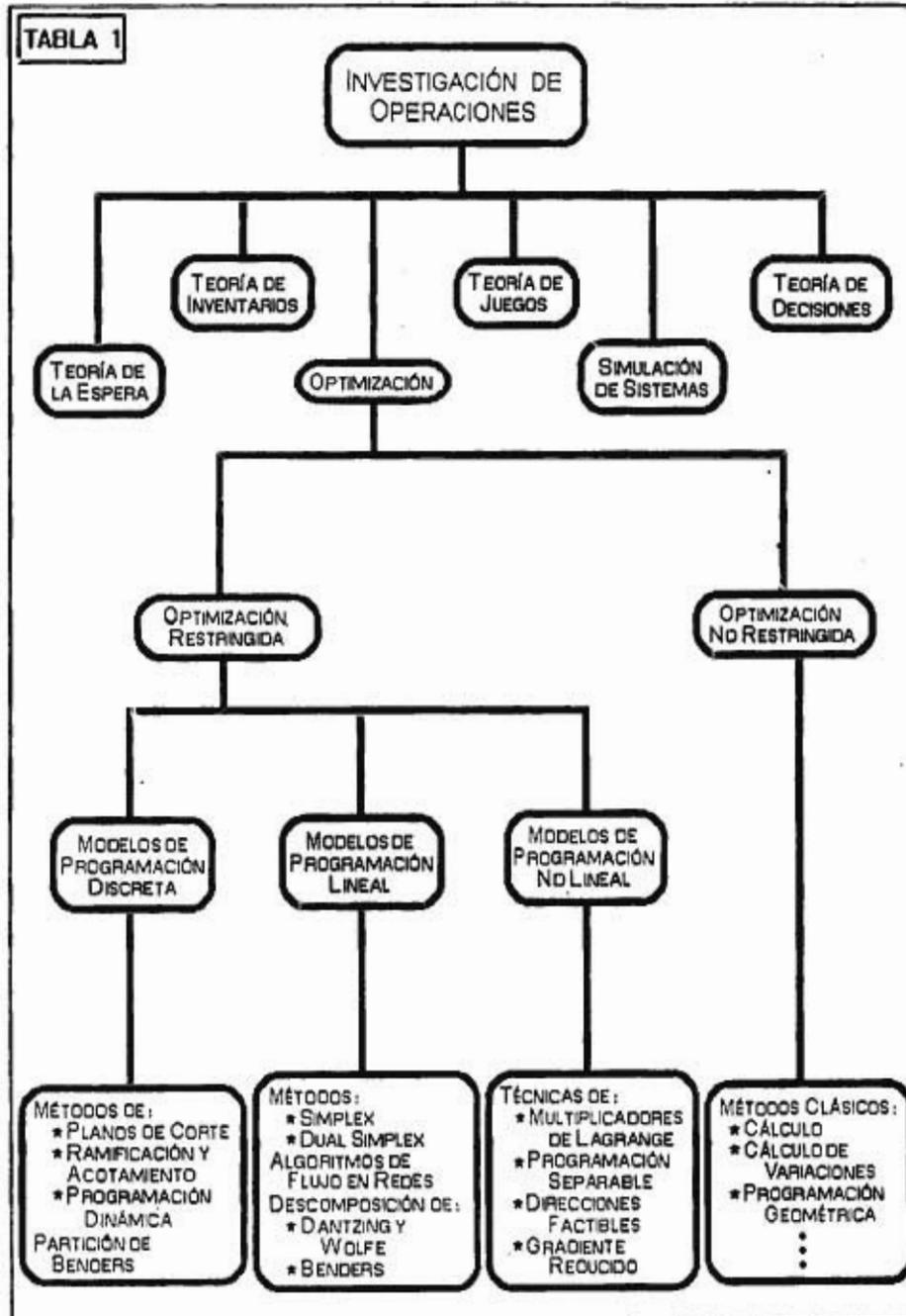
La tabla 1 muestra de manera esquematizada la idea anterior, además, debido a su relación con el presente trabajo, se disgrega el módulo de Optimización.

Los modelos matemáticos utilizados para la solución de problemas de optimización, se pueden dividir de forma natural en dos categorías, los que tienen variables continuas y los que tienen las variables discretas (aunque puede haber mixtos).

En los problemas de variables continuas se busca, generalmente, una solución en un conjunto continuo de posibles soluciones de números reales; en cambio, en los problemas de variables discretas (o de optimización combinatoria), se busca una solución en un conjunto finito o infinito numerable de posibles soluciones (números enteros, gráficas, etc.). Un tipo de problema que juega un papel único en la teoría de la optimización es la programación lineal; el cual, es un problema de optimización continua pero también, por los resultados fundamentales de su teoría, es considerado un problema de optimización combinatoria.

Los problemas de optimización continua se estudian básicamente por el Cálculo y el Análisis Matemático, no así los de optimización combinatoria, donde el estudio de las Matemáticas Discretas y la construcción de algoritmos juegan un papel muy importante.

Uno de los enfoques más generales para la solución de problemas de variables discretas, es el de Ramificación y Acotamiento (R-A), o, en palabras de Bertier y Roy [6], "separación y evaluación progresiva".



Al igual que la Programación Dinámica, la Ramificación y Acotamiento es una búsqueda estructurada e inteligente de todas las soluciones factibles; más específicamente: el espacio de soluciones factibles es dividido repetidamente en subconjuntos cada vez más pequeños, a la vez que una cota es calculada para cada subconjunto; después de cada división o partición, aquellos subconjuntos que excedan (en caso de estar minimizando) el valor de una solución factible conocida, son excluidos de particiones posteriores; finalmente, la partición continúa hasta que se encuentra una solución factible cuyo valor no sea mayor que la cota de cualquier subconjunto. Estas ideas básicas de Ramificación y Acotamiento son formalizadas a lo largo del capítulo 1 de este trabajo.

El objetivo principal de la presente tesis, es mostrar como los métodos de Ramificación y Acotamiento pueden ser utilizados en la resolución de problemas no combinatorios (de variables continuas); lo cual se pretende conseguir con la presentación de un algoritmo, tipo R-A, que resuelve problemas de minimización de funciones no convexas, y en donde, las cotas son calculadas resolviendo problemas cuya función objetivo sí es convexa.

Otro objetivo perseguido durante la elaboración de este trabajo, es el de lograr que aún las personas poco familiarizadas con los temas aquí involucrados, entiendan la filosofía de la Ramificación y Acotamiento; para lograr lo anterior, se incluye un árbol de búsqueda (definido en el capítulo 1) para cada iteración de los algoritmos.

El contenido del trabajo se resume como sigue: en el capítulo 1, se explican de manera general los métodos de Ramificación y Acotamiento; en el capítulo 2 se enuncian y ejemplifican dos algoritmos tipo R-A; en el capítulo 3 se presentan conceptos y resultados en convexidad de conjuntos y funciones y; en el capítulo 4 se enuncia el algoritmo principal de la tesis. Se incluye dentro del capítulo 4, un ejemplo de cómo se aplica el algoritmo y, además, se enuncian las conclusiones. Finalmente, cabe mencionar que al final se da un apéndice de resultados de Análisis Matemático, los cuales son utilizados para la comprensión, la construcción y la prueba de convergencia del algoritmo.

Capítulo 1

RAMIFICACIÓN Y ACOTAMIENTO

Una gran variedad de algoritmos de ramificación y acotamiento, han sido descritos a lo largo de la historia de la Optimización. En el presente capítulo se proporciona una descripción general de estos, teniendo, como objetivo principal, mostrar su gran aplicación en la resolución de problemas combinatorios; además, se da una explicación de dos problemas combinatorios clásicos: el problema de la mochila, y el del agente viajero.

1.1 PROBLEMAS COMBINATORIOS

Un problema combinatorio puede ser definido de la siguiente manera: dada una asignación de valores numéricos a un conjunto finito de variables X , optimizar (maximizar o minimizar) alguna función objetivo $Z(X)$, la cual esta sujeta a que las variables satisfagan un conjunto de restricciones.

La característica de los problemas de optimización combinatoria, de tener un conjunto finito de soluciones, hace que la obtención de la mejor solución sea un problema que podría presentar poco interés matemático (dado que la búsqueda exhaustiva permitiría resolver el problema); sin embargo, es frecuente que problemas de este tipo, cuyo tamaño es n , tenga $n!$ soluciones factibles; así, si una computadora pudiera ser programada para examinar y comparar soluciones a razón de mil millones de soluciones por segundo, la computadora podría terminar su tarea, para $n = 20$, en alrededor de 800 años; para $n = 21$, en alrededor de 16,800 años, y así sucesivamente. *¡ No*

tiene sentido resolver de esa forma un problema, si al interesado no le alcanza su vida para ver la respuesta !.

En un problema combinatorio se tiene amplia flexibilidad, tanto en el conjunto de restricciones, como en la función objetivo, a la cual, se le pide sólo que esté determinada de manera única dado un conjunto de valores de X .

Existen muchos problemas, tratados con Ramificación y Acotamiento, en los que $Z(X)$ toma sólo los valores 0 ó ∞ ; indicando así, si una solución es factible o no.

La posibilidad de resolver problemas combinatorios, con funciones objetivo no lineales, discontinuas, e incluso no matemáticas, es sin lugar a dudas, una de las grandes ventajas de los métodos de Ramificación y Acotamiento (esta misma flexibilidad existe también en las restricciones).

Como ya se había mencionado, se han seleccionado como ejemplos, el problema de la mochila y el problema del agente viajero; los cuales, aunque no son representativos de la gran gama de posibles problemas combinatorios, sí son un buen indicativo de cómo se pueden emplear los métodos de ramificación y acotamiento en la resolución de éstos.

1.1.1 El Problema de la Mochila

Un montañista (o excursionista) se dispone a realizar una de sus tantas salidas, obviamente, no puede llevar consigo todo lo que quisiera (como por ejemplo, televisión, lavadora, secadora de pelo, el cuadro con la foto de su novia, su pata de conejo, su libro de Investigación de Operaciones, o hasta sus pantuflas de dinosaurio); es entonces que él debe asignar un valor de importancia a cada uno de los artículos deseados y necesitados, y decidir cuáles llevar. Así, él podría asignar mayor importancia a una sogá y un teléfono celular, que a una secadora de pelo y una televisión.

El problema de este montañista se conoce como el problema de la mochila, en el cual, se trata de determinar que artículos incluir en la selección, de forma que se optimice lo deseado (peso, dinero, seguridad, etc.) sin exceder la capacidad de la mochila.

Muchos problemas industriales se pueden plantear como problemas tipo mochila, como por ejemplo la selección de proyectos y el control de presupuestos .

Existen muchas versiones del problema, pero en este caso, se trabajará con la versión más popular, el problema de la mochila 0-1; el cual contiene una sola restricción lineal (que es la asociada a los costos de incluir un artículo), y cuyo planteamiento es el siguiente:

Maximizar

$$\sum_{i=1}^n p_i x_i$$

sujeto a

$$\sum_{i=1}^n w_i x_i \leq W ,$$

$$x_i \in \{0, 1\} \text{ para } i = 1, \dots, n ;$$

y en donde, p_i es la ganancia de incluir al artículo i ; w_i es el costo de incluir al artículo i ; W es la capacidad de la mochila y; la variable x_i toma el valor 0 si el artículo no se incluye, y 1 en otro caso. Además, $p_i \geq 0$, $w_i \geq 0$ y $W \geq 0$.

Aunque se trabajará el problema de la mochila tipo 0-1, cabe mencionar otros: a) el *problema de la mochila acotado*, en el que se permite a x_i ser un entero no mayor que una cota dada ($0 \leq x_i \leq b_i$, $x_i \in \mathbb{Z}$); con esta restricción se plantea la posibilidad de incluir varios artículos del mismo tipo; y b) el *problema de la mochila no acotado*, en el que se acepta $b_i = \infty$.

Algo que es importante observar, es que un problema lineal entero se puede transformar en un problema tipo mochila; para ejemplificar esto, tómesese el siguiente problema:

Maximizar

$$2x_1 - 3x_2$$

sujeto a

$$x_1 + x_2 \leq 6 \quad \dots(1) \quad (\mathcal{P})$$

$$x_1 \geq 0, x_2 \geq 0 \quad \dots(2)$$

$$x_1, x_2 \in \mathbb{Z} \quad \dots(3)$$

De (1), (2) y (3), se deduce que tanto x_1 como x_2 pueden tomar los valores 0, 1, 2, 3, 4, 5 o 6; se podrían definir entonces, las nuevas variables: $x_{1,1}$, $x_{1,2}$,

$x_{1,3}, x_{2,1}, x_{2,2}, x_{2,3} \in \{0, 1\}$; de forma tal que

$$\begin{aligned}x_{1,1} + 2x_{1,2} + 4x_{1,3} &= x_1 \\x_{2,1} + 2x_{2,2} + 4x_{2,3} &= x_2\end{aligned}$$

y así, se replantea (\mathcal{P}) como un problema tipo mochila 0-1 de la siguiente manera:

$$\begin{aligned}&\text{Maximizar} \\&2x_{1,1} + 4x_{1,2} + 8x_{1,3} - 3x_{2,1} - 6x_{2,2} - 12x_{2,3} \\&\text{sujeto a} \qquad \qquad \qquad (\mathcal{P}') \\&(1') \quad x_{1,1} + 2x_{1,2} + 4x_{1,3} + x_{2,1} + 2x_{2,2} + 4x_{2,3} \leq 6 \\&(2') \quad x_{i,j} \in \{0, 1\}.\end{aligned}$$

(Si se desea conocer más acerca de como replantear problemas lineales en problemas tipo mochila, o se interesa en el tema en general, puede consultarse [11], [12] y [15].)

La observación anterior hace que el problema de la mochila, en cualquiera de sus presentaciones, tenga un campo aún más amplio de problemas reales en los cuales puede aplicarse la teoría desarrollada.

1.1.2 El Problema del Agente Viajero

Imagínese que una persona desea visitar n ciudades, de cada una de las cuales se puede trasladar a cualquier otra; es natural que el costo (tiempo, dinero, distancia, etc.) de viajar de la ciudad i hacia otras sea distinto, más que eso, podría ser que viajar de A a B sea más costoso que hacerlo de B a A .

El problema es entonces: empezar en alguna ciudad, visitar cada una de las otras ciudades una sola vez, y retornar a la ciudad original; lo anterior buscando que el recorrido sea de costo mínimo. Si se asocia un costo $c_{i,j}$ por viajar de i a j , y $\mathbb{X} = \{x_{i,j}\}$ es un conjunto de variables 0 ó 1 que, tienen asociado el número 0 si no se viaja de i a j , y 1 en otro caso, la función objetivo que se intenta minimizar es

$$Z(\mathbb{X}) = \sum_i \sum_j c_{i,j} x_{i,j}$$

que representa el costo total del recorrido.

El problema del agente viajero es un problema clásico de optimización combinatoria, que cuenta con gran número de aplicaciones; por ejemplo: problemas de distribución, donde se tiene que un almacén central de cierta compañía desea distribuir productos a cada una de sus sucursales a un costo mínimo; otro ejemplo es el de los carteros, ¿cómo debe un cartero recorrer la ciudad, para repartir la correspondencia utilizando el menor tiempo posible?; o considere la siguiente situación: una fábrica produce n artículos, pero se debe cambiar de montaje de maquinaria siempre que se cambia el artículo que se está produciendo; conociendo el costo (dinero, tiempo, personal, etc.) de montaje entre los artículos, se desea encontrar una secuencia óptima de producción.

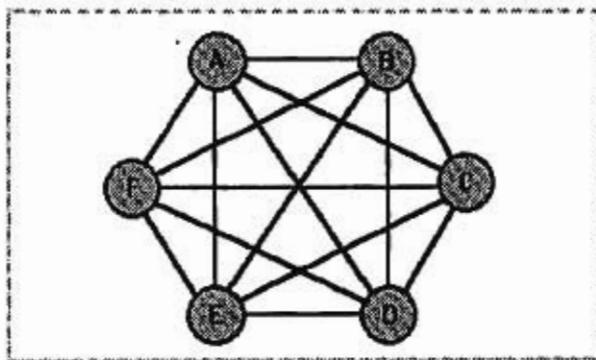


Figura 1.1 . Una gráfica completa para $n=6$.

En términos de la Teoría de Redes, el problema consiste en encontrar en una gráfica completa, con n nodos, un circuito de peso mínimo cuya longitud sea n . (Una gráfica completa es aquella en la cual entre cualquier par de nodos existe un arco que los une.) Considere, por ejemplo, un problema con 6 ciudades como se muestra en la figura 1.1, y cuya matriz de costos es

	A	B	C	D	E	F
A	-	27	43	16	30	26
B	7	-	16	1	30	25
C	20	13	-	35	5	0
D	21	16	25	-	18	18
E	12	46	27	28	-	5
F	23	5	5	9	5	-

El peso del circuito, o costo de la ruta

$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow A$$

es de 124; mientras que el circuito de peso mínimo es

$$A \rightarrow D \rightarrow C \rightarrow E \rightarrow F \rightarrow B \rightarrow A,$$

cuyo peso es 63.

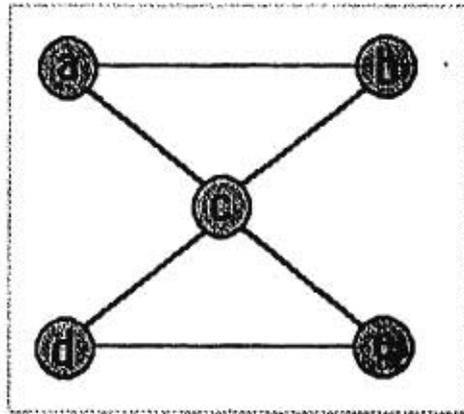


Figura 1.2 . Gráfica no completa .

Algunas veces la gráfica no es completa, esto es, existen parejas de nodos que no están directamente conectados por un arco; en tal caso, pueden no existir circuitos de longitud n (n es el número de nodos, y la longitud de un circuito es el número de arcos del circuito).

Regresando al contexto del problema del agente viajero, lo que significa que la gráfica no sea completa, es que pueden existir ciudades no comunicadas directamente por viajes (si se desea tener mayor información sobre el problema del agente viajero, puede consultarse [13] y [15]). Para ejemplificar lo anterior, véase la figura 1.2, en la cual no existen circuitos de longitud 5, y por lo tanto, el problema del agente viajero no tiene solución.

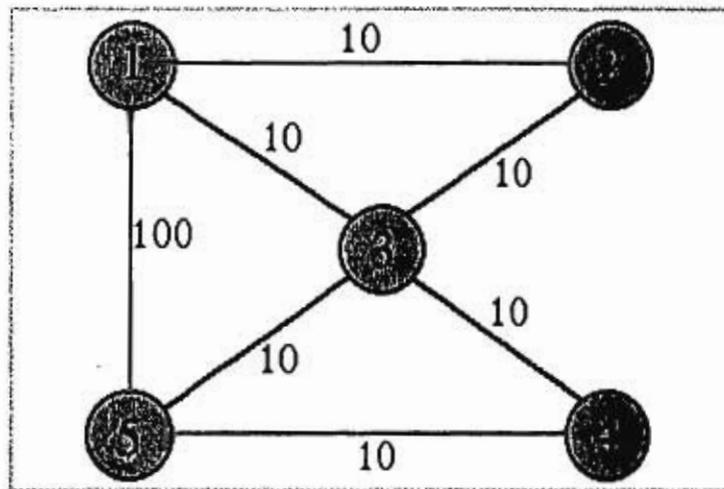


Figura 1.3. Gráfica con un único circuito de longitud 5.

Se podrá notar un cambio muy drástico, si al resolver este problema, se permite la visita a los nodos más de una vez; por ejemplo, en la figura 1.3 la única solución posible es

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$$

con costo de 140; mientras que permitiendo la visita más de una vez a los nodos, la ruta óptima es

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 1$$

(que no es un circuito) con costo de 60.

Una forma de resolver el problema del agente viajero es, sin duda, encontrar todas las posibles rutas que se pueden seguir, evaluar su costo, y tomar la de valor mínimo. El inconveniente de este método es el siguiente: una vez fijada la primera ciudad, tenemos $n - 1$ posibilidades para la segunda ciudad por visitar; una vez fijada la segunda ciudad, tenemos $n - 2$ posibilidades para la tercera; y así sucesivamente. De esta manera, podemos concluir que el total de posibles recorridos es

$$(n - 1) \times (n - 2) \times \cdots \times 2 \times 1 = (n - 1)!$$

Así, si se quiere resolver el problema de agente viajero para 21 nodos, tendrían que analizarse $20!$ posibles rutas ($20! > 10^{18}$), lo más probable es que después de analizar todas, ya no se desee viajar.

1.2 ALGORITMOS DE RAMIFICACIÓN Y ACOTAMIENTO

Con el propósito de describir los algoritmos de ramificación y acotamiento, la presente sección inicia con algunas definiciones y con un poco de notación, que permitirán un mejor manejo de los algoritmos particulares presentados en este trabajo.

1.2.1 Definiciones

RESTRICCIONES IMPLÍCITAS. Las restricciones implícitas son aquellas que deben de cumplirse por la forma en que el algoritmo se construye, por ejemplo, que las variables empiecen con valores enteros.

RESTRICCIONES EXPLÍCITAS. Las restricciones que necesitan ser verificadas como parte integral, *durante todo el algoritmo*, se denominan explícitas; por ejemplo, ecuaciones que restringen los valores que las variables pueden tomar.

SOLUCIÓN. Una solución es una asignación de valores numéricos a las variables X , y las cuales satisfacen todas las restricciones implícitas.

SOLUCIÓN FACTIBLE. Es una solución que satisface también las restricciones explícitas.

s . Se denota por s , a una asignación particular de valores numéricos de X , que además satisfacen todas las restricciones implícitas; es decir, s es una solución al problema combinatorio.

Ω . Se utiliza el símbolo Ω para representar al conjunto de soluciones de un problema combinatorio, esto es, $\Omega = \{s \mid s \text{ es solución}\}$.

\mathcal{S} . Con el símbolo \mathcal{S} se hace referencia a cualquier subconjunto de Ω .

PARTICIÓN. Una partición de Ω es una división exhaustiva de Ω en un número finito de subconjuntos $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k$; más formalmente, una partición de Ω es una colección finita $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k$ de subconjuntos de Ω , que satisfacen:

$$(1) . \mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \cup \mathcal{S}_k = \Omega$$

$$(2) . \mathcal{S}_i \cap \mathcal{S}_j = \emptyset \iff i \neq j .$$

RAMIFICACIÓN. La ramificación es el proceso de *particionamiento* de un subconjunto \mathcal{S} de Ω , en subconjuntos disjuntos \mathcal{S}_i . Este proceso de particiones sucesivas puede visualizarse mejor a través de un **ÁRBOL**, el cual se construye de la siguiente manera: el nodo inicial representa a Ω , las ramas se crean con el proceso de ramificación, y los nodos del árbol representan a los subconjuntos de Ω ; esto es, cada nodo N está asociado a algún conjunto de soluciones \mathcal{S} . Para dejar claras estas ideas véase la figura 1.4 ; en este ejemplo se utiliza $k = 2$, es decir, cada subconjunto se particiona en dos subconjuntos disjuntos; además, $N = 1$ representa a Ω , y en los nodos 3, 4, y 5 aún no se ha realizado la partición o ramificación.

La relación entre subconjuntos de Ω y la creación de un árbol , se puede formalizar con las siguientes definiciones.

$\mathcal{S}(N)$. Denota al subconjunto \mathcal{S} de Ω , que es representado por el nodo N . (En la figura 1.4, por ejemplo, $\mathcal{S}(5) = \mathcal{S}_{1,2}$.)

NODO INTERMEDIO. Un nodo intermedio, N , es aquel en el cual no se ha realizado ramificación alguna. (En la figura 1.4, 3, 4, y 5 son intermedios.)

NODO BÚSQUEDA. Es aquel nodo sobre el cual se ramificará, y al cual se le denota por i^* .

NODO FINAL. Un nodo final es un nodo intermedio para el cual $\mathcal{S}(N)$ consiste de una única solución al problema, s .

NODO ANTECESOR. El nodo antecesor de N , denotado por N^a , es el nodo inmediato superior en el árbol proporcionado por el algoritmo (a este árbol se le conoce como **ÁRBOL DE BÚSQUEDA**); en la figura 1.4, $5^a = 2$.

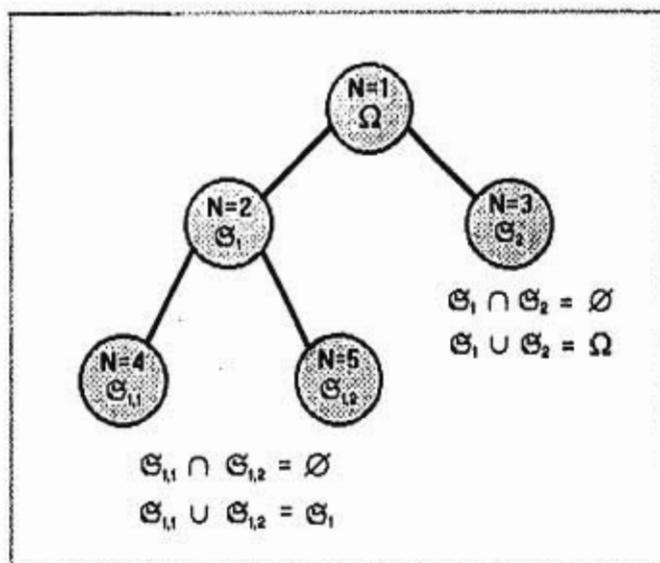


Figura 1.4 . Árbol de búsqueda .

COTA. Una cota para el nodo N , y a la cual designaremos como $\mathfrak{B}(N)$, es un valor que acota (por arriba o por abajo, dependiendo si el problema es de maximización o de minimización) el valor de la función objetivo de todas las soluciones asociadas al nodo N ; es decir, $\mathfrak{B}(N)$ es un número fijo con la propiedad de que, cualquier solución factible $s \in \mathcal{G}(N)$ tiene valor (en la función objetivo) no mejor que $\mathfrak{B}(N)$; más formalmente,

$$s \in \mathcal{G}(N) \implies Z(s) \geq \mathfrak{B}(N) \quad (\text{en caso de minimización})$$

o

$$s \in \mathcal{G}(N) \implies Z(s) \leq \mathfrak{B}(N) \quad (\text{en caso de maximización}).$$

Nótese que durante el proceso de ramificación, $\mathfrak{B}(N)$ garantiza que la búsqueda de soluciones en el nodo N , proporcionará soluciones con a lo más, tan buen valor como $\mathfrak{B}(N)$; lo anterior da lugar a una forma de decidir sobre qué rama se debe continuar ramificando, "la de mejor cota" (donde mejor quiere decir menor en caso de minimización, o mayor en caso de maximización); en otras palabras, $\mathfrak{B}(N)$ proporciona a i^* .

1.2.2 Estructura General de los Algoritmos Tipo Ramificación y Acotamiento

Usando las definiciones y la notación de la subsección anterior, se puede definir de manera general a un algoritmo de ramificación y acotamiento:

Un algoritmo de ramificación y acotamiento es un conjunto de reglas para:

- a. *Ramificar desde nodos existentes a nodos nuevos.*
- b. *Determinar cotas para los nuevos nodos.*
- c. *Decidir sobre que nodo ramificar (seleccionar i^*).*
- d. *Reconocer cuándo un nodo contiene soluciones no factibles o cuándo contiene soluciones no óptimas.*
- e. *Reconocer cuándo un nodo final contiene una solución óptima.*

1.2.3 Características de los Algoritmos R-A

Los algoritmos de ramificación y acotamiento, aplicados a problemas combinatorios, tienen dos características; la primera garantiza convergencia, mientras que la segunda admite la posibilidad de no realizar una enumeración completa de las posibles soluciones; a continuación se explican con mayor detalle estas características.

Convergencia. Dado que Ω es finito, el proceso de ramificación proporciona (a menos que se haya detenido antes) sólo nodos finales s ; esto es, el algoritmo proporciona una partición tal que $\mathcal{S}_i = \{s_i\}$, por lo tanto, todas las soluciones posibles son generadas y, la solución óptima (en caso de existir) ha sido encontrada; *el algoritmo converge en un número finito de pasos a la solución.*

Obsérvese que, de la definición de partición, debe pasar que todo \mathcal{S}_i es no vacío; así, la característica de ramificación nos garantiza que después de un número finito de pasos, se habrían examinado "todas" las posibles soluciones; entonces, ¿qué ventaja hay sobre la enumeración completa?. La siguiente característica de los algoritmos de ramificación y acotamiento nos da la ventaja.

Acotamiento. *La creación de cotas en cada nodo permite reconocer una solución óptima antes de hacer una enumeración completa.*

Para probar lo anterior, supóngase que se está minimizando $Z(\mathbf{X})$, y considérese un nodo final N^* para el cual $\mathcal{G}(N^*) = \{s^*\}$. Si $Z(s^*) \leq \mathfrak{B}(N)$ para cada nodo intermedio N , entonces s^* es óptimo para el problema y no se debe seguir ramificando; esto último dado que, por definición,

$$Z(s^*) \leq \mathfrak{B}(N) \leq \underset{s \in \mathcal{G}(N)}{\text{Min}} \{Z(s)\} \leq Z(s) \quad \forall s \in \mathcal{G}(N), \forall N;$$

o en palabras: para cualquier N y para cualquier $s \in \mathcal{G}(N)$, $Z(s)$ es a lo más tan pequeño como $\mathfrak{B}(N)$, pero $\mathfrak{B}(N)$ es más grande (por hipótesis) que $Z(s^*)$, por lo tanto $Z(s^*)$ es el mínimo de los $Z(s)$, es decir, s^* es el óptimo.

Capítulo 2

DOS ALGORITMOS TIPO R-A

Para ejemplificar el uso de los métodos de ramificación y acotamiento, en la solución de problemas combinatorios, se presentan en este capítulo dos algoritmos particulares: el primero es el de Kolesar para el problema de la mochila; y el segundo, el de Little-Murty para el problema del agente viajero. Además, con el fin de dar mayor claridad a los algoritmos, se desarrolla un ejemplo para cada uno de ellos.

2.1 RESOLVIENDO "LA MOCHILA"

Recordemos que en el problema de la mochila 0-1, lo que se debe resolver es (\mathcal{P}):

Maximizar

$$Z(X) = \sum_{i=1}^n p_i x_i$$

sujeto a

$$\sum_{i=1}^n w_i x_i \leq W$$

$$x_i \in \{0, 1\}, \text{ para } i = 1, \dots, n;$$

con:

- p_i = ganancia o valor por incluir el artículo i ;
- w_i = peso o costo en el que se incurre al incluir al artículo i ;
- W = capacidad de la mochila .

Una vez definido el problema, se da paso al algoritmo de Ramificación y Acotamiento que lo resuelve: el de Kolesar.

2.1.1 El Algoritmo

PASO 0. VERIFICACIÓN DE NO TRIVIALIDAD

- 0.1. Verificar si existe no-factibilidad trivial ; esto es , *comprobar que en la mochila quepa al menos un artículo* . Más formalmente , comprobar si: $\exists i \in \{1, \dots, n\}$ t.q. $w_i \leq W$. Si el problema es no factible, parar, en caso contrario ir a 0.2 .
- 0.2. Verificar si existe factibilidad trivial; esto es , *confirmar si en la mochila caben todos los artículos* . Más formalmente, observar si: $\sum w_i \leq W$. En caso afirmativo parar, en caso contrario ir al paso 1.

PASO 1. INICIO

- 1.1. Reordenar los artículos de manera decreciente con respecto al valor o ganancia por unidad de peso ($\frac{p_i}{w_i}$). De aquí en adelante se supondrá que los artículos están ordenados de esta forma . Ir a 1.2.
- 1.2. Crear el nodo $N = 1$, el cual representa a todas las posibles soluciones del problema. Ir al paso 2.

PASO 2. ACOTAMIENTO

- 2.1. Para cada nodo N , sin acotar, crear la correspondiente cota $\mathcal{B}(N)$ de la siguiente forma:
 - 2.1.1. Incluir en la mochila los artículos fijados (para ese subproblema) con $x_i = 1$, y borrarlos de la lista hecha en 1.1. Luego ir a 2.1.2.
 - 2.1.2. Borrar de la lista los artículos fijados con $x_i = 0$, luego pasar a 2.1.3.
 - 2.1.3. Introducir en la mochila los artículos restantes de la lista (siguiendo el orden ya fijado) hasta agotar la capaci-

dad de la mochila, o hasta acabar con la lista; en caso de ser necesario tomar una fracción del último artículo incluido; por ejemplo, si al agregar artículos llega un momento en que la capacidad es de 2, y el siguiente artículo, j , ocupa 3, tomar $x_j = \frac{2}{3}$. Ir a 2.1.4.

- 2.1.4. Calcular el valor de la mochila obtenida (i.e., $Z(\mathbf{X})$), la cota para ese nodo o subproblema es $\mathfrak{B}(N) = Z(\mathbf{X})$ (se puede probar que como problema de programación lineal se ha llegado a un óptimo).

PASO 3. BÚSQUEDA

- 3.1. Comparar las cotas de los nodos existentes. Sea N el de cota mayor o igual a todas, (si hay empate, rómpase arbitrariamente). Ir a 3.2.
- 3.2. Si \mathbf{X} , el óptimo del problema representado en N , es entero, parar, \mathbf{X} es óptimo para \mathcal{P} . En caso contrario, hacer $i^* = N$ e ir al paso 4.

PASO 4. RAMIFICACIÓN

- 4.1. Crear 2 ramas, con sus respectivos nodos, saliendo de i^* . Uno de los nuevos nodos representa al subproblema (del dado en i^*) que fija $x_j = 0$, mientras que el otro al subproblema que fija $x_j = 1$; y en donde x_j es la variable no entera en i^* .
- 4.2. Si el problema representado por un nuevo nodo es no factible, cortar la rama haciendo $\mathfrak{B}(N) = -\infty$; en caso contrario ir a 2.1.

2.1.2 Un Ejemplo Numérico

Ejemplo 1 Aplicar el algoritmo de Kolesar para resolver el problema tipo mochila 0-1 :

Maximizar

$$12x_1 + 12x_2 + 9x_3 + 15x_4 + 90x_5 + 26x_6 + 112x_7$$

sujeto a

$$3x_1 + 4x_2 + 3x_3 + 3x_4 + 15x_5 + 13x_6 + 16x_7 \leq 35$$

$$x_i \in \{0, 1\}, \quad \text{para } i = 1, \dots, 7.$$

SOLUCIÓN

Iteración 1

Paso 0. Dado que $w_1 = 3 < 35$, y $w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 = 57 > 35$, el problema es no trivial.

Paso 1. Reordenando los artículos, se tiene:

# de objeto	Peso (p_i)	Valor (v_i)	$\frac{v_i}{p_i}$
7	16	112	7
5	15	90	6
4	3	15	5
1	3	12	4
3	3	9	3
2	4	12	3
6	13	26	2

y el árbol inicia como en la figura 2.1 .



Figura 2.1. El inicial árbol de búsqueda .

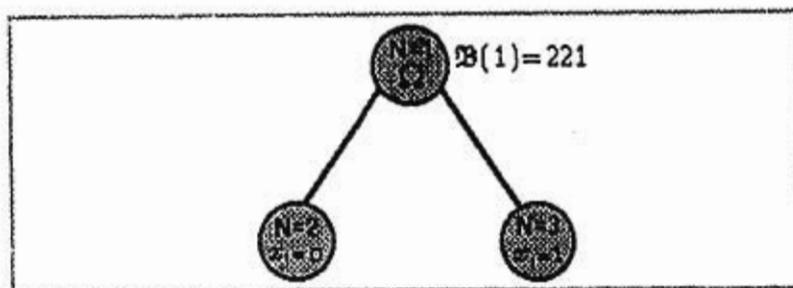


Figura 2.2 . Árbol de búsqueda al final de la primer iteración .

Paso 2. Se asigna $x_7 = 1$, $x_5 = 1$, $x_4 = 1$, $x_1 = \frac{1}{3}$; además, $\mathcal{B}(1) = 221$.

Paso 3. Se verifica que la solución en $N = 1$ no es entera, por lo que se hace $i^* = 1$ y se pasa a 4.

Paso 4. Al ramificar sobre x_1 , (que es la variable no entera en la solución de $N = 1$) el árbol resultante es el de la figura 2.2. Como ambos subproblemas son factibles, se regresa a 2.

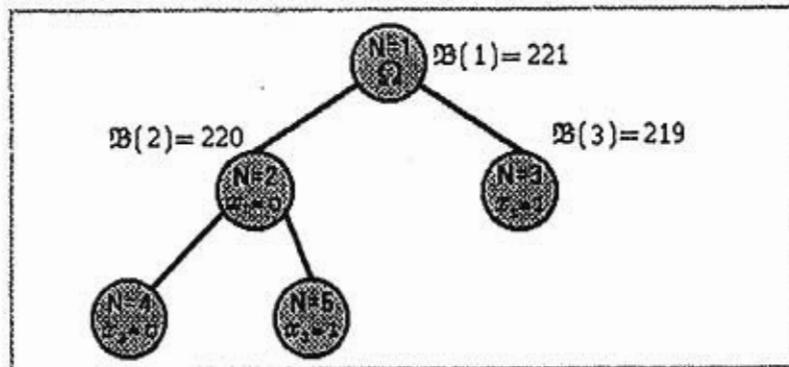


Figura 2.3. Árbol de búsqueda al final de la segunda iteración.

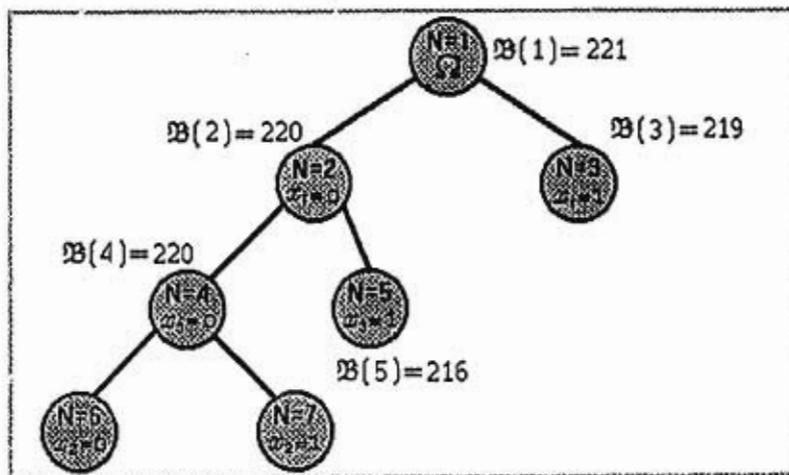


Figura 2.4. Árbol de búsqueda al final de la tercera iteración.

Segunda iteración

Paso 2. Para $N = 2$ se tiene $x_1 = 0, x_7 = 1, x_5 = 1, x_4 = 1, x_3 = \frac{1}{3}$; con $\mathcal{B}(2) = 220$. Para $N = 3$ se tiene $x_1 = 1, x_7 = 1, x_5 = 1, x_4 = \frac{1}{3}$; con $\mathcal{B}(3) = 219$.

Paso 3. Se verifica que las soluciones en $N = 2$ y $N = 3$ no son soluciones enteras. Se decide ramificar por $i^* = 2$ (pues $B(2) > B(3)$).

Paso 4. Se obtiene el árbol de la figura 2.3. Una vez más, los nuevos subproblemas son factibles, por lo que se vuelve a 2.

Iteración 3

Paso 2. Para $N = 4$ se obtiene $x_1 = 0, x_3 = 0, x_7 = 1, x_5 = 1, x_4 = 1, x_2 = \frac{1}{4}$; con $\mathcal{B}(4) = 220$. Para $N = 5$ se obtiene $x_1 = 0, x_3 = 1, x_7 = 1, x_5 = 1, x_4 = \frac{1}{3}$; con $\mathcal{B}(5) = 216$.

Paso 3. En $N = 4$ y $N = 5$ no se tienen soluciones enteras. Se ramifica por $i^* = 4$ (i.e., la variable a ramificar es x_2).

Paso 4. Se obtiene el árbol que se muestra en la figura 2.4, en el que los nuevos subproblemas son factibles; así, se regresa a 2.

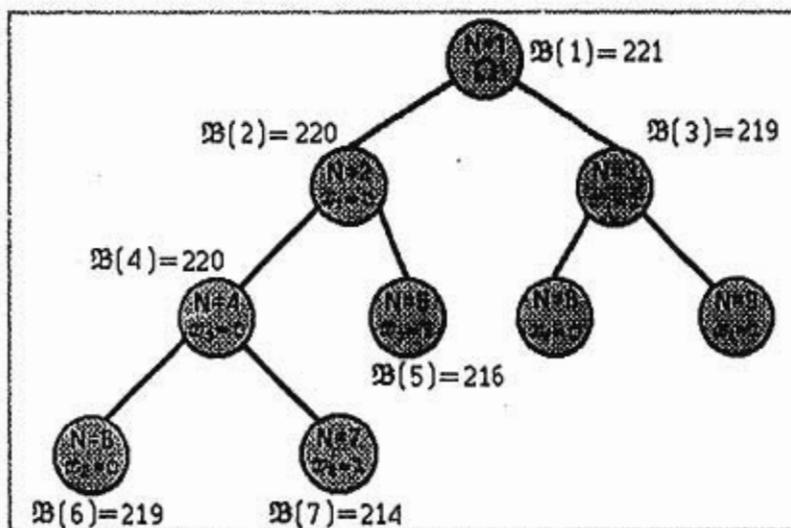


Figura 2.5. Árbol de búsqueda al final de la iteración 4.

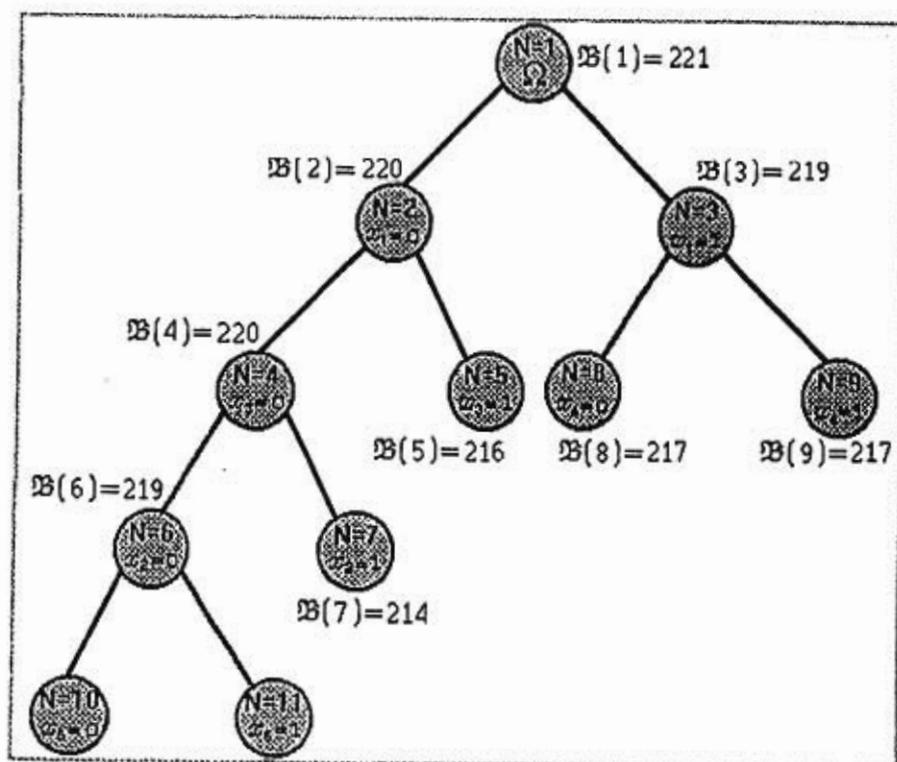


Figura 2.6. Árbol de búsqueda al final de la iteración 5.

Iteración 4

Paso 2. Para $N = 6$: $x_1 = 0$, $x_3 = 0$, $x_2 = 0$, $x_7 = 1$, $x_5 = 1$, $x_4 = 1$, $x_6 = \frac{1}{13}$ y $\mathcal{B}(6) = 219$. Para $N = 7$: $x_1 = 0$, $x_3 = 0$, $x_2 = 1$, $x_7 = 1$, $x_5 = 1$, y $\mathcal{B}(7) = 214$.

Paso 3. En $N = 6$ no se tiene una solución entera, mientras que en $N = 7$ sí. La mejor cota se encuentra en $N = 3$ y $N = 6$. Se escoge arbitrariamente $i^* = 3$.

Paso 4. Se llega a el árbol mostrado en la figura 2.5. Los nuevos nodos representan problemas factibles, por lo que se regresa a 2.

Iteración 5

Paso 2. En $N = 8$: $x_1 = 1$, $x_4 = 0$, $x_7 = 1$, $x_5 = 1$, $x_3 = \frac{1}{3}$ y $\mathcal{B}(8) = 217$. En $N = 9$: $x_1 = 1$, $x_4 = 1$, $x_7 = 1$, $x_5 = \frac{13}{15}$ y $\mathcal{B}(9) = 217$.

Paso 3. Como los nuevos nodos no tienen soluciones óptimas enteras, se comparan las cotas y se toma $i^* = 6$.

Paso 4. El árbol resultante es el de la figura 2.6. Los nuevos subproblemas son factibles y se regresa a 2.

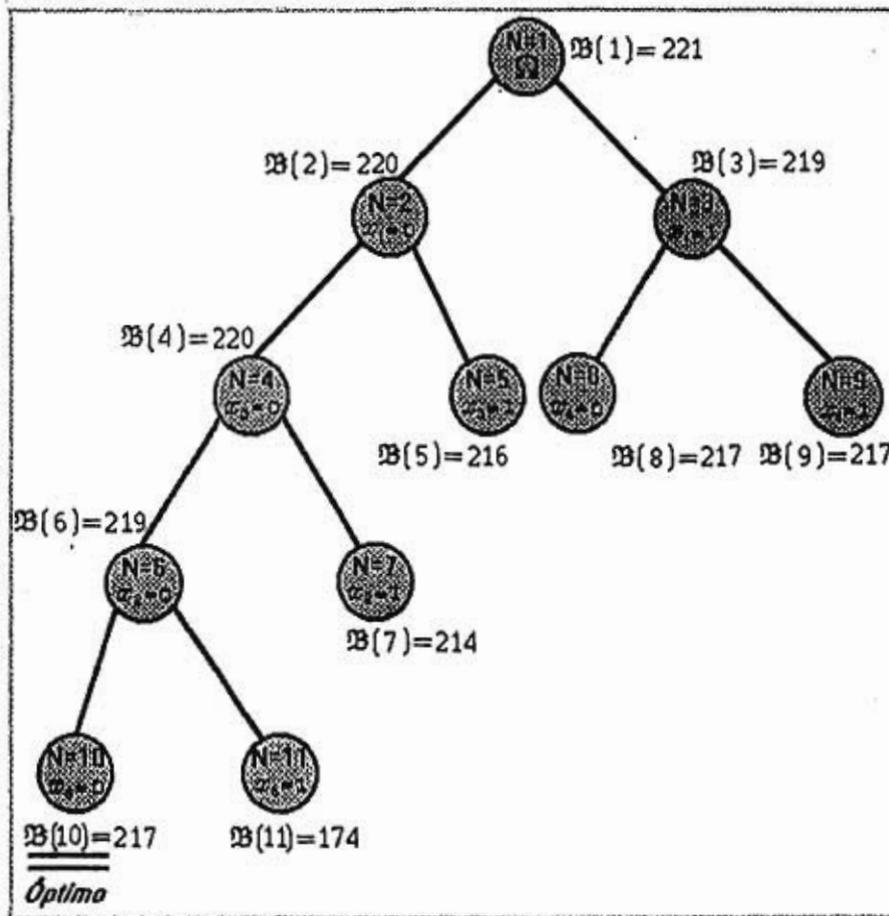


Figura 2.7. Árbol final de búsqueda.

Iteración 6

Paso 2. En $N = 10$: $x_1 = 0, x_3 = 0, x_2 = 0, x_6 = 0, x_7 = 1, x_5 = 1, x_4 = 1$
y $B(10) = 217$. En $N = 11$: $x_1 = 0, x_3 = 0, x_2 = 0, x_6 = 1, x_7 = 1, x_5 = \frac{2}{5}$
y $B(11) = 174$.

Paso 3. Al comparar las cotas, se tiene un empate en los nodos 8, 9, y 10; pero en $N = 10$ se tiene una solución entera, por lo que el algoritmo termina. La solución óptima es : $x_1 = 0, x_3 = 0, x_2 = 0, x_6 = 0, x_7 = 1, x_5 = 1, x_4 = 1$; con $Z(X) = 217$.

Para terminar con este ejemplo, se muestra el árbol de búsqueda final (figura 2.7); no sin antes mencionar que, gracias al algoritmo, no se tuvieron que evaluar todas las posibles soluciones del problema, que son $2^7 = 128$.

2.2 RESOLVIENDO "EL AGENTE VIAJERO"

El problema del agente viajero puede ser enunciado, más formalmente, de la siguiente manera:

Minimizar

$$\sum_{i=1}^n \sum_{j=1}^n c_{i,j} x_{i,j}$$

sujeto a

$$\sum_{i=1}^n x_{i,j} = 1 \quad \text{para } j = 1, \dots, n; \dots (1)$$

$$\sum_{j=1}^n x_{i,j} = 1 \quad \text{para } i = 1, \dots, n; \dots (2)$$

$$x_{i,j} \in \{0, 1\} \quad \forall i, \forall j; \dots (3)$$

$$X \text{ ruta}; \dots (4)$$

donde

$c_{i,j}$ = Costo por ir de la ciudad i a la j .

$x_{i,j}$ = Variable que indica si se hace el viaje de i a j .

(1), (2) y (3) restricciones que aseguran la visita a cada ciudad una única vez.

(4) : Restricción que obliga a seguir un circuito de longitud n .

El algoritmo presentado en esta sección se basa en en la construcción de una ruta (también conocida como circuito hamiltoniano) mediante el uso de la matriz reducida, la cual se construye de manera general en la siguiente subsección.

2.2.1 Obtención de la Matriz Reducida y la Cota Asociada a un Problema

Sea K la matriz de costos asociada al problema tratado en el nodo N . La matriz reducida del problema, así como la cota asociada al nodo, se calculan de la forma siguiente:

a. A cada renglón i de K , se le resta el mínimo de sus elementos (al cual se le denota por h_i); esto es, para cada i , renglón de K , hacer:

$$1. h_i \equiv \underset{j}{\text{Min}} K_{i,j}.$$

$$2. K_{i,j} \equiv K_{i,j} - h_i.$$

b. A cada columna j de K , se le resta el mínimo de sus elementos (denotado por h^j); más formalmente, para cada j , columna de K , hacer:

$$1. h^j \equiv \underset{i}{\text{Min}} K_{i,j}.$$

$$2. K_{i,j} \equiv K_{i,j} - h^j.$$

c. La matriz obtenida de a y b es conocida como la matriz reducida y, la cota para el problema es definida por: $\mathfrak{B}(N) = \sum h_i + \sum h^j + \mathfrak{B}(N^a)$.

Ejemplo 2 Si se tiene la matriz de costos para el subproblema dado en N

	A	B	C
A	∞	4	2
B	4	∞	6
C	2	6	∞

y $\mathfrak{B}(N^a) = 6$. Entonces $h_A = 2$, $h_B = 4$ y $h_C = 2$; por lo que después del paso a, la matriz sería

	A	B	C
A	∞	2	0
B	0	∞	2
C	0	4	∞

Al realizar el paso b se tiene $h^A = 0$, $h^B = 2$, $h^C = 0$; obteniendo la matriz reducida

	A	B	C
A	∞	0	0
B	0	∞	2
C	0	2	∞

Además, la cota para el problema es

$$\mathfrak{B}(N) = \sum h_i + \sum h^j + \mathfrak{B}(N^a) = 8 + 2 + 6 = 16 .$$

Obsérvese que, ciertamente, la cota obtenida es cota para el problema; esto es fácil de observar dado que, si en el ejemplo anterior, se tiene que salir forzosamente de la ciudad A, se incurre a lo menos en un costo de 2; así, se puede restar 2 a todo el renglón A y trabajar con un problema equivalente. Lo mismo ocurre con las columnas, por ejemplo, como se quiere visitar B, se debe incurrir en un costo mínimo de 2.

Una vez que se entiende el concepto de la matriz reducida, el algoritmo de Little-Murty será de fácil comprensión.

2.2.2 El Algoritmo

PASO 1. INICIO

- 1.1. Crear el nodo $N = 1$, que representa todas las posibles soluciones del problema.
- 1.2. Hacer $\mathfrak{B}(1^a) = 0$, e ir al paso 2.

PASO 2. ACOTAMIENTO

- 2.1. Para cada nodo N , sin acotar, calcular $\mathfrak{B}(N)$ y crear la matriz reducida del problema; luego ir al paso 3.

PASO 3. BÚSQUEDA

- 3.1. Comparar las cotas de los nodos existentes. Sea N el de cota menor o igual (en caso de empate, rómpase arbitrariamente).
- 3.2. Si $\mathfrak{G}(N)$ consta de una única solución, parar, se ha encon-

trado el óptimo; en otro caso, hacer $i^* = N$, e ir al paso 4.

PASO 4. RAMIFICACIÓN

4.1. Para cada $c_{i,j} = 0$ de i^* , calcular:

$$a. \alpha_i = \min_{l \neq j} C_{i,l}$$

$$b. \beta_j = \min_{k \neq i} C_{k,j}$$

$$c. \theta_{i,j} = \alpha_i + \beta_j$$

($\theta_{i,j}$ representa el costo por no ir de i a j , consecuencia de la manera en que se construye la matriz reducida.)

4.2. Sea $\theta_{A,B}$ el máximo de los $\theta_{i,j}$ calculados (en caso de empate, rómpase arbitrariamente).

4.3. En el árbol de búsqueda crear dos ramas saliendo de i^* , cada una con su respectivo nodo. El primer nodo representa el subproblema (del representado en i^*) que fija $x_{A,B} = 0$, y, para el cual, se construye una nueva matriz de costos, C , cambiando (a la matriz reducida de i^*) $c_{A,B}$ por ∞ ; lo cual evita ir de A a B . El segundo nodo representa el subproblema que fija $x_{A,B} = 1$, y para el cual, se calcula la nueva matriz de costos cancelando (a la matriz reducida en i^*) la columna A y el renglón B ; además de hacer $c_{i,j} = \infty$ en aquellos (i,j) necesarios para evitar recorridos de tamaño menor que n . Por ejemplo, si $x_{A,B} = 1$ y $x_{B,C} = 1$ y $n > 3$, hacer $x_{C,A} = 0$. Luego regresar al paso 2.

2.2.3 El Ejemplo Numérico

Para terminar con la presente sección y con el capítulo 1, se presenta un ejemplo en el que se utiliza el algoritmo de Little-Murty.

Ejemplo 3 Aplicar el algoritmo de Little-Murty para encontrar el recorrido del agente viajero si, se desea visitar 5 ciudades, y la matriz de costos es:

	A	B	C	D	E
A	∞	11	15	17	19
B	5	∞	16	9	8
C	3	7	∞	12	10
D	9	4	18	∞	6
E	10	8	11	6	∞

SOLUCIÓN

Iteración 1

Paso 1. Se crea el nodo $N = 1$ y el árbol de búsqueda es el de la figura 2.8.



Figura 2.8. El árbol inicial de búsqueda.

Paso 2. La matriz reducida para $N = 1$ es

	A	B	C	D	E
A	∞	0	0	6	6
B	0	∞	7	4	1
C	0	4	∞	9	5
D	5	0	10	∞	0
E	4	2	1	0	∞

con $\mathfrak{B}(1) = \sum h_i + \sum h^j + \mathfrak{B}(1^a) = 29 + 6 + 0 = 35$.

Paso 3. Dado que el nodo $N = 1$ representa a un problema con más de una solución, se hace $i^* = 1$.

Paso 4. Se calcula $\theta_{i,j}$ para $N = 1$, obteniéndose:

$$\begin{aligned} \theta_{A,B} &= \alpha_A + \beta_B = 0 + 0 = 0, & \theta_{A,C} &= \alpha_A + \beta_C = 0 + 1 = 1, \\ \theta_{B,A} &= \alpha_B + \beta_A = 1 + 0 = 1, & \theta_{C,A} &= \alpha_C + \beta_A = 4 + 0 = 4, \\ \theta_{D,B} &= \alpha_D + \beta_B = 0 + 0 = 0, & \theta_{D,E} &= \alpha_D + \beta_E = 0 + 1 = 1 \text{ y} \\ \theta_{E,D} &= \alpha_E + \beta_D = 1 + 4 = 5. \end{aligned}$$

Así, dado que el máximo es $\theta_{E,D} = 5$, se toma como variable a ramificar a $x_{E,D}$. Se obtiene el árbol de la figura 2.9, con matrices de costo

	A	B	C	D	E
A	∞	0	0	6	6
B	0	∞	7	4	1
C	0	4	∞	9	5
D	5	0	10	∞	0
E	4	2	1	∞	∞

	A	B	C	E
A	∞	0	0	6
B	0	∞	7	1
C	0	4	∞	5
D	5	0	10	∞

para $N = 2$ y $N = 3$ respectivamente. Nótese que en $N = 3$, se le asigna costo infinito al viaje de D a E , esto para evitar formar el circuito de longitud 2: $E \rightarrow D \rightarrow E$

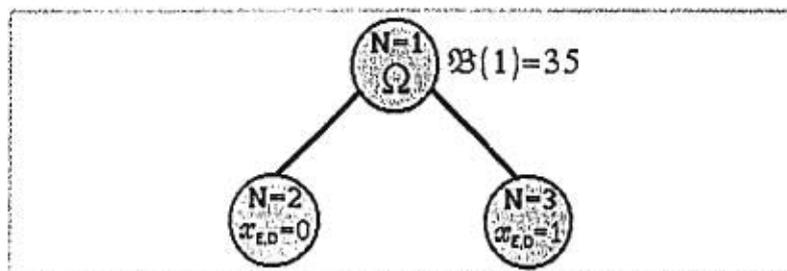


Figura 2.9. Árbol de búsqueda al final de la primera iteración.

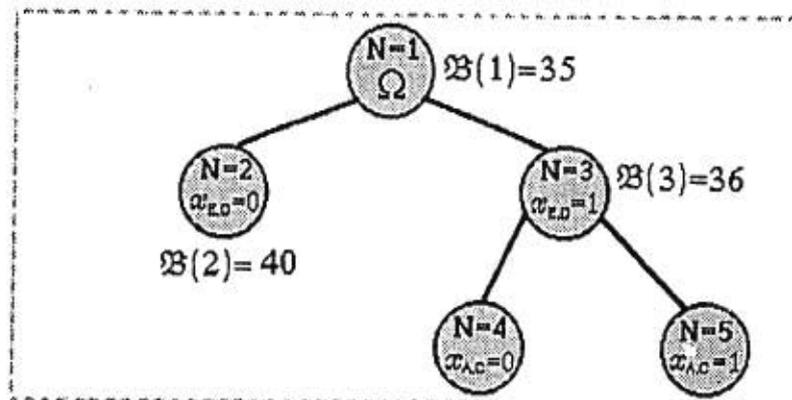


Figura 2.10. Árbol de búsqueda al final de la segunda iteración.

Iteración 2.

Paso 2. Las matrices reducidas para $N = 2$ y $N = 3$ quedan de la siguiente forma:

	A	B	C	D	E
A	∞	0	0	2	6
B	0	∞	7	0	1
C	0	4	∞	5	5
D	5	0	10	∞	0
E	3	1	0	∞	∞

	A	B	C	E
A	∞	0	0	5
B	0	∞	7	0
C	0	4	∞	4
D	5	0	10	∞

con: $\mathfrak{B}(2) = 5 + \mathfrak{B}(2^a) = 5 + \mathfrak{B}(1) = 40$ y $\mathfrak{B}(3) = 1 + \mathfrak{B}(3^a) = 1 + \mathfrak{B}(1) = 36$.

Paso 3. Como $\mathfrak{B}(3) < \mathfrak{B}(2)$, y en $N = 3$ no se representa una única solución, se toma $i^* = 3$.

Paso 4. Al realizar el cálculo de los $\theta_{i,j}$, en la matriz asociada a $N = 3$, se tiene que el máximo es $\theta_{A,C} = 7$; por lo cual, se ramifica en $x_{A,C}$. Se obtiene el árbol de la figura 2.10, con matrices de costo, para $N = 4$ y $N = 5$:

	A	B	C	E
A	∞	0	∞	5
B	0	∞	7	0
C	0	4	∞	4
D	5	0	10	∞

	A	B	E
B	0	∞	0
C	∞	4	4
D	5	0	∞

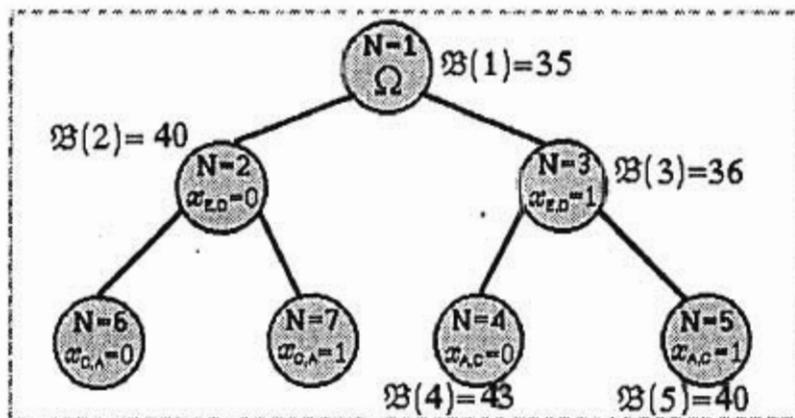


Figura 2.11. Árbol de búsqueda al final de la iteración 3.

Iteración 3

Paso 2. Las matrices reducidas para $N = 4$ y $N = 5$ son

	A	B	C	E
A	∞	0	∞	5
B	0	∞	0	0
C	0	4	∞	4
D	5	0	3	∞

	A	B	E
B	0	∞	0
C	∞	0	0
D	5	0	∞

y donde: $\mathfrak{B}(4) = 7 + \mathfrak{B}(3) = 43$ y $\mathfrak{B}(5) = 4 + \mathfrak{B}(3) = 40$.

Paso 3. Dado que existe un empate en las cotas de $N=2$ y de $N=5$, se escoge de manera arbitraria $i^* = 2$.

Paso 4. El máximo de los $\theta_{i,j}$, en $N = 2$, es $\theta_{C,A} = 4$; así, se ramifica en $x_{C,A}$. Se obtiene el árbol de la figura 2.11, con matrices de costo para $N = 6$ y $N = 7$:

	A	B	C	D	E
A	∞	0	0	2	6
B	0	∞	7	0	1
C	∞	4	∞	5	5
D	5	0	10	∞	0
E	3	1	0	∞	∞

	B	C	D	E
A	0	∞	2	6
B	∞	7	0	1
D	0	10	∞	0
E	1	0	∞	∞

Iteración 4

Paso 2. Las matrices reducidas para $N=6$ y $N=7$ quedan como

	A	B	C	D	E
A	∞	0	0	2	6
B	0	∞	7	0	1
C	∞	0	∞	1	1
D	5	0	10	∞	0
E	3	1	0	∞	∞

	B	C	D	E
A	0	∞	2	6
B	∞	7	0	1
D	0	10	∞	0
E	1	0	∞	∞

y donde: $\mathfrak{B}(6) = 4 + \mathfrak{B}(2) = 44$, $\mathfrak{B}(7) = 0 + \mathfrak{B}(2) = 40$.

Paso 3. De nueva cuenta existe un empate, ahora en $N = 5$ y $N = 7$. Ninguno de los dos nodos representa una única solución, así, se toma de forma arbitraria $i^* = 5$.

Paso 4. El máximo de los $\theta_{i,j}$, en $N = 5$, es $\theta_{B,A} = 5$ y $\theta_{D,B} = 5$; se toma $x_{B,A}$ para ramificar. El árbol de búsqueda es el de la figura 2.12, con matrices de costo para $N = 8$ y $N = 9$:

	A	B	E
B	∞	∞	0
C	∞	0	0
D	5	0	∞

	B	E
C	∞	0
D	0	∞

Nótese que se dió costo infinito a ir de C a B, esto para evitar el circuito de longitud 3 : $A \rightarrow C \rightarrow B \rightarrow A$.

Iteración 5

Paso 2. Las matrices reducidas para $N = 8$ y $N = 9$ son

	A	B	E
B	∞	∞	0
C	∞	0	0
D	0	0	∞

	B	E
C	∞	0
D	0	∞

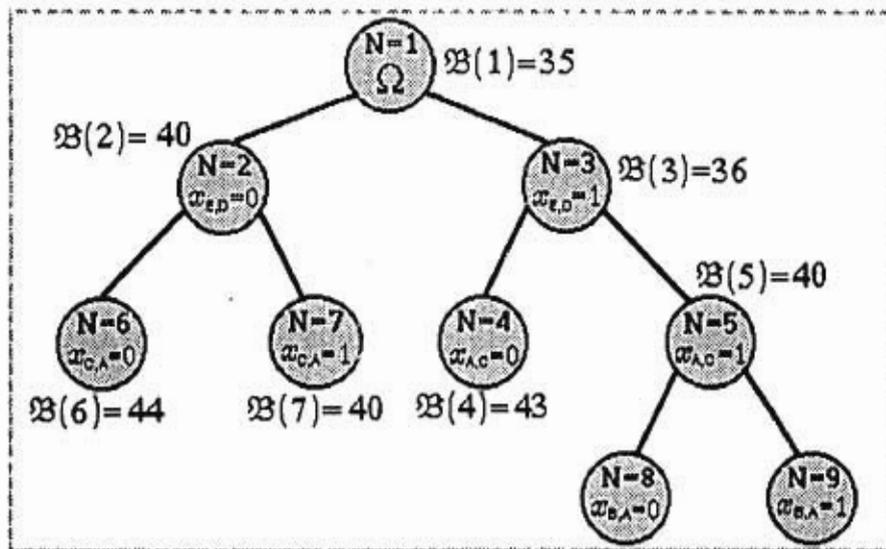


Figura 2.12. Árbol de búsqueda al final de la iteración 4 .

donde: $B(8) = 5 + B(5) = 45$, $B(9) = 0 + B(5) = 40$.

Paso 3. Nuevamente existe un empate en las cotas, en este caso para $N = 7$ y $N = 9$; sin embargo, en $N = 9$ se tiene una única solución factible (ir de C a E y de D a B); por lo tanto, se ha encontrado el óptimo, en el cual:

$$x_{C,E} = 1, x_{D,B} = 1, x_{B,A} = 1, x_{A,C} = 1, x_{E,D} = 1.$$

Un recorrido óptimo es

$$A \rightarrow C \rightarrow E \rightarrow D \rightarrow B \rightarrow A$$

con costo 40.

El árbol de búsqueda final, proporcionado por el algoritmo, es el de la figura 2.13. Finalmente, nótese que no se evaluaron las 4! posibles soluciones del problema.

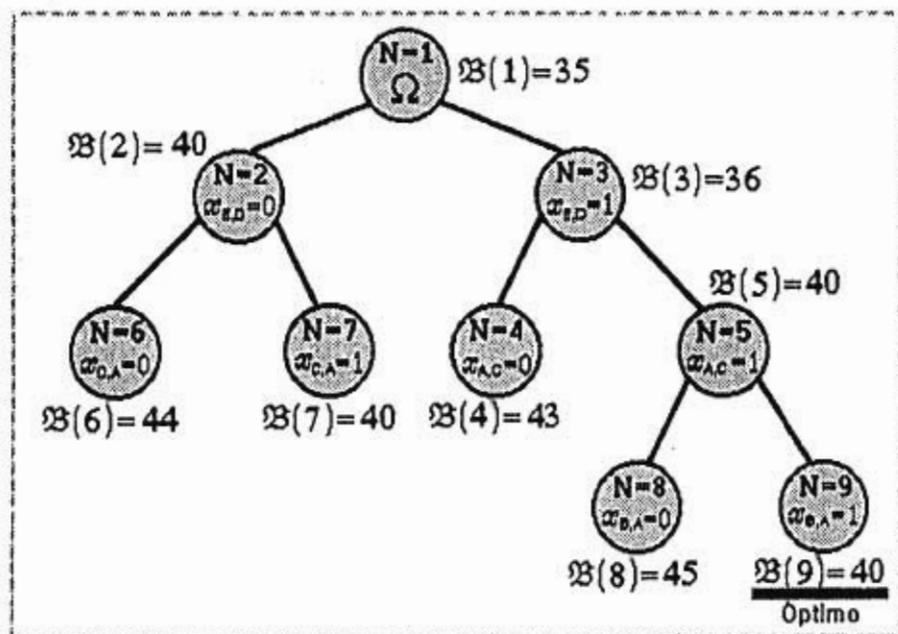


Figura 2.13. Árbol final de búsqueda.

Capítulo 3

CONJUNTOS Y FUNCIONES CONVEXAS

Intuitivamente, un conjunto es convexo cuando para cualquier par de puntos en él, se puede ir de uno a otro en línea recta "sin salirse del conjunto". Aunque el concepto de convexidad se puede definir en cualquier espacio vectorial, para los fines del presente trabajo se trabajará, casi siempre, en un espacio vectorial particular: \mathbb{R}^2 .

El objetivo de este capítulo es proporcionar la herramienta necesaria para la comprensión del algoritmo principal de la tesis.

3.1 CONCEPTOS Y RESULTADOS

Para dejar claros los siguientes conceptos, considerense los ejemplos presentados en la figura 3.1.

Definición 4 *Dados x y y en \mathbb{R}^2 , se define el segmento que une a x con y como el conjunto $[x,y] = \{\alpha x + (1 - \alpha)y \mid 0 \leq \alpha \leq 1\}$.*

Definición 5 *Un conjunto S en \mathbb{R}^2 se dice **convexo** si: dados cualesquiera dos puntos en S , el segmento que los une está totalmente contenido en S ; esto es: $\forall x, y \in S, [x,y] \subseteq S$.*

Una vez entendido el concepto de conjunto convexo, se puede definir lo que es una función convexa.

Definición 6 Una función $f : D_f \rightarrow \mathbb{R}$, con $D_f \subseteq \mathbb{R}$, se dice **convexa** (**cóncava**) si $\forall x, y \in D_f$ & $\forall \lambda \in [0, 1]$ se cumple:

$$f(\lambda x + (1 - \lambda)y) \underset{(\geq)}{\leq} \lambda f(x) + (1 - \lambda)f(y).$$

Geométicamente, una función es convexa (cóncava) si los segmentos que unen a $(x, f(x))$ con $(y, f(y))$, siempre están por encima (debajo) de la gráfica de f (ver figura 3.2).

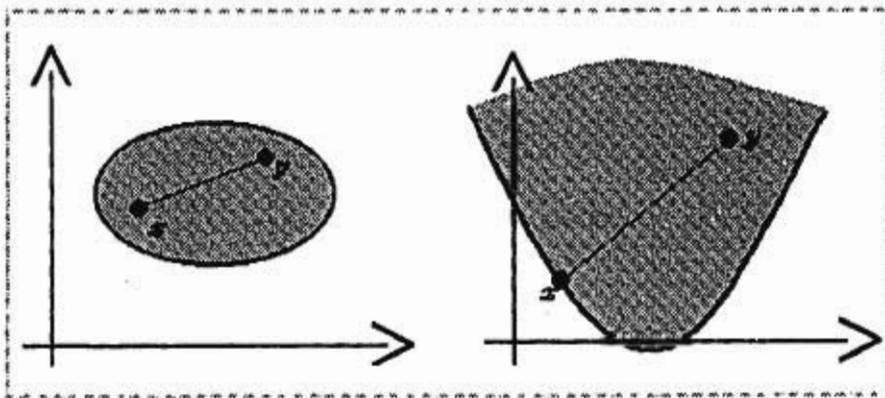


Figura 3.1.A. Conjuntos convexos .

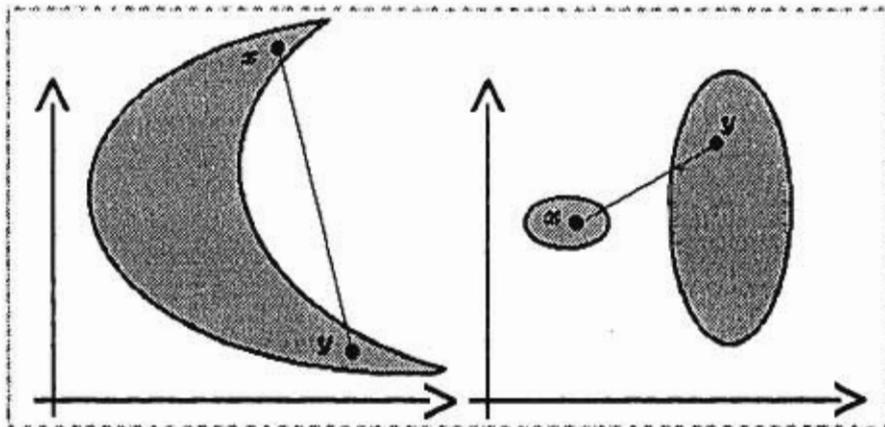


Figura 3.1.B. Conjuntos no convexos .

La relación entre los conjuntos convexos y las funciones convexas es la siguiente: una función $f : D_f \rightarrow \mathbb{R}$ ($D_f \subseteq \mathbb{R}$) es convexa si el conjunto de puntos que están por encima de su gráfica es convexo. A este conjunto se le conoce como *EPÍGRAFO* y, dado que no es relevante para el desarrollo de este trabajo, no se incluye su demostración; sin embargo, para mayores detalles puede consultarse [18] y [22].

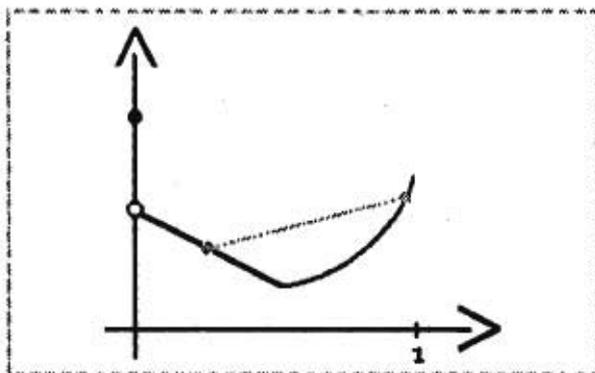


Figura 3.2.A. Función convexa $f: [0, 1] \rightarrow \mathbb{R}$.

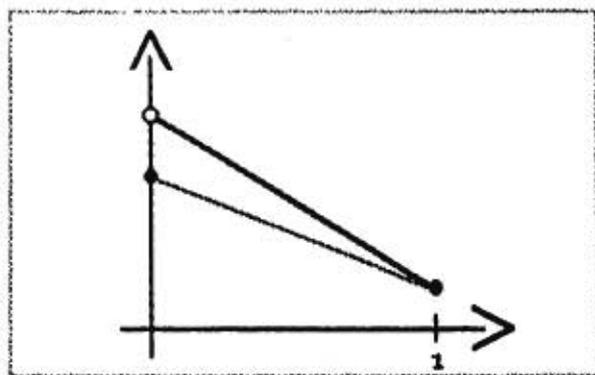


Figura 3.2.B. Función cóncava $f: [0, 1] \rightarrow \mathbb{R}$.

Nótese que una función convexa no tiene que ser continua en su dominio (véase nuevamente la figura 3.2), menos aún diferenciable. Dado lo anterior, se presenta ahora un resultado de gran utilidad (si no se manejan del todo bien los conceptos involucrados en el enunciado o en su demostración, se puede referir al apéndice [44], [46] y [47]).

Proposición 7 Si $f : [a, b] \rightarrow \mathbb{R}$ es convexa, entonces, f es continua en (a, b) .

DEMOSTRACIÓN

Sean $a < x_1 < x_2 < x_3 < x_4 < b$. Defínase α y β como

$$\alpha = \frac{x_3 - x_2}{x_3 - x_1}, \quad \beta = \frac{x_2 - x_1}{x_3 - x_1}.$$

Nótese que $\alpha > 0$, $\beta > 0$ y $\alpha + \beta = 1$ (por lo tanto $\beta = 1 - \alpha$); además,

$$\alpha x_1 + \beta x_3 = \frac{x_3 - x_2}{x_3 - x_1} x_1 + \frac{x_2 - x_1}{x_3 - x_1} x_3 = \frac{x_3 x_1 - x_2 x_1 + x_2 x_3 - x_1 x_3}{x_3 - x_1} = \frac{x_2(x_3 - x_1)}{x_3 - x_1} = x_2.$$

Entonces

$$\frac{f(x_3) - f(x_2)}{x_3 - x_2} - \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{(x_2 - x_1)(f(x_3) - f(x_2)) - (x_3 - x_2)(f(x_2) - f(x_1))}{(x_3 - x_2)(x_2 - x_1)} =$$

$$\frac{(x_2 - x_1)(f(x_3) - f(x_2)) - (x_3 - x_2)(f(x_2) - f(x_1))}{\left(\frac{x_3 - x_2}{x_3 - x_1}\right)\left(\frac{x_2 - x_1}{x_3 - x_1}\right)(x_3 - x_1)^2} =$$

$$\frac{\left(\frac{x_2 - x_1}{x_3 - x_1}\right)(f(x_3) - f(x_2)) - \left(\frac{x_3 - x_2}{x_3 - x_1}\right)(f(x_2) - f(x_1))}{\alpha\beta(x_3 - x_1)} =$$

$$\frac{\beta(f(x_3) - f(x_2)) - \alpha(f(x_2) - f(x_1))}{\alpha\beta(x_3 - x_1)} = \frac{\alpha f(x_1) + \beta f(x_3) - (\alpha + \beta)f(x_2)}{\alpha\beta(x_3 - x_1)} =$$

$$\frac{\alpha f(x_1) + (1 - \alpha)f(x_3) - f(\alpha x_1 + \beta x_3)}{\alpha\beta(x_3 - x_1)} \geq 0;$$

(donde la desigualdad se sigue de que f es convexa). Por lo tanto,

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1} \leq \frac{f(x_3) - f(x_2)}{x_3 - x_2};$$

y, de manera semejante,

$$\frac{f(x_3) - f(x_2)}{x_3 - x_2} \leq \frac{f(x_4) - f(x_3)}{x_4 - x_3}.$$

Lo anterior asegura que, los cocientes (pendientes) de la función convexa, formados por intervalos ajenos, son crecientes. Se deduce entonces que, dados $a < x_2 < x_3 < b$, ocurre

$$\frac{f(x_2)-f(x_1)}{x_2-x_1} \leq \frac{f(x_3)-f(x_2)}{x_3-x_2} \leq \frac{f(x_4)-f(x_3)}{x_4-x_3} .$$

así, tomando

$$M = \text{Max} \left\{ \left| \frac{f(x_2)-f(x_1)}{x_2-x_1} \right| , \left| \frac{f(x_4)-f(x_3)}{x_4-x_3} \right| \right\}$$

(donde x_1, x_4 son fijos, y sólo se pide que $a < x_1 < x_2 < x_3 < x_4 < b$); se concluye que: dado $\varepsilon > 0$ arbitrario,

$$|x_3 - x_2| < \frac{\varepsilon}{M} \implies |f(x_3) - f(x_2)| \leq M|x_3 - x_2| < \varepsilon .$$

Esta última implicación asegura que f es uniformemente continua en (a, b) y, por lo tanto, f es continua en (a, b) .

f

El resultado anterior se puede generalizar para funciones de \mathbb{R}^n en \mathbb{R} , sin embargo, para los fines perseguidos, es suficiente tener en mente que se cumple para funciones de \mathbb{R} en \mathbb{R} . En seguida se enuncia otro resultado importante de convexidad de funciones.

Proposición 8 Sean $D_1, D_2 \subseteq \mathbb{R}$; y sean $f_1 : D_1 \rightarrow \mathbb{R}$ y $f_2 : D_2 \rightarrow \mathbb{R}$. Si f_1 y f_2 son convexas, entonces la función $f : D_1 \times D_2 \rightarrow \mathbb{R}$, definida por $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$ es convexa.

DEMOSTRACIÓN

Sean $x = (x_1, x_2)$, $y = (y_1, y_2) \in D_1 \times D_2$, y sea $\lambda \in [0, 1]$. Entonces, por definición de f , y dado que tanto f_1 como f_2 son convexas, se sigue que

$$f(\lambda x + (1 - \lambda)y) = f(\lambda(x_1, x_2) + (1 - \lambda)(y_1, y_2)) =$$

$$f(\lambda x_1 + (1 - \lambda)y_1, \lambda x_2 + (1 - \lambda)y_2) =$$

$$f_1(\lambda x_1 + (1 - \lambda)y_1) + f_2(\lambda x_2 + (1 - \lambda)y_2) \leq$$

$$\lambda f_1(x_1) + (1 - \lambda)f_1(y_1) + \lambda f_2(x_2) + (1 - \lambda)f_2(y_2) =$$

$$\lambda(f_1(x_1) + f_2(x_2)) + (1 - \lambda)(f_1(y_1) + f_2(y_2)) =$$

$$\lambda f(x_1, x_2) + (1 - \lambda)f(y_1, y_2) = \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y}).$$

Con lo cual queda demostrado que f es convexa. \mathbf{f}

Una generalización inmediata al resultado anterior es la siguiente.

Corolario 9 Dadas n funciones convexas $f_i : D_i \rightarrow \mathbb{R}$ (con $D_i \subseteq \mathbb{R}$), la función $f : \prod D_i \rightarrow \mathbb{R}$ definida por $f(x_1, \dots, x_n) = \sum f_i(x_i)$ es convexa. \mathbf{f}

3.2 CUBIERTAS CONVEXAS

De manera poco formal, la cubierta convexa de una función f , es la más grande función convexa por debajo de f ; y precisamente, del hecho que sea una función por debajo de f , se desprende rápidamente una forma de acotar el mínimo de f : con el mínimo de la cubierta convexa. Este último hecho será utilizado más adelante para obtener las cotas en el algoritmo presentado en el capítulo 4.

Antes de definir formalmente cubierta convexa, se necesita el siguiente resultado.

Proposición 10 Sea $D \subseteq \mathbb{R}^n$ y $\mathfrak{F} = \{g_i : D \rightarrow \mathbb{R} \mid i \in I \text{ \& } g_i \text{ es convexa}\}$. Entonces la función $f : D \rightarrow \mathbb{R}$ definida por $f(\mathbf{x}) = \sup_{i \in I} g_i(\mathbf{x})$, es convexa.

DEMOSTRACIÓN

Sean $\mathbf{x}, \mathbf{y} \in D$; y $\lambda \in (0, 1)$. Tómese $\varepsilon > 0$. Por definición de f , existe $g_i \in \mathfrak{F}$ tal que

$$f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) - \varepsilon \leq g_i(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y});$$

y, dado que g_i es convexa

$$g_i(\lambda x + (1 - \lambda)y) \leq \lambda g_i(x) + (1 - \lambda)g_i(y).$$

Nuevamente por definición de f y g_i , también ocurre que

$$\lambda g_i(x) + (1 - \lambda)g_i(y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Por lo tanto, para $\varepsilon > 0$ arbitraria, se tiene

$$f(\lambda x + (1 - \lambda)y) - \varepsilon \leq \lambda f(x) + (1 - \lambda)f(y).$$

Así, se concluye que

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y),$$

y por lo tanto, f es convexa. **f**

Con el resultado anterior, es posible definir ahora la cubierta convexa de una función.

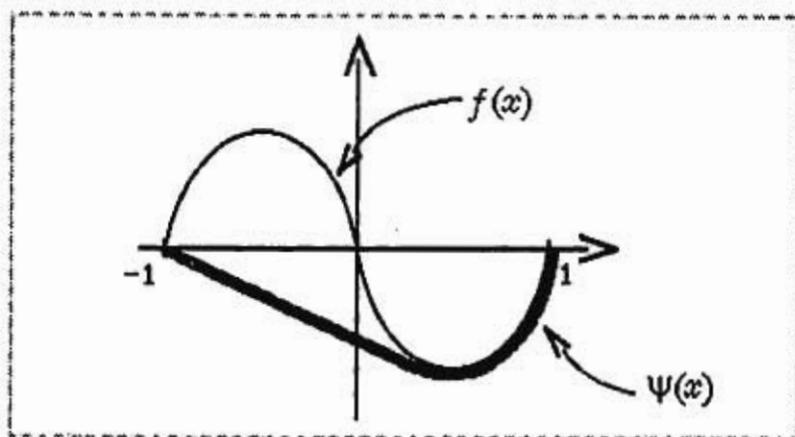


Figura 3.3.A. Cubierta convexa de $f(x) = x^3 - x$ en $[-1, 1]$.

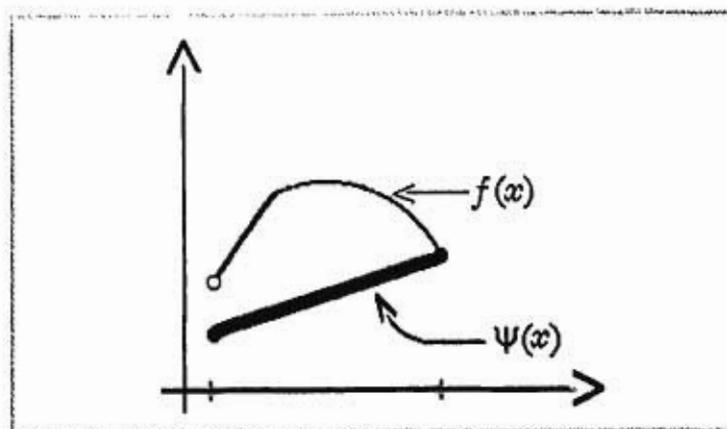


Figura 3.3.B. Cubierta convexa de una función cóncava.

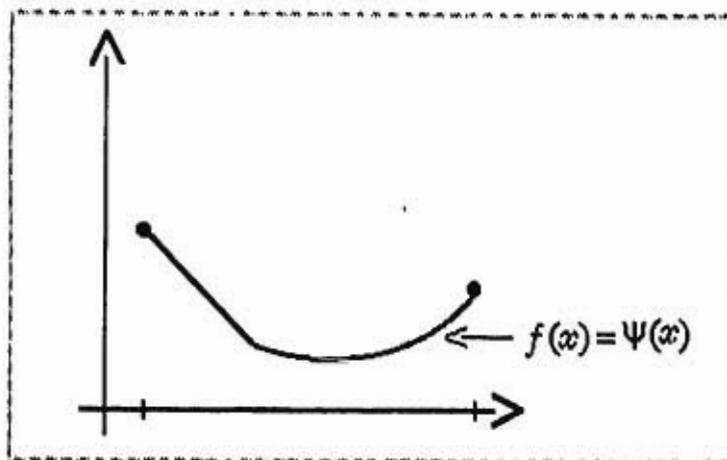


Figura 3.3.C. Cubierta convexa de una función convexa.

Definición 11 Dada una función $f : D \rightarrow \mathbb{R}$ ($D \subseteq \mathbb{R}^n$), se define su **cubierta convexa**, ψ , como "la más grande función convexa por debajo de f ". Más formalmente, sea

$\mathfrak{F}_f = \{g : D \rightarrow \mathbb{R} \mid g \text{ es convexa} \ \& \ \forall y \in D, g(y) \leq f(y)\}$,
entonces $\psi : D \rightarrow \mathbb{R}$ se define como $\psi(x) = \sup_{g \in \mathfrak{F}_f} g(x)$.

En las figuras 3.3 se muestra la cubierta convexa de tres funciones de variable real.

De la proposición 10, es claro que ψ es convexa; además, dado que el supremo de un conjunto es menor que cualquier cota del conjunto, entonces $\psi(x) \leq f(x) \quad \forall x \in D$; esto es: como para toda $g \in \mathfrak{F}_f$, $g(x) \leq f(x)$, entonces $\text{Sup}_{g \in \mathfrak{F}_f} g(x) \leq f(x)$.

Es obvio que, la cubierta convexa de una función convexa es la misma función (ver figura 3.3.C); y, para funciones de \mathbb{R} en \mathbb{R} , la cubierta convexa de una función cóncava es la recta que une sus puntos extremos (ver figura 3.3.B).

Utilizando la proposición 8, es fácil demostrar que dada una función $f: D \rightarrow \mathbb{R}$ (con $D \subseteq \mathbb{R}^n$), la cual satisface: $f(x_1, \dots, x_n) = \sum f_i(x_i) \dots (*)$, (con $f_i: D_i \rightarrow \mathbb{R}$), debe cumplir que su cubierta convexa es la suma de las cubiertas convexas de las f_i (a una función que satisface $(*)$ se le denomina *separable*). Esta última afirmación es consecuencia del siguiente resultado.

Proposición 12 Sea $f: D \rightarrow \mathbb{R}$ ($D \subseteq \mathbb{R}^2$) una función separable, con $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$; entonces, $\psi = \psi_1 + \psi_2$ (donde ψ es la cubierta convexa de f , y ψ_i es la cubierta convexa de f_i).

DEMOSTRACIÓN

Por definición, tanto ψ_1 como ψ_2 son convexas y satisfacen que

$$\psi_1(x_1) \leq f_1(x_1) \text{ y } \psi_2(x_2) \leq f_2(x_2);$$

así, $\forall (x_1, x_2)$,

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2) \geq \psi_1(x_1) + \psi_2(x_2);$$

con lo cual, por la proposición 8, $\psi_1 + \psi_2$ es una función convexa por debajo de f ; y entonces

$$(1) \dots \psi \geq \psi_1 + \psi_2.$$

Por otra parte, dada $\varepsilon > 0$ arbitraria, se cumple que, $\forall (x_1, x_2)$,

$$\psi_1(x_1) + \psi_2(x_2) + \varepsilon =$$

$$\psi_1(x_1) + \frac{\varepsilon}{2} + \psi_2(x_2) + \frac{\varepsilon}{2} \geq$$

$$f_1(x_1) + f_2(x_2) = f(x_1, x_2) \geq \psi(x_1, x_2).$$

Por lo tanto, $\forall \varepsilon > 0$,

$$\psi_1(x_1) + \psi_2(x_2) + \varepsilon \geq \psi(x_1, x_2),$$

y en consecuencia, $\forall (x_1, x_2)$,

$$\psi_1(x_1) + \psi_2(x_2) \geq \psi(x_1, x_2);$$

esto es:

$$(2) \dots \psi_1 + \psi_2 \geq \psi.$$

Finalmente, de (1) y (2) se sigue que $\psi_1 + \psi_2 = \psi$. **f**

El resultado de esta última proposición también se puede generalizar, gracias al corolario 9, para funciones f de \mathbb{R}^n en \mathbb{R} de la siguiente manera.

Corolario 13 Dada $f : \mathbb{R}^n \rightarrow \mathbb{R}$ separable como $f(x_1, \dots, x_n) = \sum f_i(x_i)$, se cumple que $\psi = \sum \psi_i$. **f**

Antes de anunciar el resultado principal de este capítulo, es necesario hacer unas últimas observaciones. Para mayor comprensión de los conceptos involucrados en la siguiente afirmación, ver el apéndice [43].

Proposición 14 Sea $f : [a, b] \rightarrow \mathbb{R}$ una **función semicontinua por debajo**, entonces ψ (su cubierta convexa) alcanza un mínimo en $[a, b]$.

DEMOSTRACIÓN

Por el resultado de la proposición 7, ψ es continua en (a, b) ; por lo que, $\forall \alpha > 0$, ψ es continua en $[a + \alpha, b - \alpha]$; y ahora, por la proposición [48] del apéndice, ψ alcanza su mínimo en cada intervalo de esta forma. Debido a lo anterior, la única posibilidad de que ψ no alcance su mínimo en $[a, b]$ es que

$$\lim_{x \rightarrow a} \psi(x) < \psi(a) \dots (*) \quad \text{ó} \quad \lim_{x \rightarrow b} \psi(x) < \psi(b) \dots (**)$$

Supóngase que pasa (*) (para un mejor entendimiento del siguiente argumento, véase la figura 3.4), entonces sea $\varepsilon = \psi(a) - \lim_{x \rightarrow a} \psi(x)$. Por ser f semicontinua por debajo, $\exists \delta_1 > 0$ tal que

$$|x - a| < \delta_1 \implies f(x) > f(a) - \frac{\varepsilon}{4};$$

y, por lo tanto, por definición de ψ

$$|x - a| < \delta_1 \implies f(x) > f(a) - \frac{\varepsilon}{4} \geq \psi(a) - \frac{\varepsilon}{4}.$$

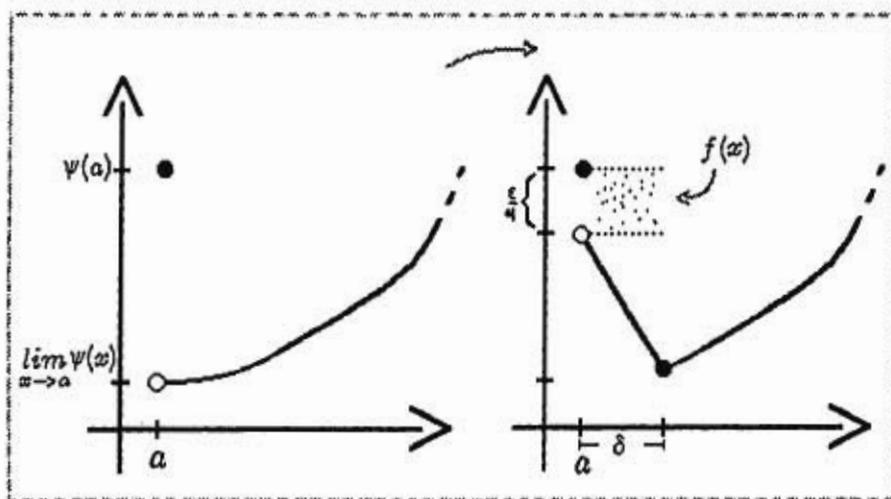


Figura 3.4. Construcción de g .

Por otra parte, por definición de límite, $\exists \delta_2 > 0$ tal que

$$0 < |x - a| < \delta_2 \implies |\psi(x) - \text{Lim } \psi(x)| < \frac{\epsilon}{4}.$$

Sea $\delta = \text{Min } \{\delta_1, \delta_2\}$, defínase la función g , de la siguiente forma:

- (i) $g(a) = \psi(a)$.
- (ii) En $(a, a + \delta]$ como la recta que pasa por $(a, \psi(a) - \frac{\epsilon}{4})$ y por $(a + \delta, \psi(a + \delta))$.
- (iii) En $(a + \delta, b]$, $g(x) = \psi(x)$.

Nótese que por construcción, g es convexa; además:

- (i)* En a , $g(a) = \psi(a) \leq f(a)$.
- (ii)* En $(a, a + \delta)$, $\psi(x) < g(x) \leq f(x)$.
- (iii)* En $[a + \delta, b]$, $g(x) = \psi(x) = f(x)$.

Por lo tanto, $g \leq f$, $g \geq \psi$ y $g \neq \psi$. Esto contradice (por (ii)*) la definición de ψ . La contradicción viene de suponer (*), por lo que (*) es imposible. De manera análoga se prueba que (**) no es posible, por lo que se concluye que ψ alcanza su mínimo.

f

Se finaliza con un resultado que será utilizado para calcular cotas en el algoritmo tipo ramificación y acotamiento presentado en el siguiente capítulo.

Teorema 15 Sea $f : D \rightarrow \mathbb{R}$ separable (donde $D = [a_1, b_1] \times \dots \times [a_n, b_n]$ y $f_i : [a_i, b_i] \rightarrow \mathbb{R}$). Entonces, si cada f_i es semicontinua por debajo, ψ alcanza su mínimo en D .

DEMOSTRACIÓN

Por el corolario 13, $\psi = \sum \psi_i$; ahora, por el resultado de la proposición 14, cada ψ_i alcanza su mínimo en $[a_i, b_i]$. Así, dado que

$$\text{Min}_{x \in D} \psi(x) = \text{Min}_{x_i \in [a_i, b_i]} \left(\sum_{i=1}^n \psi_i(x_i) \right) = \sum_{i=1}^n \left(\text{Min}_{x_i \in [a_i, b_i]} \psi_i(x_i) \right),$$

ψ alcanza su mínimo.

f

Capítulo 4

R-A EN UN PROBLEMA NO COMBINATORIO

En el presente capítulo se considera un algoritmo que resuelve el problema de encontrar $\mathbf{X} = (X_1, \dots, X_n)$, tal que minimice

$$\Phi(\mathbf{X}) = \sum_{i=1}^n \varphi_i(X_i)$$

sujeto a

$$\mathbf{X} \in G \quad \& \quad l \leq \mathbf{X} \leq L;$$

donde G es cerrado y cada φ_i es semicontinua por debajo en el intervalo $[l_i, L_i]$. Bajo estos últimos supuestos, se asegura que Φ tiene un mínimo en $G \cap C$, donde $C = \{\mathbf{X} \mid l \leq \mathbf{X} \leq L\}$. (Obviamente, también se asume que $G \cap C$ es no vacío pues, de lo contrario, el problema sería no factible.)

El algoritmo propuesto es del tipo de ramificación y acotamiento, y en él, en cada nodo se resuelve un problema cuya función objetivo es convexa. Cada uno de estos problemas restringe a la variable X_i a pertenecer a un subintervalo de $[l_i, L_i]$ y reemplaza a φ_i con su cubierta convexa sobre el subintervalo.

El algoritmo es bastante similar a un método sugerido por Lawler y Wood [1] en su estudio de métodos de ramificación y acotamiento y, a diferencia del dado por Jones y Soland [7] y del procedimiento esbozado por Beale [8], es aplicable a una clase más amplia de problemas.

Otra gran diferencia del presente algoritmo es la siguiente: en la mayoría

de los algoritmos de ramificación y acotamiento, la convergencia a la solución óptima es garantizada por la existencia de un número finito de soluciones factibles; mientras que la convergencia del algoritmo presentado aquí, no es obvia, por lo que se prueba bajo diferentes supuestos (en general, el algoritmo no converge en un número finito de pasos).

4.1 EL ALGORITMO

Se resolverá:

Minimizar

$$\Phi(\mathbf{X}) = \sum_{i=1}^n \varphi_i(\mathbf{X}_i)$$

sujeto a (P)

$$\begin{aligned} \mathbf{X} &\in G \\ l &\leq \mathbf{X} \leq L, \end{aligned}$$

donde se supone que G es cerrado y que Φ es semicontinua por debajo en $C = \{\mathbf{X} \mid l \leq \mathbf{X} \leq L\}$. Con esto último, como ya se había mencionado antes, se asegura que \mathcal{P} tiene solución si $C \cap G \neq \emptyset$.

El algoritmo es de naturaleza iterativa, y en él, se produce una sucesión de puntos $\{\mathbf{X}^k\}$. Estos puntos se determinan de forma tal que, cada \mathbf{X}^k es solución a un problema \mathcal{P}^k que trata la minimización de una función convexa a trozos sobre $C \cap G$. El problema \mathcal{P}^{k+1} es obtenido a partir del \mathcal{P}^k por ramificación y acotamiento, a través de un refinamiento de C en rectángulos cada vez más pequeños.

El problema \mathcal{P}^1 tiene la forma:

Minimizar

$$\Psi^1(\mathbf{X}) = \Psi(\mathbf{X}) = \sum_{i=1}^n \psi_i(\mathbf{X}_i)$$

sujeto a (P¹)

$$\begin{aligned} X &\in G \\ l &\leq X \leq L, \end{aligned}$$

donde Ψ es la cubierta convexa de Φ (ver corolario 13).

Si X^1 es solución de \mathcal{P}^1 y $\Psi^1(X^1) = \Phi(X^1)$, es claro que X^1 resuelve a \mathcal{P} (dado que en $C \cap G$, $\Psi^1(X^1) \leq \Psi(X) \leq \Phi(X)$).

En caso contrario (si $\Psi^1(X^1) < \Phi(X^1)$), se determina una partición de C en dos o más subconjuntos rectangulares; la cubierta convexa de Φ es entonces minimizada en cada uno de estos conjuntos, guardándose los respectivos mínimos. El proceso continúa particionando aquel subconjunto de C que corresponde al menor de los mínimos existentes y, de nueva cuenta, minimizando la cubierta convexa de Φ sobre cada uno de los conjuntos resultantes.

La estructura general del algoritmo es la que se enuncia a continuación:

PASO 1. INICIO

1.1. Crear el nodo $N = 1$, éste representa todas las posibles soluciones $X \in C \cap G$. Ir al paso 2.

PASO 2. ACOTAMIENTO

Para cada nodo sin acotar, N , hacer:

2.1. Si $\mathfrak{S}(N) = \emptyset$, $\mathfrak{B}(N) = \infty$; en otro caso ir a 2.2.

2.2. Crear la cubierta convexa de Φ en $\mathfrak{S}(N)$; sea ésta Ψ . Ir a 2.3.

2.3. Calcular el mínimo de Ψ en $\mathfrak{S}(N)$, éste es una cota para Φ en $\mathfrak{S}(N)$. (i.e., $\mathfrak{B}(N) = \text{Min } \Psi$ en $\mathfrak{S}(N)$). Ir al paso 3.

PASO 3. BÚSQUEDA

3.1. Comparar las cotas existentes. Sea N el de cota menor o igual (en caso de empate, rómpase arbitrariamente). Sea también X^0 quien minimiza a Ψ en $\mathfrak{S}(N)$. Ir a 3.2.

3.2. Si $\Phi(X^0) = \Psi(X^0)$ entonces X^0 es el óptimo, en caso contrario, hacer $i^* = N$. Ir al paso 4.

PASO 4. RAMIFICACIÓN

4.1. Particionar $\mathfrak{S}(N)$ siguiendo cualquiera de las dos reglas descritas posteriormente. Ir a 4.2.

4.2. En el árbol de búsqueda crear tantas ramas (y nodos) como

elementos de la partición. Regresar al paso 2.

De manera general, al final de la k -ésima etapa del algoritmo, el conjunto C ha sido particionado en p_k subconjuntos $C^{k_1}, \dots, C^{k_{p_k}}$. Denótese esta partición por π^k :

$$\pi^k = \{C^{k_1}, \dots, C^{k_{p_k}}\}.$$

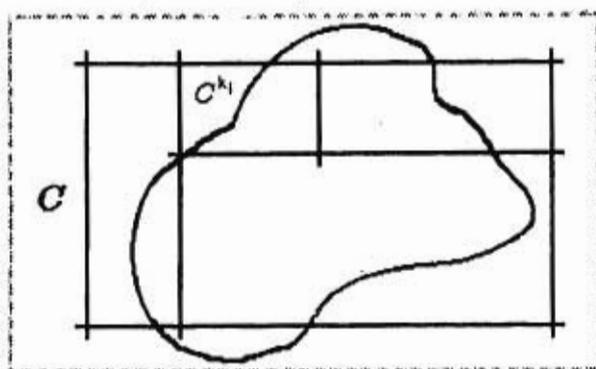


Figura 4.1. Una partición de C en \mathbb{R}^2 .

Cada C^{k_j} es descrito por dos vectores, los cuales representan las cotas superior e inferior de la variables $X \in C^{k_j}$ (ver figura 4.1). El punto $X^{k_j} \in C^{k_j}$ es determinado de manera que minimiza la cubierta convexa Ψ^{k_j} de Φ sobre $C^{k_j} \cap G$; esto es, X^{k_j} es solución del problema

$$\begin{aligned} & \text{Min } \Psi^{k_j}(X) \\ & \text{s.a.} \\ & X \in G \\ & l^{k_j} \leq X \leq L^{k_j}. \end{aligned}$$

El número $\mu^{k_j} = \Psi^{k_j}(X^{k_j})$ representa una cota inferior para Φ sobre $C^{k_j} \cap G$. Si $C^{k_j} \cap G$ es vacío, se toma $\mu^{k_j} = \infty$. El menor de los μ^{k_j} 's es entonces una cota inferior para Φ sobre $C \cap G$.

Sean

$$\mu^k = \text{Min} \{ \mu^{k_j} \mid j = 1, \dots, p_k \}$$

y

$$X^k = X^{k_{j_k}}$$

de forma tal que

$$\Psi^{k_{j_k}}(X^{k_{j_k}}) = \mu^k.$$

Nótese que puede existir más de un X^{k_j} elegible para ser X^k , en este caso se elige uno de manera arbitraria.

En el caso de que $\Phi(X^k) = \mu^k$, X^k resuelve a \mathcal{P} ; dado que es factible y $\mu^k \leq \Psi^{k_j}(X) \leq \Phi(X)$ para todo $X \in C^{k_j} \cap G$, y toda $j = 1, \dots, p_k$.

Por otro lado, si $\Phi(X^k) > \mu^k$, debe refinarse π^k para obtener un nuevo punto X^{k+1} .

Para ilustrar el procedimiento de refinación de π^k , se introduce algo de notación adicional.

La "completación" de π^k , denotada por $\bar{\pi}^k$, es obtenida extendiendo todas las particiones de los C^{k_j} 's sobre el total de C (ver figura 4.2).

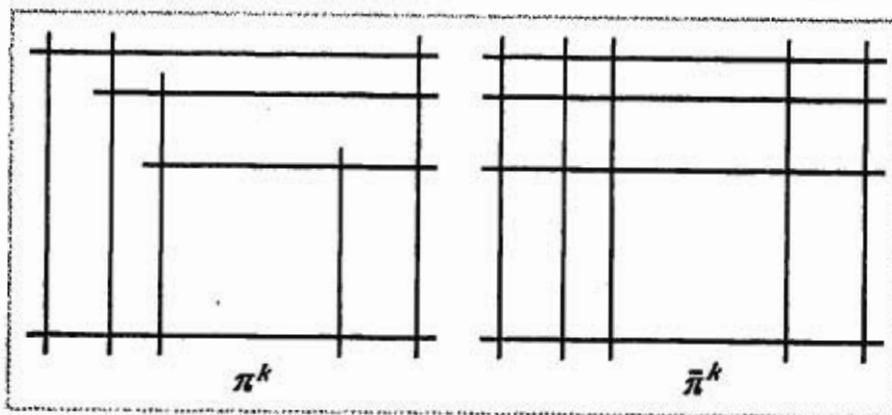


Figura 4.2. Una partición y su completación.

Sea $\pi^k = \{D^{k_1}, \dots, D^{k_{q_k}}\}$ y $\bar{\Psi}^{k_j}$ la cubierta convexa de Ψ^{k_j} sobre D^{k_j} . Dado que $\bar{\pi}^k$ es un refinamiento completo, es posible expresar la colección de funciones $\bar{\Psi}^{k_j}$ ($j = 1, \dots, q_k$) como una única función separable

$$\bar{\Psi}^k(X) = \sum \bar{\psi}_i^k(X_i);$$

donde $\bar{\psi}_i^k$ es la función *convexa a pedazos* definida en $[l_i, L_i]$, y cuya representación entre dos puntos adyacentes de la subdivisión, x_i^k, x_i^{k+1} , está dada por la cubierta convexa de φ_i sobre $[x_i^k, x_i^{k+1}]$.

No es necesario calcular enteramente $\bar{\Psi}^k$ en el paso k , pero el punto \mathbf{X}^k debe ser determinado de forma que minimice $\bar{\Psi}^k$ sobre $C \cap G$; es decir, se asume de manera explícita que $C^{k_{j_k}} \in \bar{\pi}^k$.

Es claro, dado que $\bar{\pi}^k$ es la completación de π^k , que $\bar{\Psi}^k(\mathbf{X}) \geq \Psi^{k_j}(\mathbf{X})$ para toda $\mathbf{X} \in C^{k_j}$ y para cualquier $j = 1, \dots, p_k$. Además, la construcción de $\bar{\Psi}^k$ y Ψ^{k_j} garantiza que si $C^{k_j} \in \bar{\pi}^k$ entonces $\bar{\Psi}^k(\mathbf{X}) = \Psi^{k_j}(\mathbf{X})$ para toda $\mathbf{X} \in C^{k_j}$.

Resumiendo, en la etapa k se tiene:

- Una partición $\pi^k = \{C^{k_1}, \dots, C^{k_{p_k}}\}$.
- Una colección de funciones $\{\Psi^{k_1}, \dots, \Psi^{k_{p_k}}\}$ donde cada Ψ^{k_j} es la cubierta convexa de Φ sobre C^{k_j} .
- Una colección de puntos $\{\mathbf{X}^{k_j}\}_{j=1}^{p_k}$, donde \mathbf{X}^{k_j} es solución del problema P^{k_j} .
- Un punto $\mathbf{X}^k = \mathbf{X}^{k_{j_k}}$ que satisface $\Psi^{k_{j_k}}(\mathbf{X}^{k_{j_k}}) \leq \Psi^{k_j}(\mathbf{X}^{k_j}) \forall j = 1, \dots, p_k$ y en donde $C^{k_{j_k}} \in \bar{\pi}^k$.

Así entonces, \mathbf{X}^k es solución del problema

$$\begin{aligned} \text{Min } \bar{\Psi}^k(\mathbf{X}) &= \sum \bar{\psi}_i^k(\mathbf{X}_i) \\ \text{s.t. } \mathbf{X} &\in C \cap G. \end{aligned}$$

A fin de pasar de la etapa k a la $k+1$, se debe dividir el conjunto $C^{k_{j_k}} = \{\mathbf{X} \mid l^{k_{j_k}} \leq \mathbf{X} \leq L^{k_{j_k}}\}$ en dos o más subconjuntos rectangulares de acuerdo a una de las siguientes reglas:

Regla de refinamiento débil. Escoger una i que maximice la diferencia $\varphi_i(\mathbf{X}_i^k) - \psi_i^{k_{j_k}}(\mathbf{X}_i^k)$ ($i = 1, \dots, n$), y dividir el correspondiente intervalo $[l_i^{k_{j_k}}, L_i^{k_{j_k}}]$ en los intervalos $[l_i^{k_{j_k}}, \mathbf{X}_i^k]$ y $[\mathbf{X}_i^k, L_i^{k_{j_k}}]$.

Regla de refinamiento fuerte. Para cada i que satisfaga $\varphi_i(\mathbf{X}_i^k) -$

$\psi_i^{k_{j_k}}(\mathbf{X}_i^k) > 0$ dividir el correspondiente intervalo $[l^{k_{j_k}}, L^{k_{j_k}}]$ en los intervalos $[l^{k_{j_k}}, \mathbf{X}_i^k]$ y $[\mathbf{X}_i^k, L^{k_{j_k}}]$.

Obsérvese que la segunda regla puede dividir $C^{k_{j_k}}$ hasta en 2^n subconjuntos rectangulares, mientras que la primera proporciona sólo dos subconjuntos.

Cada subdivisión de $C^{k_{j_k}}$ representa un refinamiento de la partición π^k (y por lo tanto de $\bar{\pi}^k$). El refinamiento de π^k se define como π^{k+1} . La función $\bar{\Psi}^{k+1}$ se define de manera semejante a $\bar{\Psi}^k$, y el punto \mathbf{X}^{k+1} es obtenido de la minimización de $\bar{\Psi}^{k+1}$ sobre $C \cap G$.

Para que esto último quede más claro, supóngase que $C^{k_{j_k}} = C^{k_{p_k}}$; después de la aplicación de una de las reglas de refinamiento, se pueden numerar los miembros de la partición resultante de la siguiente forma:

$$C^{k+1,1}, \dots, C^{k+1,p_k-1}, \dots, C^{k+1,t_k};$$

donde

$$C^{k+1,j} = C^{k_j} \quad \text{para } j = 1, \dots, p_k - 1;$$

y

$$\cup C^{k+1,j} = C^{k_{p_k}} \quad \text{para } j = p_k, \dots, t_k.$$

Los problemas $\mathcal{P}^{k+1,j}$ ($j = 1, \dots, p_k - 1$) ya se resolvieron en la etapa anterior y, entonces, ya se tienen a la mano sus cotas inferiores $\mu^{k+1,j}$. Se resuelven después los problemas $\mathcal{P}^{k+1,j}$ ($j = p_k, \dots, t_k$) para obtener sus respectivas cotas.

Si

$$\mu^{k+1,l} = \text{Min} \{ \mu^{k+1,j} \mid j = 1, \dots, t_k \} \text{ y } l \in \{p_k, \dots, t_k\},$$

hacer

$$\mathbf{X}^{k+1} = \mathbf{X}^{k+1,l}$$

(ya que $C^{k+1,l} \in \bar{\pi}^{k+1}$). Sin embargo, puede pasar que $l \in \{1, \dots, p_k - 1\}$ (y por lo tanto $C^{k+1,l} \notin \bar{\pi}^{k+1}$), en tal caso se debe subdividir $C^{k+1,l}$ en dos o más subconjuntos (añadiendo hiperplanos que partirán a $C^{k+1,l}$ en subconjuntos más pequeños). Al hacer lo anterior se forman nuevos subproblemas y, por lo tanto, deben calcularse las nuevas cotas. (Nótese que en general no es necesario tomar todos los elementos de $\bar{\pi}^{k+1}$.)

Es hasta el momento en que $X^{k+1} \in C^{k+1,l} \in \bar{\pi}^{k+1}$ que se encuentra una solución para \mathcal{P}^{k+1} (en los teoremas de convergencia de la siguiente sección, esto es necesario para asegurar que el algoritmo converge a un óptimo). La partición $\bar{\pi}^{k+1}$ es el último refinamiento necesario para encontrar X^{k+1} .

El método de determinación de X^{k+1} puede ser ilustrado con un ejemplo en \mathbb{R}^2 . Véase la figura 4.3, en ésta, el número en cada rectángulo corresponde a la cota inferior asociada con el rectángulo; además obsérvese que la regla de refinamiento fuerte es usada para dividir el conjunto con la cota más pequeña. Los refinamientos adicionales son obtenidos añadiendo las líneas apropiadas de $\bar{\pi}^{k+1}$.

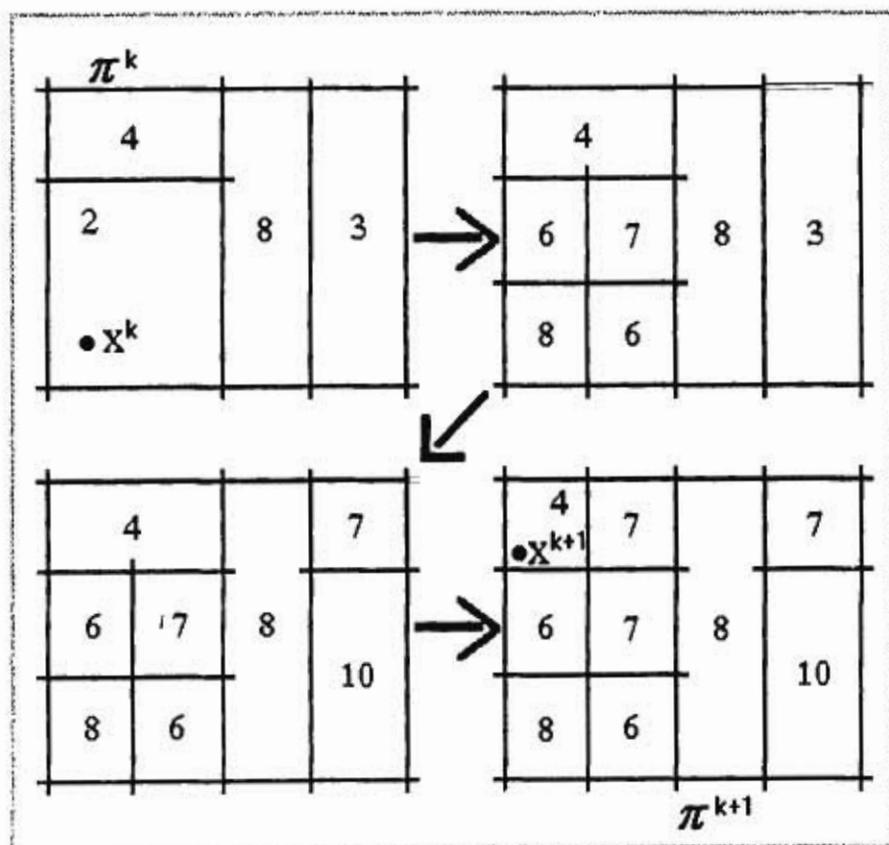


Figura 4.3. Determinación de X^{k+1} .

4.2 TEOREMAS DE CONVERGENCIA

El propósito de esta sección es demostrar que el algoritmo presentado y explicado en la sección anterior, converge a un óptimo; para esto obsérvese que el proceso de determinación del elemento de $\bar{\pi}^{k+1}$ en el que X^{k+1} se encuentra, es un procedimiento finito (dado que existe un número finito de elementos en $\bar{\pi}^{k+1}$); sin embargo, la sucesión de puntos $\{X^k\}$ puede ser infinita; en tal caso, del hecho que $C \cap G$ sea compacto, se deduce que tiene al menos un punto de acumulación X^* (ver apéndice [33]).

Por lo anterior son necesarios los teoremas de convergencia citados más adelante, en los cuales se probará que, bajo determinadas hipótesis, X^* es solución de \mathcal{P} . Antes de enunciar estos teoremas y de dar su demostración, nótese que:

"Si la sucesión $\{X^k\}$ es finita, su último punto X^k debe ser solución del problema \mathcal{P} , pues, el algoritmo se detiene sólo cuando $\Psi^k(X^k) = \Phi(X^k)$."

Se pasa ahora a la prueba de la convergencia del algoritmo, en estas pruebas se hará uso de resultados vistos en el capítulo 3, así como resultados de Análisis Matemático, los cuales son citados en el apéndice.

Sea τ la subsucesión convergente de $\{X^k\}$ (apéndice [40]), y sea T el correspondiente conjunto de índices ($T \subseteq N$); así,

$$X^* = \lim_{k \in T} X^k$$

También por el resultado del apéndice [40], $X^* \in C \cap G$, lo que significa que **" X^* es factible"**.

Las sucesiones $\tau_i = \{X_i^k \mid k \in T\}$ son sucesiones de reales que convergen a X_i^* respectivamente (apéndice [36]). Por comodidad (extrayendo subsucesiones si es necesario) se puede suponer que cada τ_i es monótona; no sólo eso, sin pérdida de generalidad se puede suponer que son monótonas decrecientes; esto es, que las sucesiones τ_i satisfacen:

$$k < l \implies X_i^k \geq X_i^l.$$

Otro hecho que debe resaltarse, es que las funciones $\bar{\Psi}^k$, definidas en la sección anterior, forman una sucesión de funciones crecientes y acotadas por

Φ ; por lo tanto, $\{\bar{\Psi}^k\}$ converge puntualmente a alguna función $\bar{\Psi}$ (apéndice [42]); esto es, $\exists \bar{\Psi}$ t.q. $\forall X \in C$ se cumple:

$$\lim_{k \rightarrow \infty} \bar{\Psi}^k(X) = \bar{\Psi}(X).$$

Una vez hechas las definiciones y observaciones anteriores, se pasa a la parte densa del trabajo.

Lema 16 $\forall X \in C, \bar{\Psi}(X) \leq \Phi(X)$

DEMOSTRACIÓN

$$\bar{\Psi}^k(X) \leq \Phi(X) \quad \forall X \in C;$$

por lo tanto $\forall X \in C$

$$\bar{\Psi}(X) = \lim_{k \rightarrow \infty} \bar{\Psi}^k(X) \leq \Phi(X).$$

f

Lema 17 $\lim_{k \in \mathbb{T}} \bar{\Psi}^k(X^k) = \bar{\Psi}(X^*)$

DEMOSTRACIÓN

Dado que $\sigma = \{\bar{\Psi}^k(X^k)\}$ es una sucesión monótona no decreciente y acotada por $\Phi(X^*)$ (donde X^* es el óptimo de \mathcal{P}), esta sucesión tiene límite; lo que se pretende probar es que este límite coincide con $\bar{\Psi}(X^*)$.

Como para $k \in \mathbb{T}$, pasa que $\bar{\Psi}^k \leq \bar{\Psi}$; y además σ es monótona no decreciente, se tiene que

$$\bar{\Psi}^k(X^k) \leq \bar{\Psi}^k(X^*) \leq \bar{\Psi}(X^*).$$

Así pues,

$$\lim_{k \in T} \bar{\Psi}^k(\mathbf{X}^k) \leq \bar{\Psi}(\mathbf{X}^*) .$$

Para demostrar la desigualdad opuesta, tómesese $\varepsilon > 0$, arbitraria. Por definición de $\bar{\Psi}$, existe k_1 tal que

$$k > k_1 \implies \bar{\Psi}^k(\mathbf{X}^*) > \bar{\Psi}(\mathbf{X}^*) - \varepsilon .$$

Defínase

$$A = \{X \in C \mid \bar{\Psi}^{k_1}(X) > \bar{\Psi}(\mathbf{X}^*) - \varepsilon\} .$$

Por el resultado de la proposición 7, $\bar{\Psi}^{k_1}$ es continua (es una función convexa), además, dado que $\mathbf{X}^* \in A$, A es no vacío y abierto con respecto a C (ver apéndice [24]). Por otra parte, τ converge a \mathbf{X}^* . De estas dos últimas observaciones se concluye que existe $k_2 \geq k_1$ tal que

$$k \geq k_2 \ \& \ k \in T \implies \mathbf{X}^k \in A ;$$

es decir,

$$k \geq k_2 \ \& \ k \in T \implies \bar{\Psi}(\mathbf{X}^*) - \bar{\Psi}^{k_1}(\mathbf{X}^k) \leq \varepsilon .$$

Ahora, dado que para $k \geq k_2$ pasa que

$$\bar{\Psi}^k(\mathbf{X}^k) \geq \bar{\Psi}^{k_1}(\mathbf{X}^k) ,$$

se tiene:

$$k \geq k_2 \ \& \ k \in T \implies 0 \leq \bar{\Psi}(\mathbf{X}^*) - \bar{\Psi}^k(\mathbf{X}^k) \leq \varepsilon .$$

Por lo tanto, se ha probado que para $\varepsilon > 0$ arbitraria,

$$\bar{\Psi}^k(\mathbf{X}^k) \geq \bar{\Psi}(\mathbf{X}^*) - \varepsilon ,$$

y con esto,

$$\lim_{k \in \mathbb{T}} \bar{\Psi}^k(X^k) \geq \bar{\Psi}(X^*) - \varepsilon$$

de lo cual se deduce:

$$\lim_{k \in \mathbb{T}} \bar{\Psi}^k(X^k) \geq \bar{\Psi}(X^*)$$

con lo que se concluye la demostración.

f

Lema 18 X^* minimiza $\bar{\Psi}$ sobre $C \cap G$

DEMOSTRACIÓN

Para toda $X \in C \cap G$ ocurre que, si $k \in \mathbb{T}$

$$\bar{\Psi}^k(X^k) \leq \bar{\Psi}^k(X) \leq \bar{\Psi}(X),$$

por lo que

$$\lim_{k \in \mathbb{T}} \bar{\Psi}^k(X^k) \leq \bar{\Psi}(X),$$

y por el lema 17

$$\bar{\Psi}(X^*) \leq \bar{\Psi}(X).$$

f

Lema 19 Si $\bar{\Psi}(X^*) = \Phi(X^0)$ con $X^0 \in C \cap G$, entonces X^0 es solución de \mathcal{P} .

DEMOSTRACIÓN

Por el lema 16, para toda $X \in C$

$$\bar{\Psi}(X) \leq \Phi(X);$$

y por el lema 18, también para toda $X \in C$,

$$\bar{\Psi}(X^*) \leq \bar{\Psi}(X).$$

Por lo tanto, para toda $X \in C$

$$\bar{\Psi}(X^*) \leq \Phi(X),$$

pero

$$\bar{\Psi}(X^*) = \Phi(X^0);$$

y así, para toda $X \in C$

$$\Phi(X^0) \leq \Phi(X),$$

y como por hipótesis $X^0 \in C \cap G$, X^0 es solución de \mathcal{P} .

f

Teorema 20 *Cualquier punto de acumulación de $\{X^k\}$ es solución de \mathcal{P} si se cumplen las condiciones:*

- a. Φ es semicontinua por la izquierda,
- b. G es cerrado, y
- c. se usa la regla de refinamiento fuerte para generar $\{X^k\}$.

DEMOSTRACIÓN

Como cada φ_i es semicontinua por debajo en X_i^* (véase apéndice [43]), dada $\varepsilon > 0$ existe δ tal que, para cada $i = 1, \dots, n$

$$|X_i - X_i^*| < \delta \implies \varphi_i(X_i) \geq \varphi_i(X_i^*) - \frac{\varepsilon}{n}.$$

Por otra parte, $\{X^k \mid k \in \mathbb{T}\}$ converge a X^* , por lo que existe k_1 tal que

$$k \geq k_1 \implies |X_i^k - X_i^*| < \delta.$$

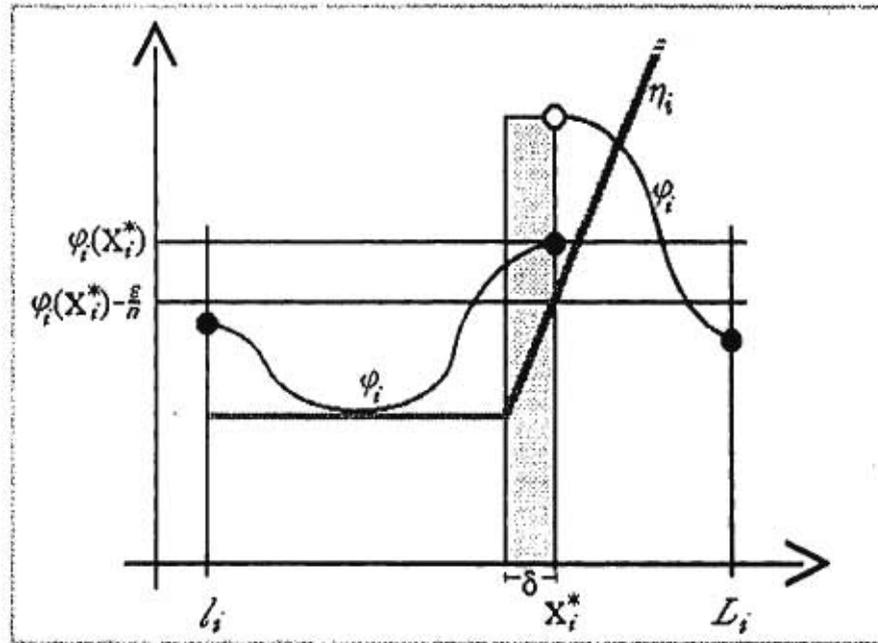


Figura 4.4. Definición de η_i .

Construyanse las funciones η_i de la siguiente manera (véase la figura 4.4): entre l_i y $X_i^* - \delta$, $\eta_i(X_i)$ es constante con valor

$$\text{Min} \left\{ \text{Min} \{ \varphi_i(X_i) \mid l_i \leq X_i \leq X_i^* - \delta \}, \varphi_i(X_i^*) - \frac{\varepsilon}{n} \right\}$$

(esta definición es válida aún para $X_i^* - \delta$). Además, entre $X_i^* - \delta$ y L_i , $\eta_i(X_i)$ es lineal y pasa por los puntos

$$(X_i^* - \delta, \eta_i(X_i^* - \delta)), (X_i^*, \varphi_i(X_i^*) - \frac{\varepsilon}{n}).$$

Es claro que existe $k_2 \geq k_1$ tal que para toda $X_i \in [X_i^*, X_i^k]$

$$k \geq k_2 \implies \varphi_i(X_i) \geq \eta_i(X_i)$$

(recuérdese que X_i^k es monótona decreciente y converge a X_i^* ; así, para k suficientemente grande, X_i^k está tan cerca de X_i^* que se puede asegurar la

desigualdad).

Se tiene entonces que $\bar{\psi}_i^{k+1}(\mathbf{X}_i)$ está definida sobre el intervalo $[l_i, \mathbf{X}_i^k]$ con $\bar{\psi}_i^{k+1}(l_i) = \varphi_i(l_i)$ y $\bar{\psi}_i^{k+1}(\mathbf{X}_i^k) = \varphi_i(\mathbf{X}_i^k)$ (esto último dado que se está usando la regla de refinamiento fuerte); y, dado que η_i es convexa sobre ese intervalo y $\varphi_i(\mathbf{X}_i) \geq \eta_i(\mathbf{X}_i)$, es claro que para $k \geq k_2$ y $\mathbf{X}_i \in [\mathbf{X}_i^*, \mathbf{X}_i^k]$ (con $i = 1, \dots, n$), ocurre que

$$\bar{\psi}_i^{k+1}(\mathbf{X}_i) \geq \eta_i(\mathbf{X}_i) \geq \varphi_i(\mathbf{X}_i^*) - \frac{\varepsilon}{n},$$

por lo que

$$\bar{\psi}_i^{k+1}(\mathbf{X}_i) \geq \varphi_i(\mathbf{X}_i^*) - \frac{\varepsilon}{n}, \quad \dots \text{ (Q}_i\text{)}$$

y entonces

$$\lim_{i \rightarrow \infty} \bar{\psi}_i^{k+1}(\mathbf{X}_i) \geq \varphi_i(\mathbf{X}_i^*) - \frac{\varepsilon}{n},$$

y así,

$$\bar{\psi}_i^{k+1}(\mathbf{X}_i^*) \geq \varphi_i(\mathbf{X}_i^*) - \frac{\varepsilon}{n}.$$

Por lo tanto pasa que

$$\bar{\Psi}(\mathbf{X}^*) \geq \bar{\Psi}^{k+1}(\mathbf{X}^*) = \sum \bar{\psi}_i^{k+1}(\mathbf{X}_i^*) \geq \sum \left(\varphi_i(\mathbf{X}_i^*) - \frac{\varepsilon}{n} \right) = \Phi(\mathbf{X}^*) - \varepsilon.$$

Se ha demostrado entonces que, dada $\varepsilon > 0$ arbitraria

$$\bar{\Psi}(\mathbf{X}^*) \geq \Phi(\mathbf{X}^*) - \varepsilon,$$

por lo cual se concluye que

$$\bar{\Psi}(\mathbf{X}^*) \geq \Phi(\mathbf{X}^*);$$

y, por lo tanto, ya que Φ es cota superior para $\bar{\Psi}$,

$$\bar{\Psi}(X^*) = \Phi(X^*).$$

Finalmente, del lema 19 se concluye que X^* es solución de \mathcal{P} .

f

Teorema 21 *Cualquier punto de acumulación de $\{X^k\}$ es solución de \mathcal{P} si se cumplen:*

- a. Φ es continua,
- b. G es cerrado, y
- c. se usa la regla de refinamiento débil para generar X^k .

DEMOSTRACIÓN

Tómese una sucesión τ como en la demostración del teorema 20, si ésta es finita, nada hay que probar; en otro caso, $\exists m \in \{1, \dots, n\}$ tal que, para $i = 1, \dots, n$, pasa

$$\varphi_m(X_m^k) - \bar{\psi}_m^k(X_m^k) \geq \varphi_i(X_i^k) - \bar{\psi}_i^k(X_i^k) \quad \dots (\mathbb{Q}_2)$$

Definiendo ahora sobre $[l_m, L_m]$ la función η_m (como en la demostración del teorema 20), se llega a una desigualdad parecida a (\mathbb{Q}_1) , pero sólo válida para m :

$$\bar{\psi}_m^{k+1}(X_m) \geq \varphi_m(X_m^*) - \frac{\epsilon}{n}, \quad \dots (\mathbb{Q}_3)$$

para toda $X_m \in [X_i^*, X_i^k]$, $k \geq k_2$ y $k \in \mathbb{T}$.

Fíjese entonces $k = k_0 \in \mathbb{T}$ con $k_0 \geq k_2$. Dado que $\{\bar{\psi}_m^l\}$ es monótona no decreciente, (\mathbb{Q}_3) implica que, para $l > k_0$, $l \in \mathbb{T}$ y $X_m \in [X_i^*, X_i^{k_0}]$

$$\bar{\psi}_m^l(X_m) \geq \varphi_m(X_m^*) - \frac{\epsilon}{n}.$$

En particular

$$\bar{\psi}_m^l(X_m^l) \geq \varphi_m(X_m^*) - \frac{\epsilon}{n} \quad \dots (\mathbb{Q}_4)$$

Ahora, como (A₂) es válida para $k = l$, se puede sumar con (A₄) para obtener que, para toda $l > k_0$ y $l \in \mathbb{T}$

$$\varphi_m(\mathbf{X}_m^l) - \varphi_m(\mathbf{X}_m^*) + \bar{\psi}_i^l(\mathbf{X}_i^l) \geq \varphi_i(\mathbf{X}_i^l) - \frac{\varepsilon}{n}.$$

De lo anterior se sigue que

$$n [\varphi_m(\mathbf{X}_m^l) - \varphi_m(\mathbf{X}_m^*)] + \bar{\Psi}^l(\mathbf{X}^l) \geq \Phi(\mathbf{X}^l) - \varepsilon,$$

y, tomando el límite cuando $l \rightarrow \infty$ con $l \in \mathbb{T}$ (dado que Φ es continua y, gracias al lema 17), se tiene

$$\bar{\Psi}(\mathbf{X}^*) \geq \Phi(\mathbf{X}^*) - \varepsilon$$

con $\varepsilon > 0$, arbitraria. Por lo tanto

$$\bar{\Psi}(\mathbf{X}^*) \geq \Phi(\mathbf{X}^*).$$

Se concluye entonces que

$$\bar{\Psi}(\mathbf{X}^*) = \Phi(\mathbf{X}^*)$$

y, por el lema 19, \mathbf{X}^* es un óptimo del problema \mathcal{P} .

f

Teorema 22 $\{\mathbf{X}^k\}$ es finita y termina en una solución de \mathcal{P} si se cumplen:

- Φ es cóncava (a lo más discontinua a lo largo de la frontera de C),
- G es un poliedro lineal, y
- se usa la regla de refinamiento débil para generar $\{\mathbf{X}^k\}$

DEMOSTRACIÓN

Cada $\bar{\psi}_i^k$ es lineal a pedazos y cóncava, por lo que cada \mathbf{X}^k debe encontrarse en un vértice de $C \cap G$; pero, como existe un número finito de vértices, el algoritmo termina en un número finito de pasos; y por lo tanto, como ya

se había mencionado antes, el algoritmo termina en una solución de P .

f

4.3 EL EJEMPLO

Sea G el polígono convexo definido por

$$\begin{aligned} -X_1 + 3X_2 &\leq 5 \\ 2X_1 - X_2 &\leq 5 \\ -2X_1 + X_2 &\leq 0 \\ X_1 - 3X_2 &\leq 0. \end{aligned}$$

El conjunto C está definido como

$$\begin{aligned} 0 &\leq X_1 \leq 6 \\ 0 &\leq X_2 \leq 5; \end{aligned}$$

$G \cap C$ se muestra en la gráfica de la figura 4.5. (Para este ejemplo, el teorema 22 garantiza convergencia finita.)

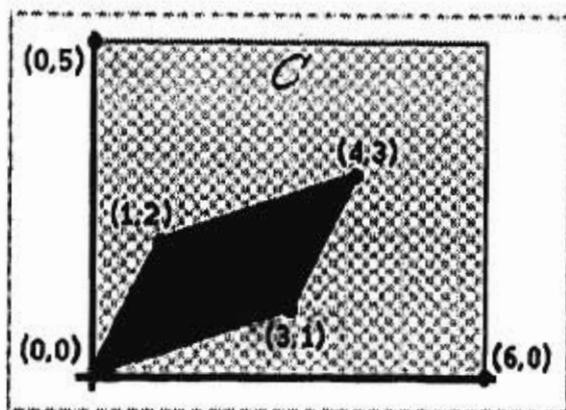


Figura 4.5. Región factible ($G \cap C$).

La función objetivo es la siguiente:

$$\Phi(\mathbf{X}) = \varphi_1(\mathbf{X}_1) + \varphi_2(\mathbf{X}_2),$$

con

$$\varphi_1(\mathbf{X}_1) = \begin{cases} 0 & \text{si } \mathbf{X}_1 = 0 \\ -\mathbf{X}_1^2 + 4\mathbf{X}_1 + 2 & \text{si } 0 < \mathbf{X}_1 \leq 6 \end{cases}$$

$$\varphi_2(\mathbf{X}_2) = \begin{cases} 2 & \text{si } \mathbf{X}_2 = 0 \\ -\mathbf{X}_2^2 + 2\mathbf{X}_2 + 4 & \text{si } 0 < \mathbf{X}_2 \leq 5 \end{cases}$$

SOLUCIÓN

Iteración 1 ($k = 1$)

Paso 1. Se crea el nodo $N = 1$, el cual representa a todas las posibles soluciones del problema. El árbol de búsqueda es el mostrado en la figura 4.6.



Figura 4.6. Inicial árbol de búsqueda.

Paso 2. $\mathcal{G}(N) \neq \emptyset$ pues $G \cap C$ es no vacío; así, se calcula la cubierta convexa de Φ , que no es otra cosa que la suma de las cubiertas convexas de φ_1 y φ_2 . Estas cubiertas se muestran en la figura 4.7. y están dadas por:

$$\begin{aligned} \psi_1^1(\mathbf{X}_1) &= -\frac{5}{3}\mathbf{X}_1 \\ \psi_2^1(\mathbf{X}_2) &= -\frac{13}{5}\mathbf{X}_2 + 2. \end{aligned}$$

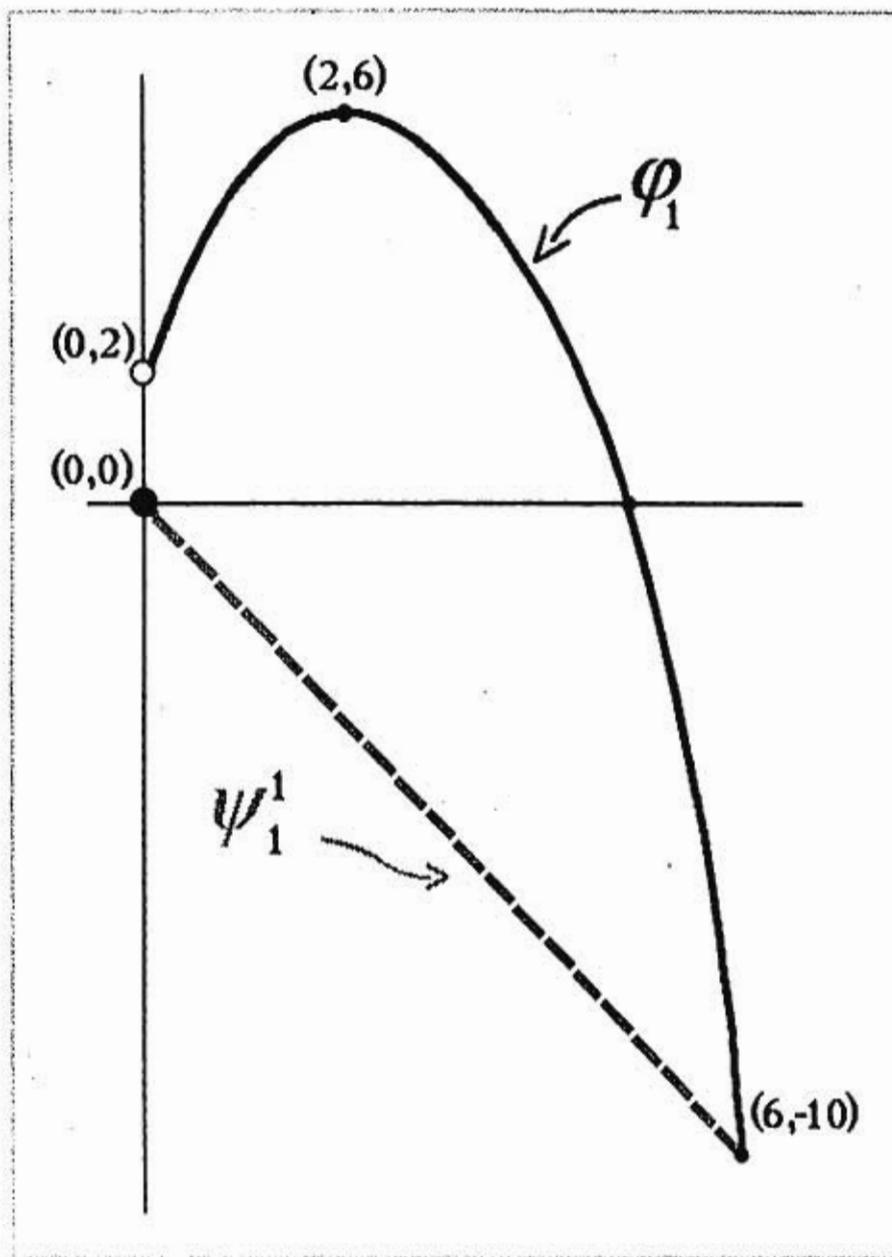


Figura 4.7.A. Cubierta convexa de φ_1 en la primer iteración.

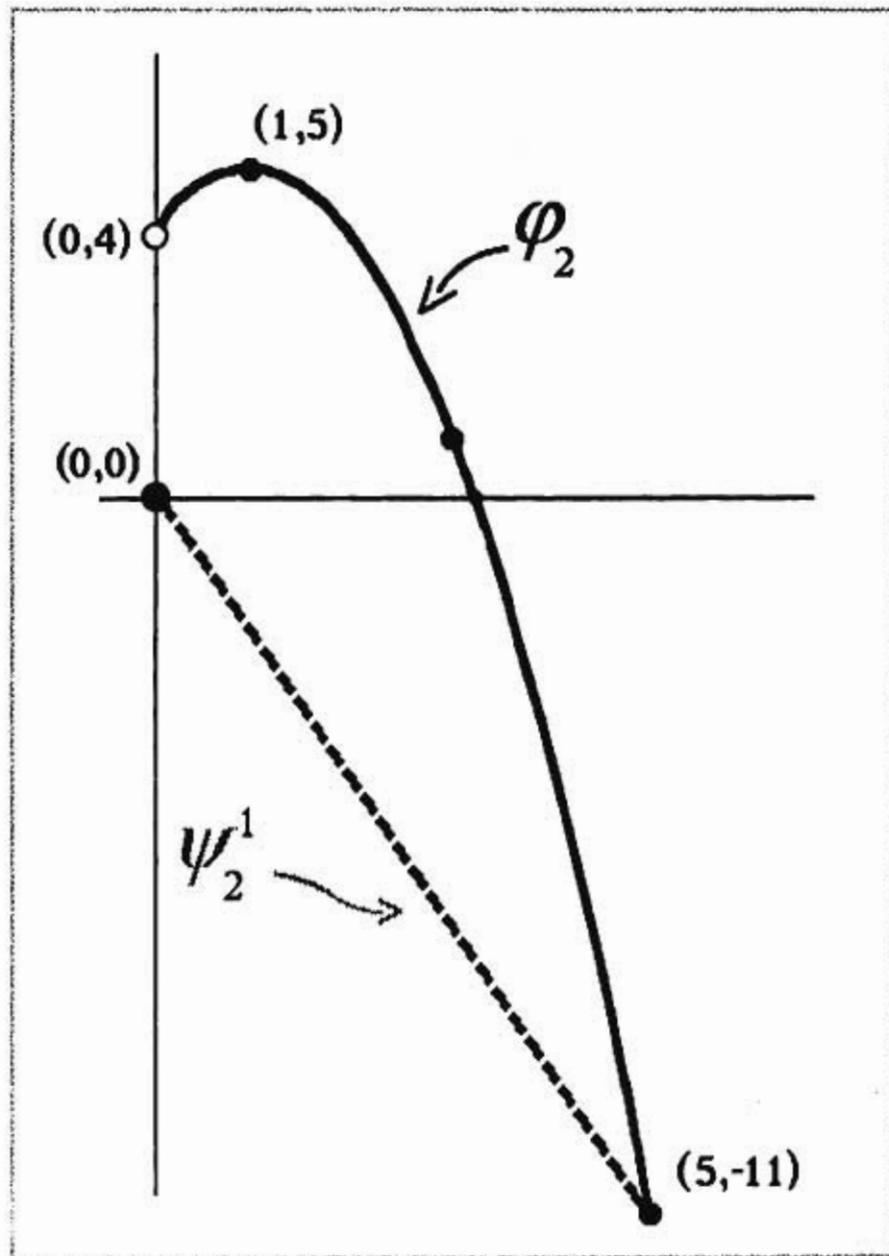


Figura 4.7.B. Cubierta convexa de φ_2 en la primer iteración.

Se resuelve entonces el problema

$$\text{Min } -\frac{5}{3}X_1 - \frac{13}{5}X_2 + 2$$

sujeeto a

$$-X_1 + 3X_2 \leq 5$$

$$2X_1 - X_2 \leq 5$$

$$-2X_1 + X_2 \leq 0$$

$$X_1 - 3X_2 \leq 0$$

$$0 \leq X_1 \leq 6$$

$$0 \leq X_2 \leq 5;$$

el cual es un clásico problema de programación lineal, y que tiene como solución: $X_1 = 4$, $X_2 = 3$, con valor -12.47 .

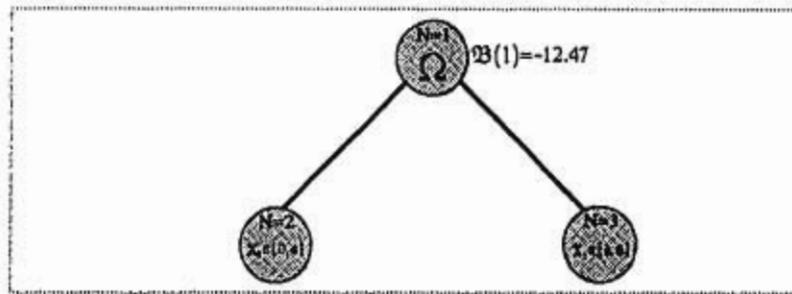


Figura 4.8. Árbol de búsqueda al final de la primer iteración.

Paso 3. Dado que existe un único nodo, $N = 1$, se buscará en él al óptimo. Después, como $\Phi(X^k) = 3 > -12.47 = \Psi(X^k)$, se hace $i^* = 1$.

Así, para $k = 1$ se tiene:

$$X^k = (4, 3),$$

$$\mu^k = \mathfrak{B}(1) = -12.47,$$

$$\Phi(X^k) = 3.$$

Paso 4. Se utiliza la regla de refinamiento débil, por lo que se calculan las diferencias

$$\varphi_1(X_1^k) - \psi_1(X_1^k) = 2 - (-0.60) = 8.60,$$

$$\varphi_2(\mathbf{X}_2^k) - \psi_2(\mathbf{X}_2^k) = 1 - (-5.88) = 6.8.$$

Dado que $8.66 > 6.8$, se ramifica por \mathbf{X}_1 , partiendo $[0, 6]$ en $[0, 4]$, $[4, 6]$; y se regresa a 2. El árbol de búsqueda obtenido es el que se muestra en la figura 4.8.

Iteración 2 ($k = 2$)

Paso 2. En ninguno de los dos casos $\mathcal{G}(N)$ es vacío por lo que se pasa a 2.3 en cada nodo.

Para $N = 2$ se tiene

$$\begin{aligned}\psi_1^2(\mathbf{X}_1) &= \frac{1}{2}\mathbf{X}_1 \\ \psi_2^2(\mathbf{X}_2) &= -\frac{13}{5}\mathbf{X}_2 + 2.\end{aligned}$$

(La nueva ψ_1^k se muestra en la figura 4.9.)
Se resuelve el problema

$$\begin{aligned}Min \quad & \frac{1}{2}\mathbf{X}_1 - \frac{13}{5}\mathbf{X}_2 + 2 \\ \text{sujeto a} \quad & -\mathbf{X}_1 + 3\mathbf{X}_2 \leq 5 \\ & 2\mathbf{X}_1 - \mathbf{X}_2 \leq 5 \\ & -2\mathbf{X}_1 + \mathbf{X}_2 \leq 0 \\ & \mathbf{X}_1 - 3\mathbf{X}_2 \leq 0 \\ & 0 \leq \mathbf{X}_1 \leq 4 \\ & 0 \leq \mathbf{X}_2 \leq 5;\end{aligned}$$

obteniéndose el valor óptimo -3.8 para $\mathbf{X} = (4, 3)$.

Por otra parte, para $N = 3$ se tiene

$$\begin{aligned}\psi_1^2(\mathbf{X}_1) &= -6\mathbf{X}_1 + 26 \\ \psi_2^2(\mathbf{X}_2) &= -\frac{13}{5}\mathbf{X}_2 + 2.\end{aligned}$$

Se resuelve el problema

$$\begin{aligned}Min \quad & -6\mathbf{X}_1 - \frac{13}{5}\mathbf{X}_2 + 28 \\ \text{sujeto a} \quad & \end{aligned}$$

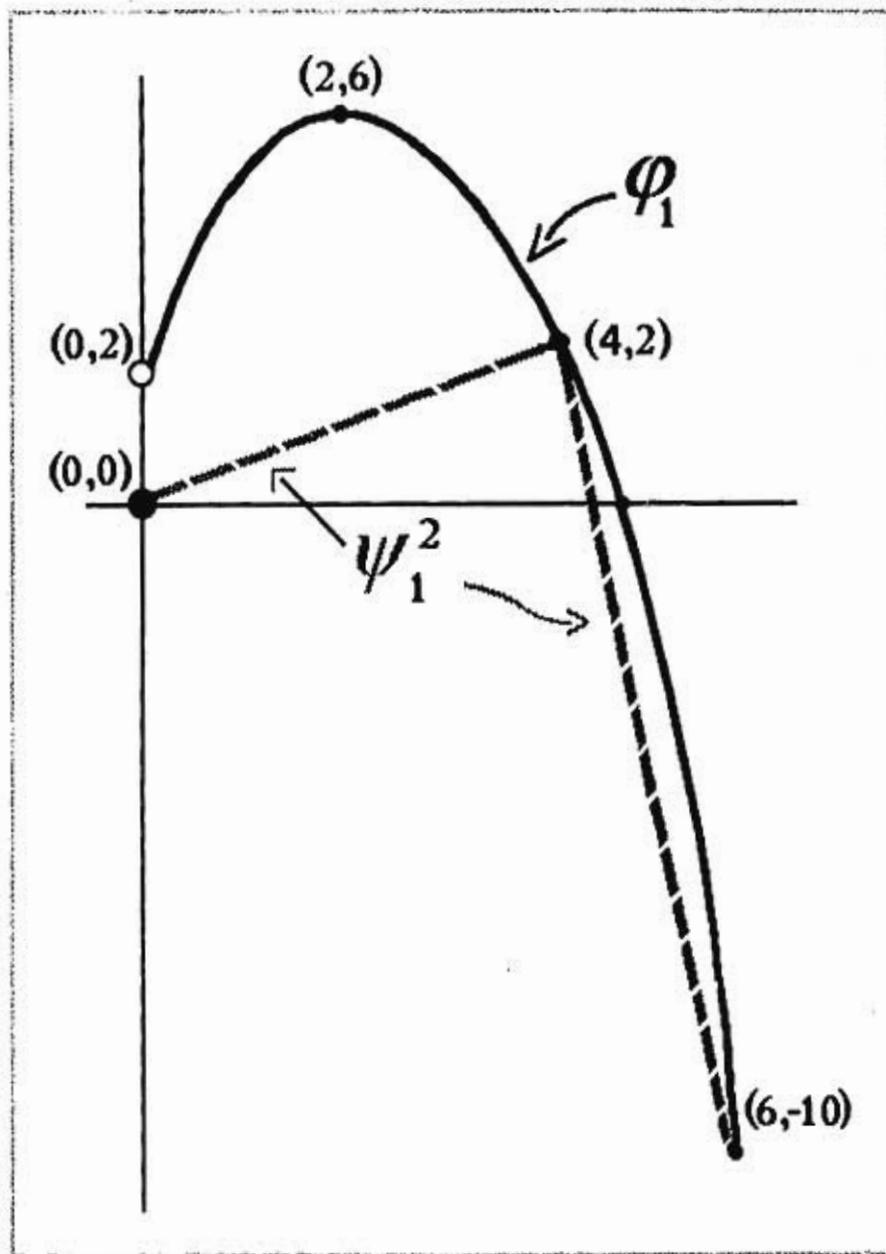


Figura 4.9. Cubierta convexa de φ_1 en la segunda iteración.

$$\begin{aligned}
 -X_1 + 3X_2 &\leq 5 \\
 2X_1 - X_2 &\leq 5 \\
 -2X_1 + X_2 &\leq 0 \\
 X_1 - 3X_2 &\leq 0 \\
 4 &\leq X_1 \leq 6 \\
 0 &\leq X_2 \leq 5 ;
 \end{aligned}$$

que arroja el valor óptimo -3.8 para $X = (4, 3)$

Así, se tienen las cotas

$$\begin{aligned}
 \mathfrak{B}(2) &= -3.8 , \\
 \mathfrak{B}(3) &= -3.8 .
 \end{aligned}$$

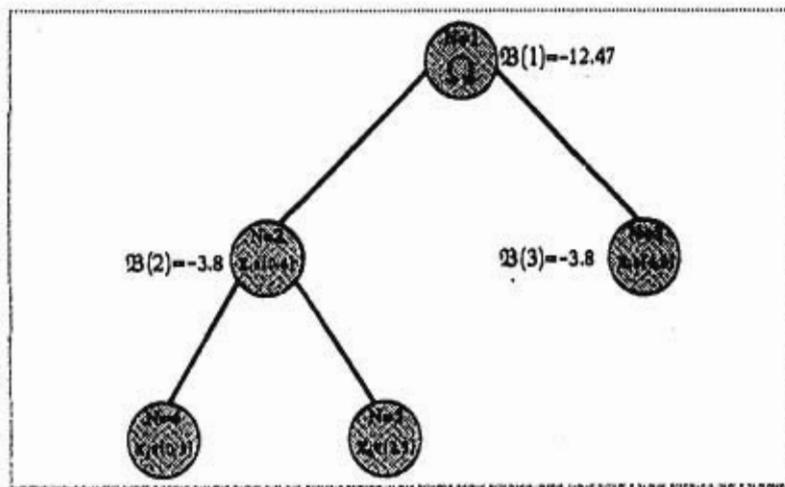


Figura 4.10. Árbol de búsqueda al final de la segunda iteración.

Paso 3. Como existe un empate, se decide (arbitrariamente) tomar $N = 2$, por lo que, en resumen, para $k = 2$ se tiene

$$\begin{aligned}
 X^k &= (4, 3) , \\
 \mu^k &= \mathfrak{B}(2) = -3.8 , \\
 \Phi(X^k) &= 3 .
 \end{aligned}$$

Dado que $\Phi(X^k) = 3 > -3.8 = \Psi(X^k)$, se hace $i^* = 2$.

Paso 4. Se tiene que

$$\begin{aligned}\varphi_1(X_1^k) - \psi_1(X_1^k) &= 2 - (0) = 2, \\ \varphi_2(X_2^k) - \psi_2(X_2^k) &= 1 - (-5.88) = 6.8;\end{aligned}$$

por lo que se decide ramificar por X_2 , partiéndose $[0, 5]$ en $[0, 3]$, $[3, 5]$.
El árbol de búsqueda es el que se muestra en la figura 4.10.

Iteración 3 ($k = 3$)

Paso 2. En ninguno de los dos casos $\mathfrak{S}(N)$ es vacío por lo que se pasa a 2.3 en cada nodo.

Para $N = 4$ se tiene

$$\begin{aligned}\psi_1^3(X_1) &= \frac{1}{2}X_1 \\ \psi_2^3(X_2) &= -\frac{1}{3}X_2 + 2.\end{aligned}$$

(La nueva ψ_2^k se muestra en la figura 4.11.)

Se resuelve el problema

$$\begin{aligned}\text{Min } & \frac{1}{2}X_1 - \frac{1}{3}X_2 + 2 \\ \text{sujeto a } & \\ & -X_1 + 3X_2 \leq 5 \\ & 2X_1 - X_2 \leq 5 \\ & -2X_1 + X_2 \leq 0 \\ & X_1 - 3X_2 \leq 0 \\ & 0 \leq X_1 \leq 4 \\ & 0 \leq X_2 \leq 3;\end{aligned}$$

obteniéndose el valor óptimo 1.83 para $\mathbf{X} = (1, 2)$.

Por otra parte, para $N = 5$ se tiene

$$\begin{aligned}\psi_1^3(X_1) &= \frac{1}{2}X_1 \\ \psi_2^3(X_2) &= -6X_2 + 19.\end{aligned}$$

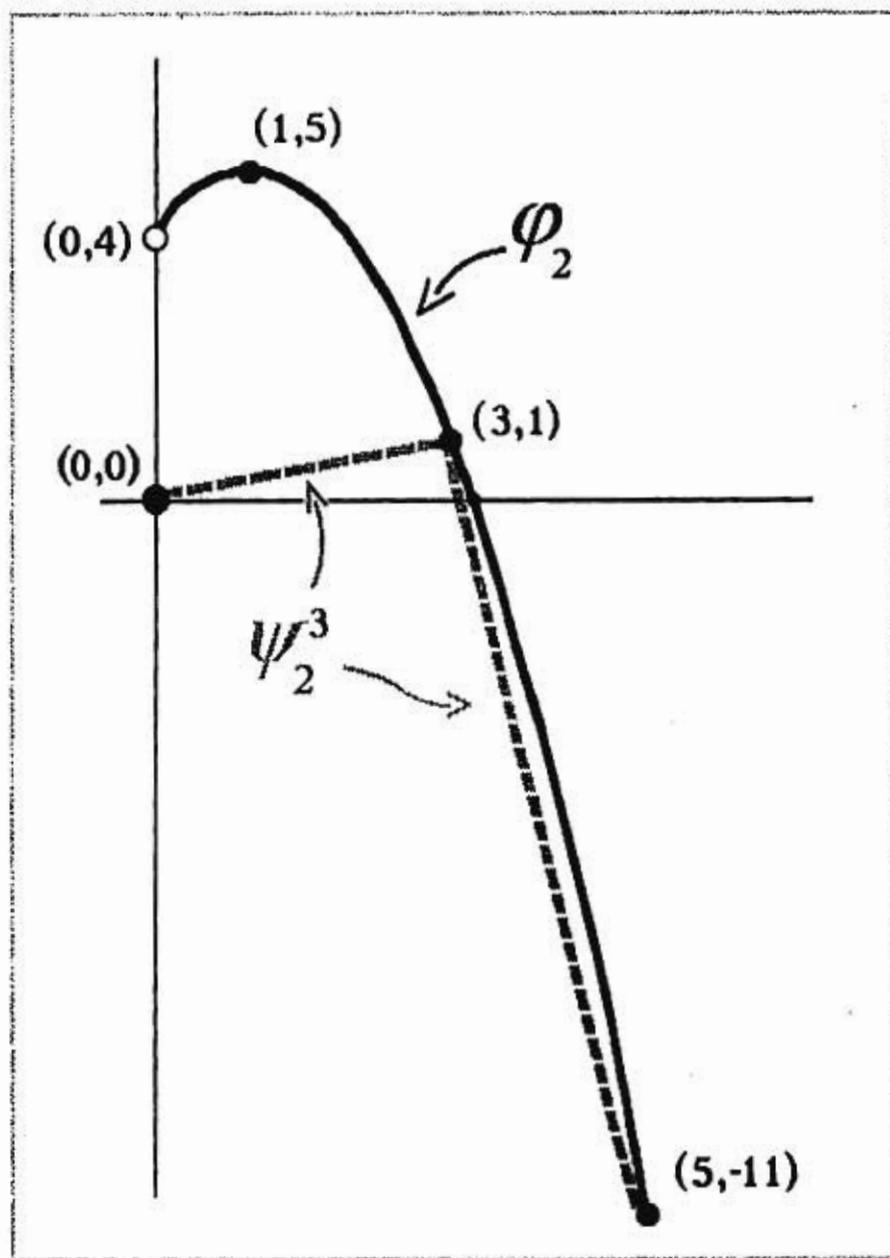


Figura 4.11. Cubierta convexa de φ_2 en la iteración 3.

$$\text{Min } \frac{1}{2}X_1 - 6X_2 + 19$$

sujeto a

$$-X_1 + 3X_2 \leq 5$$

$$2X_1 - X_2 \leq 5$$

$$-2X_1 + X_2 \leq 0$$

$$X_1 - 3X_2 \leq 0$$

$$0 \leq X_1 \leq 4$$

$$3 \leq X_2 \leq 5;$$

que arroja el valor óptimo 3 para $\mathbf{X} = (4, 3)$

Así, se tienen las cotas

$$\mathfrak{B}(4) = 1.83,$$

$$\mathfrak{B}(5) = 3.$$

Paso 3. El de cota menor o igual es $N = 3$, por lo que para $k = 3$ se tiene

$$\mathbf{X}^k = (4, 3),$$

$$\mu^k = \mathfrak{B}(3) = -3.8,$$

$$\Phi(\mathbf{X}^k) = 3.$$

Dado que $\Phi(\mathbf{X}^k) = 3 > -3.8 = \Psi(\mathbf{X}^k)$, se hace $i^* = 3$.

Paso 4. Se tiene que

$$\varphi_1(\mathbf{X}_1^k) - \psi_1(\mathbf{X}_1^k) = 2 - (2) = 0$$

$$\varphi_2(\mathbf{X}_2^k) - \psi_2(\mathbf{X}_2^k) = 1 - (-5.88) = 6.8;$$

por lo que se decide ramificar por \mathbf{X}_2 , partiéndose $[0, 5]$ en $[0, 3]$, $[3, 5]$.

El árbol de búsqueda es el que se muestra en la figura 4.12.

Iteración 4 ($k = 4$)

Paso 2. En ninguno de los dos casos $\Theta(N)$ es vacío por lo que se pasa a 2.3 en cada nodo.

Para $N = 6$ se tiene

$$\begin{aligned}\psi_1^4(X_1) &= -6X_1 + 26 \\ \psi_2^4(X_2) &= -\frac{1}{3}X_2 + 2.\end{aligned}$$

Se resuelve el problema

$$\begin{aligned}\text{Min } & -6X_1 - \frac{1}{3}X_2 + 28 \\ \text{sujeto a } & \\ & -X_1 + 3X_2 \leq 5 \\ & 2X_1 - X_2 \leq 5 \\ & -2X_1 + X_2 \leq 0 \\ & X_1 - 3X_2 \leq 0 \\ & 4 \leq X_1 \leq 6 \\ & 0 \leq X_2 \leq 3;\end{aligned}$$

obteniéndose el valor óptimo 3 para $X = (4, 3)$.

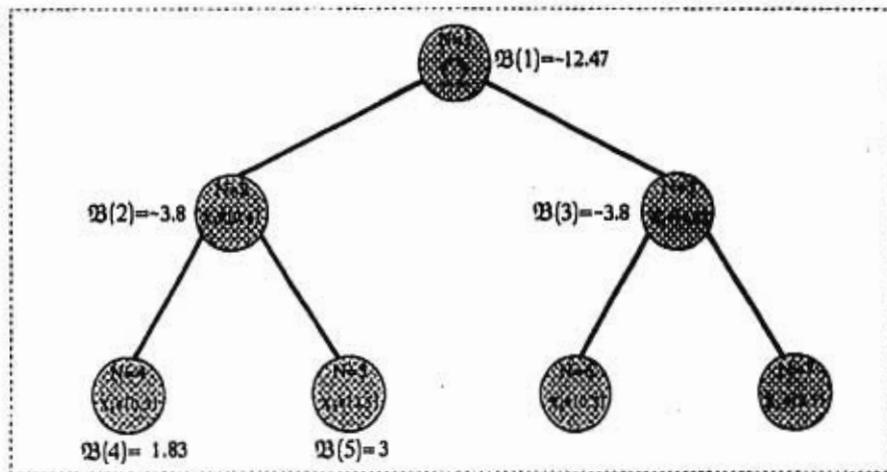


Figura 4.12. Árbol de búsqueda al final de la iteración 3.

Por otra parte, para $N = 7$ se tiene

$$\begin{aligned}\psi_1^4(X_1) &= -6X_1 + 26 \\ \psi_2^4(X_2) &= -6X_2 + 19.\end{aligned}$$

Se resuelve el problema

$$\text{Min } -6X_1 - 6X_2 + 45$$

sujeto a

$$-X_1 + 3X_2 \leq 5$$

$$2X_1 - X_2 \leq 5$$

$$-2X_1 + X_2 \leq 0$$

$$X_1 - 3X_2 \leq 0$$

$$4 \leq X_1 \leq 6$$

$$3 \leq X_2 \leq 5;$$

que arroja el valor óptimo 3 para $X = (4, 3)$.

Así, se tienen las cotas

$$\mathfrak{B}(6) = 3,$$

$$\mathfrak{B}(7) = 3.$$

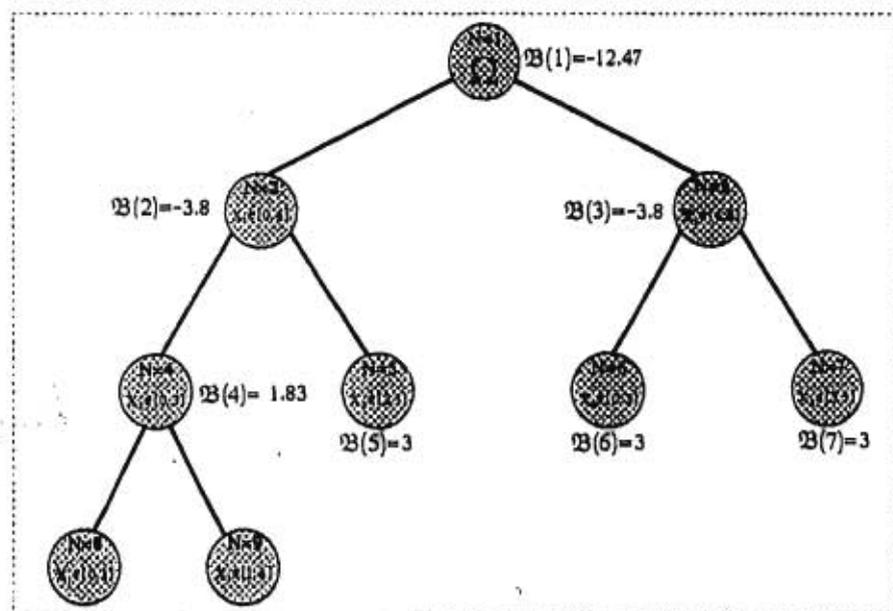


Figura 4.13. Árbol de búsqueda al final de la iteración 4.

Paso 3. El de cota menor o igual es $N = 4$, por lo que para $k = 4$ se tiene

$$\begin{aligned} \mathbf{X}^k &= (1, 2), \\ \mu^k &= \mathfrak{B}(4) = 1.83, \\ \Phi(\mathbf{X}^k) &= 9. \end{aligned}$$

Dado que $\Phi(\mathbf{X}^k) = 9 > 1.83 = \Psi(\mathbf{X}^k)$, se hace $i^* = 4$.

Paso 4. Se tiene que

$$\begin{aligned} \varphi_1(\mathbf{X}_1^k) - \psi_1(\mathbf{X}_1^k) &= 5 - (0.5) = 4.5 \\ \varphi_2(\mathbf{X}_2^k) - \psi_2(\mathbf{X}_2^k) &= 4 - (1.33) = 2.66; \end{aligned}$$

por lo que se decide ramificar por X_1 , partiéndose $[0, 4]$ en $[0, 1]$, $[1, 4]$.

El árbol de búsqueda es el que se muestra en la figura 4.13.

Iteración 5 ($k = 5$)

Paso 2. En ninguno de los dos casos $\mathfrak{S}(N)$ es vacío por lo que se pasa a 2.3 en cada nodo.

Para $N = 8$ se tiene

$$\begin{aligned} \psi_1^5(\mathbf{X}_1) &= 5X_1 \\ \psi_2^5(\mathbf{X}_2) &= -\frac{1}{3}X_2 + 2. \end{aligned}$$

(La nueva ψ_1^k se muestra en la figura 4.14.)

Se resuelve el problema

$$\begin{aligned} \text{Min } & 5X_1 - \frac{1}{3}X_2 + 2 \\ \text{sujeto a } & \\ & -X_1 + 3X_2 \leq 5 \\ & 2X_1 - X_2 \leq 5 \\ & -2X_1 + X_2 \leq 0 \\ & X_1 - 3X_2 \leq 0 \\ & 0 \leq X_1 \leq 1 \\ & 0 \leq X_2 \leq 3; \end{aligned}$$

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

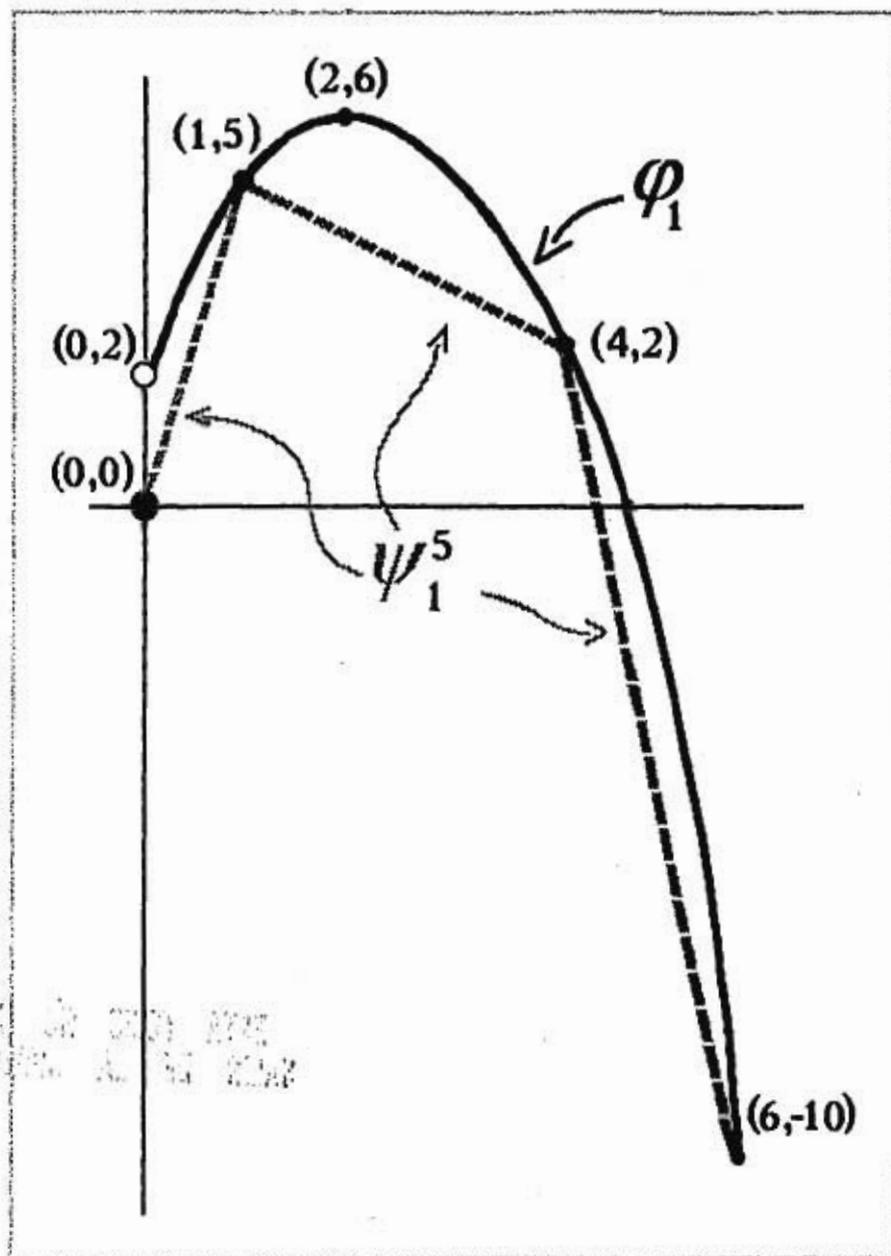


Figura 4.14. Cubierta convexa de φ_1 en la iteración 5.

obteniéndose el valor óptimo 2 para $\mathbf{X} = (0, 0)$.

Por otra parte, para $N = 9$ se tiene

$$\begin{aligned}\psi_1^5(\mathbf{X}_1) &= -\mathbf{X}_1 + 6 \\ \psi_2^5(\mathbf{X}_2) &= -\frac{1}{3}\mathbf{X}_2 + 2.\end{aligned}$$

Se resuelve el problema

$$\begin{aligned}\text{Min } & -\mathbf{X}_1 - \frac{1}{3}\mathbf{X}_2 + 8 \\ \text{sujeto a } & \\ & -\mathbf{X}_1 + 3\mathbf{X}_2 \leq 5 \\ & 2\mathbf{X}_1 - \mathbf{X}_2 \leq 5 \\ & -2\mathbf{X}_1 + \mathbf{X}_2 \leq 0 \\ & \mathbf{X}_1 - 3\mathbf{X}_2 \leq 0 \\ & 1 \leq \mathbf{X}_1 \leq 4 \\ & 0 \leq \mathbf{X}_2 \leq 3;\end{aligned}$$

que arroja el valor óptimo 3 para $\mathbf{X} = (4, 3)$

Así, se tienen las cotas

$$\begin{aligned}\mathfrak{B}(8) &= 0, \\ \mathfrak{B}(9) &= 3.\end{aligned}$$

Paso 3. El de cota menor o igual es $N = 8$, por lo que para $k = 5$ se tiene

$$\begin{aligned}\mathbf{X}^k &= (0, 0), \\ \mu^k &= \mathfrak{B}(8) = 2, \\ \Phi(\mathbf{X}^k) &= 2.\end{aligned}$$

Dado que $\Phi(\mathbf{X}^k) = 2 = \Psi(\mathbf{X}^k)$, se ha alcanzado el óptimo.
El árbol final de búsqueda es el que se ilustra en la figura 4.15.

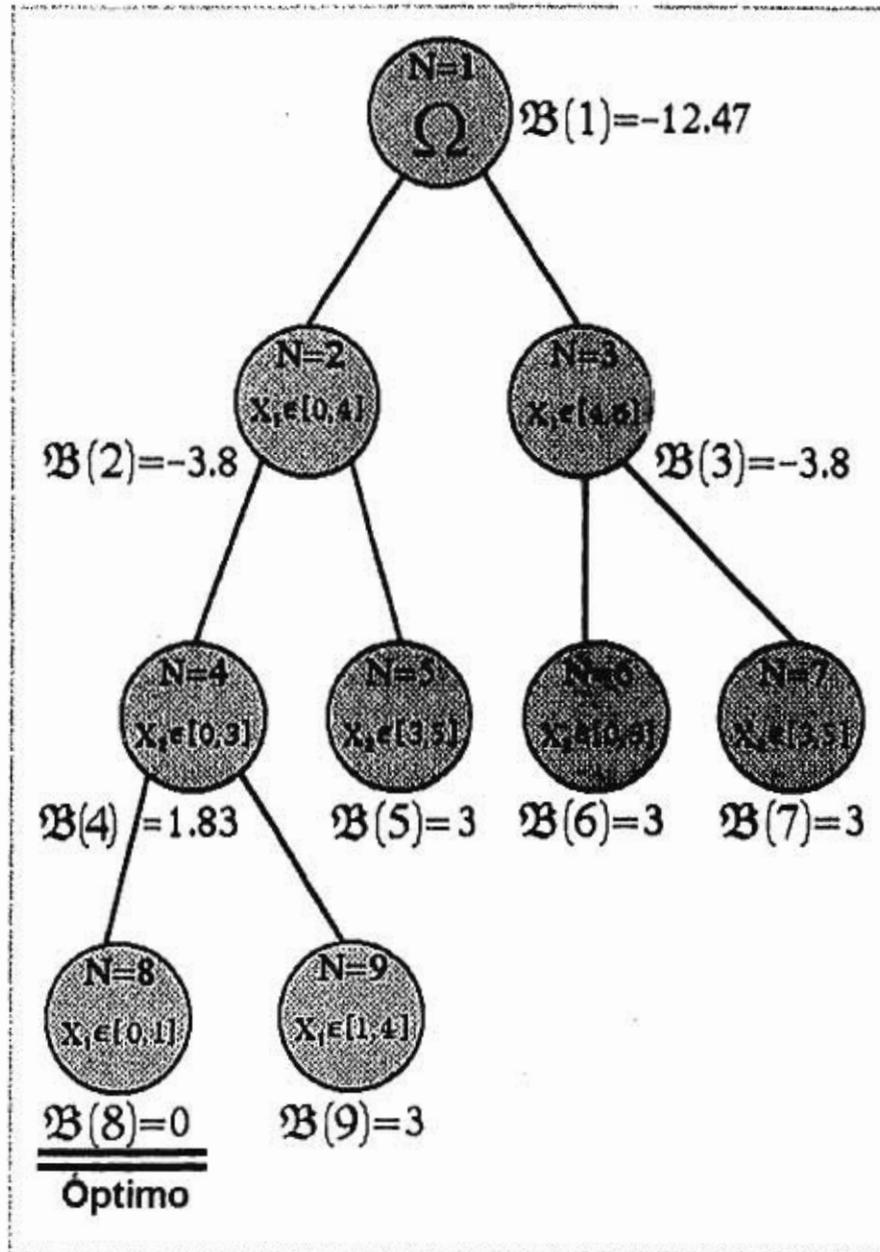


Figura 4.15. Árbol final de búsqueda.

El algoritmo también puede ser resumido con la siguiente tabla:

k	\mathbf{x}^k	$\mu^k = \psi(\mathbf{x}^k)$	$\Phi(\mathbf{x}^k)$
1	(4, 3)	-12.47	3
2	(4, 3)	-3.8	3
3	(4, 3)	-3.8	3
4	(1, 2)	1.83	9
5	(0, 0)	2	2

Capítulo 5

CONCLUSIONES

La ramificación y acotamiento es una herramienta que no ha sido aplicada, aún, en toda su capacidad, ya que ésta también es aplicable a problemas no combinatorios; sin embargo, existe una gran desventaja al resolver problemas que poseen un número infinito de soluciones con un algoritmo tipo R-A: la convergencia del algoritmo no es intuitiva (a diferencia de los problemas combinatorios); aunque para el caso del algoritmo presentado aquí, los resultados de Análisis Matemático y de Conjuntos Convexos, ayudan a esclarecer su funcionamiento.

La posibilidad de aplicar a una gran variedad de problemas, hacen muy atractivo el uso del algoritmo desarrollado en este trabajo. Por ejemplo, en problemas en los que la función objetivo no es continua. Para ilustrar esto, tómesese una clásica función definida por:

$$f(x) = \begin{cases} 0, & x = 0 \\ x + 1, & 0 < x \leq 1 \\ x + \frac{1}{2}, & 1 < x \leq 2 \end{cases}$$

Una función como esta es fácil de encontrar en problemas de minimización de costos; y en donde, el costo "es menor" conforme x crece.

El algoritmo presentado se puede modificar para resolver problemas enteros; por ejemplo, si se quiere resolver

$$\begin{aligned} & \text{Min } \sum C_i X_i \\ & \text{sujeto a} \\ & \quad X_i \in \mathbb{Z} \end{aligned}$$

$$l_i \leq X_i \leq L_i$$

donde cada C_i es una constante real; se construye la cubierta convexa de $C_i X_i$, la cual, es muy fácil de reconocer, es simplemente $C_i X_i$ con X_i variable real. (Esta última afirmación se relaciona con un algoritmo entero mixto propuesto por Beale y Small en [10].)

Otra fácil observación, es que la construcción de cubiertas convexas no es un procedimiento sencillo; sin embargo, para funciones suficientemente "suaves", el procedimiento de particionar C , hace que las cubiertas degeneren en rectas o en trozos de la misma función (utilizando el hecho de que la cubierta convexa de una función cóncava es la recta que une sus puntos extremos, y la de una función convexa, es ella misma). Gracias a este último hecho, el algoritmo podría ser modificado para iniciar siempre con problemas en los que ya se conocen las cubiertas convexas. Por ejemplo, para la cubierta convexa de $x^3 - x$ en $[-1, 1]$, se puede partir el dominio en $[-1, 0]$ y $[0, 1]$ y, dado que en el primero la función es cóncava, la cubiera convexa es la recta $y = 0$; mientras que en $[0, 1]$ la función es convexa y, por lo tanto, coincide con su cubierta (véase la figura 5.1).

Aunque en un principio podría parecer que el algoritmo tiene una gran desventaja en el hecho de que no se puede iterar un número infinito de veces, es claro que al menos se podrían tener soluciones tan cercanas al óptimo como se desee; esto es, se puede dar un criterio de paro del algoritmo pidiendo que si ocurre $\|X^{k+1} - X^k\| < \epsilon$, no se itere más.

Una pregunta que puede surgir es la siguiente, ¿qué pasa si en el teorema 22 sólo se pide que G sea cerrado?. La respuesta es que, aún aceptando convergencia infinita, el teorema no se cumple. Para observar lo anterior, se toma el siguiente ejemplo.

$$\begin{array}{l} \text{Min } \varphi_1(X_1) + \varphi_2(X_2) + \varphi_3(X_3) \\ \text{sujeto a} \\ \quad X_2^2 \leq 4X_1 \quad \} \quad : G \end{array}$$

$$l = (0, 0, -2), \text{ \& } L = (1, 2, 1)$$

donde

$$\varphi_1(X_1) = \begin{cases} 0 & \text{si } X_1 \leq 0 \\ 4 - X_1 & \text{si } X_1 > 0 \end{cases}$$

$$\varphi_2(X_2) = -X_2$$

$$\varphi_3(X_3) = -X_3^2.$$

Si se usa la regla de refinamiento débil, todas las ramificaciones se hacen en X_1 , llegando al punto limite $(0, 0, 0)$; sin embargo, la solución es $(0, 0, 1)$.

Finalmente, considero que la incorporación de algoritmos como el presentado aquí, a los cursos impartidos en la maestría de Investigación de Operaciones, en la División de Estudios de Posgrado de la Facultad de Ingeniería (UNAM), impulsarían el desarrollo de la ramificación y acotamiento en problemas no combinatorios.

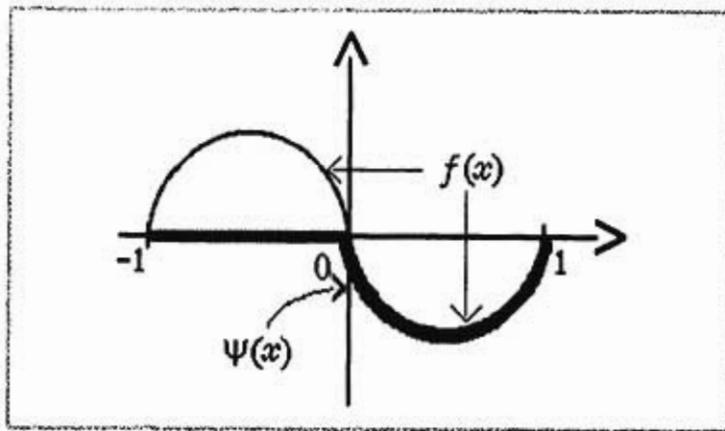


Figura 5.1. Cubierta convexa "a trozos".

Apéndice

Definición 23 $A \subseteq \mathbb{R}^p$ es **abierto** en \mathbb{R}^p (o simplemente **abierto**) si: para cada punto X en A existe un número real $\varepsilon > 0$ tal que $V_\varepsilon(X) \subseteq A$.

Definición 24 Dados A y B subconjuntos de \mathbb{R}^p , se dice que B es **abierto relativo a A** si: $\exists C$ abierto en \mathbb{R}^p tal que $B = A \cap C$.

Definición 25 $F \subseteq \mathbb{R}^p$ es **cerrado** si F^c es abierto.

Definición 26 $X \in \mathbb{R}^p$ es **punto de acumulación** de A si: para cada $\varepsilon > 0$, $\overset{\circ}{V}_\varepsilon(X) \cap A \neq \emptyset$. Se denota a los puntos de acumulación de un conjunto A por A' .

Teorema 27 F es cerrado $\iff F' \subseteq F$.

Definición 28 $A \subseteq \mathbb{R}^p$ es **acotado** si $\exists M$ tal que $V_M(0) \supseteq A$.

Definición 29 Se dice que un conjunto K es **compacto** si siempre que está contenido en la unión de una colección $\mathcal{G} = \{G_\alpha\}$ de conjuntos abiertos, también está contenido en la unión de algún número finito de conjuntos de \mathcal{G} .

Teorema 30 $K \subseteq \mathbb{R}^p$ es compacto $\iff K$ es cerrado y acotado.

Proposición 31 K_1, K_2 compactos $\implies K_1 \times K_2$ compacto.

Corolario 32 Dados n intervalos cerrados $\{[a_i, b_i]\}_{i=1}^n$, $\prod [a_i, b_i]$ es compacto.

Teorema 33 $A \subseteq K$ tales que A es infinito y K es compacto $\implies \exists X \in A'$ tal que $X \in K$.

Definición 34 Una sucesión en \mathbb{R}^p es una función cuyo dominio es \mathbb{N} y cuyo rango está contenido en \mathbb{R}^p . A una sucesión $\tau : \mathbb{N} \rightarrow \mathbb{R}^p$ también se le denota por $\tau = \{X^n\}_{n \in \mathbb{N}}$, donde $X^n = \tau(n)$.

Definición 35 Sea $\tau = \{X^n\}$ una sucesión en \mathbb{R}^p . Se dice que un elemento X de \mathbb{R}^p es *límite* de τ si, para cada vecindad V de X hay un número natural K tal que para toda $n \geq K$, X^n pertenece a V . Si X es límite de τ también se dice que τ **converge** a X . Si una sucesión tiene límite se dice que la sucesión es **convergente**. Si una sucesión no tiene límite se dice que es **divergente**. Lo anterior se suele denotar por $\text{Lim } X^n = X$.

Teorema 36 Una sucesión $\{X^n = (X_1^n, \dots, X_p^n)\}$ en \mathbb{R}^p converge a un elemento $X = (X_1, \dots, X_p) \iff$ las p sucesiones $\{X_i^n\}$ convergen a X_i respectivamente.

Definición 37 Sea $D \subseteq \mathbb{R}^p$, supóngase que para cada número natural n hay una función f^n con dominio D y rango en \mathbb{R}^q ; se dice que $\{f^n\}$ es una **sucesión de funciones** en D . Se debe comprender que para cualquier punto X en D , dicha sucesión de funciones da una sucesión de puntos en \mathbb{R}^q ; específicamente, la sucesión $\{f^n(X)\}$ que se obtiene calculando cada una de las funciones en X .

Definición 38 Sea $\{f^n\}$, con $f^n : D \rightarrow \mathbb{R}^q$, una sucesión de funciones ($D \subseteq \mathbb{R}^p$); y sea $f : D \rightarrow \mathbb{R}^q$. Se dice que la sucesión $\{f^n\}$ **converge** a f si para cada X en D , $\{f^n(X)\}$ converge a $f(X)$. En este caso, a la función f se le llama el **límite** de la sucesión. Cuando existe dicha función f se dice que la sucesión $\{f^n\}$ **converge a f** , o simplemente, que la sucesión es **convergente**. Lo anterior se suele denotar por $\text{Lim } f^n = f$.

Definición 39 Si $\tau = \{X^n\}$ es una sucesión en \mathbb{R}^p y si $r_1 < r_2 < \dots < r_n < \dots$ es una sucesión de números naturales estrictamente creciente, entonces la sucesión $\bar{\tau}$ en \mathbb{R}^p dada por $\bar{\tau} = \{X^{r_i}\}$ se llama una **subsucesión de τ** .

Teorema 40 $\tau = \{X^n\} \subseteq K \subseteq \mathbb{R}^p$ & K es compacto $\implies \tau$ tiene una subsucesión convergente a un punto en K .

Corolario 41 $\tau = \{X^n\} \subseteq \mathbb{R}^p$ & $X^* \in \tau'$ (i.e. es un punto de acumulación)
 $\implies \exists \tau$ subsucesión de τ tal que $\text{Lim } X^n = X^*$.

Teorema 42 $\{f^n\}$ sucesión de funciones crecientes y acotadas $\implies \exists f^*$ tal
 que $\text{Lim } f^n = f^*$.

Definición 43 Dada $f : D \rightarrow \mathbb{R}$, con $D \subseteq \mathbb{R}$, se dice que es *semi-continua por debajo* en X_0 si $f(X_0) < \infty$ & $\forall \varepsilon > 0, \exists \delta > 0$ tal que
 $|X - X_0| < \delta \implies f(X) > f(X_0) - \varepsilon$.

Definición 44 Dada $f : D \rightarrow \mathbb{R}^q$, con $D \subseteq \mathbb{R}^p$, se dice que es *continua*
 en X_0 si: $\forall \varepsilon > 0, \exists \delta > 0$ tal que $\|X - X_0\| < \delta \implies \|f(X) - f(X_0)\| \leq \varepsilon$.

Proposición 45 $f : D \rightarrow \mathbb{R}$, con $D \subseteq \mathbb{R}$, entonces: f continua en
 $X_0 \implies f$ semicontinua por debajo en X_0 .

Definición 46 Dada $f : D \rightarrow \mathbb{R}^q$, con $D \subseteq \mathbb{R}^p$, se dice que es *uniformemente continua*
 en D si: $\forall \varepsilon > 0, \exists \delta_\varepsilon > 0$ tal que $\|X_1 - X_2\| < \delta \implies \|f(X_1) - f(X_2)\| \leq \varepsilon$.

Proposición 47 $f : D \rightarrow \mathbb{R}^q$, con $D \subseteq \mathbb{R}^p$, entonces: f uniformemente
 continua $\implies f$ continua en todo $X_0 \in D$.

Teorema 48 $f : K \rightarrow \mathbb{R}$ y K compacto, con $K \subseteq \mathbb{R}^p$, entonces f alcanza
 un mínimo en K y un máximo en K .

BIBLIOGRAFÍA

- [1]. LAWLER, E. L. & WOOD, D. E., "Branch-and Bound Methods: A Survey", *Operations Research* **14**, 699-719 (1966).
- [2]. AGIN, N., "Optimum Seeking with Branch and Bound", *Operations Research* **13**, B-176-B-185 (1966).
- [3]. BALAS, E., "A Note on the Branch and Bound Principle", *Operations Research* **16**, 442-445, (1968).
- [4]. MITTEN, L. G. , " Branch and Bound Methods: General Formulation and Properties " , *Operations Research*, 24-35, (1969).
- [5]. LARNDER, H., "The Origen of Operational Research", *Operations Research* **32**, 465-475 (1966).
- [6]. BERTIER, P. & ROY, B., "Procédure de Résolution pour une Classe de Problèmes pouvant avoir un Caractère Combinatoire", *Cahiers du Centre D'Études de Recherche Opérationnelle* **6**, 202-208 (1964).
- [7]. JONES, A. P. & SOLAND, R. M. , " A Branch-and-Bound Algorithm for Multilevel-Fixed Cost Problems " , RAC-TP-285, Research Analysis Corporation, McLean, Va. (1967).
- [8]. BEALE, E. M. L., "Two Transportation Problems", *Proceedings of the 3rd International Conference on Operational Research*, Kreweras, G. and Morlot, G., 780-788, Dunod, Paris.
- [9]. FALK, J. & SOLAND, R. M., "An Algorithm for Separable Nonconvex Programming Problems", *Management Science* **15**, (1969).

- [10]. BEALE, E. M. L. & SMALL, R. E., "Mixed Integer Programming by Branch and Bound Technique", Information Processing, Kalenick, W. A., Vol.2, 450-451, Spartan Books, Washington (1965).
- [11]. KOLESAR, P. J. , "A Branch an Bound Algorithm for the Knapsack Problem", *Management Science* **13**, 723-735 (1967).
- [12]. SHAPIRO, G. F., "Dynamic Programming Algorithms for the Integer Programming Problem I; The Integer Programming Problem Viewed as a Knapsack Type Problem", *Operations Research* **16**, 103-121 (1968).
- [13]. LITTLE, J., MURTY, K., SWEENEY, D. & KAREL, C., "An Algorithm for the Traveling Salesman Problem", *Operations Research* **34**, 698-717(1963).
- [14]. FLORES, I., "Técnicas de Ramificación y Acotamiento: Análisis, Metodología y Aplicaciones", Tesis de Maestría UNAM, (1990).
- [15]. CHRISTOFIDES, N., "Combinatorial Optimization", John Wiley & Sons, Ltd., (1979).
- [16]. GUTIÉRREZ, M. A., et al, "Complejidad Computacional de Algoritmos", UAM-Azcapotzalco, 1ª Edición, (1993).
- [17]. PRIETO, C., "Apuntes para la Clase de Conjuntos Convexos", Facultad de Ciencias, UNAM, (1972).
- [18]. ROCKAFELLAR, R. T., "Convex Analysis", Princeton University Press, 2ª Edición, (1970).
- [19]. RUDIN, W., "Principles of Mathematical Analysis", McGraw-Hill, 2ª Edición, (1964).
- [20]. BARTLE, R. G., "Introducción al Análisis Matemático", Limusa, 1ª Edición, (1980).

- [21]. HEWITT, E. & STROMBERG, K., "Real and abstract Analysis : a Modern Treatment of the Theory of Functions of a Real Variable", Springer Verlag, Berlin, (1965).
- [22]. CAMERON, N., "Introduction to Linear and Convex Programming", Cambridge University Press, (1985), Great Britain.