



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

**FACULTAD DE CIENCIAS**

**SOFTWARE INTERACTIVO AUXILIAR  
DE PROGRAMACIÓN LINEAL Y  
REDES**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE:**

**A C T U A R I A**

**P R E S E N T A:**

**ASTRID MINERVA DUQUE MARTÍNEZ**



**FACULTAD DE CIENCIAS  
UNAM**

**Tutor: MAT. ESTEBAN RUBÉN HURTADO CRUZ**

**2006**



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.

## Hoja de Datos del Jurado

1. Duque  
Martínez  
Astrid Minerva  
56 64 83 83  
Universidad Nacional Autónoma de México  
Facultad de Ciencias  
Actuaría  
09909747-7
  
2. Mat.  
Esteban Rubén  
Hurtado  
Cruz
  
3. Mat.  
Adrián  
Girard  
Islas
  
4. Dr.  
Carlos  
Hernández  
Garcíadiego
  
5. Fís. Mat.  
Héctor de Jesús  
Argueta  
Villamar
  
6. Mat.  
María Juana  
Linares  
Altamirano
  
7. Software interactivo auxiliar de programación lineal y redes  
57 p  
2006

Dedicado a mis padres

***Minerva y Humberto***

y a mis hermanas

***Itzel y Diana***

# ÍNDICE

<b>1. JUSTIFICACIÓN.....</b>	<b>1</b>
<b>2. EJECUCIÓN DEL PROGRAMA</b>	
2.1. Requerimientos de Hardware y Software.....	2
2.2. Ejecución del programa y de las Fuente.....	2
<b>3. DESCRIPCIÓN DEL PROGRAMA</b>	
3.1 Navegación dentro del programa.....	3
3.2 Interactividad con el usuario.....	4
3.2.1 Áreas activas.....	5
3.2.2 Arrastre de objetos.....	6
3.2.3 Botones y flechas de acción.....	7
3.2.4 Selección de flechas.....	8
3.2.5 Áreas de texto.....	9
3.3 Estructura temática del programa.....	10
3.3.1 Programación lineal.....	10
3.3.2 Redes.....	11
<b>4. NAVEGACIÓN DENTRO DEL PROGRAMA</b>	
4.1 Pantalla de presentación.....	12
4.1.1 Botón: Antes de comenzar de la Pantalla de Presentación.....	13
4.2 Selección del botón: Programación Lineal.....	14
4.2.1 Navegando por el menú: Método Simplex.....	15
4.2.2 Áreas activas en el menú Método Simplex.....	16
4.2.3 Botón: Propiedades en el menú Método Simplex.....	17
4.2.4 Pantalla Propiedades.....	18
4.2.5 Botón: Menú Anterior de la pantalla Propiedades.....	19
4.2.6 Botón: Transición de lo geométrico a lo algebraico en el menú Método Simplex.....	20
4.2.7 Pantalla Transición de lo geométrico a lo algebraico.....	21
4.2.8 Botón: Naturaleza Iterativa de la Pantalla Transición de lo geométrico a lo algebraico.....	22
4.2.9 Pantalla Naturaleza iterativa del Método Simplex.....	23
4.2.10 Botón: Menú Anterior de la Pantalla Naturaleza Iterativa del Método Simplex.....	24
4.2.11 Botón: Menú Anterior de la Pantalla Transición de lo geométrico a lo algebraico.....	25
4.2.12 Botón: Método simplex del Menú Método simplex.....	26
4.2.13 Pantalla Procedimiento del Método Simplex.....	27
4.2.14 Botón: Menú Anterior de la Pantalla Procedimiento del Método Simplex.....	28

4.2.15 Botón: Otra solución inicial del menú Método Simplex	29
4.2.16 Pantalla Otra solución inicial.....	30
4.2.17 Botón: Método de dos fases de la pantalla Otra solución inicial.....	31
4.2.18 Pantalla Método de dos fases.....	32
4.2.19 Botón: Ejemplo de la pantalla Método de dos fases.....	33
4.2.20 Pantalla Ejemplo del Método de dos fases.....	34
4.2.21 Botón: Menú Anterior de la Pantalla Ejemplo del Método de dos fases.....	35
4.2.22 Botón: Menú Anterior de la Pantalla Método de dos fases.....	36
4.2.23 Botón: Menú Anterior de la Pantalla Método de dos fases.....	37
4.2.24 Botón: Casos especiales del Método Simplex.....	38
4.2.25 Botón: Óptimos alternativos de la pantalla Casos Especiales.....	39
4.2.26 Pantalla Óptimos alternativos.....	40
4.2.27 Botón: Menú Anterior de la pantalla Óptimos Alternativos.....	41
4.2.28 Botón: Menú Anterior de la pantalla Casos especiales	42
4.3 Selección del botón: Redes.....	43
4.3.1 Botón: Terminología del menú Principal de Redes.....	44
4.3.2 Pantalla Terminología.....	45
4.3.3 Botón: Menú Anterior de la Pantalla Terminología.....	46
4.3.4 Botón Árbol de peso mínimo del menú Principal de Redes.....	47
4.3.5 Pantalla Árbol de peso mínimo.....	48
4.3.6 Botón: Algoritmo de Kruskal del menú Árbol de peso mínimo.....	49
4.3.7 Pantalla Algoritmo de Kruskal.....	50
4.3.8 Botón: Menú Anterior del Algoritmo de Kruskal.....	51
4.3.9 Botón: Algoritmo de Primm de la pantalla Árbol de peso mínimo.....	52
4.3.10 Pantalla Algoritmo de Primm.....	53
4.3.11 Botón Menú Anterior de la Pantalla Algoritmo de Primm.....	54
4.3.12 Botón Ejercicios del Menú Árbol de peso mínimo.....	55
4.3.13 Pantalla Ejercicios.....	56
<b>5. BIBLIOGRAFÍA.....</b>	<b>57</b>

## 1. JUSTIFICACIÓN

En el taller de matemáticas existe el proyecto de crear material que apoye el proceso enseñanza-aprendizaje a través de propuestas como la que presento. Anteriormente se habían hecho trabajos de este tipo enfocados al cálculo integral y diferencial, ahora es el turno de la programación lineal y redes.

Considero que es importante emplear los programas computacionales existentes para crear propuestas alternativas en la enseñanza, ya que estas ofrecen al estudiante un mejor panorama de los temas. En este trabajo, *Software interactivo auxiliar de programación lineal y redes*, utilizó los programas: Authorware, The Geometer's Sketchpad y Flash.

Este material lo hice con el propósito de apoyar a los estudiantes de las asignaturas de programación lineal, investigación de operaciones, análisis de redes, etc.

Mi propuesta es un material didáctico que abarca los fundamentos de la programación lineal y redes. El software incluye conceptos, definiciones, procedimientos, propiedades, ejercicios y ejemplos, varios de los cuales fueron elaborados por mí.

Al presentar los diferentes temas en este material didáctico lo hago con la intención de que resulten visualmente atractivos. Un software manejado en forma autodidacta, donde los estudiantes puedan interactuar con lo que se presenta en cada pantalla, con el propósito de apoyar el aprendizaje de los temas.

## **2. EJECUCIÓN DEL PROGRAMA**

### **2.1. Requerimientos de Hardware y Software**

- *Hardware*  
15 MB de RAM
- *Sistema operativo*  
Windows 98, 2000 o superior
- *Adaptador y monitor compatible con uno de los siguientes:*  
SVGA, Plug and Play

Para una mejor apreciación del programa se recomienda una resolución de la pantalla de 1024x768 píxeles o mayores. Ya que en resoluciones menores el programa no se ve completo.

### **2.2. Ejecución del Programa y de las Fuentes**

- Insertar el CD del programa
- Ejecutar el programa TESIS.exe (No es necesaria ninguna instalación en disco duro)
- Si se quiere instalar el programa, copiar todo el contenido del CD en una carpeta en el disco duro.
- Si al ejecutar el programa las fuentes (funciones matemáticas) no se ven bien), instalar las fuentes Math Type ejecutando el archivo Wintruetype que se encuentra en el CD del programa.

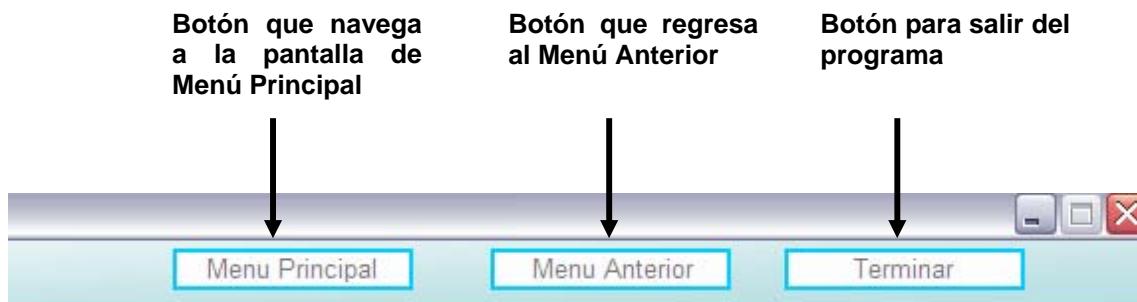


### 3. DESCRIPCIÓN DEL PROGRAMA

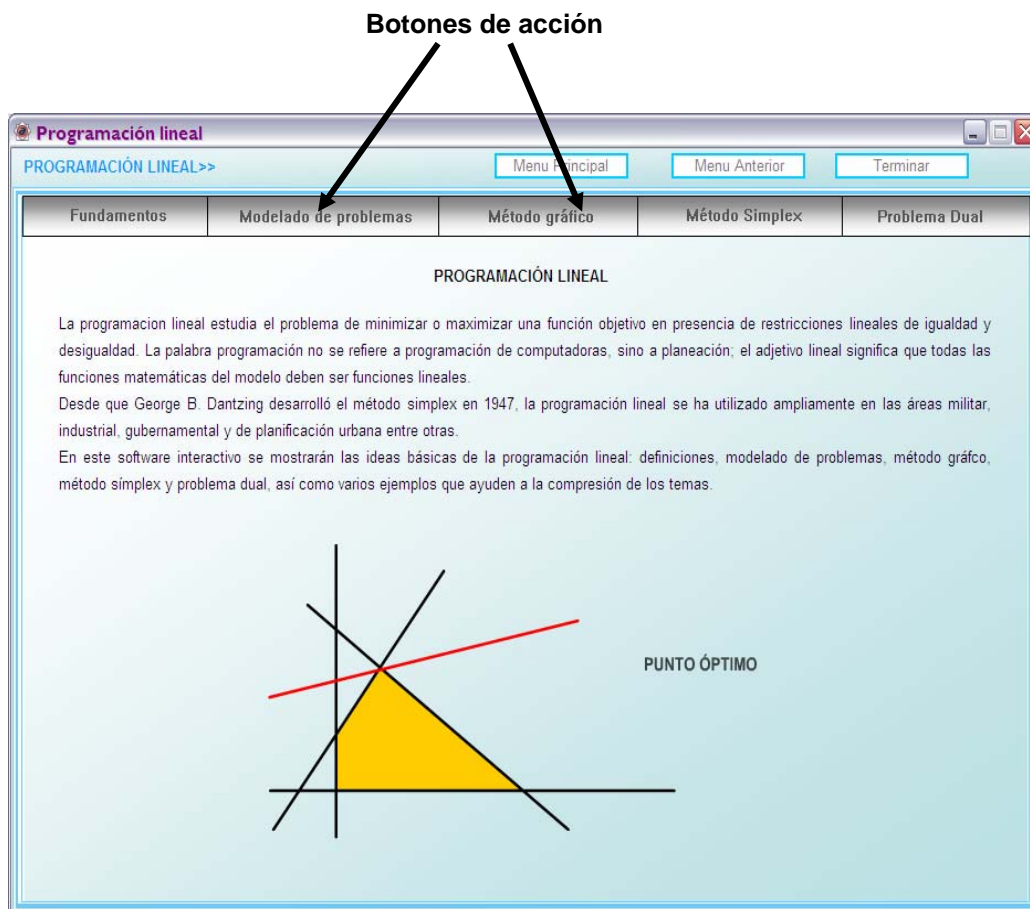
#### 3.1 Navegación dentro del programa

La navegación dentro del programa se lleva a cabo mediante botones de acción.

En todas las pantallas aparecen estos tres botones:



En algunas pantallas aparece una barra-menú de color gris, en la que cada botón indica una opción diferente de navegación.



## 3.2 Interactividad con el usuario

En cada pantalla aparecen las instrucciones de cada interacción en color rojo.

**Instrucciones**

DEFINICIONES

Solución factible:      Solución no factible:      Región factible:      Solución óptima:

*Haz clic sobre cada concepto para ver su definición*

Con el siguiente problema mencionaremos algunas definiciones:

Maximizar  $2x_1 + x_2$   
sujeta a  $-x_1 + x_2 \leq 1$   
 $\frac{2}{5}x_1 + x_2 \leq 3$   
 $x_1, x_2 \geq 0$

*Mueve el mouse sobre la región del cuadrado*

### 3.2.1 Áreas activas

En algunas pantallas aparecen palabras de color azul marino, al situar el puntero del ratón sobre ellas, se muestra información adicional.

#### Pantalla Relaciones entre el primal y dual

Software interactivo auxiliar de programación lineal y redes

PROBLEMA DUAL>> Menu Principal Menu Anterior Terminar

Relaciones entre el primal y el dual

La definición que se ha elegido para el problema dual lleva a muchas relaciones importantes entre los problemas lineales primal y dual. Las siguientes propiedades se cumplen sin importar cuál de los dos problemas se etiqueta como problema primal (mientras la dirección de las desigualdades de los problemas se expresen según la pantalla anterior). *Coloca el mouse sobre cada propiedad para ver su contenido.*

Propiedad de dualidad débil      Propiedad de dualidad fuerte      Propiedad de soluciones complementarias

Propiedad de soluciones complementarias óptimas      **Teorema fundamental de dualidad**

Ejemplo

PRIMAL	Minimizar	$Z = 2x_1 + 5x_2$	DUAL	Maximizar	$F = y_1 + 2y_2$
	sujeta a	$-x_1 + 3x_2 \geq 1$		sujeta a	$-y_1 + y_2 \leq 2$
		$x_1 + x_2 \geq 2$			$3y_1 + y_2 \leq 5$
		$x_1 \geq 0, x_2 \geq 0$			$y_1 \geq 0, y_2 \geq 0$

Al colocar el mouse sobre las palabras en azul marino 'Teorema fundamental de dualidad' se muestra el teorema y un ejemplo en la parte inferior de la pantalla

Software interactivo auxiliar de programación lineal y redes

PROBLEMA DUAL>> Menu Principal Menu Anterior Terminar

Relaciones entre el primal y el dual

La definición que se ha elegido para el problema dual lleva a muchas relaciones importantes entre los problemas lineales primal y dual. Las siguientes propiedades se cumplen sin importar cuál de los dos problemas se etiqueta como problema primal (mientras la dirección de las desigualdades de los problemas se expresen según la pantalla anterior). *Coloca el mouse sobre cada propiedad para ver su contenido.*

Propiedad de dualidad débil      Propiedad de dualidad fuerte      Propiedad de soluciones complementarias

Propiedad de soluciones complementarias óptimas      **Teorema fundamental de dualidad**

Con respecto a los problemas de programación lineal primal y dual, exactamente una de las siguientes proposiciones es verdadera:

1. Ambos problemas tienen soluciones óptimas  $x^*$  y  $y^*$ , con  $cx^* = y^*b$ .
2. Uno de los problemas tiene un valor objetivo no acotado, en cuyo caso el otro problema debe ser no factible.
3. Ambos problemas son no factibles.

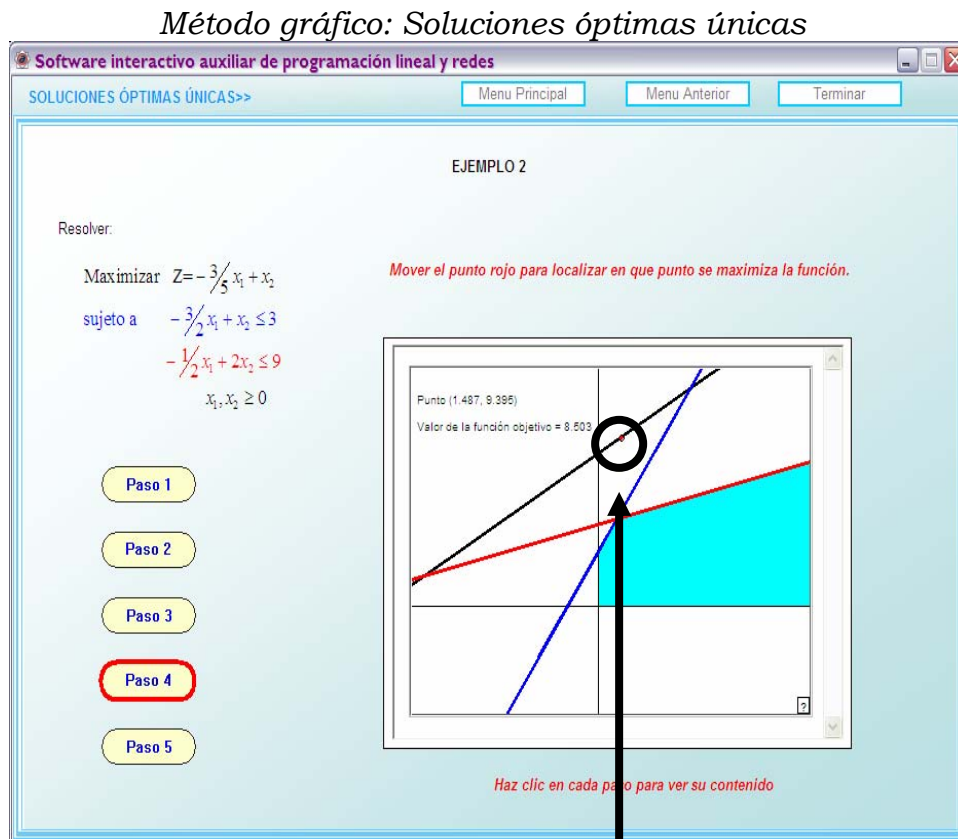
Ejemplo

PRIMAL	Minimizar	$Z = 2x_1 + 5x_2$	DUAL	Maximizar	$F = y_1 + 2y_2$
	sujeta a	$-x_1 + 3x_2 \geq 1$		sujeta a	$-y_1 + y_2 \leq 2$
		$x_1 + x_2 \geq 2$			$3y_1 + y_2 \leq 5$
		$x_1 \geq 0, x_2 \geq 0$			$y_1 \geq 0, y_2 \geq 0$

Con respecto al teorema fundamental de dualidad, en este ejemplo, la primera proposición es la que se cumple, ambos problemas tienen soluciones óptimas y los valores de las funciones objetivas son los mismos.

### 3.2.2 Arrastre de objetos

En algunas gráficas existe la posibilidad de arrastrar un punto rojo que facilita visualizar la posible solución de un problema.



Al arrastrar el punto rojo se muestran sus coordenadas, así como el valor de la función objetivo evaluado en él.

### 3.2.3 Botones y flechas de acción

En el programa existen botones de flecha, botones con fondo amarillo y flechas azules que al presionarlos despliegan información.

#### Pantalla Transición de lo geométrico a lo algebraico

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menu Principal Menu Anterior Terminar

TRANSICIÓN DE LO GEOMÉTRICO A LO ALGEBRAICO

Las ideas contenidas en el método gráfico son la base para desarrollar el método simplex. A continuación se muestra un esquema de como trabajan los métodos. *Haz clic en las flechas azules. Al colocar el mouse sobre algunas flechas aparecerá información adicional en la parte inferior.*

MÉTODO GRÁFICO (MG)	MÉTODO SIMPLEX (MS)
Grafica todas las restricciones, incluyendo las de no negatividad	
El espacio de soluciones consiste en una infinidad de puntos <b>factibles</b>	
Identifica <b>puntos esquina (o extremos) factibles</b> del espacio de soluciones	
Los candidatos a la solución óptima corresponden a una cantidad finita de <b>puntos esquina (o extremos)</b>	
Se usa la función objetivo para determinar el <b>punto esquina óptimo</b> entre todos los candidatos	

A continuación se desarrollan las ideas anteriores con más detalle:

Motivación Geométrica Naturaleza Iterativa

Al hacer clic sobre la flecha azul se muestra más información

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menu Principal Menu Anterior Terminar

TRANSICIÓN DE LA SOLUCIÓN GRÁFICA A LA SOLUCIÓN ALGEBRAICA (SIMPLEX)

Las ideas contenidas en el método gráfico son la base para desarrollar el método simplex. A continuación se muestra un esquema de como trabajan los métodos. *Haz clic en las flechas azules. Al colocar el mouse sobre algunas flechas aparecerá información adicional en la parte inferior.*

MÉTODO GRÁFICO (MG)	MÉTODO SIMPLEX (MS)
Grafica todas las restricciones, incluyendo las de no negatividad	Representa el espacio de soluciones con $m$ ecuaciones con $n$ variables, y restringe a todas las variables a valores no negativos; $m < n$ .
El espacio de soluciones consiste en una infinidad de puntos <b>factibles</b>	
Identifica <b>puntos esquina (o extremos) factibles</b> del espacio de soluciones	
Los candidatos a la solución óptima corresponden a una cantidad finita de <b>puntos esquina (o extremos)</b>	
Se usa la función objetivo para determinar el <b>punto esquina óptimo</b> entre todos los candidatos	

A continuación se desarrollan las ideas anteriores con más detalle:

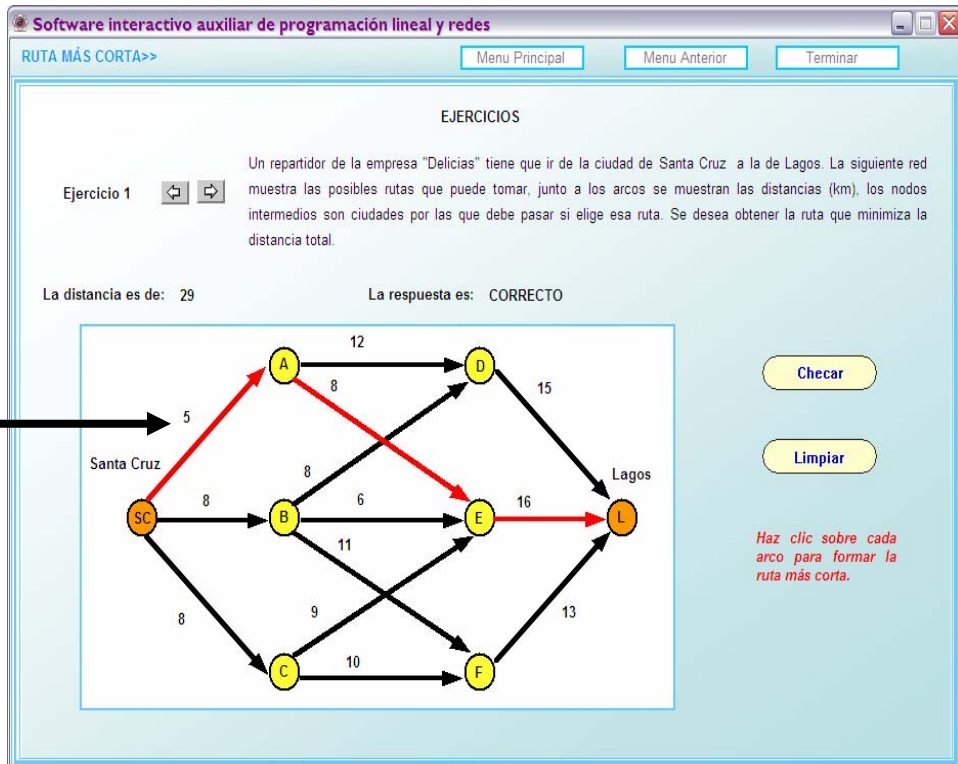
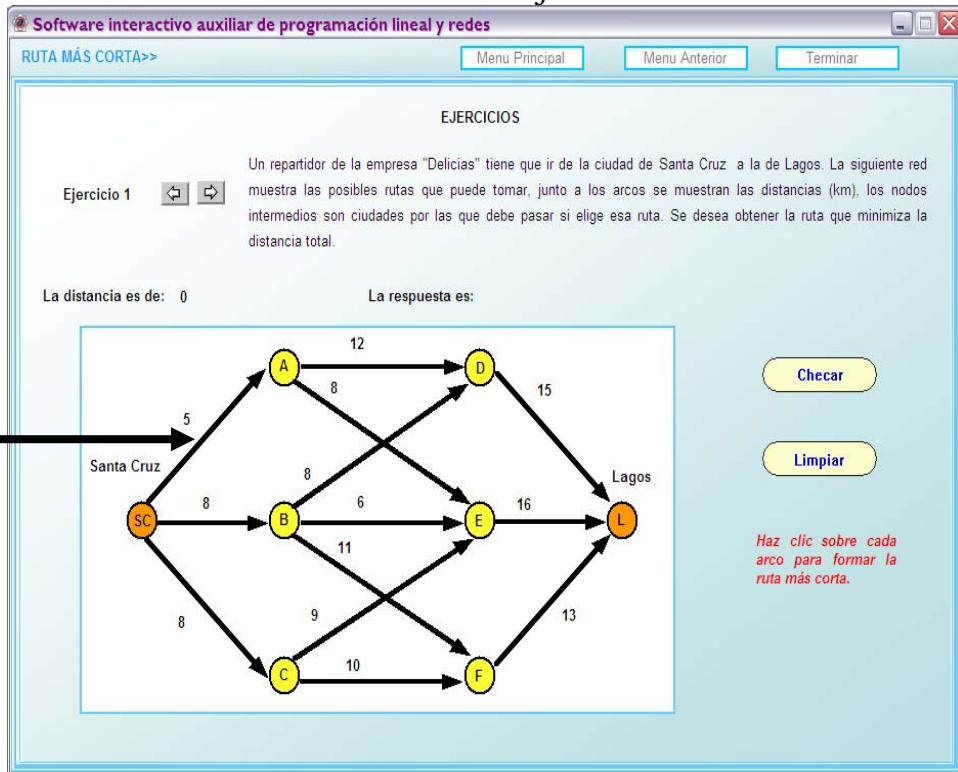
Motivación geométrica Naturaleza iterativa

### 3.2.4 Selección de flechas

En la parte de ejercicios de árbol de peso mínimo y ruta más corta de Redes existe la posibilidad de que el usuario seleccione las flechas con los valores que él considere convenientes para construir la solución de los ejercicios.

#### Ruta más corta: Ejercicio 1

Al ir seleccionando las flechas se va construyendo la solución del ejercicio

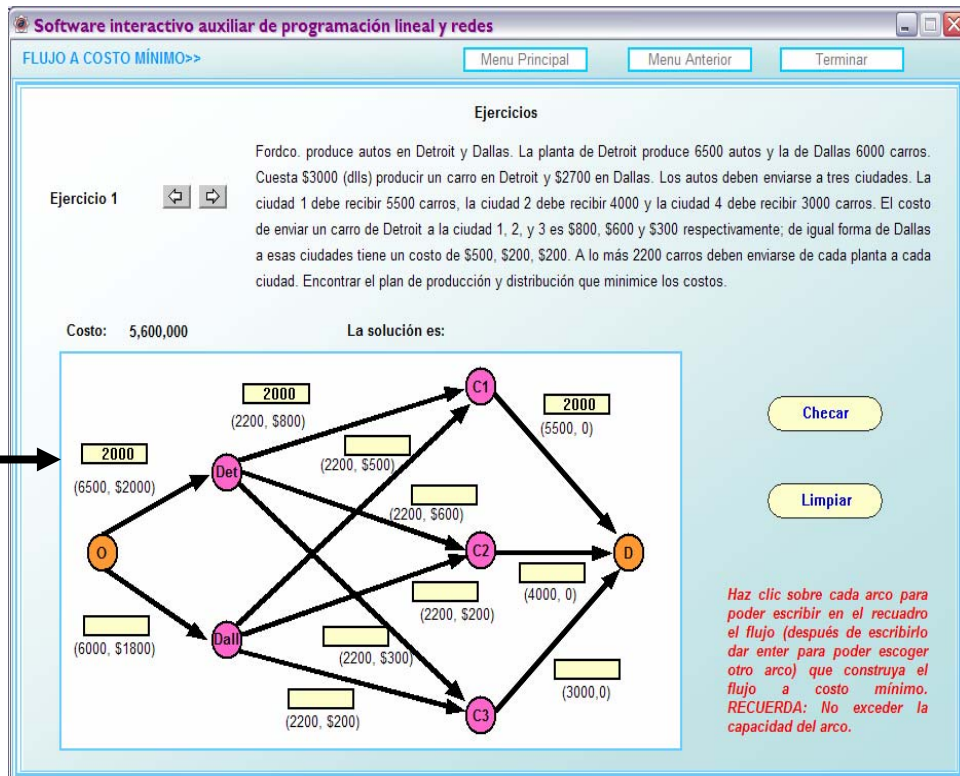
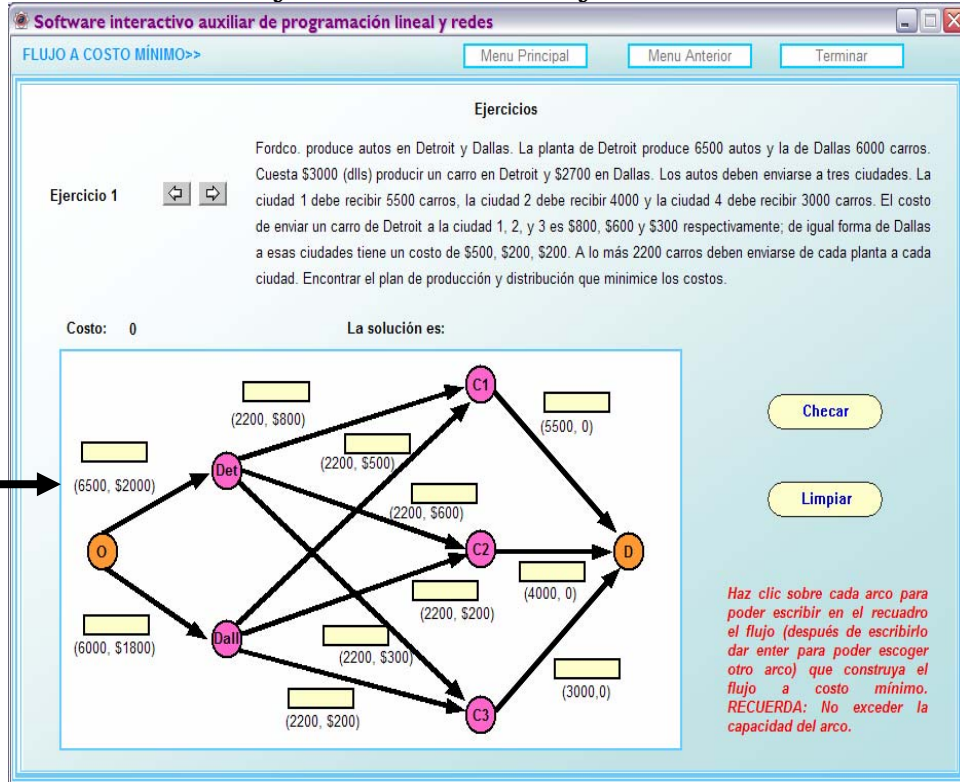


### 3.2.5 Áreas de texto

En la parte de ejercicios de flujo máximo y flujo a costo mínimo de Redes existe la posibilidad de que el usuario asigne valores que él considere convenientes para construir la solución de los ejercicios.

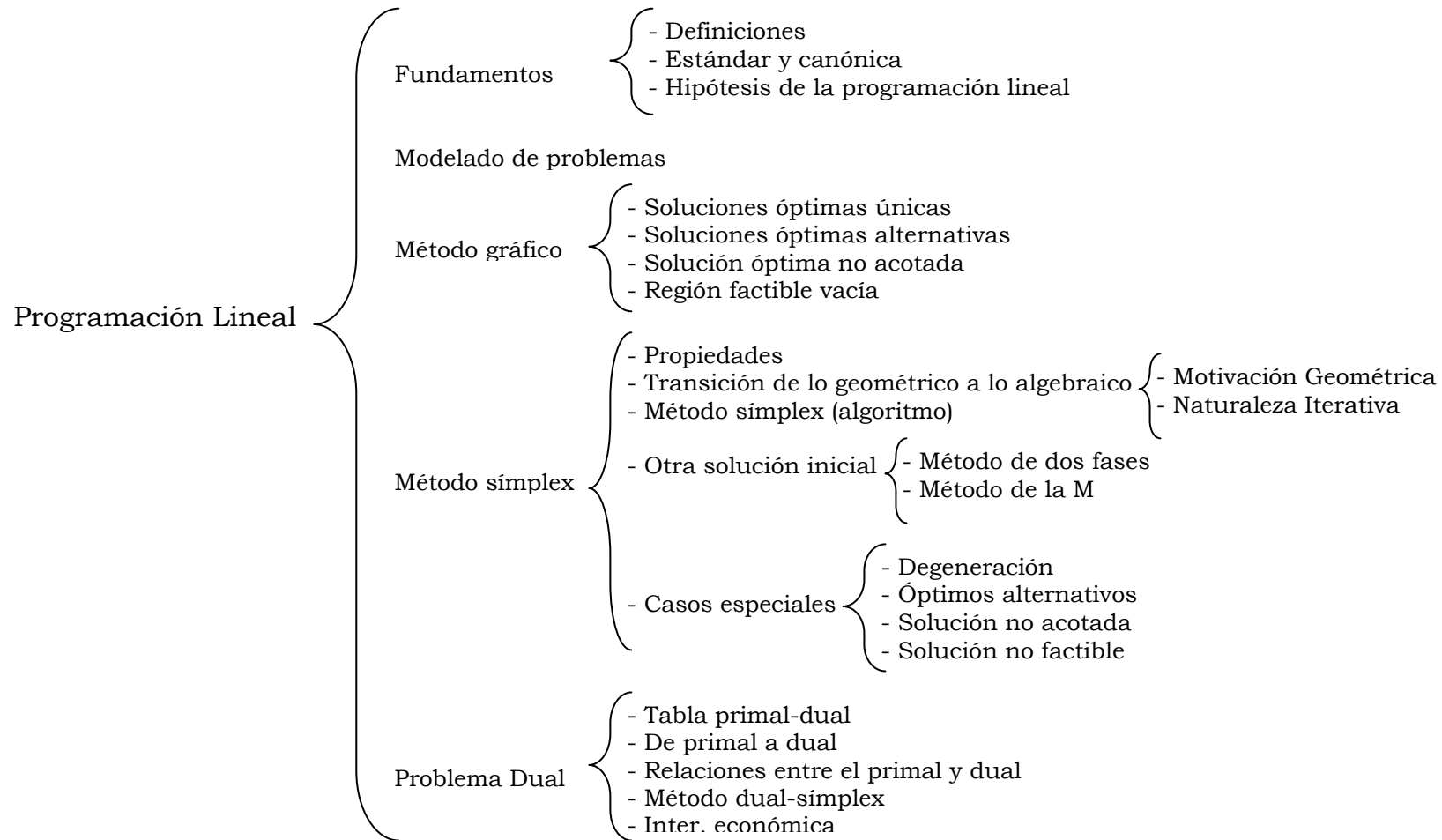
#### Flujo a costo mínimo: Ejercicio 1

Al asignar valores en las casillas se va construyendo la solución del ejercicio



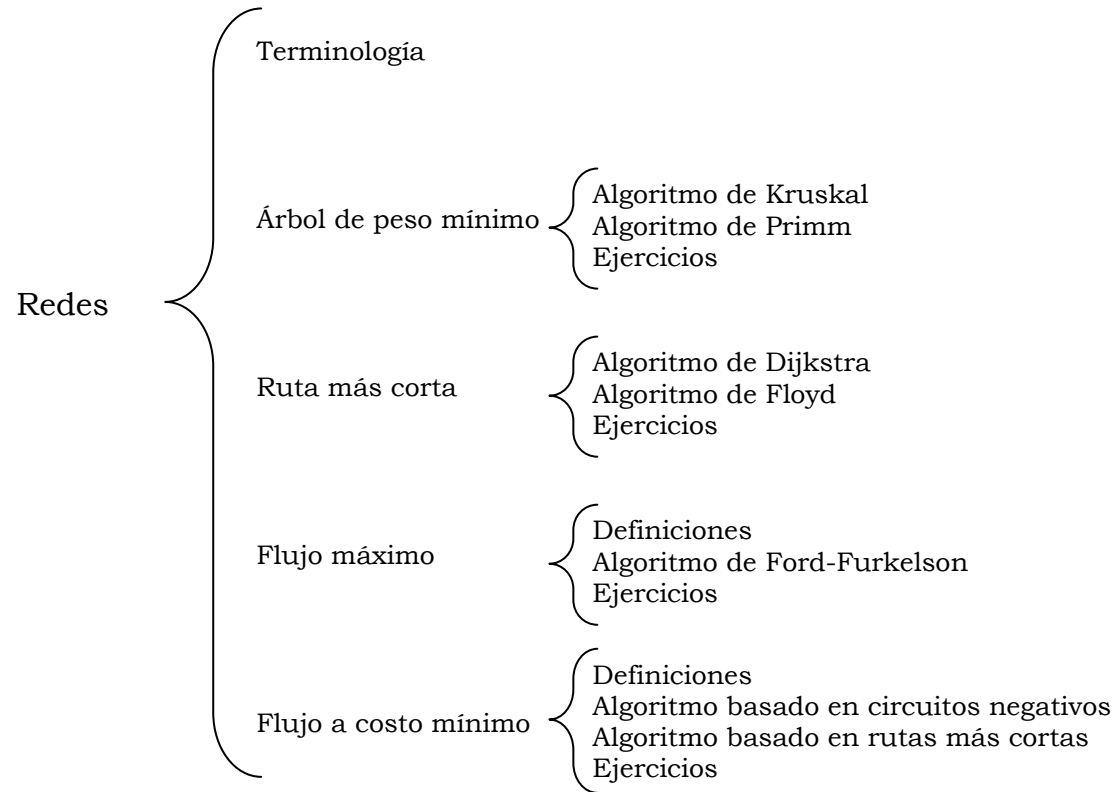
### 3.3 Estructura temática del programa

#### 3.3.1 Programación Lineal





### 3.3.2 Redes



## 4. NAVEGACIÓN DENTRO DEL PROGRAMA

### 4.1 Pantalla de presentación

Al correr el programa se despliega la siguiente pantalla, que muestra el título de la tesis, el nombre del autor en la parte inferior y muestra varios botones: Programación Lineal, Redes, Antes de comenzar y Terminar.



#### 4.1.1 Botón: Antes de comenzar de la Pantalla de presentación

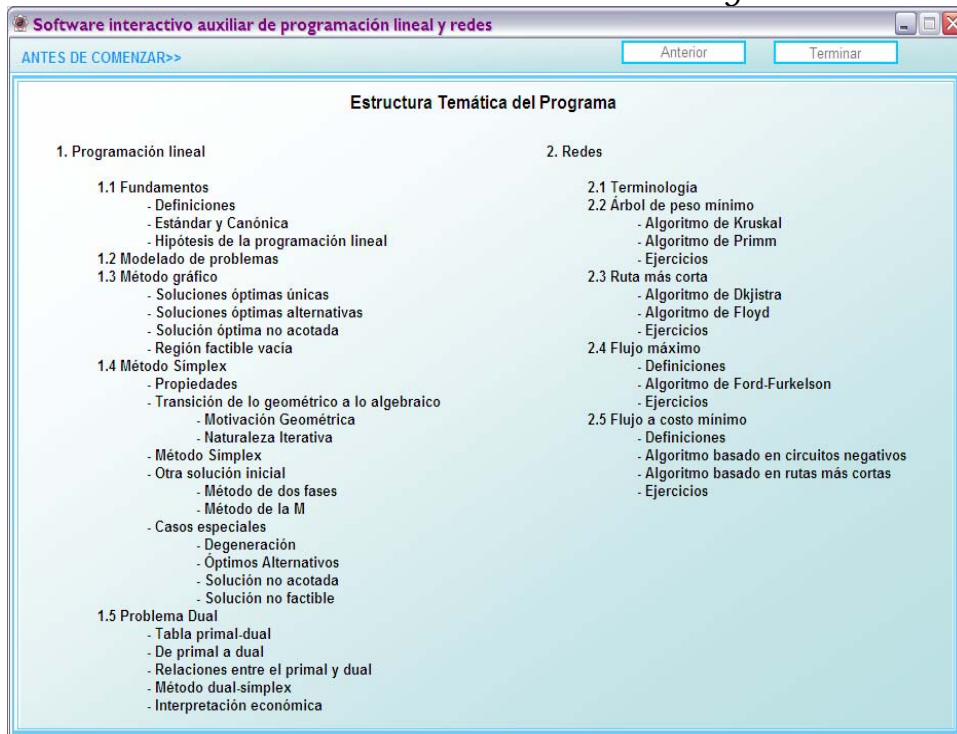
En esta pantalla se muestra el nombre del autor, el tema, el nombre del director de tesis, una breve descripción de a quién va dirigido el trabajo, la bibliografía y un botón llamado 'Estructura' que al presionarlo muestra otra pantalla con la estructura temática del programa.

*Pantalla Antes de comenzar*



Al oprimir el botón 'Estructura Temática' se pasa a la pantalla

*Pantalla Estructura Temática del Programa*



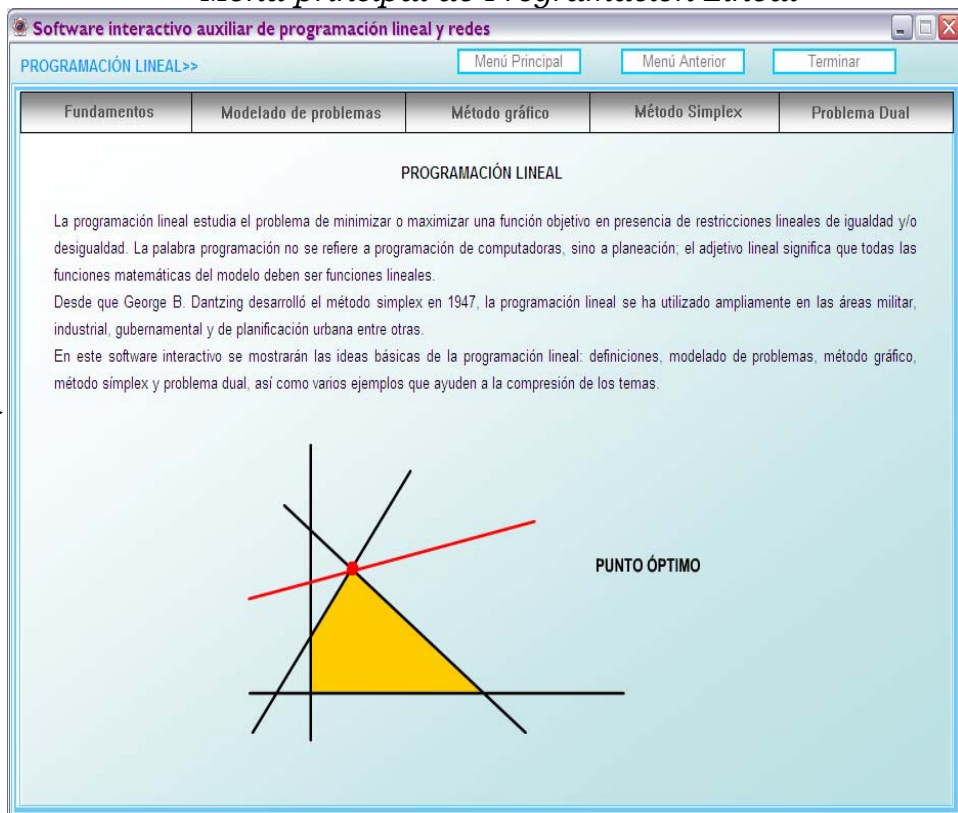
## 4.2 Selección del botón: Programación Lineal

*Pantalla de presentación*

Al oprimir el botón Programación Lineal se pasa a la pantalla



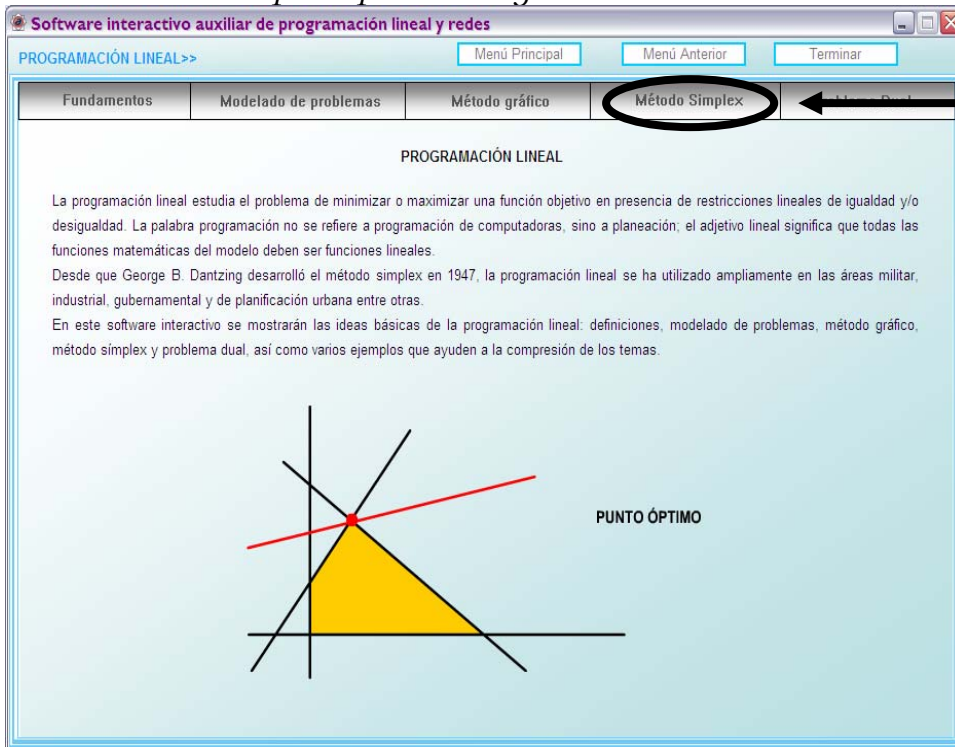
*Menú principal de Programación Lineal*



### 4.2.1 Navegando por el menú: Método Simplex

En este manual sólo se muestra la manera de navegar por el menú *Método Simplex*, el acceso a los demás subtemas del menú principal de Programación Lineal se hace de manera similar.

#### Menú principal de Programación Lineal



Al oprimir el botón **Método Simplex** se pasa a la pantalla

#### Menú Método Simplex

Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.

Ecuación	Condiciones						Solución inicial
	Z	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	
Básicas	Z	-5	-6	0	0	0	0
Variables básicas	$h_1$	0	1	3	1	0	0
	$h_2$	0	1	1	0	1	0
	$h_3$	0	4	1	0	0	1
							Solución
							0
							9
							5
							16

Los términos independientes son no negativos

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

## 4.2.2 Áreas activas en el menú Método Simplex

En esta pantalla hay tres áreas activas asociadas con las palabras: Ecuación, Condiciones y Solución inicial.

### Menú Método Simplex

Software interactivo auxiliar de programación lineal y redes

PROGRAMACIÓN LINEAL >>

Menú Principal Menú Anterior Terminar

Propiedades Transición de lo geométrico a lo algebraico Método Simplex Otra solución inicial Casos especiales

#### MÉTODO SIMPLEX

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

**Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.**

Ecuación Condiciones Solución inicial

Variables no básicas

Básicas	Z	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	Solución
Z	1	-5	-6	0	0	0	0
$h_1$	0	1	3	1	0	0	9
$h_2$	0	1	1	0	1	0	5
$h_3$	0	4	1	0	0	1	16

Variables básicas

Los términos independientes son no negativos

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

Al colocar el mouse sobre la frase 'Solución inicial' se despliega un cubo con información en la pantalla

### Áreas activas: palabras en azul marino

Software interactivo auxiliar de programación lineal y redes

PROGRAMACIÓN LINEAL >>

Menu Principal Menu Anterior Terminar

Propiedades Transición de lo geométrico a lo algebraico Método Simplex Otra solución inicial Casos especiales

#### MÉTODO SIMPLEX

En el diseño de la tabla se especifica el conjunto de variables básicas y no básicas, y también se muestra la solución básica factible asociada con la tabla, las variables básicas tienen el valor de los términos independientes, mientras que las variables no básicas tienen el valor de 0.

Solución básica factible asociada a la tabla.

$h_1 = 9$   
 $h_2 = 5$   
 $h_3 = 16$   
 $Z = 0$   
 $x_1 = 0$   
 $x_2 = 0$

Ecuación Condiciones Solución inicial

Variables no básicas

Básicas	Z	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	Solución
Z	1	-5	-6	0	0	0	0
$h_1$	0	1	3	1	0	0	9
$h_2$	0	1	1	0	1	0	5
$h_3$	0	4	1	0	0	1	16

Variables básicas

Los términos independientes son no negativos

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

### 4.2.3 Botón: Propiedades en el menú Método Simplex

#### Menú Método Simplex

Al oprimir el botón 'Propiedades' se pasa a la pantalla

**MÉTODO SIMPLEX**

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

*Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.*

Ecuación	Condiciones						Solución inicial
Básicas	Z	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	Solución
Z	1	-5	-6	0	0	0	0
$h_1$	0	1	3	1	0	0	9
$h_2$	0	1	1	0	1	0	5
$h_3$	0	4	1	0	0	1	16

Los términos independientes son no negativos

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

#### Pantalla Propiedades

**PROPIEDADES**

Para trabajar con el método simplex es necesario que el problema de programación lineal se ponga en forma estándar.

Las propiedades de un problema de programación lineal en forma estándar son:

1. Todas las restricciones son ecuaciones con términos independientes no negativos
2. Todas las variables son no negativas
3. La función objetivo puede ser la maximización o minimización.

Coloca el mouse sobre cada propiedad para que se muestre la información.

## 4.2.4 Pantalla Propiedades

### Pantalla Propiedades

Al colocar el mouse sobre la frase 'Todas las variables son no negativas' se despliega más información en pantalla.

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

PROPIEDADES

Para trabajar con el método simplex es necesario que el problema de programación lineal se ponga en **forma estándar**.

Las propiedades de un problema de programación lineal en forma estándar son:

1. Todas las restricciones son ecuaciones con términos independientes no negativos
2. Todas las variables son no negativas
3. La función objetivo puede ser la maximización o minimización.

Coloca el mouse sobre cada propiedad para que se muestre la información.

### Áreas activas: frases en azul marino

MÉTODO SIMPLEX>> Menu Principal Menu Anterior Terminar

PROPIEDADES

Para trabajar con el método simplex es necesario que el problema de programación lineal se ponga en **forma estándar**.

Las propiedades de un problema de programación lineal en forma estándar son:

1. Todas las restricciones son ecuaciones con términos independientes no negativos
2. Todas las variables son no negativas
3. La función objetivo puede ser la maximización o minimización.

Las variables del problema deben ser no negativas es decir  $x_i \geq 0$  para toda  $i$ . En caso de tener una variable irrestricta (o no restringida)  $y_i$ , puede expresarse en términos de dos variables no negativas mediante la sustitución:

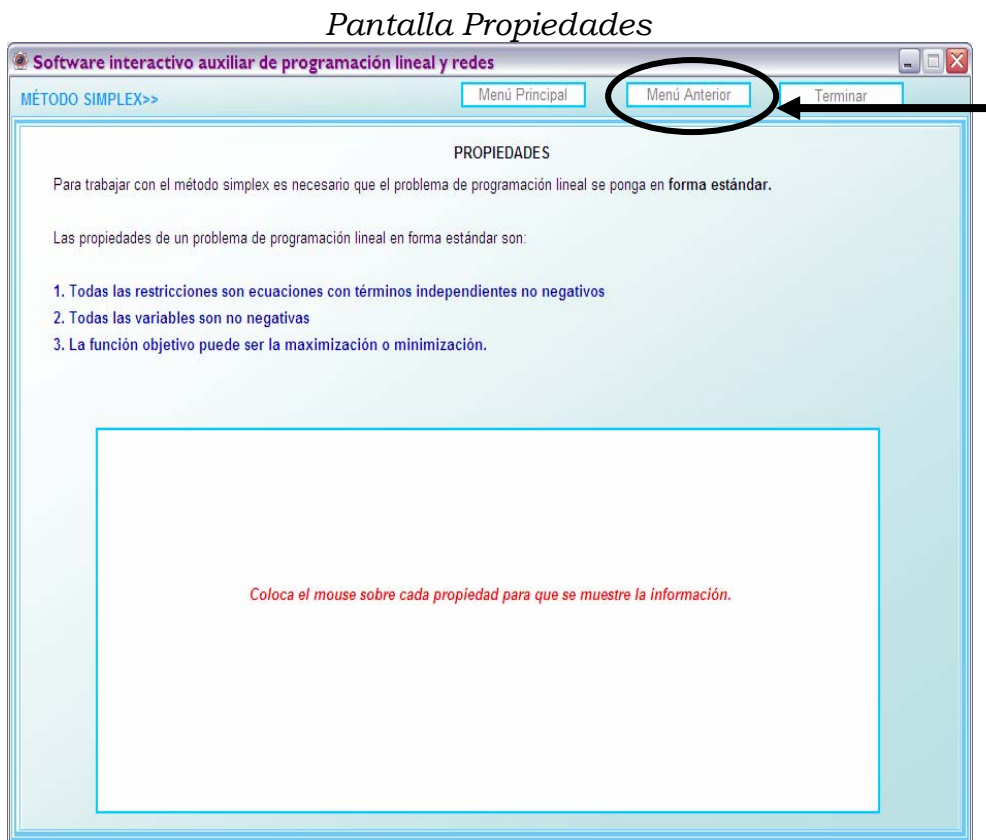
$$y_i = y_i' - y_i'' \quad \text{con} \quad y_i', y_i'' \geq 0$$

El problema de PL se resuelve ahora en términos de  $y_i', y_i''$ , y  $y_i$  se determina por sustitución inversa. Esta sustitución debe efectuarse en todas las restricciones y en la función objetivo. *Por ejemplo:*

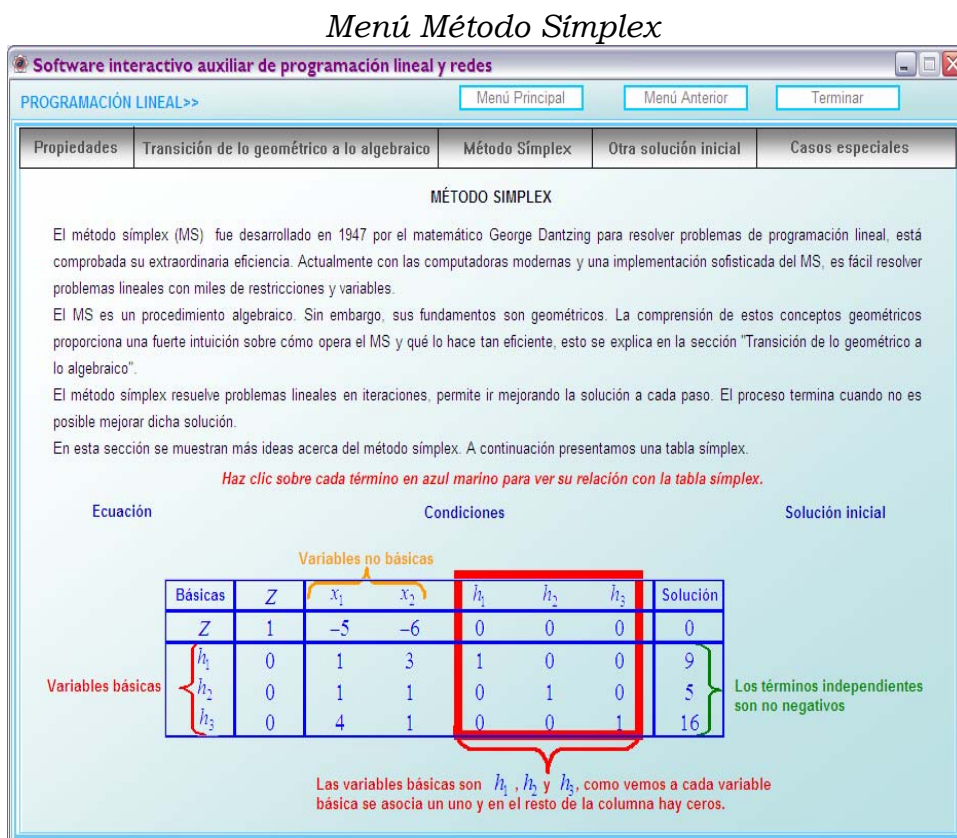
Mínimizar	$3x_1 + 2x_2$	La variable $x_2$ es una variable irrestricta. Hacemos la sustitución $x_2 = x_2' - x_2''$ y obtenemos el siguiente problema:	Mínimizar	$3x_1 + 2x_2' - 2x_2''$
sujeta a	$-x_1 + x_2 \leq 3$		sujeta a	$-x_1 + x_2' - x_2'' \leq 3$
	$5x_1 + 12x_2 \leq 20$			$5x_1 + 12x_2' - 12x_2'' \leq 20$
	$x_1 \geq 0$			$x_1, x_2', x_2'' \geq 0$



#### 4.2.5 Botón: Menú Anterior de la pantalla Propiedades



Al oprimir el botón 'Menú Anterior' se pasa a la pantalla del menú del 'Método Simplex'



#### 4.2.6 Botón: Transición de lo geométrico a lo algebraico en el menú Método Simplex

Al oprimir el botón 'Transición de lo geométrico a lo algebraico' se pasa a la siguiente pantalla

#### Menú Método Simplex

**MÉTODO SIMPLEX**

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

*Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.*

Ecuación	Condiciones						Solución inicial
	Z	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	Solución
Z	1	-5	-6	0	0	0	0
$h_1$	0	1	3	1	0	0	9
$h_2$	0	1	1	0	1	0	5
$h_3$	0	4	1	0	0	1	16

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

Los términos independientes son no negativos

#### Pantalla Transición de lo geométrico a lo algebraico

**TRANSICIÓN DE LO GEOMÉTRICO A LO ALGEBRAICO**

Las ideas contenidas en el método gráfico son la base para desarrollar el método simplex. A continuación se muestra un esquema de como trabajan los métodos. *Haz clic en las flechas azules. Al colocar el mouse sobre algunas flechas aparecerá información adicional en la parte inferior.*

MÉTODO GRÁFICO (MG)	MÉTODO SIMPLEX (MS)
Grafica todas las restricciones, incluyendo las de no negatividad	→
El espacio de soluciones consiste en una infinidad de puntos factibles	→
Identifica puntos esquina (o extremos) factibles del espacio de soluciones	→
Los candidatos a la solución óptima corresponden a una cantidad finita de puntos esquina (o extremos)	→
Se usa la función objetivo para determinar el punto esquina óptimo entre todos los candidatos	→

A continuación se desarrollan las ideas anteriores con más detalle:

**Motivación Geométrica**      **Naturaleza Iterativa**

## 4.2.7 Pantalla Transición de lo geométrico a lo algebraico

### Pantalla Transición de lo geométrico a lo algebraico

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

TRANSICIÓN DE LO GEOMÉTRICO A LO ALGEBRAICO

Las ideas contenidas en el método gráfico son la base para desarrollar el método simplex. A continuación se muestra un esquema de como trabajan los métodos. *Haz clic en las flechas azules. Al colocar el mouse sobre algunas flechas aparecerá información adicional en la parte inferior.*

MÉTODO GRÁFICO (MG)	MÉTODO SIMPLEX (MS)
Grafica todas las restricciones, incluyendo las de no negatividad	
El espacio de soluciones consiste en una infinidad de puntos factibles	
Identifica puntos esquina (o extremos) factibles del espacio de soluciones	
Los candidatos a la solución óptima corresponden a una cantidad finita de puntos esquina (o extremos)	
Se usa la función objetivo para determinar el punto esquina óptimo entre todos los candidatos	

A continuación se desarrollan las ideas anteriores con más detalle:

Motivación Geométrica Naturaleza Iterativa

Al hacer clic sobre cada flecha azul se despliega información adicional

### Flechas de acción azules

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

TRANSICIÓN DE LO GEOMÉTRICO A LO ALGEBRAICO

Las ideas contenidas en el método gráfico son la base para desarrollar el método simplex. A continuación se muestra un esquema de como trabajan los métodos. *Haz clic en las flechas azules. Al colocar el mouse sobre algunas flechas aparecerá información adicional en la parte inferior.*

MÉTODO GRÁFICO (MG)	MÉTODO SIMPLEX (MS)
Grafica todas las restricciones, incluyendo las de no negatividad	Representa el espacio de soluciones con $m$ ecuaciones con $n$ variables, y restringe a todas las variables a valores no negativos, $m < n$ .
El espacio de soluciones consiste en una infinidad de puntos factibles	
Identifica puntos esquina (o extremos) factibles del espacio de soluciones	
Los candidatos a la solución óptima corresponden a una cantidad finita de puntos esquina (o extremos)	
Se usa la función objetivo para determinar el punto esquina óptimo entre todos los candidatos	

A continuación se desarrollan las ideas anteriores con más detalle:

Motivación Geométrica Naturaleza Iterativa

## 4.2.8 Botón: Naturaleza Iterativa de la Pantalla Transición de lo geométrico a lo algebraico

### Pantalla Transición de lo geométrico a lo algebraico

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

TRANSICIÓN DE LO GEOMÉTRICO A LO ALGEBRAICO

Las ideas contenidas en el método gráfico son la base para desarrollar el método simplex. A continuación se muestra un esquema de como trabajan los métodos. *Haz clic en las flechas azules. Al colocar el mouse sobre algunas flechas aparecerá información adicional en la parte inferior.*

MÉTODO GRÁFICO (MG)	MÉTODO SIMPLEX (MS)
Grafica todas las restricciones, incluyendo las de no negatividad	
El espacio de soluciones consiste en una infinidad de puntos factibles	
Identifica puntos esquina (o extremos) factibles del espacio de soluciones	
Los candidatos a la solución óptima corresponden a una cantidad finita de puntos esquina (o extremos)	
Se usa la función objetivo para determinar el punto esquina óptimo entre todos los candidatos	

A continuación se desarrollan las ideas anteriores con más detalle:

Motivación Geométrica Naturaleza Iterativa

Al oprimir el botón 'Naturaleza Iterativa' se pasa a la siguiente pantalla

### Pantalla Naturaleza Iterativa del Método Simplex

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

NATURALEZA ITERATIVA DEL MÉTODO SIMPLEX

En el método gráfico nos dimos cuenta que la solución óptima era un vértice de la región factible, es decir un punto extremo. Como ya se ha mencionado si existe una solución óptima entonces también existe un punto extremo óptimo. En base a esto el método simplex resuelve los problemas de programación lineal moviéndose de un punto extremo al siguiente hasta encontrar el óptimo. Se analiza con el siguiente ejemplo como trabaja el método simplex, en iteraciones, hasta encontrar la solución óptima.

Maximizar  $Z=2x_1 + 3x_2$   
 sujeta a  $x_1 + x_2 \leq 6$   
 $-x_1 + 2x_2 \leq 8$   
 $x_1, x_2 \geq 0$

En la gráfica se muestran los puntos candidatos a ser los óptimos: A:(0,0), B:(0,4), C:(4/3, 14/3), D:(6,0).

Normalmente el MS comienza en el origen donde  $x_1 = 0$  y  $x_2 = 0$ . En este punto de inicio el valor de la función objetivo  $Z=0$ . Se analiza si ese valor mejora (aumenta, pues es un problema de maximización) con un aumento de  $x_1$  y/o  $x_2$  respecto a sus valores actuales de cero.

Entre las dos aristas de la región factible que salen de (0,0), el MS se mueve a través de la arista que aumenta más rápido el valor de Z en este caso  $x_2$ , porque por cada unidad de aumento en esta variable Z aumenta tres unidades. Así se llega al vértice B y  $Z=12$ , escogiendo  $x_1$  se hubiera llegado al vértice D y el valor de Z también hubiera sido de 12.

Siguiente

## 4.2.9 Pantalla Naturaleza Iterativa del Método Simplex

### Pantalla Naturaleza Iterativa del Método Simplex

Software interactivo auxiliar de programación lineal y redes

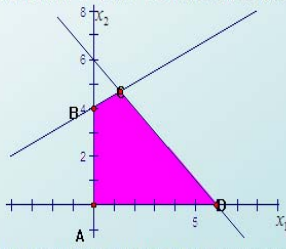
MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

NATURALEZA ITERATIVA DEL MÉTODO SIMPLEX

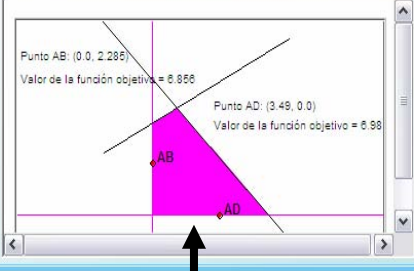
En el **método gráfico** nos dimos cuenta que la solución óptima era un vértice de la región factible, es decir un punto extremo.  
Como ya se ha mencionado si existe una solución óptima entonces también existe un punto extremo óptimo. En base a esto el método simplex resuelve los problemas de programación lineal moviéndose de un punto extremo al siguiente hasta encontrar el óptimo. Se analiza con el siguiente ejemplo como trabaja el método simplex, en iteraciones, hasta encontrar la solución óptima.

Maximizar  $Z=2x_1 + 3x_2$   
sujeta a  $x_1 + x_2 \leq 6$   
 $-x_1 + 2x_2 \leq 8$   
 $x_1, x_2 \geq 0$

En la gráfica se muestran los puntos candidatos a ser los óptimos: A:(0,0), B:(0,4), C:(4/3, 14/3), D:(6,0).



Normalmente el MS comienza en el origen donde  $x_1 = 0$  y  $x_2 = 0$ . En este punto de inicio el valor de la función objetivo  $Z=0$ . Se analiza si ese valor mejora (aumenta, pues es un problema de maximización) con un aumento de  $x_1$  y/o  $x_2$  respecto a sus valores actuales de cero.



Entre las dos aristas de la región factible que salen de (0,0), el MS se mueve a través de la arista que aumenta más rápido el valor de Z en este caso  $x_2$ , porque por cada unidad de aumento en esta variable Z aumenta tres unidades. Así se llega al vértice B y  $Z=12$ , escogiendo  $x_1$  se hubiera llegado al vértice D y el valor de Z también hubiera sido de 12.

Siguiente

En esta gráfica se pueden arrastrar los puntos rojos, y al moverse se muestran sus coordenadas y el valor de la función objetivo.

#### 4.2.10 Botón: Menú Anterior de la Pantalla Naturaleza iterativa del Método Simplex

Pantalla Naturaleza Iterativa del Método Simplex

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

NATURALEZA ITERATIVA DEL MÉTODO SIMPLEX

En el **método gráfico** nos dimos cuenta que la solución óptima era un vértice de la región factible, es decir un punto extremo. Como ya se ha mencionado si existe una solución óptima entonces también existe un punto extremo óptimo. En base a esto el método simplex resuelve los problemas de programación lineal moviéndose de un punto extremo al siguiente hasta encontrar el óptimo. Se analiza con el siguiente ejemplo como trabaja el método simplex, en iteraciones, hasta encontrar la solución óptima.

Maximizar  $Z=2x_1+3x_2$   
 sujeta a  $x_1+x_2 \leq 6$   
 $-x_1+2x_2 \leq 8$   
 $x_1, x_2 \geq 0$

En la gráfica se muestran los puntos candidatos a ser los óptimos: A:(0,0), B:(0,4), C:(4/3, 14/3), D: (6,0).

Normalmente el MS comienza en el origen donde  $x_1=0$  y  $x_2=0$ . En este punto de inicio el valor de la función objetivo  $Z=0$ . Se analiza si ese valor mejora (aumenta, pues es un problema de maximización) con un aumento de  $x_1$  y/o  $x_2$  respecto a sus valores actuales de cero.

Entre las dos aristas de la región factible que salen de (0,0), el MS se mueve a través de la arista que aumenta más rápido el valor de Z en este caso  $x_2$ , porque por cada unidad de aumento en esta variable Z aumenta tres unidades. Así se llega al vértice B y  $Z=12$ , escogiendo  $x_1$  se hubiera llegado al vértice D y el valor de Z también hubiera sido de 12.

Siguiete

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla 'Transición de lo geométrico a lo algebraico'

Pantalla Transición de lo geométrico a lo algebraico

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

TRANSICIÓN DE LO GEOMÉTRICO A LO ALGEBRAICO

Las ideas contenidas en el método gráfico son la base para desarrollar el método simplex. A continuación se muestra un esquema de como trabajan los métodos. Haz clic en las flechas azules. Al colocar el mouse sobre algunas flechas aparecerá información adicional en la parte inferior.

MÉTODO GRÁFICO (MG)	MÉTODO SIMPLEX (MS)
Grafica todas las restricciones, incluyendo las de no negatividad	
El espacio de soluciones consiste en una infinidad de puntos factibles	
Identifica puntos esquina (o extremos) factibles del espacio de soluciones	
Los candidatos a la solución óptima corresponden a una cantidad finita de puntos esquina (o extremos)	
Se usa la función objetivo para determinar el punto esquina óptimo entre todos los candidatos	

A continuación se desarrollan las ideas anteriores con más detalle:

Motivación Geométrica Naturaleza Iterativa

### 4.2.11 Botón: Menú Anterior de la Pantalla Transición de lo geométrico a lo algebraico

Pantalla Naturaleza Iterativa del Método Simplex

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

TRANSICIÓN DE LO GEOMÉTRICO A LO ALGEBRAICO

Las ideas contenidas en el método gráfico son la base para desarrollar el método simplex. A continuación se muestra un esquema de como trabajan los métodos. Haz clic en las flechas azules. Al colocar el mouse sobre algunas flechas aparecerá información adicional en la parte inferior.

MÉTODO GRÁFICO (MG) MÉTODO SIMPLEX (MS)

- Grafica todas las restricciones, incluyendo las de no negatividad
- El espacio de soluciones consiste en una infinidad de puntos factibles
- Identifica puntos esquina (o extremos) factibles del espacio de soluciones
- Los candidatos a la solución óptima corresponden a una cantidad finita de puntos esquina (o extremos)
- Se usa la función objetivo para determinar el punto esquina óptimo entre todos los candidatos

A continuación se desarrollan las ideas anteriores con más detalle:

Motivación Geométrica Naturaleza Iterativa

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla del menú del 'Método simplex'

Menú Método Simplex

PROGRAMACIÓN LINEAL>> Menú Principal Menú Anterior Terminar

Propiedades Transición de lo geométrico a lo algebraico Método Simplex Otra solución inicial Casos especiales

MÉTODO SIMPLEX

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.

Ecuación Condiciones Solución inicial

Básicas	Z	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	Solución
Z	1	-5	-6	0	0	0	0
$h_1$	0	1	3	1	0	0	9
$h_2$	0	1	1	0	1	0	5
$h_3$	0	4	1	0	0	1	16

Variables básicas } Los términos independientes son no negativos

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

#### 4.2.12 Botón: Método Simplex del menú Método Simplex

##### Menú Método Simplex

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

*Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.*

Ecuación		Condiciones			Solución inicial		
	Básicas	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	Solución
Z	Z	1	-5	-6	0	0	0
Variables básicas	$h_1$	0	1	3	1	0	9
	$h_2$	0	1	1	0	1	5
	$h_3$	0	4	1	0	0	16

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

Los términos independientes son no negativos

Al oprimir el botón 'Método Simplex' se pasa a la pantalla

##### Pantalla Procedimiento del Método Simplex

Para mostrar cuál es el procedimiento del Método Simplex se utiliza un ejemplo:

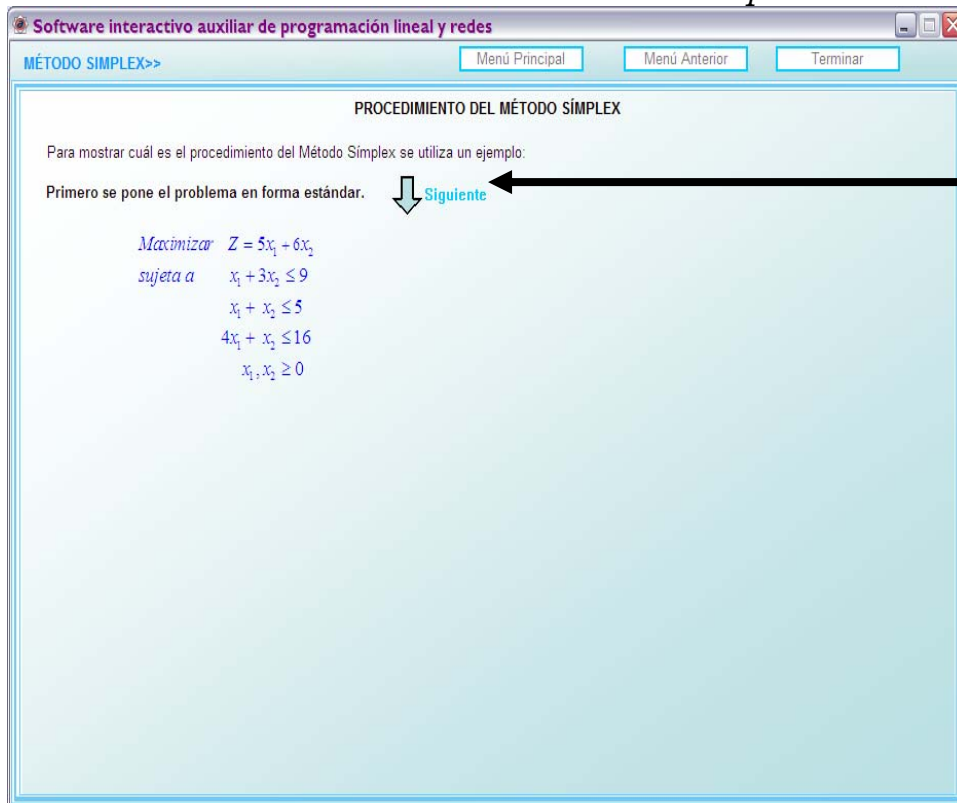
Primero se pone el problema en forma estándar. Siguiente

Maximizar  $Z = 5x_1 + 6x_2$   
 sujeta a  $x_1 + 3x_2 \leq 9$   
 $x_1 + x_2 \leq 5$   
 $4x_1 + x_2 \leq 16$   
 $x_1, x_2 \geq 0$



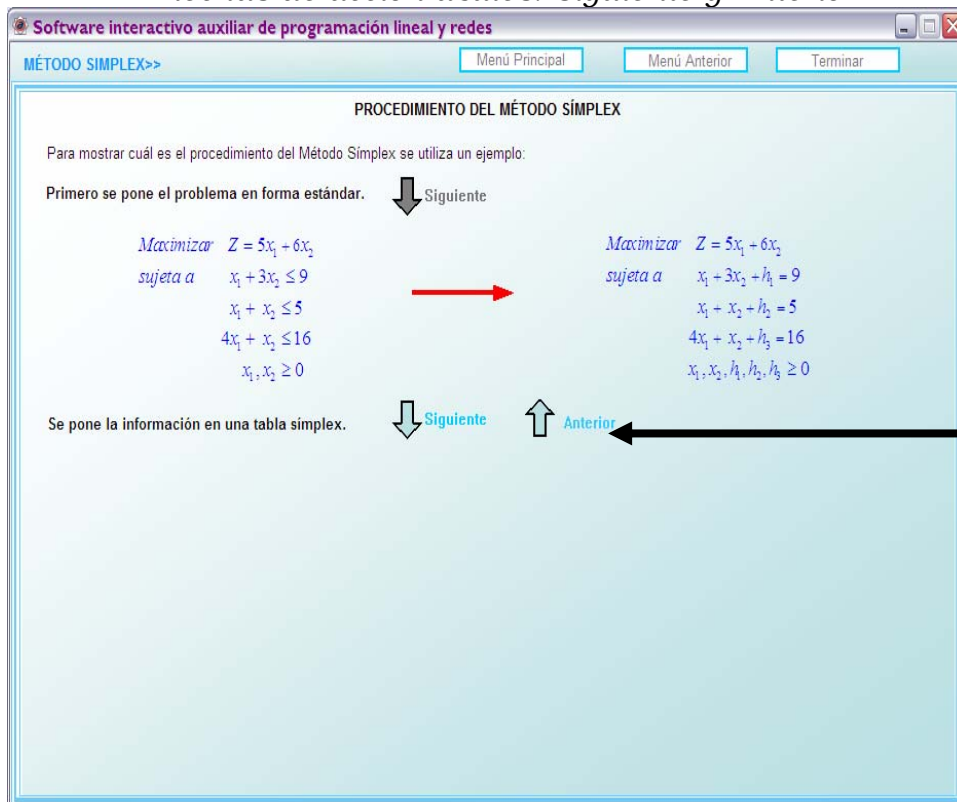
### 4.2.13 Pantalla Procedimiento del Método Simplex

#### Pantalla Procedimiento del Método Simplex



Al hacer clic sobre la flecha azul llamada 'Siguiente' se muestra más información en pantalla

#### Flechas de acción azules: Siguiente y Anterior



Al presionar el botón 'Anterior' se regresa a la pantalla anterior

#### 4.2.14 Botón: Menú Anterior de la Pantalla Procedimiento del Método Simplex

Pantalla Procedimiento del Método Simplex

Software interactivo auxiliar de programación lineal y redes

MÉTODO SÍMPLEX>> Menú Principal Menú Anterior Terminar

PROCEDIMIENTO DEL MÉTODO SÍMPLEX

Para mostrar cuál es el procedimiento del Método Simplex se utiliza un ejemplo:

Primero se pone el problema en forma estándar. ↓ Siguiete

$$\begin{aligned} \text{Maximizar } Z &= 5x_1 + 6x_2 \\ \text{sujeta a } x_1 + 3x_2 &\leq 9 \\ x_1 + x_2 &\leq 5 \\ 4x_1 + x_2 &\leq 16 \\ x_1, x_2 &\geq 0 \end{aligned}$$

→

$$\begin{aligned} \text{Maximizar } Z &= 5x_1 + 6x_2 \\ \text{sujeta a } x_1 + 3x_2 + h_1 &= 9 \\ x_1 + x_2 + h_2 &= 5 \\ 4x_1 + x_2 + h_3 &= 16 \\ x_1, x_2, h_1, h_2, h_3 &\geq 0 \end{aligned}$$

Se pone la información en una tabla simplex. ↓ Siguiete ↑ Anterior

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla del menú del 'Método Simplex'

Menú Método Simplex

Software interactivo auxiliar de programación lineal y redes

PROGRAMACIÓN LINEAL>> Menú Principal Menú Anterior Terminar

Propiedades Transición de lo geométrico a lo algebraico Método Simplex Otra solución inicial Casos especiales

MÉTODO SÍMPLEX

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

*Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.*

Ecuación		Condiciones			Solución inicial			
	Básicas	Z	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	Solución
	Z	1	-5	-6	0	0	0	0
Variables básicas	$h_1$	0	1	3	1	0	0	9
	$h_2$	0	1	1	0	1	0	5
	$h_3$	0	4	1	0	0	1	16

Los términos independientes son no negativos

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

## 4.2.15 Botón: Otra solución inicial del menú Método Simplex

### Menú Método Simplex

Software interactivo auxiliar de programación lineal y redes

PROGRAMACIÓN LINEAL >>

Menú Principal Menú Anterior Terminar

Propiedades Transición de lo geométrico a lo algebraico Método Simplex **Otra solución inicial** Casos especiales

MÉTODO SIMPLEX

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

*Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.*

Ecuación	Condiciones						Solución inicial
	Z	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	Solución
Z	1	-5	-6	0	0	0	0
$h_1$	0	1	3	1	0	0	9
$h_2$	0	1	1	0	1	0	5
$h_3$	0	4	1	0	0	1	16

Variables básicas  $\left\{ \begin{matrix} h_1 \\ h_2 \\ h_3 \end{matrix} \right.$

Variables no básicas  $\left\{ \begin{matrix} x_1 \\ x_2 \end{matrix} \right.$

Los términos independientes son no negativos

Las variables básicas son  $h_1, h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

Al oprimir el botón 'Otra solución inicial' se pasa a la pantalla

### Pantalla Otra solución inicial

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX >>>

Menú Principal Menú Anterior Terminar

OTRA SOLUCIÓN INICIAL

Hasta ahora se ha trabajado con problemas de programación lineal en los que todas las restricciones son  $\leq$  y los términos independientes son no negativos, pues ofrecen una cómoda **solución factible básica de inicio** con todas las holguras. Los modelos donde intervienen restricciones del tipo ( $\geq$ ) o ( $=$ ) no poseen esta propiedad, por lo que en esta sección se mostrará como hacer los ajustes requeridos para poder trabajar dicho tipo de problemas.

Para los problemas con restricciones funcionales del tipo ( $\geq$  ó  $=$ ) se necesita identificar una solución inicial básica factible, esto se resuelve al introducir una variable artificial en la restricción que lo requiera. Esta nueva variable se introduce con el fin de ser la variable básica inicial para esa ecuación.

A continuación se muestra un ejemplo de como se introducen las variables artificiales:

*Pasa el mouse sobre cada conjunto de problemas para ver la explicación*

$\begin{aligned} \text{Minimizar } Z &= 5x_1 + 4x_2 \\ \text{sujeta a } x_1 + 5x_2 &= 3 \\ 3x_1 + 4x_2 &\leq 7 \\ 2x_1 + 3x_2 &\geq 2 \\ x_1, x_2 &\geq 0 \end{aligned}$	$\begin{aligned} x_1 + 5x_2 &= 3 \\ 3x_1 + 4x_2 + h_1 &= 7 \\ 2x_1 + 3x_2 - h_2 &= 2 \\ x_1, x_2, h_1, h_2 &\geq 0 \end{aligned}$	$\begin{aligned} x_1 + 5x_2 + a_1 &= 3 \\ 3x_1 + 4x_2 + h_1 &= 7 \\ 2x_1 + 3x_2 - h_2 + a_2 &= 2 \\ x_1, x_2, h_1, h_2, a_1, a_2 &\geq 0 \end{aligned}$
--	---	---

Las variables artificiales se introducen con el fin de ser las variables básicas iniciales para la ecuación que lo requiera. Las restricciones usuales de no negatividad también se aplican sobre estas variables y la función objetivo se modifica para que imponga una penalización exorbitante en el caso de que adquieran valores mayores a cero. Las iteraciones del MS automáticamente fuerzan a las variables a desaparecer (a volverse cero) una a una, hasta que todas quedan fuera de la solución; después de esto se resuelve el problema real.

En esta sección mostraremos el *Método de dos fases* y el *Método de la M*, para este tipo de problemas.

Método de dos fases      Método de la M

## 4.2.16 Pantalla Otra solución inicial

### Pantalla Otra solución inicial

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

OTRA SOLUCIÓN INICIAL

Hasta ahora se ha trabajado con problemas de programación lineal en los que todas las restricciones son  $\leq$  y los términos independientes son no negativos, pues ofrecen una cómoda **solución factible básica de inicio** con todas las holguras. Los modelos donde intervienen restricciones del tipo  $(\geq)$  o  $(=)$  no poseen esta propiedad, por lo que en esta sección se mostrará como hacer los ajustes requeridos para poder trabajar dicho tipo de problemas.

Para los problemas con restricciones funcionales del tipo  $(\geq)$  o  $(=)$  se necesita identificar una solución inicial básica factible, esto se resuelve al introducir una variable artificial en la restricción que lo requiera. Esta nueva variable se introduce con el fin de ser la variable básica inicial para esa ecuación.

A continuación se muestra un ejemplo de como se introducen las variables artificiales:

Pasa el mouse sobre cada conjunto de ecuaciones para ver la explicación

Minimizar  $Z = 5x_1 + 4x_2$   
 sujeta a  $x_1 + 5x_2 = 3$   
 $3x_1 + 4x_2 \leq 7$   
 $2x_1 + 3x_2 \geq 2$   
 $x_1, x_2 \geq 0$

$x_1 + 5x_2 = 3$   
 $3x_1 + 4x_2 + h_1 = 7$   
 $2x_1 + 3x_2 - h_2 = 2$   
 $x_1, x_2, h_1, h_2 \geq 0$

$x_1 + 5x_2 + a_1 = 3$   
 $3x_1 + 4x_2 + h_1 = 7$   
 $2x_1 + 3x_2 - h_2 + a_2 = 2$   
 $x_1, x_2, h_1, h_2, a_1, a_2 \geq 0$

Las variables artificiales se introducen con el fin de ser las variables básicas iniciales para la ecuación que lo requiera. Las restricciones usuales de no negatividad también se aplican sobre estas variables y la función objetivo se modifica para que imponga una penalización exorbitante en el caso de que adquieran valores mayores a cero. Las iteraciones del MS automáticamente fuerzan a las variables a desaparecer (a volverse cero) una a una, hasta que todas quedan fuera de la solución; después de esto se resuelve el problema real.

En esta sección mostraremos el *Método de dos fases* y el *Método de la M*, para este tipo de problemas.

Método de dos fases Método de la M

Al colocar el mouse sobre el conjunto de ecuaciones se muestra dos cubos con información

### Áreas activas: conjuntos de ecuaciones

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

OTRA SOLUCIÓN INICIAL

Hasta ahora se ha trabajado con problemas de programación lineal en los que todas las restricciones son  $\leq$  y los términos independientes son no negativos, pues ofrecen una cómoda **solución factible básica de inicio** con todas las holguras. Los modelos donde intervienen restricciones del tipo  $(\geq)$  o  $(=)$  no poseen esta propiedad, por lo que en esta sección se mostrará como hacer los ajustes requeridos para poder trabajar dicho tipo de problemas.

Para los problemas con restricciones funcionales del tipo  $(\geq)$  o  $(=)$  se necesita identificar una solución inicial básica factible, esto se resuelve al introducir una variable artificial en la restricción que lo requiera. Esta nueva variable se introduce con el fin de ser la variable básica inicial para esa ecuación.

A continuación se muestra un ejemplo de como se introducen las variables artificiales:

Pasa el mouse sobre cada conjunto de ecuaciones para ver la explicación

Minimizar  $Z = 5x_1 + 4x_2$   
 sujeta a  $x_1 + 5x_2 = 3$   
 $3x_1 + 4x_2 \leq 7$   
 $2x_1 + 3x_2 \geq 2$   
 $x_1, x_2 \geq 0$

Se introducen dos variables: una de holgura en la segunda ecuación y una de exceso en la tercera, porque resta el exceso del lado izquierdo sobre el derecho.

Estas ecuaciones todavía no tienen una solución básica factible, porque en la primera y tercera ecuación no se tienen variables de holgura para usar como variables básicas iniciales.

$x_1 + 5x_2 = 3$   
 $3x_1 + 4x_2 + h_1 = 7$   
 $2x_1 + 3x_2 - h_2 = 2$   
 $x_1, x_2, h_1, h_2 \geq 0$

$x_1 + 5x_2 + a_1 = 3$   
 $3x_1 + 4x_2 + h_1 = 7$   
 $2x_1 + 3x_2 - h_2 + a_2 = 2$   
 $x_1, x_2, h_1, h_2, a_1, a_2 \geq 0$

Las variables artificiales se introducen con el fin de ser las variables básicas iniciales para la ecuación que lo requiera. Las restricciones usuales de no negatividad también se aplican sobre estas variables y la función objetivo se modifica para que imponga una penalización exorbitante en el caso de que adquieran valores mayores a cero. Las iteraciones del MS automáticamente fuerzan a las variables a desaparecer (a volverse cero) una a una, hasta que todas quedan fuera de la solución; después de esto se resuelve el problema real.

En esta sección mostraremos el *Método de dos fases* y el *Método de la M*, para este tipo de problemas.

Método de dos fases Método de la M

#### 4.2.17 Botón: Método de dos fases de la pantalla Otra solución inicial

Pantalla Otra solución inicial

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

OTRA SOLUCIÓN INICIAL

Hasta ahora se ha trabajado con problemas de programación lineal en los que todas las restricciones son  $\leq$  y los términos independientes son no negativos, pues ofrecen una cómoda **solución factible básica de inicio** con todas las holguras. Los modelos donde intervinieren restricciones del tipo  $(\geq)$  o  $(=)$  no poseen esta propiedad, por lo que en esta sección se mostrará como hacer los ajustes requeridos para poder trabajar dicho tipo de problemas.

Para los problemas con restricciones funcionales del tipo  $(\geq)$  o  $(=)$  se necesita identificar una solución inicial básica factible, esto se resuelve al introducir una variable artificial en la restricción que lo requiera. Esta nueva variable se introduce con el fin de ser la variable básica inicial para esa ecuación.

A continuación se muestra un ejemplo de como se introducen las variables artificiales:

*Pasa el mouse sobre cada conjunto de problemas para ver la explicación*

Minimizar $Z = 5x_1 + 4x_2$ sujeta a $x_1 + 5x_2 = 3$ $3x_1 + 4x_2 \leq 7$ $2x_1 + 3x_2 \geq 2$ $x_1, x_2 \geq 0$	$\longrightarrow$	$x_1 + 5x_2 = 3$ $3x_1 + 4x_2 + h_1 = 7$ $2x_1 + 3x_2 - h_2 = 2$ $x_1, x_2, h_1, h_2 \geq 0$	$\longrightarrow$	$x_1 + 5x_2 + a_1 = 3$ $3x_1 + 4x_2 + h_1 = 7$ $2x_1 + 3x_2 - h_2 + a_2 = 2$ $x_1, x_2, h_1, h_2, a_1, a_2 \geq 0$
---	-------------------	---	-------------------	---

Las variables artificiales se introducen con el fin de ser las variables básicas iniciales para la ecuación que lo requiera. Las restricciones usuales de no negatividad también se aplican sobre estas variables y la función objetivo se modifica para que imponga una penalización exorbitante en el caso de que adquieran valores mayores a cero. Las iteraciones del MS automáticamente fuerzan a las variables a desaparecer (a volverse cero) una a una, hasta que todas quedan fuera de la solución; después de esto se resuelve el problema real.

En esta sección mostraremos el *Método de dos fases* y el *Método de la M*, para este tipo de problemas.

Método de dos fases Método de la M

Al oprimir el botón 'Método de dos fases' se pasa a la pantalla

Pantalla Método de dos fases

Software interactivo auxiliar de programación lineal y redes

OTRA SOLUCIÓN INICIAL>> Menu Principal Menu Anterior Terminar

MÉTODO DE DOS FASES

Este método permite que variables artificiales desempeñen el trabajo de holguras en la primera iteración, para después en alguna iteración posterior desecharlas de forma legítima.

Como su nombre lo indica el Método de dos fases resuelve los problemas de programación lineal en dos fases:

**FASE I:** El problema se pone en forma de ecuación y se agregan a las restricciones las variables artificiales necesarias para asegurar una solución básica de inicio. Se determina una solución básica de las ecuaciones resultantes que minimice la suma de las variables artificiales. Si el valor de la suma es positivo, el problema de programación lineal no tiene solución factible, y termina el proceso (una variable artificial positiva significa que no se satisface una restricción original). En caso contrario (valor de cero en la suma) se prosigue con la fase II.

**FASE II:** Se usa la solución factible de la fase I como solución básica factible de inicio para el problema original.

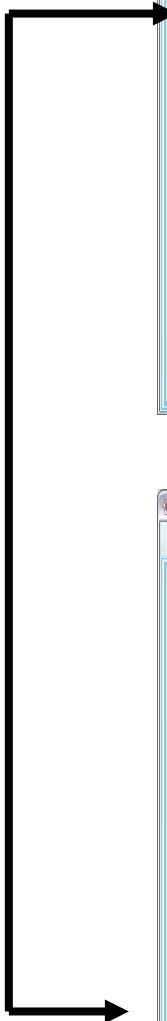
*Coloca el mouse sobre "FASE I" y "FASE II" para mostrar algunas ideas sobre estas fases.*

Se explica el procedimiento del *Método de dos fases* mediante un ejemplo. Ejemplo

## 4.2.18 Pantalla Método de dos fases

### Pantalla Método de dos fases

Al colocar el mouse sobre las palabras en azul marino se muestra más información



### Áreas activas: palabras en azul marino

#### 4.2.19 Botón: Ejemplo de la pantalla Método de dos fases

##### Pantalla Método de dos fases

Software interactivo auxiliar de programación lineal y redes

OTRA SOLUCIÓN INICIAL>>    Menu Principal    Menu Anterior    Terminar

**MÉTODO DE DOS FASES**

Este método permite que variables artificiales desempeñen el trabajo de holguras en la primera iteración, para después en alguna iteración posterior desecharlas de forma legítima.

Como su nombre lo indica el Método de dos fases resuelve los problemas de programación lineal en dos fases:

**FASE I:** El problema se pone en forma de ecuación y se agregan a las restricciones las variables artificiales necesarias para asegurar una solución básica de inicio. Se determina una solución básica de las ecuaciones resultantes que minimice la suma de las variables artificiales. Si el valor de la suma es positivo, el problema de programación lineal no tiene solución factible, y termina el proceso (una variable artificial positiva significa que no se satisface una restricción original). En caso contrario (valor de cero en la suma) se prosigue con la fase II.

**FASE II:** Se usa la solución factible de la fase I como solución básica factible de inicio para el problema original.

Después de algunas iteraciones, como la solución es cero, se pasa a la siguiente fase.

Básicas	$x_1$	$x_2$	$h_1$	$h_2$	$a_1$	$a_2$	Solución
W	0	0	0	0	-1	-1	0
$h_1$	0	0	1	2/9	-2/9	-1/9	14/3
$x_2$	0	1	0	-5/9	5/9	-2/9	39/9
$x_1$	1	0	0	1/3	-1/3	1/3	1

Se explica el procedimiento del Método de dos fases mediante un ejemplo.

Ejemplo

Al oprimir el botón 'Ejemplo' se pasa a la pantalla

##### Pantalla Ejemplo del Método de dos fases

Software interactivo auxiliar de programación lineal y redes

MÉTODO DE DOS FASES>>    Menú Principal    Menú Anterior    Terminar

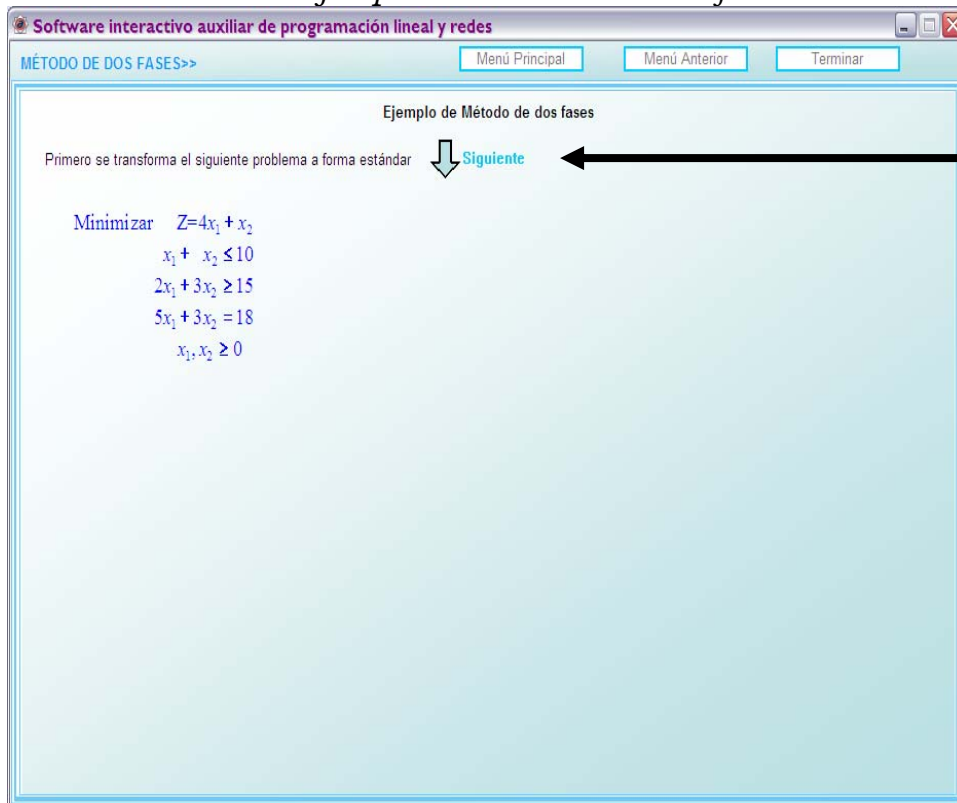
**Ejemplo de Método de dos fases**

Primero se transforma el siguiente problema a forma estándar ↓ **Siguiente**

Minimizar  $Z=4x_1 + x_2$   
 $x_1 + x_2 \leq 10$   
 $2x_1 + 3x_2 \geq 15$   
 $5x_1 + 3x_2 = 18$   
 $x_1, x_2 \geq 0$

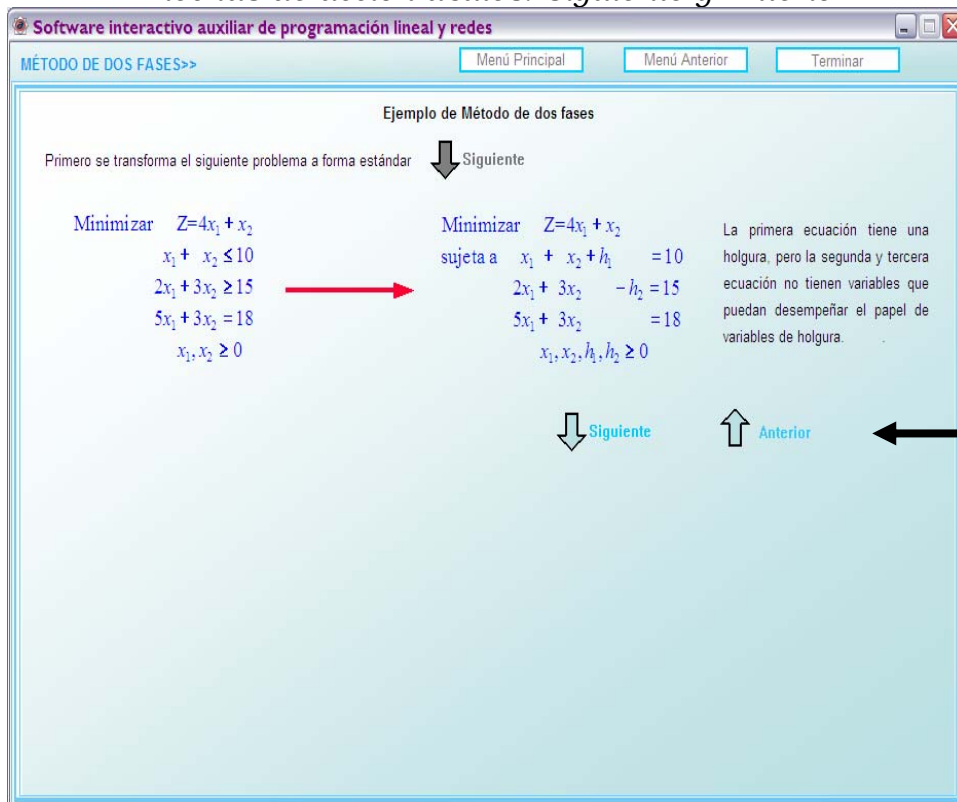
#### 4.2.20 Pantalla Ejemplo del Método de dos fases

##### Pantalla Ejemplo del Método de dos fases



Al hacer clic sobre la flecha azul llamada 'Siguiente' se muestra más información

##### Flechas de acción azules: Siguiente y Anterior

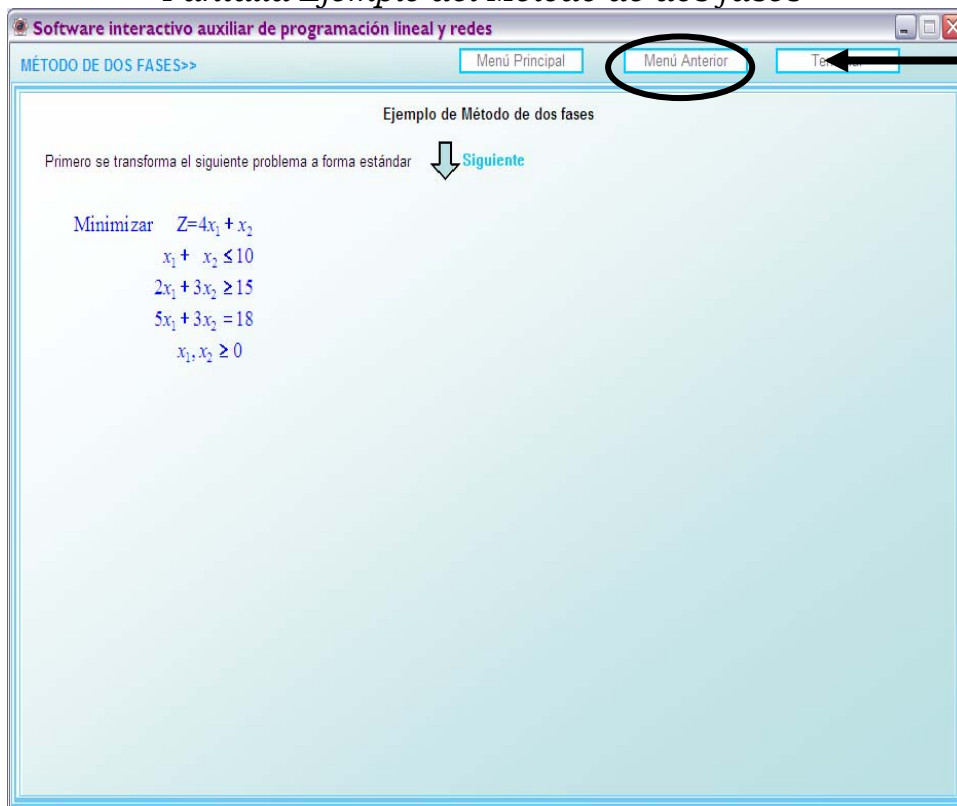


Al presionar el botón 'Anterior' se regresa a la pantalla anterior



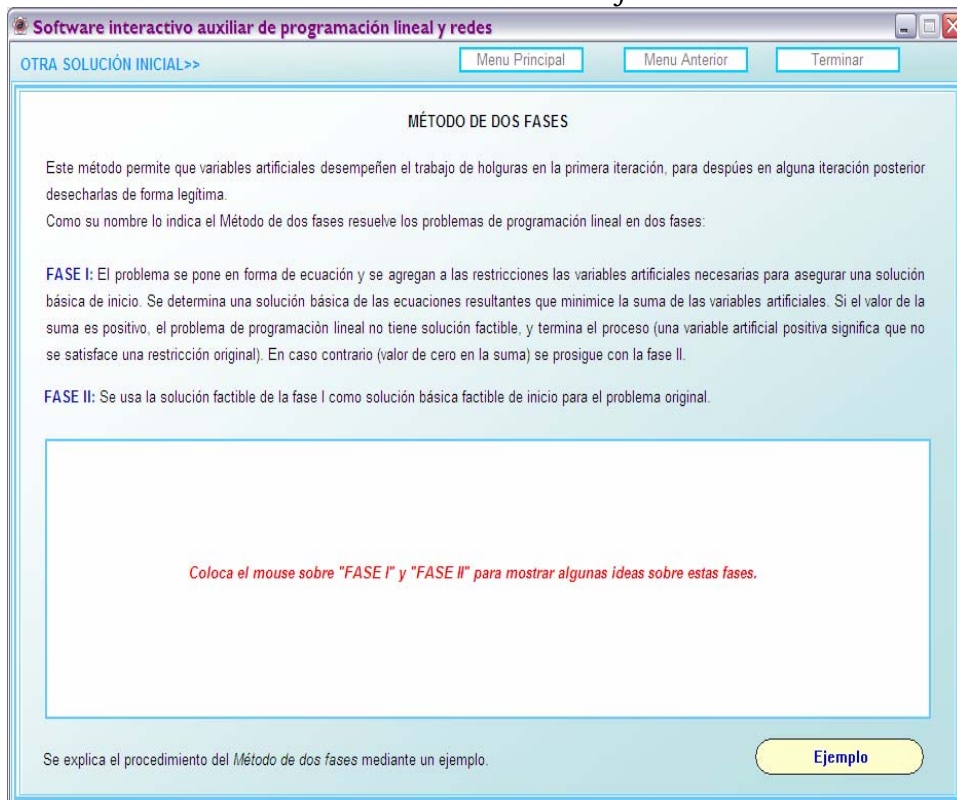
#### 4.2.21 Botón: Menú Anterior de la Pantalla Ejemplo del Método de dos fases

Pantalla Ejemplo del Método de dos fases



Al oprimir el botón 'Menú Anterior' se pasa a la pantalla del menú del 'Método de dos Fases'

Pantalla Método de dos fases



#### 4.2.22 Botón: Menú Anterior de la Pantalla Método de dos fases

*Pantalla Método de dos fases*

MÉTODO DE DOS FASES

Este método permite que variables artificiales desempeñen el trabajo de holguras en la primera iteración, para después en alguna iteración posterior desecharlas de forma legítima.

Como su nombre lo indica el Método de dos fases resuelve los problemas de programación lineal en dos fases:

**FASE I:** El problema se pone en forma de ecuación y se agregan a las restricciones las variables artificiales necesarias para asegurar una solución básica de inicio. Se determina una solución básica de las ecuaciones resultantes que minimice la suma de las variables artificiales. Si el valor de la suma es positivo, el problema de programación lineal no tiene solución factible, y termina el proceso (una variable artificial positiva significa que no se satisface una restricción original). En caso contrario (valor de cero en la suma) se prosigue con la fase II.

**FASE II:** Se usa la solución factible de la fase I como solución básica factible de inicio para el problema original.

*Coloca el mouse sobre "FASE I" y "FASE II" para mostrar algunas ideas sobre estas fases.*

Ejemplo

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla 'Otra solución inicial'

*Pantalla Otra solución inicial*

OTRA SOLUCIÓN INICIAL

Hasta ahora se ha trabajado con problemas de programación lineal en los que todas las restricciones son  $\leq$  y los términos independientes son no negativos, pues ofrecen una cómoda **solución factible básica de inicio** con todas las holguras. Los modelos donde intervienen restricciones del tipo ( $\geq$ ) o ( $=$ ) no poseen esta propiedad, por lo que en esta sección se mostrará como hacer los ajustes requeridos para poder trabajar dicho tipo de problemas.

Para los problemas con restricciones funcionales del tipo ( $\geq$  ó  $=$ ) se necesita identificar una solución inicial básica factible, esto se resuelve al introducir una variable artificial en la restricción que lo requiera. Esta nueva variable se introduce con el fin de ser la variable básica inicial para esa ecuación.

A continuación se muestra un ejemplo de como se introducen las variables artificiales:

*Pasa el mouse sobre cada conjunto de problemas para ver la explicación*

<p>Minimizar <math>Z = 5x_1 + 4x_2</math></p> <p>sujeta a <math>x_1 + 5x_2 = 3</math></p> <p><math>3x_1 + 4x_2 \leq 7</math></p> <p><math>2x_1 + 3x_2 \geq 2</math></p> <p><math>x_1, x_2 \geq 0</math></p>	$\longrightarrow$	<p><math>x_1 + 5x_2 = 3</math></p> <p><math>3x_1 + 4x_2 + h_1 = 7</math></p> <p><math>2x_1 + 3x_2 - h_2 = 2</math></p> <p><math>x_1, x_2, h_1, h_2 \geq 0</math></p>	$\longrightarrow$	<p><math>x_1 + 5x_2 + a_1 = 3</math></p> <p><math>3x_1 + 4x_2 + h_1 = 7</math></p> <p><math>2x_1 + 3x_2 - h_2 + a_2 = 2</math></p> <p><math>x_1, x_2, h_1, h_2, a_1, a_2 \geq 0</math></p>
---	-------------------	--	-------------------	--

Las variables artificiales se introducen con el fin de ser las variables básicas iniciales para la ecuación que lo requiera. Las restricciones usuales de no negatividad también se aplican sobre estas variables y la función objetivo se modifica para que imponga una penalización exorbitante en el caso de que adquieran valores mayores a cero. Las iteraciones del MS automáticamente fuerzan a las variables a desaparecer (a volverse cero) una a una, hasta que todas quedan fuera de la solución; después de esto se resuelve el problema real.

En esta sección mostraremos el *Método de dos fases* y el *Método de la M*, para este tipo de problemas.

Método de dos fases      Método de la M

#### 4.2.23 Botón: Menú Anterior de la Pantalla Método de dos fases

##### Pantalla Otra solución inicial

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

OTRA SOLUCIÓN INICIAL

Hasta ahora se ha trabajado con problemas de programación lineal en los que todas las restricciones son  $\leq$  y los términos independientes son no negativos, pues ofrecen una cómoda **solución factible básica de inicio** con todas las holguras. Los modelos donde intervienen restricciones del tipo ( $\geq$ ) o ( $=$ ) no poseen esta propiedad, por lo que en esta sección se mostrará como hacer los ajustes requeridos para poder trabajar dicho tipo de problemas.

Para los problemas con restricciones funcionales del tipo ( $\geq$  ó  $=$ ) se necesita identificar una solución inicial básica factible, esto se resuelve al introducir una variable artificial en la restricción que lo requiera. Esta nueva variable se introduce con el fin de ser la variable básica inicial para esa ecuación.

A continuación se muestra un ejemplo de como se introducen las variables artificiales:

*Pasa el mouse sobre cada conjunto de problemas para ver la explicación*

Minimizar  $Z = 5x_1 + 4x_2$

sujeta a  $x_1 + 5x_2 = 3$        $x_1 + 5x_2 = 3$        $x_1 + 5x_2 + a_1 = 3$

$3x_1 + 4x_2 \leq 7$        $3x_1 + 4x_2 + h_1 = 7$        $3x_1 + 4x_2 + h_1 = 7$

$2x_1 + 3x_2 \geq 2$        $2x_1 + 3x_2 - h_2 = 2$        $2x_1 + 3x_2 - h_2 + a_2 = 2$

$x_1, x_2 \geq 0$        $x_1, x_2, h_1, h_2 \geq 0$        $x_1, x_2, h_1, h_2, a_1, a_2 \geq 0$

Las variables artificiales se introducen con el fin de ser las variables básicas iniciales para la ecuación que lo requiera. Las restricciones usuales de no negatividad también se aplican sobre estas variables y la función objetivo se modifica para que imponga una penalización exorbitante en el caso de que adquieran valores mayores a cero. Las iteraciones del MS automáticamente fuerzan a las variables a desaparecer (a volverse cero) una a una, hasta que todas quedan fuera de la solución; después de esto se resuelve el problema real.

En esta sección mostraremos el *Método de dos fases* y el *Método de la M*, para este tipo de problemas.

Método de dos fases      Método de la M

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla del menú del 'Método Simplex'

##### Menú Método Simplex

Software interactivo auxiliar de programación lineal y redes

PROGRAMACIÓN LINEAL>> Menú Principal Menú Anterior Terminar

Propiedades | Transición de lo geométrico a lo algebraico | **Método Simplex** | Otra solución inicial | Casos especiales

MÉTODO SIMPLEX

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

*Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.*

Ecuación	Condiciones		Solución inicial			
	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	
Básicas						
Z	1	-5	-6	0	0	0
$h_1$	0	1	3	1	0	9
$h_2$	0	1	1	0	1	5
$h_3$	0	4	1	0	0	16

Variables no básicas:  $x_1, x_2$

Variables básicas:  $h_1, h_2, h_3$

Los términos independientes son no negativos

Las variables básicas son  $h_1, h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

## 4.2.24 Botón: Casos especiales del Método Simplex

### Menú Método Simplex

Software interactivo auxiliar de programación lineal y redes

PROGRAMACIÓN LINEAL >>

Menú Principal Menú Anterior Terminar

Propiedades Transición de lo geométrico a lo algebraico Método Simplex Otra solución inicial **Casos especiales**

#### MÉTODO SIMPLEX

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

*Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.*

Ecuación	Condiciones		Solución inicial			
	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$	
Básicas						
Z	1	-5	-6	0	0	0
$h_1$	0	1	3	1	0	9
$h_2$	0	1	1	0	1	5
$h_3$	0	4	1	0	0	16

Variables básicas:  $h_1, h_2, h_3$

Variables no básicas:  $x_1, x_2$

Los términos independientes son no negativos

Las variables básicas son  $h_1, h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.

Al oprimir el botón 'Casos especiales' se pasa a la pantalla

### Pantalla Casos especiales

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX >>

Menú Principal Menú Anterior Terminar

#### CASOS ESPECIALES

En esta sección van a mostrar cuatro casos especiales que se pueden presentar al aplicar el método simplex:

**Degeneración**

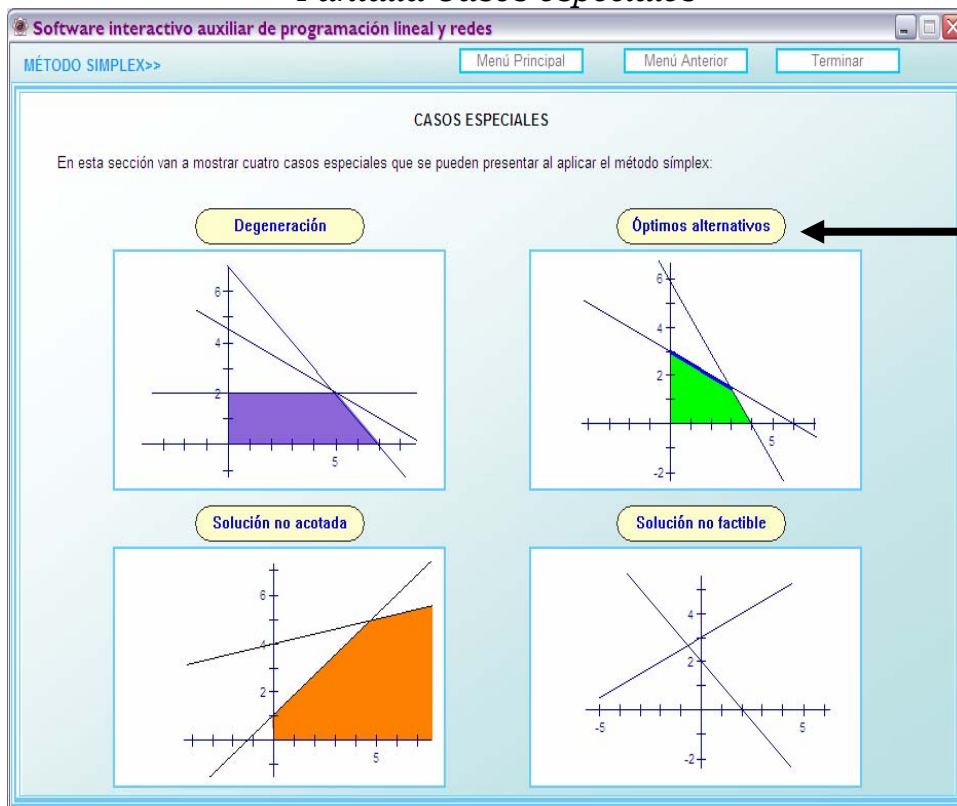
**Óptimos alternativos**

**Solución no acotada**

**Solución no factible**

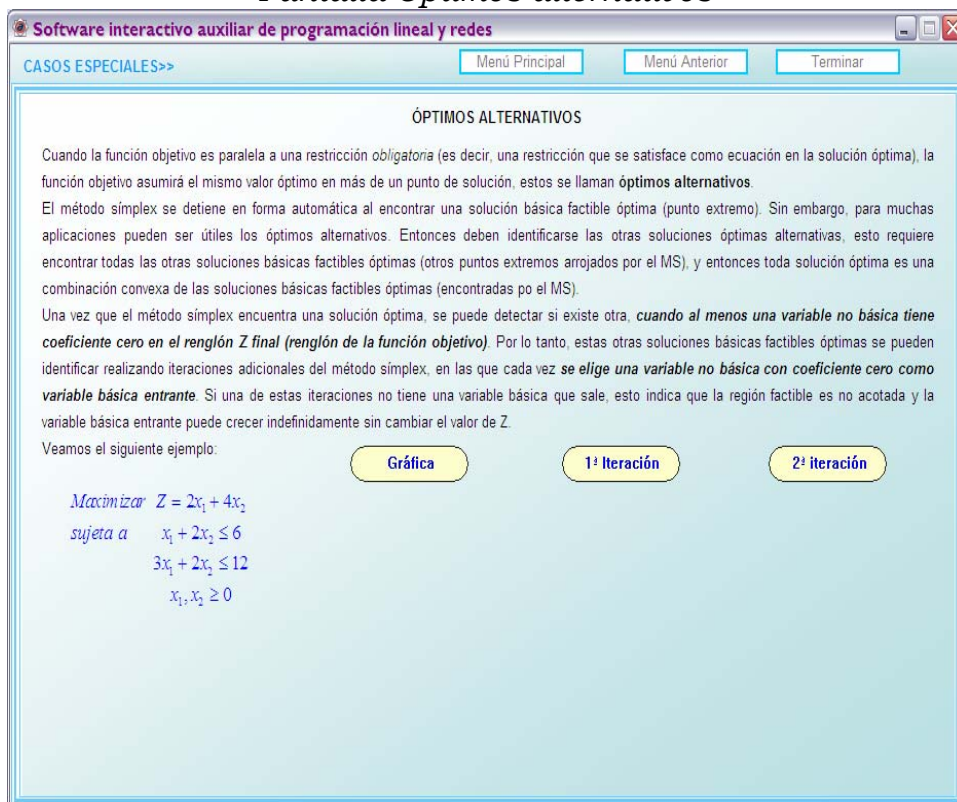
#### 4.2.25 Botón: Óptimos alternativos de la pantalla Casos especiales

Pantalla Casos especiales



Al oprimir el botón 'Óptimos alternativos' se pasa a la pantalla

Pantalla Óptimos alternativos



## 4.2.26 Pantalla Óptimos alternativos

### Pantalla Óptimos alternativos

Software interactivo auxiliar de programación lineal y redes

CASOS ESPECIALES>> Menú Principal Menú Anterior Terminar

#### ÓPTIMOS ALTERNATIVOS

Cuando la función objetivo es paralela a una restricción obligatoria (es decir, una restricción que se satisface como ecuación en la solución óptima), la función objetivo asumirá el mismo valor óptimo en más de un punto de solución, estos se llaman **óptimos alternativos**.

El método simplex se detiene en forma automática al encontrar una solución básica factible óptima (punto extremo). Sin embargo, para muchas aplicaciones pueden ser útiles los óptimos alternativos. Entonces deben identificarse las otras soluciones óptimas alternativas, esto requiere encontrar todas las otras soluciones básicas factibles óptimas (otros puntos extremos arrojados por el MS), y entonces toda solución óptima es una combinación convexa de las soluciones básicas factibles óptimas (encontradas por el MS).

Una vez que el método simplex encuentra una solución óptima, se puede detectar si existe otra, **cuando al menos una variable no básica tiene coeficiente cero en el renglón Z final (renglón de la función objetivo)**. Por lo tanto, estas otras soluciones básicas factibles óptimas se pueden identificar realizando iteraciones adicionales del método simplex, en las que cada vez **se elige una variable no básica con coeficiente cero como variable básica entrante**. Si una de estas iteraciones no tiene una variable básica que sale, esto indica que la región factible es no acotada y la variable básica entrante puede crecer indefinidamente sin cambiar el valor de Z.

Veamos el siguiente ejemplo:

$$\begin{aligned} \text{Maximizar } Z &= 2x_1 + 4x_2 \\ \text{sujeta a } x_1 + 2x_2 &\leq 6 \\ 3x_1 + 2x_2 &\leq 12 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Al oprimir el botón 2ª iteración se despliega más información en pantalla

### Botones de acción

Software interactivo auxiliar de programación lineal y redes

CASOS ESPECIALES>> Menú Principal Menú Anterior Terminar

#### ÓPTIMOS ALTERNATIVOS

Cuando la función objetivo es paralela a una restricción obligatoria (es decir, una restricción que se satisface como ecuación en la solución óptima), la función objetivo asumirá el mismo valor óptimo en más de un punto de solución, estos se llaman **óptimos alternativos**.

El método simplex se detiene en forma automática al encontrar una solución básica factible óptima (punto extremo). Sin embargo, para muchas aplicaciones pueden ser útiles los óptimos alternativos. Entonces deben identificarse las otras soluciones óptimas alternativas, esto requiere encontrar todas las otras soluciones básicas factibles óptimas (otros puntos extremos arrojados por el MS), y entonces toda solución óptima es una combinación convexa de las soluciones básicas factibles óptimas (encontradas por el MS).

Una vez que el método simplex encuentra una solución óptima, se puede detectar si existe otra, **cuando al menos una variable no básica tiene coeficiente cero en el renglón Z final (renglón de la función objetivo)**. Por lo tanto, estas otras soluciones básicas factibles óptimas se pueden identificar realizando iteraciones adicionales del método simplex, en las que cada vez **se elige una variable no básica con coeficiente cero como variable básica entrante**. Si una de estas iteraciones no tiene una variable básica que sale, esto indica que la región factible es no acotada y la variable básica entrante puede crecer indefinidamente sin cambiar el valor de Z.

Veamos el siguiente ejemplo:

$$\begin{aligned} \text{Maximizar } Z &= 2x_1 + 4x_2 \\ \text{sujeta a } x_1 + 2x_2 &\leq 6 \\ 3x_1 + 2x_2 &\leq 12 \\ x_1, x_2 &\geq 0 \end{aligned}$$

**Segunda iteración**

Como vemos en esta iteración se encontró otra solución alternativa, con el mismo valor en la función objetivo.

Básicas	$x_1$	$x_2$	$h_1$	$h_2$	Solución
Z	0	0	2	0	12
$x_2$	0	1	3/4	-1/4	3/2
$x_1$	1	0	-1/2	1/2	3

En la primera iteración vemos que el punto que se encuentra es el A (Gráfica), en la segunda iteración es el B. En realidad todos los puntos del segmento azul son óptimos alternativos, pero como ya se ha mencionado el método simplex sólo se mueve sobre las esquinas de la región factible, por ello el MS sólo encuentra dos soluciones óptimas, los extremos del segmento azul.

El conjunto de soluciones óptimas puede representarse como el promedio ponderado no negativo de los puntos A y B.

#### 4.2.27 Botón: Menú Anterior de la pantalla Óptimos Alternativos

### Pantalla Óptimos alternativos

Software interactivo auxiliar de programación lineal y redes

CASOS ESPECIALES>> Menú Principal Menú Anterior Terminar

### ÓPTIMOS ALTERNATIVOS

Cuando la función objetivo es paralela a una restricción obligatoria (es decir, una restricción que se satisface como ecuación en la solución óptima), la función objetivo asumirá el mismo valor óptimo en más de un punto de solución, estos se llaman **óptimos alternativos**.

El método simplex se detiene en forma automática al encontrar una solución básica factible óptima (punto extremo). Sin embargo, para muchas aplicaciones pueden ser útiles los óptimos alternativos. Entonces deben identificarse las otras soluciones óptimas alternativas, esto requiere encontrar todas las otras soluciones básicas factibles óptimas (otros puntos extremos arrojados por el MS), y entonces toda solución óptima es una combinación convexa de las soluciones básicas factibles óptimas (encontradas por el MS).

Una vez que el método simplex encuentra una solución óptima, se puede detectar si existe otra, **cuando al menos una variable no básica tiene coeficiente cero en el renglón Z final (renglón de la función objetivo)**. Por lo tanto, estas otras soluciones básicas factibles óptimas se pueden identificar realizando iteraciones adicionales del método simplex, en las que cada vez **se elige una variable no básica con coeficiente cero como variable básica entrante**. Si una de estas iteraciones no tiene una variable básica que sale, esto indica que la región factible es no acotada y la variable básica entrante puede crecer indefinidamente sin cambiar el valor de Z.

Veamos el siguiente ejemplo:

Gráfica 1ª Iteración 2ª Iteración

Maximizar  $Z = 2x_1 + 4x_2$   
sujeta a  $x_1 + 2x_2 \leq 6$   
 $3x_1 + 2x_2 \leq 12$   
 $x_1, x_2 \geq 0$

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla de 'Casos especiales'

### Pantalla Casos especiales

Software interactivo auxiliar de programación lineal y redes

MÉTODO SIMPLEX>> Menú Principal Menú Anterior Terminar

### CASOS ESPECIALES

En esta sección van a mostrar cuatro casos especiales que se pueden presentar al aplicar el método simplex:

Degeneración Óptimos alternativos

Solución no acotada Solución no factible

#### 4.2.28 Botón: Menú Anterior de la pantalla Casos especiales

Pantalla Casos especiales

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla del menú del 'Método Simplex'

Menú Método Simplex

El método simplex (MS) fue desarrollado en 1947 por el matemático George Dantzing para resolver problemas de programación lineal, está comprobada su extraordinaria eficiencia. Actualmente con las computadoras modernas y una implementación sofisticada del MS, es fácil resolver problemas lineales con miles de restricciones y variables.

El MS es un procedimiento algebraico. Sin embargo, sus fundamentos son geométricos. La comprensión de estos conceptos geométricos proporciona una fuerte intuición sobre cómo opera el MS y qué lo hace tan eficiente, esto se explica en la sección "Transición de lo geométrico a lo algebraico".

El método simplex resuelve problemas lineales en iteraciones, permite ir mejorando la solución a cada paso. El proceso termina cuando no es posible mejorar dicha solución.

En esta sección se muestran más ideas acerca del método simplex. A continuación presentamos una tabla simplex.

*Haz clic sobre cada término en azul marino para ver su relación con la tabla simplex.*

Ecuación	Condiciones		Solución inicial					
	$x_1$	$x_2$	$h_1$	$h_2$	$h_3$			
Básicas	$Z$	$Z$	$h_1$	$h_2$	$h_3$	Solución		
	$Z$	1	-5	-6	0	0	0	
Variables básicas	$h_1$	0	1	3	1	0	0	9
	$h_2$	0	1	1	0	1	0	5
	$h_3$	0	4	1	0	0	1	16

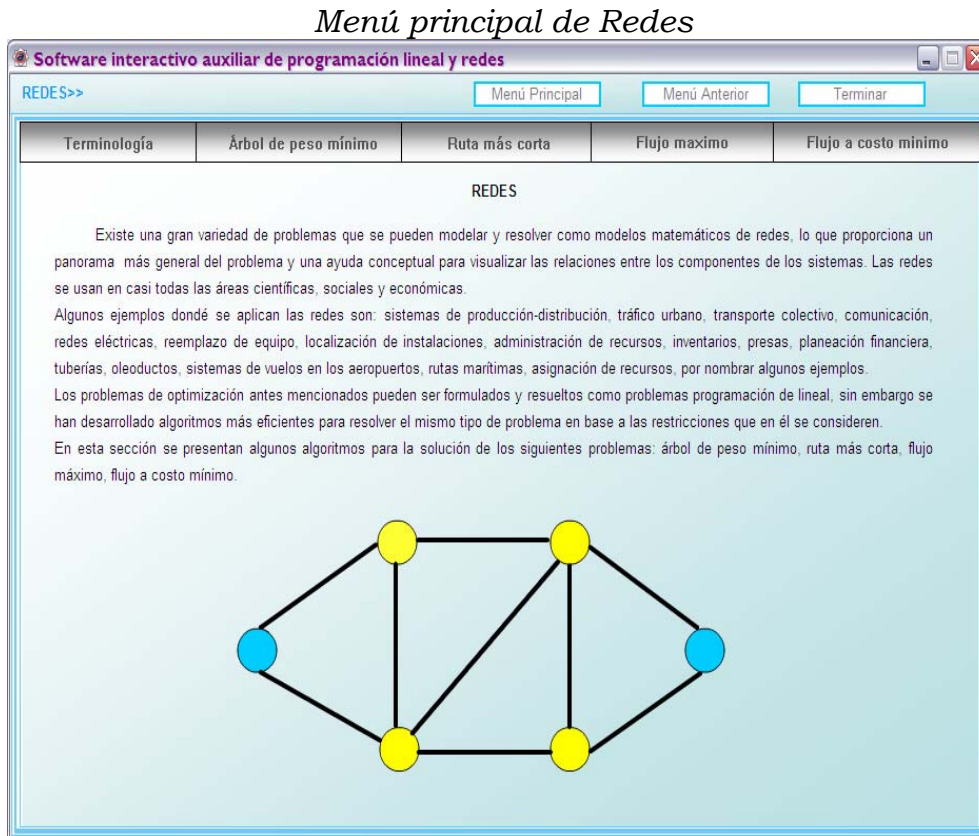
Los términos independientes son no negativos

Las variables básicas son  $h_1$ ,  $h_2$  y  $h_3$ , como vemos a cada variable básica se asocia un uno y en el resto de la columna hay ceros.



### 4.3 Selección del botón: Redes

En la pantalla se muestra una barra de color gris con cinco botones: Terminología, Árbol de peso mínimo, Ruta más corta, Flujo máximo y Flujo a costo mínimo. También se presenta una breve introducción a las redes.

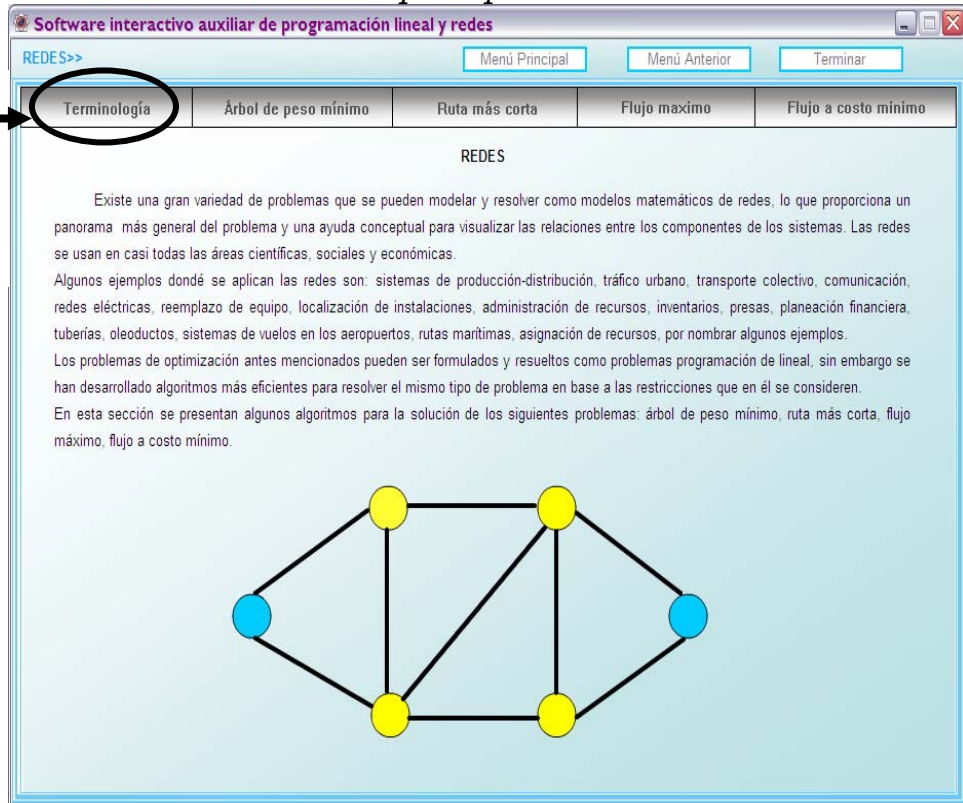


A continuación se muestra la manera de navegar por el menú *Terminología* y *Árbol de peso mínimo*, el acceso a los demás subtemas del menú principal de Redes se hace de manera similar.

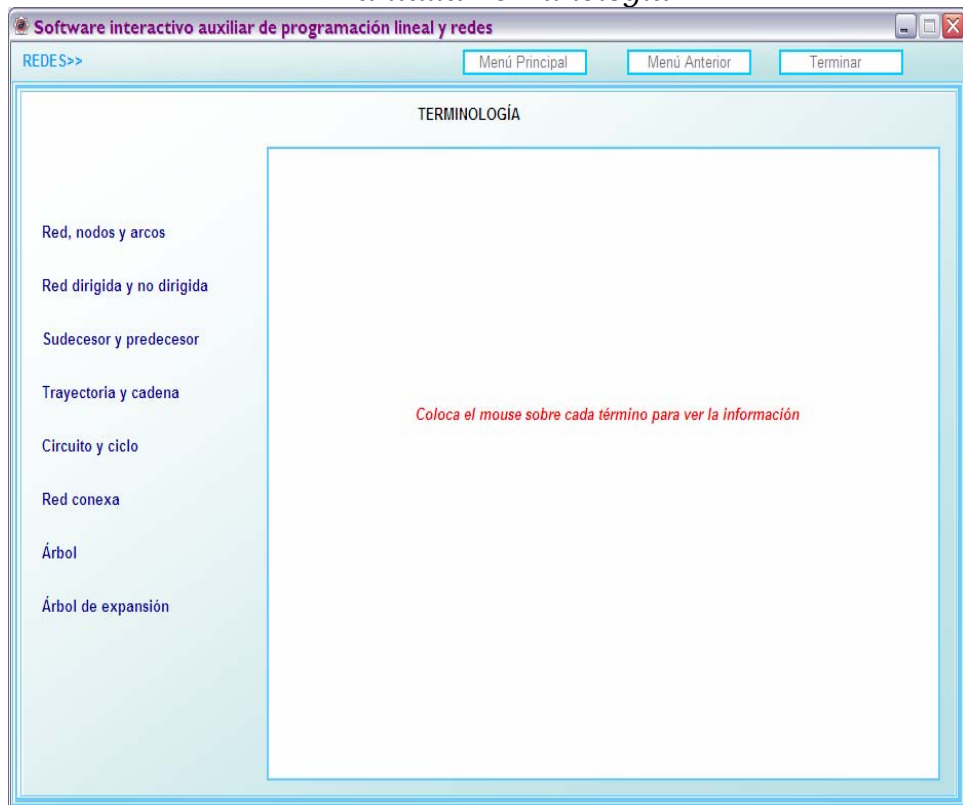
### 4.3.1 Botón: Terminología del menú Principal de Redes

*Menú principal de Redes*

Al oprimir el botón Terminología se pasa a la pantalla



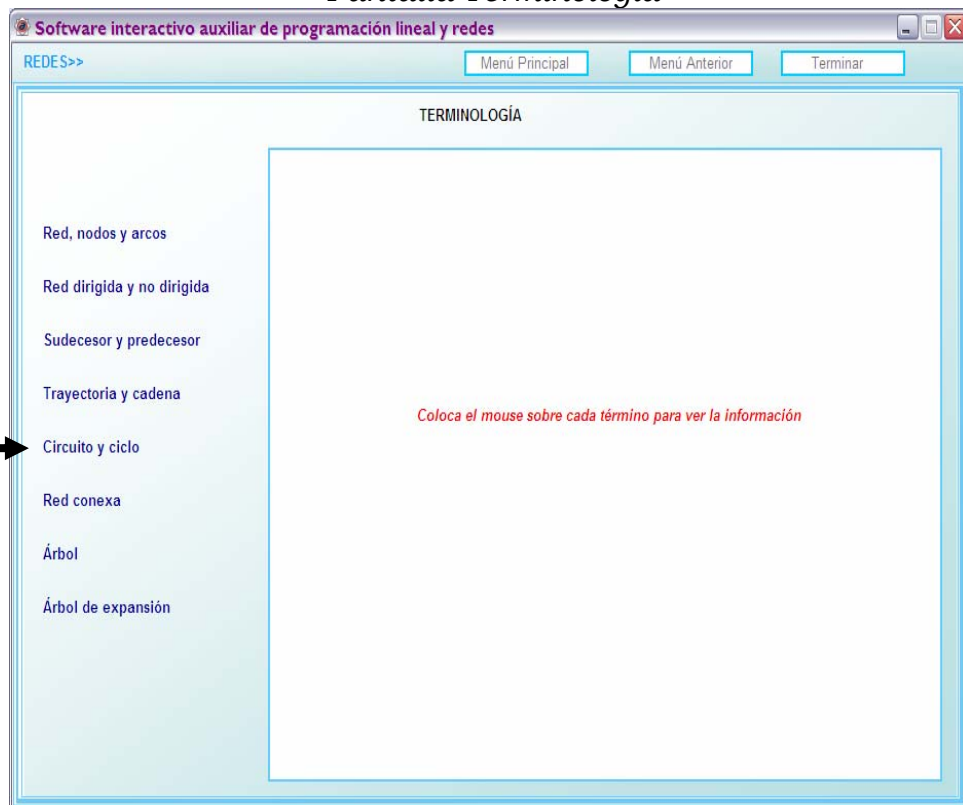
*Pantalla Terminología*



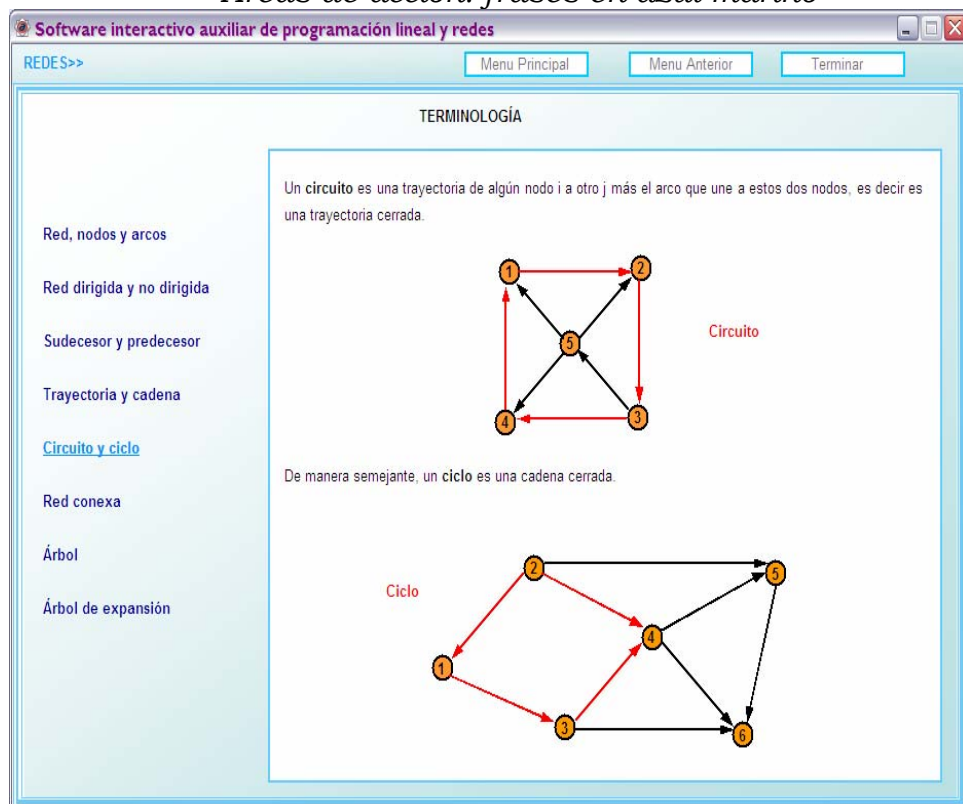
### 4.3.2 Pantalla Terminología

#### Pantalla Terminología

Al colocar el mouse sobre las palabras en azul marino 'Circuito y ciclo' se despliega información en pantalla

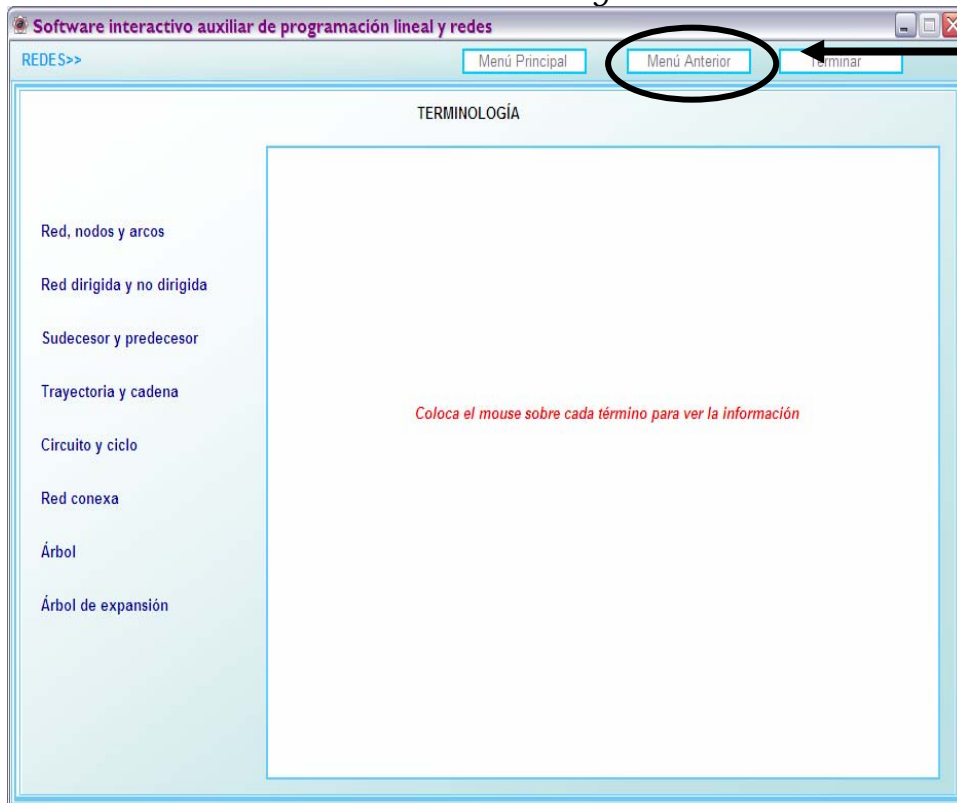


#### Áreas de acción: frases en azul marino



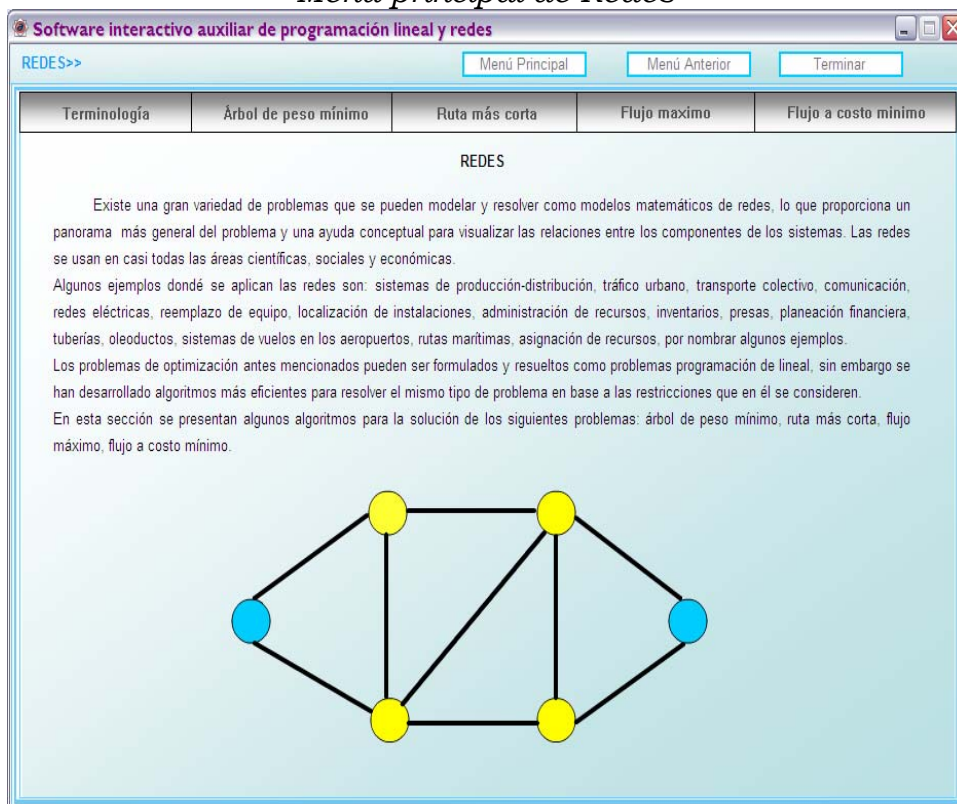
### 4.3.3 Botón: Menú Anterior de la Pantalla Terminología

*Pantalla Terminología*



Al oprimir el botón 'Menú Anterior' se pasa a la pantalla del menú principal de Redes

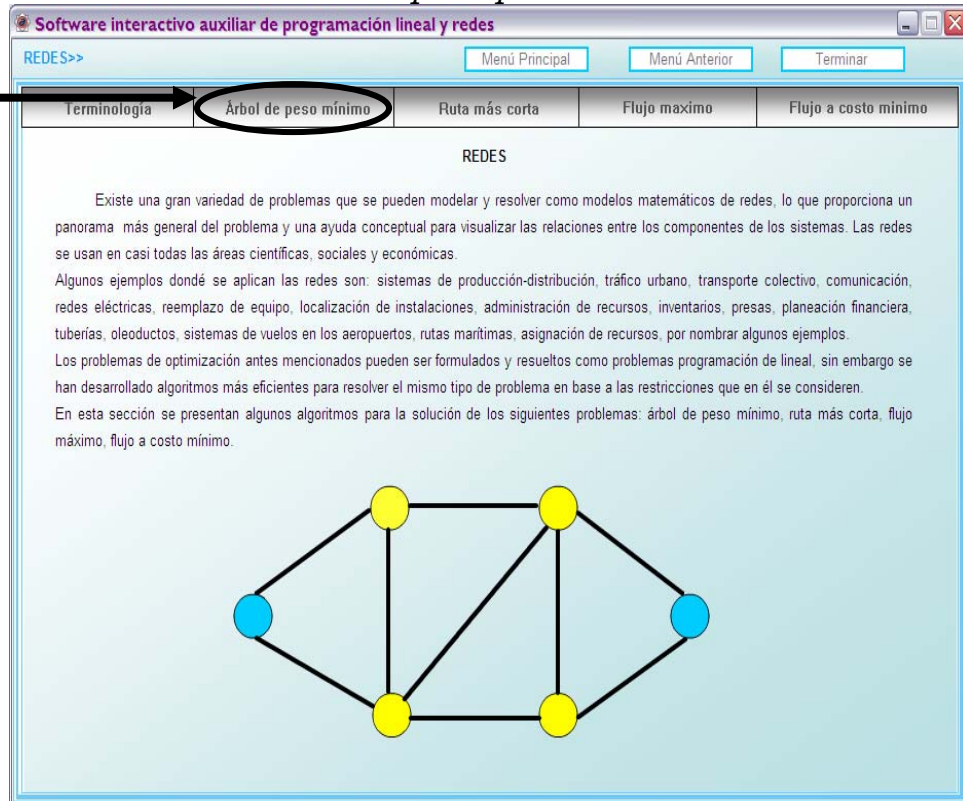
*Menú principal de Redes*



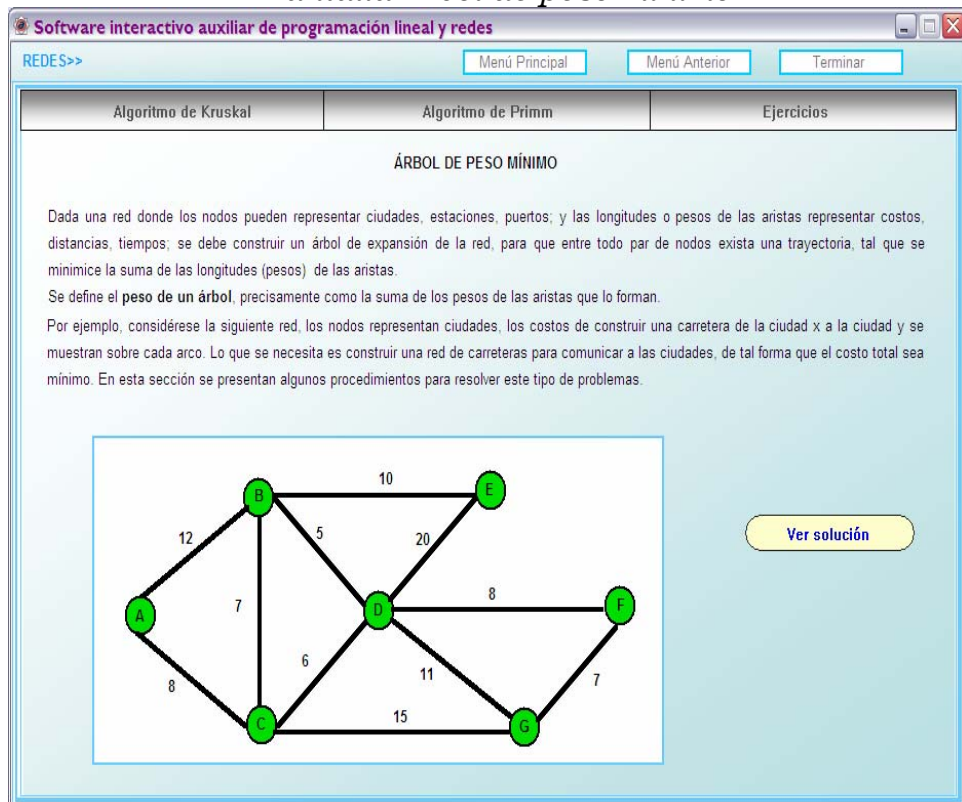
#### 4.3.4 Botón: *Árbol de peso mínimo del menú Principal de Redes*

Al oprimir el botón 'Árbol de peso mínimo' se pasa a la pantalla

*Menú principal de Redes*



*Pantalla Árbol de peso mínimo*



### 4.3.5 Pantalla Árbol de peso mínimo

#### Pantalla Árbol de peso mínimo

Software interactivo auxiliar de programación lineal y redes

REDES>>    Menú Principal    Menú Anterior    Terminar

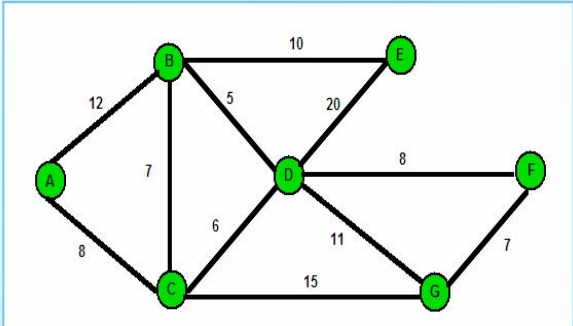
Algoritmo de Kruskal	Algoritmo de Primm	Ejercicios
----------------------	--------------------	------------

ÁRBOL DE PESO MÍNIMO

Dada una red donde los nodos pueden representar ciudades, estaciones, puertos; y las longitudes o pesos de las aristas representar costos, distancias, tiempos; se debe construir un árbol de expansión de la red, para que entre todo par de nodos exista una trayectoria, tal que se minimice la suma de las longitudes (pesos) de las aristas.

Se define el **peso de un árbol**, precisamente como la suma de los pesos de las aristas que lo forman.

Por ejemplo, considérese la siguiente red, los nodos representan ciudades, los costos de construir una carretera de la ciudad x a la ciudad y se muestran sobre cada arco. Lo que se necesita es construir una red de carreteras para comunicar a las ciudades, de tal forma que el costo total sea mínimo. En esta sección se presentan algunos procedimientos para resolver este tipo de problemas.



Ver solución

Al oprimir el botón 'Ver solución' se muestra la solución del problema presentado

#### Botón de acción: solución al problema

Software interactivo auxiliar de programación lineal y redes

REDES>>    Menú Principal    Menú Anterior    Terminar

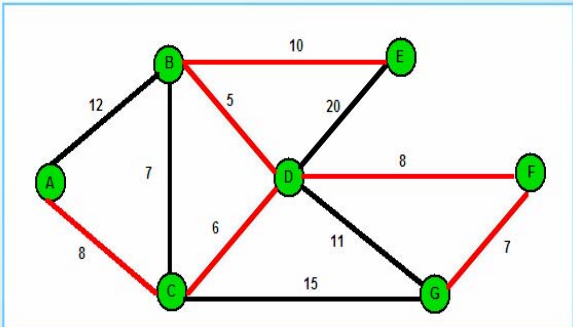
Algoritmo de Kruskal	Algoritmo de Primm	Ejercicios
----------------------	--------------------	------------

ÁRBOL DE PESO MÍNIMO

Dada una red donde los nodos pueden representar ciudades, estaciones, puertos; y las longitudes o pesos de las aristas representar costos, distancias, tiempos; se debe construir un árbol de expansión de la red, para que entre todo par de nodos exista una trayectoria, tal que se minimice la suma de las longitudes (pesos) de las aristas.

Se define el **peso de un árbol**, precisamente como la suma de los pesos de las aristas que lo forman.

Por ejemplo, considérese la siguiente red, los nodos representan ciudades, los costos de construir una carretera de la ciudad x a la ciudad y se muestran sobre cada arco. Lo que se necesita es construir una red de carreteras para comunicar a las ciudades, de tal forma que el costo total sea mínimo. En esta sección se presentan algunos procedimientos para resolver este tipo de problemas.



Ver solución

Esta solución puede comprobarse al aplicar alguno de los algoritmos de esta sección.

### 4.3.6 Botón: Algoritmo de Kruskal del menú Árbol de peso mínimo

Pantalla Árbol de peso mínimo

Al oprimir el botón 'Algoritmo de Kruskal' se pasa a la pantalla

Software interactivo auxiliar de programación lineal y redes

REDES>> Menu Principal Menu Anterior Terminar

**Algoritmo de Kruskal** | Algoritmo de Primm | Ejercicios

ÁRBOL DE PESO MÍNIMO

Dada una red donde los nodos pueden representar ciudades, estaciones, puertos; y las longitudes o pesos de las aristas representar costos, distancias, tiempos; se debe construir un árbol de expansión de la red, para que entre todo par de nodos exista una trayectoria, tal que se minimice la suma de las longitudes (pesos) de las aristas.

Se define el **peso de un árbol**, precisamente como la suma de los pesos de las aristas que lo forman.

Por ejemplo, considérese la siguiente red, los nodos representan ciudades, los costos de construir una carretera de la ciudad x a la ciudad y se muestran sobre cada arco. Lo que se necesita es construir una red de carreteras para comunicar a las ciudades, de tal forma que el costo total sea mínimo. En esta sección se presentan algunos procedimientos para resolver este tipo de problemas.

[Ver solución](#)

Pantalla Algoritmo de Kruskal

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO>> Menú Principal Menú Anterior Terminar

ALGORITMO DE KRUSKAL

Paso 0. Ordenar el conjunto de aristas de manera creciente con respecto a su peso. Sean  $a_1, a_2, \dots, a_m$  las aristas ordenadas. Hacer  $k = j = 0$  y  $A = \{\emptyset\}$ .

Paso 1. Hacer  $j = j + 1$ . Si la arista  $a_j$  no forma un ciclo con las aristas de  $A$  entonces  $A = A \cup a_j$ . Hacer  $k = k + 1$ . Si la arista  $a_j$  forma un ciclo, ir al paso 2.

Paso 2. Si  $k = n - 1$  terminar. Se ha encontrado el árbol expandido. Si  $k < n - 1$  regresar al paso 2.

A continuación se muestra como funciona el algoritmo con el siguiente ejemplo:

PASO 0 ← →

Se ordenan las aristas en forma creciente y se inicia con  $k=j=0$ .

### 4.3.7 Pantalla Algoritmo de Kruskal

Pantalla Algoritmo de Kruskal

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO >> Menú Principal Menú Anterior Terminar

**ALGORITMO DE KRUSKAL**

**Paso 0.** Ordenar el conjunto de aristas de manera creciente con respecto a su peso. Sean  $a_1, a_2, \dots, a_m$  las aristas ordenadas. Hacer  $k = j = 0$  y  $A = \{\emptyset\}$ .

**Paso 1.** Hacer  $j = j + 1$ . Si la arista  $a_j$  no forma un ciclo con las aristas de  $A$  entonces  $A = A \cup a_j$ . Hacer  $k = k + 1$ . Si la arista  $a_j$  forma un ciclo, ir al paso 2.

**Paso 2.** Si  $k = n - 1$  terminar. Se ha encontrado el árbol expandido. Si  $k < n - 1$  regresar al paso 2.

A continuación se muestra como funciona el algoritmo con el siguiente ejemplo:

PASO 0

Se ordenan las aristas en forma creciente y se inicia con  $k=j=0$ .

Al hacer clic sobre las flechas se pasa de una iteración a otra

Botones de flecha: iteraciones del algoritmo

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO >> Menú Principal Menú Anterior Terminar

**ALGORITMO DE KRUSKAL**

**Paso 0.** Ordenar el conjunto de aristas de manera creciente con respecto a su peso. Sean  $a_1, a_2, \dots, a_m$  las aristas ordenadas. Hacer  $k = j = 0$  y  $A = \{\emptyset\}$ .

**Paso 1.** Hacer  $j = j + 1$ . Si la arista  $a_j$  no forma un ciclo con las aristas de  $A$  entonces  $A = A \cup a_j$ . Hacer  $k = k + 1$ . Si la arista  $a_j$  forma un ciclo, ir al paso 2.

**Paso 2.** Si  $k = n - 1$  terminar. Se ha encontrado el árbol expandido. Si  $k < n - 1$  regresar al paso 2.

A continuación se muestra como funciona el algoritmo con el siguiente ejemplo:

ITERACIÓN 1

Hacemos:

$j = 1$   
 $A = \{a_1\}$   
 $k = 1$



### 4.3.8 Botón Menú Anterior del Algoritmo de Kruskal

Pantalla Algoritmo de Kruskal

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO >>

Menú Principal Menú Anterior Terminar

ALGORITMO DE KRUSKAL

Paso 0. Ordenar el conjunto de aristas de manera creciente con respecto a su peso. Sean  $a_1, a_2, \dots, a_n$  las aristas ordenadas. Hacer  $k = j = 0$  y  $A = \{\emptyset\}$ .

Paso 1. Hacer  $j = j + 1$ . Si la arista  $a_j$  no forma un ciclo con las aristas de  $A$  entonces  $A = A \cup a_j$ . Hacer  $k = k + 1$ . Si la arista  $a_j$  forma un ciclo, ir al paso 2.

Paso 2. Si  $k = n - 1$  terminar. Se ha encontrado el árbol expandido. Si  $k < n - 1$  regresar al paso 2.

A continuación se muestra como funciona el algoritmo con el siguiente ejemplo:

PASO 0

Se ordenan las aristas en forma creciente y se inicia con  $k=j=0$ .

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla principal de 'Árbol de peso mínimo'

Pantalla Árbol de peso mínimo

Software interactivo auxiliar de programación lineal y redes

REDES >>

Menú Principal Menú Anterior Terminar

Algoritmo de Kruskal Algoritmo de Prim Ejercicios

ÁRBOL DE PESO MÍNIMO

Dada una red donde los nodos pueden representar ciudades, estaciones, puertos; y las longitudes o pesos de las aristas representar costos, distancias, tiempos; se debe construir un árbol de expansión de la red, para que entre todo par de nodos exista una trayectoria, tal que se minimice la suma de las longitudes (pesos) de las aristas.

Se define el **peso de un árbol**, precisamente como la suma de los pesos de las aristas que lo forman.

Por ejemplo, considérese la siguiente red, los nodos representan ciudades, los costos de construir una carretera de la ciudad x a la ciudad y se muestran sobre cada arco. Lo que se necesita es construir una red de carreteras para comunicar a las ciudades, de tal forma que el costo total sea mínimo. En esta sección se presentan algunos procedimientos para resolver este tipo de problemas.

Ver solución

### 4.3.9 Botón: Algoritmo de Primm de la pantalla Árbol de peso mínimo

Pantalla Árbol de peso mínimo

Software interactivo auxiliar de programación lineal y redes

REDES>> Menú Principal Menú Anterior Terminar

Algoritmo de Kruskal **Algoritmo de Primm** Ejercicios

ÁRBOL DE PESO MÍNIMO

Dada una red donde los nodos pueden representar ciudades, estaciones, puertos; y las longitudes o pesos de las aristas representar costos, distancias, tiempos; se debe construir un árbol de expansión de la red, para que entre todo par de nodos exista una trayectoria, tal que se minimice la suma de las longitudes (pesos) de las aristas.

Se define el **peso de un árbol**, precisamente como la suma de los pesos de las aristas que lo forman.

Por ejemplo, considérese la siguiente red, los nodos representan ciudades, los costos de construir una carretera de la ciudad x a la ciudad y se muestran sobre cada arco. Lo que se necesita es construir una red de carreteras para comunicar a las ciudades, de tal forma que el costo total sea mínimo. En esta sección se presentan algunos procedimientos para resolver este tipo de problemas.

Ver solución

Al oprimir el botón 'Algoritmo de Primm' se pasa a la pantalla

Pantalla Algoritmo de Primm

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO>> Menú Principal Menú Anterior Terminar

ALGORITMO DE PRIMM

Paso 0. Sea  $n_k$  cualquier nodo de la red. Sea  $N_0 = \{n_k\}$ . Hacer  $k = 0$ .

Paso 1. Hacer  $k = k + 1$ . Sea  $C_k$  el conjunto de aristas que tienen exactamente un extremo en  $N_{k-1}$ . Sea  $a_k$  la arista de peso mínimo de  $C_k$  y sea  $n_k$  el extremo de  $a_k$  que no pertenece a  $N_{k-1}$ . Hacer:

$$N_k = N_{k-1} \cup \{n_k\}$$

Paso 2. Si  $k < n - 1$  ir al paso 2. Si  $k = n - 1$ , terminar.

A continuación mostraremos como funciona el algoritmo con el siguiente ejemplo:

PASO 0

Elegimos un nodo arbitrariamente, por ejemplo el 4. Sea:

$$n_k = 4$$

$$k = 0$$

### 4.3.10 Pantalla Algoritmo de Primm

#### Pantalla Algoritmo de Primm

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO>> Menú Principal Menú Anterior Terminar

ALGORITMO DE PRIMM

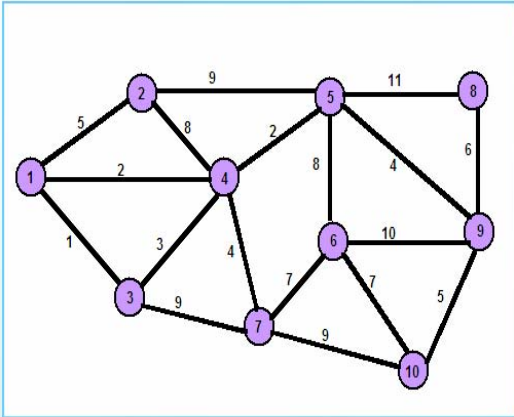
Paso 0. Sea  $n_k$  cualquier nodo de la red. Sea  $N_k = \{n_k\}$ . Hacer  $k = 0$ .

Paso 1. Hacer  $k = k + 1$ . Sea  $C_k$  el conjunto de aristas que tienen exactamente un extremo en  $N_{k-1}$ . Sea  $a_k$  la arista de peso mínimo de  $C_k$  y sea  $n_k$  el extremo de  $a_k$  que no pertenece a  $N_{k-1}$ . Hacer:  

$$N_k = N_{k-1} \cup \{n_k\}$$

Paso 2. Si  $k < n - 1$  ir al paso 1. Si  $k = n - 1$ , terminar.

A continuación mostraremos como funciona el algoritmo con el siguiente ejemplo:



PASO 0

Elegimos un nodo arbitrariamente, por ejemplo el 4. Sea:

$$n_0 = 4$$

$$k = 0$$

Al hacer clic sobre las flechas se pasa de una iteración a otra

#### Botones de flecha: iteraciones del algoritmo

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO>> Menú Principal Menú Anterior Terminar

ALGORITMO DE PRIMM

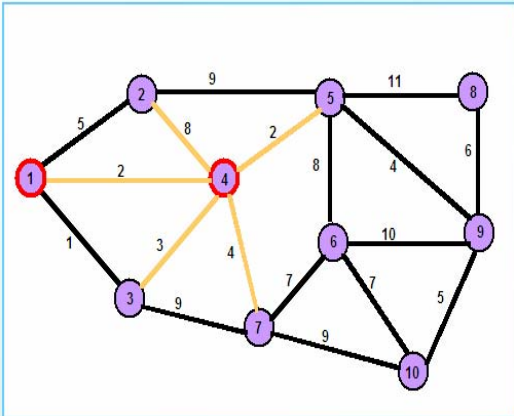
Paso 0. Sea  $n_k$  cualquier nodo de la red. Sea  $N_k = \{n_k\}$ . Hacer  $k = 0$ .

Paso 1. Hacer  $k = k + 1$ . Sea  $C_k$  el conjunto de aristas que tienen exactamente un extremo en  $N_{k-1}$ . Sea  $a_k$  la arista de peso mínimo de  $C_k$  y sea  $n_k$  el extremo de  $a_k$  que no pertenece a  $N_{k-1}$ . Hacer:  

$$N_k = N_{k-1} \cup \{n_k\}$$

Paso 2. Si  $k < n - 1$  ir al paso 1. Si  $k = n - 1$ , terminar.

A continuación mostraremos como funciona el algoritmo con el siguiente ejemplo:



ITERACIÓN 1

Hacemos:

$$N_0 = \{4\}$$

$$k = 1$$

$$C_1 = \{(2, 4), (1, 4), (3, 4), (4, 7), (4, 5)\}$$

$$a_k = (1, 4)$$

$$n_k = 1$$

$$N_1 = \{4, 1\}$$

### 4.3.11 Botón Menú Anterior de la Pantalla Algoritmo de Prim

Pantalla Algoritmo de Prim

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO>> Menú Principal Menú Anterior Terminar

ALGORITMO DE PRIMM

Paso 0. Sea  $n_k$  cualquier nodo de la red. Sea  $N_k = \{n_k\}$ . Hacer  $k = 0$ .

Paso 1. Hacer  $k = k + 1$ . Sea  $C_k$  el conjunto de aristas que tienen exactamente un extremo en  $N_{k-1}$ . Sea  $a_k$  la arista de peso mínimo de  $C_k$  y sea  $n_k$  el extremo de  $a_k$  que no pertenece a  $N_{k-1}$ . Hacer:  

$$N_k = N_{k-1} \cup \{n_k\}$$

Paso 2. Si  $k < n - 1$  ir al paso 1. Si  $k = n - 1$ , terminar.

A continuación mostraremos como funciona el algoritmo con el siguiente ejemplo:

PASO 0

Elegimos un nodo arbitrariamente, por ejemplo el 4. Sea:

$n_0 = 4$   
 $k = 0$

Al oprimir el botón 'Menú Anterior' se pasa a la pantalla principal de 'Árbol de peso mínimo'

Pantalla Árbol de peso mínimo

Software interactivo auxiliar de programación lineal y redes

REDES>> Menú Principal Menú Anterior Terminar

Algoritmo de Kruskal Algoritmo de Prim Ejercicios

ÁRBOL DE PESO MÍNIMO

Dada una red donde los nodos pueden representar ciudades, estaciones, puertos; y las longitudes o pesos de las aristas representar costos, distancias, tiempos; se debe construir un árbol de expansión de la red, para que entre todo par de nodos exista una trayectoria, tal que se minimice la suma de las longitudes (pesos) de las aristas.

Se define el **peso de un árbol**, precisamente como la suma de los pesos de las aristas que lo forman.

Por ejemplo, considérese la siguiente red, los nodos representan ciudades, los costos de construir una carretera de la ciudad x a la ciudad y se muestran sobre cada arco. Lo que se necesita es construir una red de carreteras para comunicar a las ciudades, de tal forma que el costo total sea mínimo. En esta sección se presentan algunos procedimientos para resolver este tipo de problemas.

Ver solución

Al oprimir el botón 'Ver solución' se muestra la solución del problema.

### 4.3.12 Botón: Ejercicios del Menú Árbol de peso mínimo

Pantalla Árbol de peso mínimo

Software interactivo auxiliar de programación lineal y redes

REDES>> Menú Principal Menú Anterior Terminar

Algoritmo de Kruskal Algoritmo de Primm **Ejercicios**

ÁRBOL DE PESO MÍNIMO

Dada una red donde los nodos pueden representar ciudades, estaciones, puertos; y las longitudes o pesos de las aristas representar costos, distancias, tiempos; se debe construir un árbol de expansión de la red, para que entre todo par de nodos exista una trayectoria, tal que se minimice la suma de las longitudes (pesos) de las aristas.

Se define el **peso de un árbol**, precisamente como la suma de los pesos de las aristas que lo forman.

Por ejemplo, considérese la siguiente red, los nodos representan ciudades, los costos de construir una carretera de la ciudad x a la ciudad y se muestran sobre cada arco. Lo que se necesita es construir una red de carreteras para comunicar a las ciudades, de tal forma que el costo total sea mínimo. En esta sección se presentan algunos procedimientos para resolver este tipo de problemas.

Ver solución

Al oprimir el botón 'Ejercicios' se pasa a la pantalla

Pantalla Ejercicios

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO>> Menú Principal Menú Anterior Terminar

EJERCICIOS

Ejercicio 1 ◀ ▶ Un aserreadero iniciará la tala de árboles en cinco bosques en una región. Debe desarrollar un sistema de brechas que haga a los bosques accesibles desde otros bosques. La distancia en kilómetros entre cada par de bosques se da a continuación. El problema es determinar entre qué pares de bosques se deben construir las brechas para conectarlos a todos, de tal manera que se obtenga un total mínimo de camino.

La suma es: 0 La solución es: 0

Limpia

Checar

Haz clic sobre cada arco para formar el árbol de peso mínimo.

### 4.3.13 Pantalla Ejercicios

#### Pantalla Ejercicios

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO >> Menú Principal Menú Anterior Terminar

EJERCICIOS

Ejercicio 1 Un aserreadero iniciará la tala de árboles en cinco bosques de una región. Debe desarrollar un sistema de brechas que haga a los bosques accesibles desde otros bosques. La distancia en kilómetros entre cada par de bosques se da a continuación. El problema es determinar entre qué pares de bosques se deben construir las brechas para conectarlos a todos, de tal manera que se obtenga un total mínimo de camino.

La suma es: 0 La solución es: 0

Limpiar

Checar

Haz clic sobre cada arco para formar el árbol de peso mínimo.

El botón 'Checar' sirve para verificar que la solución propuesta sea la correcta.

#### Ejercicio 1 resuelto

Software interactivo auxiliar de programación lineal y redes

ÁRBOL DE PESO MÍNIMO >> Menú Principal Menú Anterior Terminar

EJERCICIOS

Ejercicio 1 Un aserreadero iniciará la tala de árboles en cinco bosques de una región. Debe desarrollar un sistema de brechas que haga a los bosques accesibles desde otros bosques. La distancia en kilómetros entre cada par de bosques se da a continuación. El problema es determinar entre qué pares de bosques se deben construir las brechas para conectarlos a todos, de tal manera que se obtenga un total mínimo de camino.

La suma es: 82 La solución es: CORRECTA

Limpiar

Checar

Haz clic sobre cada arco para formar el árbol de peso mínimo.

## 5. BIBLIOGRAFÍA

[1] Bazaraa, M.S., Jarvis, J.J. y Sheraly, H.D., "*Programación Lineal y Flujo en Redes*", 2ª edición, Limusa, 2004.

[2] Hernández, M.C., "*Introducción a la teoría de redes*", Sociedad Matemática Mexicana, 1997.

[3] Hillier, F.S. y Lieberman, G.J., "*Investigación de Operaciones*", 7ª edición, McGraw-Hill, 2002.

[4] Taha, H.A., "*Investigación de Operaciones*", 7ª edición, Pearson Educación, 2004.