



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

Facultad de Ingeniería

SOFTWARE WINQSB APLICACIONES PRÁCTICAS EN LA
INVESTIGACIÓN DE OPERACIONES

TESIS

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO INDUSTRIAL

PRESENTA:
YAIR ABRAHAM BAZÁN TINAJERO

DIRECTOR DE TESIS:
M. en I. SILVINA HERNÁNDEZ GARCÍA.

CIUDAD UNIVERSITARIA 2009

A una maravillosa persona que con su grandeza me hizo ver lo maravilloso que es vivir.

A mis padres José Bazán Carranza y Guadalupe Tinajero Rodríguez por su amor y apoyo incondicional.

A mis hermanos Raúl y Erick porque son parte de mi motivación para continuar mis proyectos.

A la Universidad Nacional Autónoma de México por albergar mis sueños y hacerlos realidad.

A mis amigos que me hacen aún más hermoso el viaje de la vida.

A mis profesores:

Dr. Marco Antonio Murray Lasso

M. en C. Benito Marín Pinillos

M. en A. Bonifacio Román Tapia

M. en A. Víctor Manuel Rivera Romay

En especial a M. en I. Silvina Hernández García.

Agradezco a Dios por guiar mi mano al escribir la parte más difícil de mi tesis. Ésta.

ΙΤΑΚΑ

*Cuando emprendas tu viaje hacia Ítaca
debes rogar que el viaje sea largo,
lleno de peripecias, lleno de experiencias.
No has de temer ni a los lestrigones ni a los cíclopes,
ni la cólera del airado Poseidón.
Nunca tales monstruos hallarás en tu ruta
si tu pensamiento es elevado, si una exquisita
emoción penetra en tu alma y en tu cuerpo.
Los lestrigones y los cíclopes
y el feroz Poseidón no podrán encontrarte
si tú no los llevas ya dentro, en tu alma,
si tu alma no los conjura ante ti.
Debes rogar que el viaje sea largo,
que sean muchos los días de verano;
que te vean arribar con gozo, alegremente,
a puertos que tú antes ignorabas.
Que puedas detenerte en los mercados de Fenicia,
y comprar unas bellas mercancías:
madreperlas, coral, ébano, y ámbar,
y perfumes placenteros de mil clases.
Acude a muchas ciudades del Egipto
para aprender, y aprender de quienes saben.
Conserva siempre en tu alma la idea de Ítaca:
llegar allí, he aquí tu destino.
Mas no hagas con prisas tu camino;
mejor será que dure muchos años,
y que llegues, ya viejo, a la pequeña isla,
rico de cuanto habrás ganado en el camino.
No has de esperar que Ítaca te enriquezca:
Ítaca te ha concedido ya un hermoso viaje.
Sin ella, jamás habrías partido;
mas no tiene otra cosa que ofrecerte.
Y si la encuentras pobre, Ítaca no te ha engañado.
Y siendo ya tan viejo, con tanta experiencia,
sin duda sabrás ya qué significan las Ítacas.*

ESTRUCTURA DE LA TESIS

Este trabajo se encuentra integrado por cuatro partes de manera general, la primera es una breve introducción a la Investigación de Operaciones; posteriormente el contenido se estructura en tres capítulos los cuales se describen a continuación:

El Capítulo I es una Introducción al manejo del software WINQSB, en él se mencionan las especificaciones técnicas necesarias del ordenador que se pretende usar para ejecutar el programa correctamente, se exponen también los módulos que conforman su estructura y se familiariza al lector con las ventanas que son comunes a todos los módulos que integran el paquete WINQSB; finalmente se da una breve descripción del concepto de modelo matemático que servirá como introducción al Capítulo II.

El Capítulo II estudia las aplicaciones referentes a la Investigación de Operaciones I. Se abordan temas como el Método Simplex y sus complicaciones, Teoría de redes y Gestión de proyectos. En este capítulo se hace uso de los siguientes tres módulos:

1. Programación Lineal y Entera (Linear and Integer programming)
2. Modelación de redes (Network Modeling)
3. PERT y CPM (PERT_CPM)

Finalmente el Capítulo III se enfoca a la Investigación de Operaciones II en el que se muestran las aplicaciones en temas como las Cadenas de Markov, Teoría de Colas y Simulación de Sistemas de Colas. En este capítulo se hace uso de:

1. Procesos de Markov (Markov Process)
2. Análisis de Sistemas de cola (Queuing Analysis)
3. Simulación de sistemas de cola (Queuing Analysis Simulation)

Cabe señalar que no se incluye la totalidad de los temas que contempla el programa de estudios vigente para las asignaturas de Investigación de Operaciones de la carrera de Ingeniería Industrial en la Facultad de Ingeniería de la Universidad Nacional Autónoma de México.

El Objetivo fundamental del presente trabajo es:

Explicar las herramientas básicas que contempla el software WINQSB incluyendo aplicaciones prácticas en la Investigación de Operaciones, que puedan servir como guía para estas asignaturas en el Laboratorio, contribuyendo a la formación profesional del futuro Ingeniero Industrial.

Se incluye un Anexo con ejercicios propuestos para todo aquel que desee practicar y tener un mayor dominio del programa; los ejercicios están basados en apuntes de clase y algunos otros son de elaboración propia con fuente en Murray (2007), Pinillos (1994) y Colmenero (2009).

La elección del software se debe principalmente a que WINQSB es un programa al alcance de cualquier institución o persona¹ y se encuentra disponible en el equipo de cómputo del laboratorio de Ingeniería Industrial.

El lector encontrará un valor agregado ya que el uso del software le va a permitir agilizar el estudio de algunas técnicas de la Investigación de Operaciones, ahorrándole tiempo apreciable en el manejo de datos y permitiéndole analizar de una mejor manera la información para la toma de decisiones. Se pretende que esta tesis constituya un aporte significativo a la enseñanza de estas importantes asignaturas.

Es pertinente mencionar que todo aquel que consulte este trabajo debe tener las bases teóricas de los módulos que aquí se tratan. No es pretensión realizar un libro de Investigación de Operaciones en el que se expliquen los conceptos de manera exhaustiva; por tanto se asume que el lector cuenta con la formación básica en estas disciplinas. Se ofrece sin embargo, la bibliografía de referencia utilizada para el desarrollo de este trabajo que puede ser consultada por cualquiera que desee tener una mejor comprensión de los conceptos.

Las traducciones para describir las herramientas de cada uno de los módulos son elaboración propia con fuente en Quesada *et al.*, (2008). Agradezco profundamente las revisiones de los sinodales: Dr. Marco Antonio Murray Lasso, M. en C. Benito Marín

¹ El lector interesado puede descargar el software en: http://dimei.fi-b.unam.mx/industrial/catedra_software.html

INDICE

BREVE INTRODUCCIÓN A LA INVESTIGACIÓN DE OPERACIONES

CAPÍTULO I.- INTRODUCCIÓN AL MANEJO DEL WINQSB

I-1.1 DESCRIPCIÓN DE WINQSB.....	1
1-1.2 CONCEPTO DE MODELO MATEMÁTICO.....	7
I.1.2.1 EJEMPLO DE FORMULACIÓN DE UN MODELO MATEMÁTICO.....	10

CAPÍTULO II.- INVESTIGACIÓN DE OPERACIONES I

II-1 PROGRAMACIÓN LINEAL.....	17
II-1.2 COMPLICACIONES DEL MÉTODO SIMPLEX.....	29
II-1.2.1 MINIMIZACIÓN.....	30
II-1.2.2 DESIGUALDAD CON SENTIDO INVERTIDO.....	35
II-1.2.3 CONSTANTES NEGATIVAS ($b_i < 0$).....	35
II-1.2.4 IGUALDADES.....	38
II-1.2.5 VARIABLES NO RESTRINGIDAS EN SIGNO.....	39
II-1.2.6 EMPATE PARA ENTRAR CON VARIABLES BÁSICAS.....	39
II-1.2.7 EMPATE PARA DEJAR LA BASE (DEGENERACIÓN).....	40
II-1.2.8 SOLUCIONES MÚLTIPLES.....	42
II-1.3 OBTENCIÓN DEL MODELO DUAL	45
II-2 MODELO DE REDES.....	50
II-2.1 MODELO DE TRASBORDO.....	52
II-2.2 PROBLEMA DE TRANSPORTE.....	59
II-2.3 PROBLEMA DE ASIGNACIÓN.....	72
II-2.4 PROBLEMA DE LA RUTA MÁS CORTA.....	77
II-2.5 PROBLEMA DE FLUJO MÁXIMO.....	94
II-2.6 MÍNIMO ÁRBOL DE EXPANSIÓN.....	98
II-2.7 PROBLEMA DEL AGENTE VIAJERO.....	102
II-3 PERT – CPM.....	106
II-3.1 CPM.....	109
II-3.2 PERT.....	115

CAPÍTULO III- INVESTIGACIÓN DE OPERACIONES II

III-1 PROCESO DE MARKOV.....	119
III-2 TEORÍA DE COLAS.....	132
III-3 SIMULACIÓN DE COLAS.....	146

CONCLUSIONES

ANEXO

BREVE INTRODUCCIÓN A LA INVESTIGACIÓN DE OPERACIONES

La Ingeniería Industrial ha evolucionado en gran medida, anteriormente se le relacionaba con la producción y el estudio de tiempos y movimientos; mientras algunas otras ingenierías eran especialistas en su área, no existía en ese entonces personal preparado que aparte de entender los términos de las otras disciplinas ingenieriles, pudiera controlar administrativamente tales procesos. El Ingeniero Industrial se ha convertido en la persona encargada del control y la optimización de los procesos productivos, tareas que normalmente no realizan las otras especialidades. Otra característica importante del Ingeniero Industrial es que está en contacto directo con la alta dirección, gerentes, administradores, contadores, asesores financieros, entre otros; por lo que se considera una ciencia interdisciplinaria.

Una parte muy importante de la Ingeniería industrial y en cuya teoría se ha basado este trabajo es la Investigación de Operaciones o Investigación Operativa. En sus comienzos, la investigación de operaciones se refería a sistemas existentes de armas y a través del análisis típicamente matemático, se buscaban las políticas óptimas para la utilización de esos sistemas. El éxito de la investigación de operaciones en la esfera militar quedó bastante bien documentado hacia finales de la Segunda Guerra Mundial.

Para que un análisis de un sistema militar de operaciones fuera tecnológicamente factible, era necesario tener una comprensión técnica adecuada, que tomara en cuenta todas las subcomponentes del sistema. En consecuencia, el trabajo de equipo resultó ser tan necesario como efectivo. Churchman y Ackoff² escribieron numerosos libros sobre la aplicación de sus técnicas en beneficio de la Humanidad y de la Sociedad fuera del ámbito militar.

La Investigación Operativa es una moderna disciplina científica que se caracteriza por la aplicación de teoría, métodos y técnicas especiales, para buscar la solución de problemas de administración, organización y control que se producen en los diversos sistemas que existen en la naturaleza y los creados por el ser humano, tales como las

² Ackoff dio en ocasiones asesoramiento al Gobierno de México y en la Universidad Nacional Autónoma de México sobre métodos para mejorar la economía y la educación del pueblo de Perú, los mismos que describió en sus "Fábulas de Ackoff"

organizaciones a las que identifica como sistemas organizados, sistemas físicos, económicos, ecológicos, educacionales, de servicio social, etc.

El objetivo más importante de la aplicación de la Investigación Operativa es apoyar en la toma de decisiones en los sistemas y en la planificación de sus actividades.

Finalmente debe ponerse la máxima atención en no considerar la Investigación de Operaciones como una colección de recetas heterogéneas y aplicables sistemáticamente en situaciones determinadas, si se cae en este error, será muy difícil captar en condiciones reales los problemas que puedan deducirse de los múltiples aspectos de esta disciplina.



CAPÍTULO I.- INTRODUCCIÓN AL MANEJO DEL WINQSB

Si alguien aprende de otros pero no razona, estará desconcertado.

Si alguien razona pero no aprende de otros, estará en peligro.

Confucio

I-1.1 DESCRIPCIÓN DE WINQSB

WINQSB es un programa interactivo de ayuda a la toma de decisiones, contiene herramientas muy útiles para resolver distintos tipos de problemas en el campo de la investigación operativa. Utiliza los mecanismos típicos de la interface de Windows; es decir, ventanas, menús desplegables, barras de herramientas, etc. Por lo tanto el manejo del programa es similar a cualquier otro que utilice el entorno Windows. Para una adecuada instalación del software, y un mejor aprovechamiento de todos los recursos que ofrece, la computadora en la que se estudie deberá cumplir con las siguientes especificaciones:

- Windows 95, 98, Me, 2000 Professional, XP Edition o Windows Vista.
- Al menos 10 [Mb] libres en la memoria del disco duro
- Al menos 36 [Mb] en la memoria RAM
- Monitor a color



El software está integrado por 19 módulos, uno para cada tipo de modelo o problema.

1. Análisis de Muestreo de Aceptación (Acceptance Sampling Analysis)
2. Planeación Agregada (Aggregate Planning)
3. Análisis de Decisiones (Decision Analysis)
4. Programación Dinámica (Dynamic Programming)
5. Diseño y Localización *de Sistemas de Producción* (Facility Location and Layout)
6. Pronósticos (Forecasting)
7. Programación por Objetivos (Goal Programming)
8. Teoría y Sistemas de Inventarios (Inventory Theory and System)
9. Programación de Jornadas de trabajo (Job Scheduling)
10. Programación Lineal y Entera (Linear and Integer programming)
11. Procesos de Markov (Markov Process)
12. Planeación de Requerimientos de Materiales (Material Requirements Planning)
13. Modelación de redes (Network Modeling)
14. Programación no lineal (Nonlinear Programming)
15. PERT y CPM (PERT_CPM)
16. Programación cuadrática (Quadratic Programming)
17. Cartas de control de calidad (Quality Control Chart)
18. Análisis de Sistemas de cola (Queuing Analysis)
19. Simulación de Sistemas de cola (Queuing Analysis Simulation)

Una vez realizada la instalación en la computadora podemos tener acceso a las aplicaciones de WINQSB¹.

Es posible acceder al programa a través del botón INICIO del sistema operativo WINDOWS, dirigirse al submenú PROGRAMAS y dar clic en la carpeta WINQSB; la ubicación depende de la dirección que haya sido señalada por el usuario, si se desea se puede colocar un acceso directo en el escritorio de la computadora para facilitar la ruta de entrada.

¹ La versión utilizada en este trabajo es WINQSB 2.0.



Todos los módulos del programa tienen en común los siguientes menús desplegables:

- Archivo (File): Incluye las opciones típicas de este tipo de menús en Windows, es decir, permite crear y salvar ficheros con nuevos problemas, leer otros ya existentes o imprimirlos.
- Editar (Edit): Incluye las utilidades típicas para editar problemas, copiar, pegar, cortar o deshacer cambios. También permite cambiar los nombres de los problemas, las variables, y las restricciones. Facilita la eliminación o adición de variables y/o restricciones, y permite cambiar el sentido de la optimización.
- Formato (Format): Incluye las opciones necesarias para cambiar la apariencia de las ventanas, colores, fuentes, alineación, anchura de celdas, etc.
- Resolver y Analizar (Solve and Analyze): Esta opción incluye al menos dos comandos, dependiendo del módulo en el que se esté trabajando, uno para resolver el problema y otro para resolverlo siguiendo los pasos del algoritmo, esta opción es muy interesante ya que nos muestra las iteraciones paso a paso.
- Resultados (Results): Incluye las opciones para ver las soluciones del problema y verla de distintos modos, como la tabla resumen de las operaciones, ver la solución gráfica, por mencionar algunas.
- Utilidades (Utilities): Este menú permite acceder a una calculadora, a un reloj y a un editor de gráficas.
- Ventana (Window): Permite navegar por las distintas ventanas que van apareciendo al operar con el programa.
- WinQSB: Incluye las opciones necesarias para acceder a otro módulo del programa.



- Ayuda (Help): Permite acceder a la ayuda *en línea* sobre la utilización del programa o las técnicas utilizadas para resolver los distintos modelos; también proporciona información sobre cada una de las ventanas en la que nos encontramos, y despliega una descripción que nos sirve para orientarnos.

Al acceder a cualquiera de los módulos se abre una ventana en la que debemos elegir entre crear un nuevo problema (File > New Problem) o leer uno ya creado (File > Load Problem). Las extensiones de los ficheros con los modelos las pone el programa por defecto, por lo tanto solamente debemos preocuparnos del nombre, que no deberá tener más de 8 caracteres. Para cada uno de los módulos aparecerá la ventana (Figura 1.1) y las características descritas anteriormente son similares para todos los módulos.

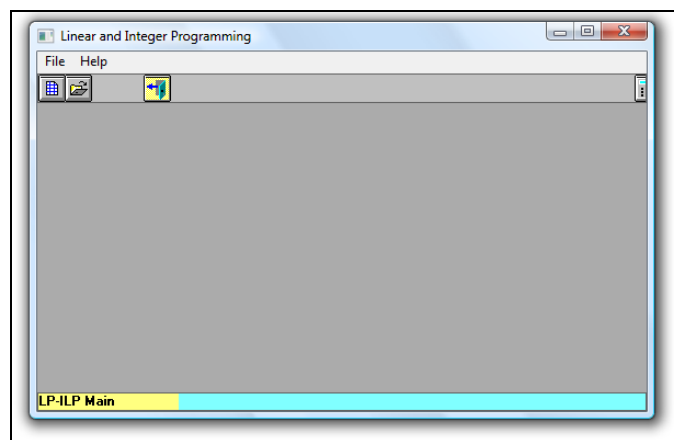


Figura 1.1

La parte superior de la ventana indica el nombre del módulo que hemos seleccionado, en este caso Programación Lineal y Entera (Linear and integer programming). Debajo de esta barra encontramos los menús Archivo (File) y Ayuda (Help). El menú archivo (Figura 1.2) comprende las siguientes opciones:

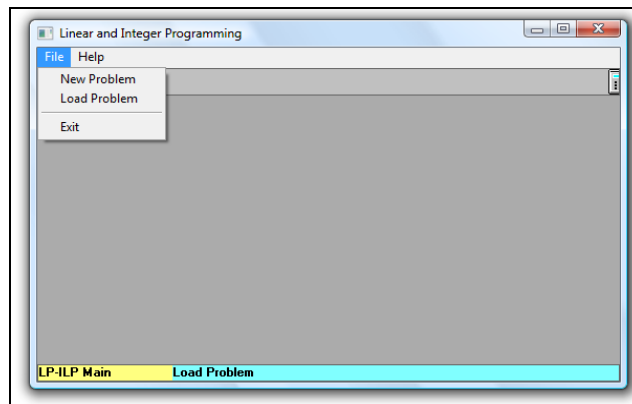


Figura 1.2

- Nuevo problema (New Problem): Permite introducir un nuevo problema.
- Abrir Problema (Load Problem): Abre un problema que se ha guardado con anterioridad.
- Salir (Exit): Sale del programa.

El menú Ayuda (Figura 1.3) lo conforman las opciones que se describe a continuación:

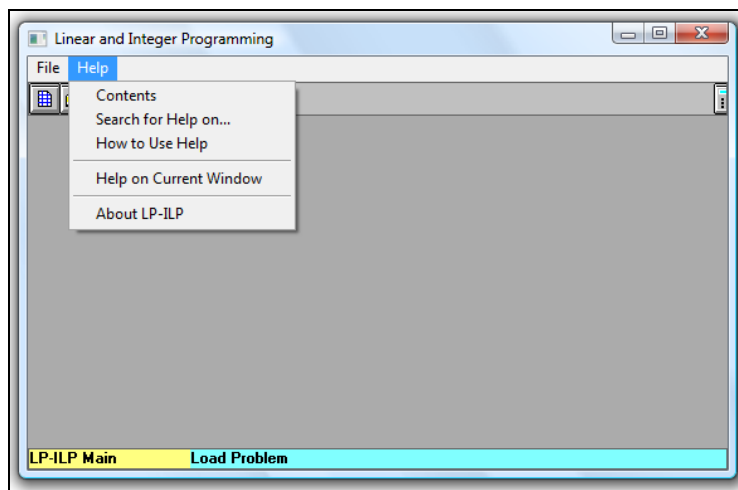


Figura 1.3

- Contenido (Contents): Permite visualizar un contenido completo de la ayuda sobre el módulo seleccionado.
- Buscar ayuda en... (Search for Help on...): Búsqueda de ayuda mediante palabras claves.






- **Cómo usar la ayuda (How to Use Help):** Muestra información de cómo se utiliza la ayuda para obtener el máximo beneficio.
- **Ayuda sobre la ventana actual (Help on Current Windows):** Muestra la ayuda sólo sobre los elementos que aparecen actualmente en la ventana.
- **Acerca de... (About LP-ILP):** Muestra datos sobre la creación del programa e información sobre la licencia.

El programa cuenta con una barra de herramientas en todos los módulos, misma que nos permite acceder a las opciones más usadas (Figura 1.4).



Figura 1.4

El primer botón  permite la creación de un nuevo problema. El segundo botón  abre un problema existente, mientras que el tercero  permite salir del programa o regresar a la ventana anterior.



I-1.2 CONCEPTO DE MODELO MATEMÁTICO

Diariamente nos enfrentamos a situaciones en las cuales los resultados no son siempre los mejores, ante esta perspectiva se encaminan acciones en busca de alternativas que ayuden a elegir de entre ellas la óptima, "¿Cómo podemos explicar que las matemáticas, un producto de la mente humana independiente de la experiencia, encajen tan bien en los objetos y elementos de la realidad?" Albert Einstein (1938).

Modelar puede ser entendido simultáneamente como ciencia y arte. Es una ciencia porque se basa en un conjunto de procesos estructurados, basando su análisis en el método científico; y es un arte porque materializa una visión o interpretación de la realidad, no siempre de manera unívoca.

La modelación matemática, es el proceso mediante el cual un conjunto de símbolos (que se definen de acuerdo al contexto en el que está situado el caso de estudio) y relaciones matemáticas traducen un fenómeno o problema de una situación real. De esta forma, el modelado es "el arte de aplicar las matemáticas a la vida real" Mogen (1991). Es importante señalar que el modelo de una situación real puede tener diferencias en la definición y el planteamiento del problema, pues dependerá de la percepción y de la apreciación del traductor. Si bien un primer modelo puede no convencer al investigador, este puede replantearlo tomando como referencia el conocimiento previo. "La modelación es un proceso iterativo en que una etapa de validación frecuentemente lleva a diferencias entre las predicciones basadas en el modelo y la realidad" O'Shea y Berry (1982).

De esta forma un modelo nace como un concepto de un fenómeno en particular; esta nueva concepción de la *realidad* es abstraída de la naturaleza a un lenguaje simbólico. Podemos concebir a un modelo matemático como una metáfora; por ejemplo, en el área de las Humanidades, particularmente en el mundo de las letras, un poeta *recrea* su percepción de la vida muy de acuerdo a las emociones que sienta en el instante de inspiración; las palabras (grafos definidos en un espacio y tiempo reconocido socialmente como un lenguaje) serán los símbolos que servirán como un *intérprete* entre el fascinante universo de las emociones humanas y el *mundo real*. Para el campo de



la Ingeniería, la modelación matemática consiste en el reemplazo del objeto cognitivo por su imagen matemática (modelo matemático). El diccionario de la Real Academia Española define modelo como: Esquema teórico generalmente en forma matemática, de un sistema o de una realidad compleja, que se elabora para facilitar su comprensión y el estudio de su comportamiento. Así mismo implica la relación entre dos figuras no necesariamente representadas por personas únicas; el modelador (encargado de la especificación y desarrollo del modelo) y el experto sobre la realidad (conocedor del problema real).

El objetivo de la modelación matemática es entender ampliamente el fenómeno de estudio y de ser posible predecir su comportamiento en el futuro. Podemos encontrar una amplia gama de bibliografía sobre el tema, existen muchos libros de ciencias como la Investigación Operativa, la Física y la Química; que en general proponen ciertos pasos para la construcción de modelos matemáticos; sin embargo en esta tesis se proponen *siete más una* etapas que componen el desarrollo de un modelo matemático y su solución, las cuales se describen a continuación:

1. Elección del tema o caso de estudio. Se debe partir de un problema del mundo real que resulte de interés particular para el investigador.
2. Identificar el problema. Se realiza la definición del problema de forma clara, en esta etapa (dependiendo de la complejidad del fenómeno) se debe tener una gran capacidad de análisis y/o síntesis (dependiendo del carácter del problema: Mecánico, Social, Operativo, etc.). Esto va a permitir la identificación de las variables más relevantes.
3. Familiarización con el tema a ser modelado (referencial teórico). Consiste en la recolección de la información necesaria para conocer a fondo el tema de estudio; los problemas reales suelen estar definidos en términos vagos e imprecisos, los datos son vitales para conseguir un realismo o aplicabilidad en las soluciones y a menudo representan el cuello de botella del proceso de modelado.



4. Formulación del modelo matemático. Gradualmente la abstracción de la realidad se da a través de procesos tales como: Hacer supuestos sobre cuáles aspectos del problema son importantes, posteriormente la generalización del problema y finalmente su formalización. Llevadas a cabo estas consideraciones realizamos la escritura matemática del problema, definiendo sus variables, sus ecuaciones, su función objetivo, sus parámetros, los objetos, los atributos de dichos objetos, etcétera. Puede haber diferentes implantaciones de un mismo caso de estudio y el tiempo de solución de un problema puede depender drásticamente de cómo esté formulado.

5. Resolución del problema a partir del modelo. Se trata de implementar un algoritmo de obtención de la solución numérica. Puede haber diferentes métodos de solución de un problema o diferentes implantaciones de un mismo método.

6.- Validar el modelo. Se hace conciencia de la solución matemática en términos de la situación real. Esta etapa conlleva la eliminación de los errores de codificación, es decir, conseguir que el modelo haga lo que se desea (depurar y verificar). Es necesario comprobar la validez de las simplificaciones realizadas a través de los resultados obtenidos.

7. Interpretación y análisis de los resultados. Esta etapa consiste en proponer soluciones y permite conocer en detalle el comportamiento del modelo al hacer un análisis de sensibilidad en los parámetros de entrada, estudiar diferentes escenarios plausibles de los parámetros y detectar soluciones alternativas cuasi-óptimas, pero suficientemente atractivas.

La última etapa del proceso; se considera que no forma parte de la modelación matemática en sentido estricto; sin embargo en la práctica la documentación sirve como soporte de una nueva propuesta; esta etapa ha sido denominada como *Documentación*, y es fundamental en el desarrollo de un modelo para garantizar su amplia difusión, la documentación ha de ser clara, precisa y completa; y se debe incluir en un documento la especificación técnica funcional, matemática e informática que se ha utilizado.



I.1.2.1 EJEMPLO DE FORMULACIÓN DE UN MODELO MATEMÁTICO

En el ejemplo siguiente se muestra un caso de aplicación de la formulación de un modelo matemático bastante simple en el cual se utiliza la metodología propuesta anteriormente. La empresa SEA. S. A., desea conocer la cantidad de productos a, b y c a producir para maximizar el beneficio, si cada unidad vendida genera en utilidad \$1500, \$2100 y \$1300 por unidad respectivamente. Cada producto pasa por tres estaciones de trabajo, restringiendo la cantidad de unidades producidas debido al tiempo disponible en cada una de ellas. La siguiente Tabla 1-1 muestra el *tiempo estándar* requerido por unidad de cada producto en cada estación y el tiempo total disponible semanalmente (tiempo dado en minutos).

Tabla **¡Error! No hay texto con el estilo especificado en el documento.-1.** Datos del problema

	Tiempo requerido estación 1	Tiempo requerido estación 2	Tiempo requerido estación 3
Producto a	10	12	8
Producto b	15	17	9
Producto c	7	7	8
Tiempo total disponible semanalmente por Estación (minutos)	3300	3500	2900



Se hace la suposición de que cada unidad producida es vendida automáticamente sin ningún costo adicional. Determinar la combinación de productos que maximicen la utilidad para la compañía. Siguiendo las etapas para el desarrollo de la modelación matemática propuestas anteriormente se tiene lo siguiente:

1. Elección del tema o caso de estudio. Se parte de un problema del mundo real. El problema ha sido tomado de una empresa mexicana, pero por razones de confidencialidad se ha cambiado el nombre de la misma y de los productos que elabora.
2. Identificar el problema. La empresa SEA S. A. desea conocer la cantidad de productos a , b y c a producir para maximizar el beneficio.
3. Familiarización con el tema a ser modelado (referencial teórico). El problema se ha definido claramente. Se ha buscado en la bibliografía y se ha confirmado que el problema puede ser modelado y resuelto por el algoritmo del Método Simplex Primal.
4. Formulación de un modelo matemático. Una vez que se ha analizado el problema, se procede a realizar el modelo matemático que lo representa. Sean:

X_1 = Unidades a producir del producto a

X_2 = Unidades a producir del producto b

X_3 = Unidades a producir del producto c

Función Objetivo: $\text{Max. } Z = \$1500X_1 + \$2100X_2 + \$1300X_3$

Restricciones (Sujeto a):

$$10X_1 + 15X_2 + 7X_3 \leq 3300 \text{ Minutos}$$

$$12X_1 + 17X_2 + 7X_3 \leq 3500 \text{ Minutos}$$

$$8X_1 + 9X_2 + 8X_3 \leq 2900 \text{ Minutos}$$

$$(X_1, X_2, X_3) \geq 0 \text{ (Restricción de no negatividad)}$$



Podemos ver claramente que estamos ante un problema de Maximización, con tres restricciones y tres variables.

5. Resolución del problema a partir del modelo. Se resolverá el problema con ayuda del módulo de Programación Lineal y Entera del software WINQSB 2.0. La descripción exhaustiva de cómo se resuelve el problema paso a paso con el programa se desarrolla en el Capítulo II. La Tabla 1-2 muestra la matriz resultante.

	14:12:29		Monday	November	10	2008
	Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status
1	a	0	1,500.0000	0	-149.3151	at bound
2	b	105.4795	2,100.0000	221,506.9000	0	basic
3	c	243.8356	1,300.0000	316,986.3000	0	basic
	Objective	Function	(Max.) =	538,493.2000		
	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price
1	C1	3,289.0410	<=	3,300.0000	10.9589	0
2	C2	3,500.0000	<=	3,500.0000	0	69.8630
3	C3	2,900.0000	<=	2,900.0000	0	101.3699

Tabla 1-2. Matriz resultante

6.- Validar el modelo. La columna Valores de la solución (Solution Value) presenta los valores óptimos encontrados. En este ejemplo se tiene que a= 0 unidades, b= 105.4795 unidades y c= 243.8356 unidades. La columna Costo o Utilidad Unitaria (Unit Cost or Profit c(j)) muestra los coeficientes de la función objetivo para cada variable. La columna Contribución Total (Total Contribution) representa el costo o utilidad generado por cada variable. Por ejemplo, si el valor de la variable b es 105.4795 unidades y la utilidad unitaria es \$2100, el beneficio total resultará de la multiplicación de ambos valores dando como resultado \$221,506.9 Justo debajo de la última contribución aparece el valor de Z óptimo (\$538,493.2). La columna Costo Reducido (Reduced Cost) identifica el costo que genera incrementar una unidad para cada variable no básica. La siguiente columna llamada Estatus de la Variable (Basis Status) muestra el estado de las variables a, b y c; es decir si es básica (Basic) o en el límite (at bound). La parte inferior de la matriz muestra la primera columna (Constraint), presenta las



variables de holgura del sistema (C1, C2, C3); la columna Lado de la mano izquierda (Left Hand Side) muestra el valor alcanzado al reemplazar los valores de a, b y c en cada restricción (es importante recordar que cada restricción se identifica con su variable de holgura). Las dos columnas siguientes (Direction y Right Hand Side) muestran las especificaciones dadas a las restricciones en cuanto al operador de relación (\leq) y los valores originales de las restricciones (3300, 3500 y 2900 minutos). La columna Holgura (Déficit) o Superávit (Slack or Surplus) muestran los valores de las variables de holgura. Finalmente la columna Precios Sombras (Shadow Price) corresponde a cuánto se estaría dispuesto a pagar por una unidad adicional de cada recurso. Los valores óptimos de las variables duales proporcionan una interpretación e información económica potencial para la toma de decisiones de una empresa, esta es proporcionada por los precios sombra o marginales o de equilibrio, los cuales permiten determinar si resulta ventajoso adquirir recursos adicionales sin necesidad de resolver nuevamente el problema; facilita el estudio del impacto sobre la optimalidad por cambios en los coeficientes del problema (análisis de sensibilidad). El precio sombra es la cantidad que mejora el valor óptimo de la función objetivo en el problema dual (un incremento de maximización o un decremento en un problema de minimización). En el renglón anterior se aclara la importancia de la última columna que hace referencia a los precios sombras y que tienen una estrecha relación con las variables duales. El lector interesado puede consultar la bibliografía propuesta en la que se explica de manera exhaustiva el concepto de dualidad, se recomienda en especial Estrada (2009).

7. Interpretación y análisis de los resultados. En el punto anterior se aprecia de manera clara que el modelo utilizado es correcto permite interpretar los resultados de una manera coherente; sin embargo se resolverá el modelo por pasos para revisar el cómputo hecho por la máquina. La primera tabla 1-3 corresponde a la tabla inicial del Simplex.

Basis	C(j)	a	b	c	Slack_C1	Slack_C2	Slack_C3	R. H. S.	Ratio
Slack_C1	0	10.0000	15.0000	7.0000	1.0000	0	0	3,300.0000	220.0000
Slack_C2	0	12.0000	17.0000	7.0000	0	1.0000	0	3,500.0000	205.8824
Slack_C3	0	8.0000	9.0000	8.0000	0	0	1.0000	2,900.0000	322.2222
	C(j)-Z(j)	1,500.0000	2,100.0000	1,300.0000	0	0	0	0	

Tabla ;Error! No hay texto con el estilo especificado en el documento.-3. Tabla inicial del método Simplex



Al pulsar sobre la opción Próxima Iteración (Next Iteration) se avanza a la siguiente tabla del Simplex, que es mostrada en la Tabla 1-4.

Basis	C(j)	a	b	c	Slack_C1	Slack_C2	Slack_C3	R. H. S.	Ratio
Slack_C1	0	-0.5882	0	0.8235	1.0000	-0.8824	0	211.7647	257.1429
b	2,100.0000	0.7059	1.0000	0.4118	0	0.0588	0	205.8824	500.0000
Slack_C3	0	1.6471	0	4.2941	0	-0.5294	1.0000	1,047.0590	243.8356
	C(j)-Z(j)	17.6471	0	435.2941	0	-123.5294	0	432,352.9000	

Tabla ¡Error! No hay texto con el estilo especificado en el documento.-4. Siguiete iteración

Es claro que se obtiene el mismo resultado, por lo que se corrobora que el modelo es correcto; sin embargo cabe mencionar, que se supuso que los tiempos para la elaboración de cada uno de los productos (a, b y c) son *tiempos estándar* y se aceptaron como válidos. Es pertinente sugerir al lector algunos conceptos que contribuyen a ampliar su criterio para el desarrollo de modelos de esta índole; ya que en la práctica surgirán las siguientes interrogantes: ¿cómo sabemos cuál es el tiempo teórico con el cual se debe hacer el modelo matemático? y ¿cómo sabremos que nuestros resultados son válidos?

El software sólo devuelve resultados que se basan en un algoritmo diseñado por el programador, pero si introducimos datos erróneos, las consecuencias pueden ser graves. Lo correcto es cerciorarse de la validez de los datos, ya sea que la empresa cuente con estudios del trabajo y *tiempos estándar*, o en caso contrario mediante una metodología que nos permita tomar una muestra y aceptar que los tiempos con los cuales se va a obtener el modelo matemático son válidos, cumplen con alguna distribución conocida y que puede usarse para describir el proceso; Murray (2007) señala que “En muchas situaciones experimentales el investigador supone a priori, por consideraciones teóricas, que la población de las cuales obtiene la muestra tiene una distribución predeterminada y solamente necesita estimar sus parámetros. Sin embargo, es conveniente tener un procedimiento de prueba para comprobar que tan adecuada es la suposición”.



Se presenta a continuación una secuencia de pasos sugerida por Guasch *et al.*, (2002). Que se deben realizar para efectuar el test de hipótesis:

- Formular la hipótesis que se pretende diagnosticar (H_0).
- Determinar el riesgo aceptable de rechazar la hipótesis (inicializar el valor de alfa)
- Seleccionar el test estadístico apropiado para diagnosticar la hipótesis H_0 .
- Fijar la región crítica para la cual H_0 será rechazada.
- Seleccionar una muestra de valores de un determinado tamaño, ejecutar el test y evaluar la hipótesis.

La selección del test estadístico dependerá del tipo de información de la que se disponga sobre el proceso. En general se consideran dos tipos de test que son los paramétricos y los no-paramétricos.

- Paramétricos: Son aquellos en los que la función de distribución a la que corresponde el proceso es conocida y se emplean para evaluar hipótesis sobre los parámetros de la función de distribución, estos test son por ejemplo, t-Student, F y Parámetro de Poisson. Si la distribución es normal y cumple algunas otras condiciones, utilizaremos los denominados test paramétricos.
- No Paramétricos: se emplean para evaluar las hipótesis que hacen referencia al tipo de distribución que puede utilizarse para describir el proceso. Dos de los cuales son el test de Chi-cuadrado y el de Kolmogorov-Smirnov. Si la distribución no es normal y no cumple dichos criterios tenemos la opción de usar los llamados test no paramétricos o de distribución libre, que se basan en los rangos de distribución de la variable como se mencionó anteriormente.



La prueba estadística que se utiliza para contrastar la hipótesis nula de igualdad de medias para una muestra o entre dos muestras o grupos es la *t* de Student. Para la aplicación de la *t* de Student se requiere que la distribución de la variable cuantitativa sea normal en ambos grupos de comparación. Cuando no se cumplen los criterios de normalidad, se utilizan test no paramétricos, tal como la prueba de Mann-Whitney, para el caso de muestras independientes y la prueba de Wilcoxon para muestras apareadas, entre otras. La prueba estadística que se utiliza para contrastar la hipótesis nula de independencia de ambas variables es la prueba Chi-cuadrado de Pearson u otras similares. Cuando los elementos están dados por otras variables que pueden influir, se suele utilizar la prueba exacta de McNemar. *El lector interesado puede consultar la bibliografía de referencia para una explicación exhaustiva de cada uno de las pruebas mencionadas, así como de muchas más que no se han señalado.*

La Documentación, como se ha mencionado es una etapa fundamental del desarrollo de un modelo para garantizar su amplia difusión. Ésta última etapa se ha llevado a cabo al ir registrando paso a paso lo que se ha hecho para resolver el problema de la empresa BAZÁN. S. A., y anotando las referencias bibliográficas pertinentes. Después de esta exhaustiva explicación es posible decir que un modelo es una representación matemática simplificada de una realidad compleja.

La modelación matemática facilita la comprensión de las relaciones entre las diversas actividades y el impacto que tienen entre sí. Puede señalar otras áreas problemáticas, o de ayuda en el modelo mismo, además de revelar necesidades de recopilación de datos: indicadores de insumos, procesos y productos. Un modelo es en definitiva, una herramienta de ayuda a la toma de decisiones. Por esta razón, sus resultados deben ser inteligibles y útiles.



CAPÍTULO II.- INVESTIGACIÓN DE OPERACIONES I

Un viaje de 1000 Kilómetros comienza dando el primer paso.

Proverbio chino

II-1 PROGRAMACIÓN LINEAL

La programación lineal tiene un gran campo de aplicación. "En donde se debe llevar a cabo un cierto número de actividades, pero existen limitaciones en la cantidad de recursos o en el modo de utilizarlos que nos impiden desarrollar cada actividad de la manera que se considera más efectiva. En tales situaciones queremos distribuir los recursos disponibles entre las actividades, de tal forma que se optimice la efectividad total" Pinillos (1994).

En este capítulo se explica paso a paso como usar el módulo Programación Lineal y Entera (Linear and Integer programming); se resuelven diversos ejemplos para mostrar las aplicaciones del módulo, y se describe de manera general como WINQSB solventa las posibles complicaciones que se presentan en el algoritmo del Método Simplex o en el modelo mismo.



Para crear un nuevo problema de programación lineal o entera, lo primero que debemos realizar es seleccionar el módulo correspondiente que es Programación Lineal y Entera (Linear and Integer programming), y posteriormente dar un clic en la opción Nuevo Problema (New Problem) que generará una plantilla (Figura 2.1) en la cual se introducirán las características de nuestro problema:

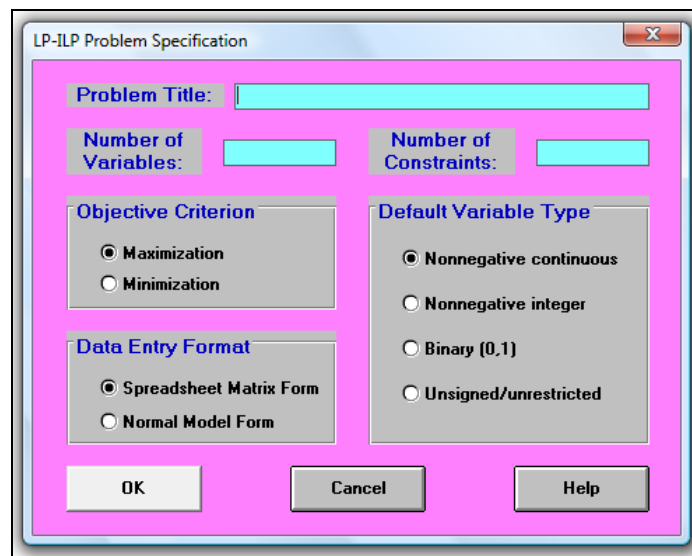


Figura 2.1

La figura 2.1 cuenta con los campos que se describen a continuación:

- Título del problema (Problem Title): Se escribe el título con el que deseamos identificar el problema.
- Número de variables (Number of Variables): Se escribe el número de variables con que cuenta el sistema, sin involucrar las variables de holgura.
- Número de restricciones (Number of Constraints): Se escribe la cantidad de restricciones con que cuenta el modelo (no se debe contar la restricción de no negatividad para las variables).



- **Objetivo (Objective Criterion):** Los problemas de programación lineal y entera se clasifican en maximización (Maximization) y minimización (Minimization), y dependerá del problema en particular que estemos resolviendo.

- **Formato de entrada de datos (Data Entry Format):** Permite elegir entre dos plantillas distintas para introducir los datos del modelo. La primera alternativa se asemeja a una hoja de cálculo de Excel, mientras que la segunda es una plantilla prediseñada del programa.

- **Tipo de variable (Default Variable Type):** En esta parte se indica el tipo de variable que deseamos utilizar:
 - Continuas no negativas (Nonnegative continuous): Indica que el modelo lo componen variables continuas no negativas (iguales o mayores a cero).

 - Enteras no negativas (Nonnegative Integer): Variables enteras no negativas.

 - Binarias (Binary): Variables de tipo binario; es decir cuyos valores serán 0 o 1.

 - Sin asignar / Irrestringidas (Unsigned/unrestricted): Variables irrestringidas, sin restricción.

Mediante el siguiente ejemplo se mostrará la manera en la que se deben introducir los datos del modelo en la plantilla de WINQSB. El problema es el que se ha usado en la sección I.1.2.1.

A continuación se muestra el enunciado que describe el caso que estudiaremos.

La empresa SEA S. A. desea conocer la cantidad de productos a, b y c a producir para maximizar el beneficio, si cada unidad vendida genera en utilidad \$1500, \$2100 y \$1300 por unidad respectivamente. Cada producto pasa por 3 estaciones de trabajo, restringiendo la cantidad de unidades producidas debido al tiempo disponible en cada una de ellas. La siguiente tabla muestra el tiempo estándar requerido por unidad de cada producto en cada estación y el tiempo total disponible semanalmente (tiempo dado en minutos):



	Tiempo requerido estación 1	Tiempo requerido estación 2	Tiempo requerido estación 3
Producto a	10	12	8
Producto b	15	17	9
Producto c	7	7	8
Tiempo total disponible semanalmente por Estación (minutos)	3300	3500	2900

Se hace la suposición de que cada unidad producida es vendida automáticamente sin ningún costo adicional. Determinar la combinación de productos que maximicen la utilidad para la compañía. Una vez que se ha analizado el problema, procedemos a realizar el modelo matemático que lo representa.

Sean:

X_1 = Unidades a producir del producto a

X_2 = Unidades a producir del producto b

X_3 = Unidades a producir del producto c



Función Objetivo: $\text{Max. } Z = \$1500X_1 + \$2100X_2 + \$1300X_3$

Restricciones (Sujeto a):

$$10X_1 + 15X_2 + 7X_3 \leq 3300 \text{ Minutos}$$

$$12X_1 + 17X_2 + 7X_3 \leq 3500 \text{ Minutos}$$

$$8X_1 + 9X_2 + 8X_3 \leq 2900 \text{ Minutos}$$

$$(X_1, X_2, X_3) \geq 0 \text{ (Restricción de no negatividad)}$$

Podemos ver claramente que estamos ante un problema de Maximización, con tres restricciones y tres variables (las cuales trabajaremos como variables continuas de tipo No Negativas). Se procede entonces a capturar los datos del problema en la plantilla, tal como lo muestra la figura 2.2.

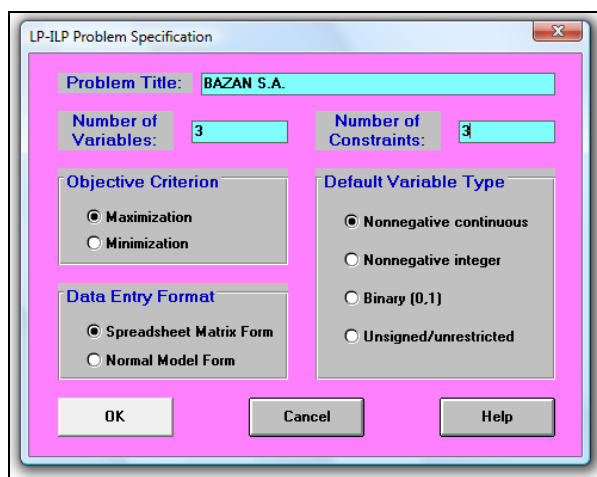


Figura 2.2

Una vez que se han completado los campos pulsamos el botón "OK", si optamos por la plantilla tipo hoja de cálculo (Spreadsheet Matrix Form) en el Formato de entrada de datos, se mostrará una nueva ventana (Figura 2.3), en la cual procederemos a introducir el modelo matemático planteado con anterioridad.



Variable -->	X1	X2	X3	Direction	R. H. S.
Maximize					
C1				<=	
C2				<=	
C3				<=	
LowerBound	0	0	0		
UpperBound	M	M	M		
VariableType	Continuous	Continuous	Continuous		

Figura 2.3

La primera fila (Variable -->) corresponde a los encabezados de las variables definidas automáticamente por el sistema como X1, X2 y X3 (que son las tres variables del ejemplo: a, b y c), el nombre de las variables se puede cambiar accediendo al submenú Nombre de variables (Variables Names) del menú Editar (Edit), seguido por el operador de relación (Direction) y los límites de las restricciones o Lado de la mano derecha (Right Hand Side -R. H. S).

La segunda fila (Maximize) permite introducir los coeficientes de la función objetivo.

En las 3°, 4° y 5° filas aparecen identificadas por C1, C2 Y C3, las cuales corresponden a la cantidad de restricciones con que cuenta el modelo.

Las últimas tres filas en las cuales definimos el valor mínimo aceptado por cada Variable: El valor mínimo (Lower Bound), el valor máximo (Upper Bound) y el tipo de variable (Variable Type).

En el caso del valor máximo, M significa que la variable podrá recibir valores muy grandes (tendientes a infinito). Para ingresar el modelo seguimos estos cuatro pasos:

1. Llenar la segunda fila con los coeficientes de la función objetivo.
2. Llenar la tercera, cuarta y quinta fila con los coeficientes de las restricciones: C1, C2 y C3
3. Llenar el lado derecho de las restricciones que nos indica la capacidad total disponible para las operaciones.
4. Ajustar los operadores de relación pulsando dos veces seguidas sobre ellos con el botón izquierdo del Mouse. Las otras filas se mantienen iguales.



Los resultados se muestran en la figura 2.4

Variable -->	a	b	c	Direction	R. H. S.
Maximize	1500	2100	1300		
C1	10	15	7	<=	3300
C2	12	17	7	<=	3500
C3	8	9	8	<=	2900
LowerBound	0	0	0		
UpperBound	M	M	M		
VariableType	Continuous	Continuous	Continuous		

Figura 2.4

Una vez que hemos ingresado el modelo en la plantilla, se podrán utilizar las herramientas que provee el menú Resolver y Analizar (Solve and Analyze) que se muestra en la figura 2.5.

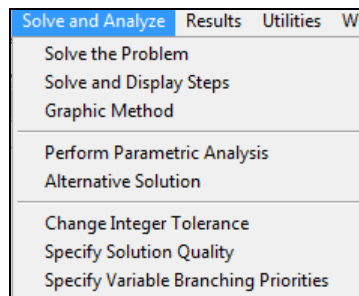


Figura 2.5

A continuación se describen las tres primeras opciones de este submenú:

- Resolver el problema (Solve the Problem): Resuelve el problema mediante el método Simplex Primal. Muestra la solución final completa.
- Resolver y mostrar los pasos (Solve and Display Steps): Muestra cada un de los pasos o las iteraciones realizadas por el Simplex hasta llegar a la solución óptima; es posible verlo por pasos también de modo gráfico, y esto permite comprender mucho mejor el proceso que sigue el algoritmo.



- Método Gráfico (Graphic Method): Resuelve el problema de programación lineal mediante el método gráfico (para problemas que trabajan con dos variables).

Para resolver el ejemplo que hemos estado trabajando seleccionamos la primera opción del menú Resolver y Analizar (Solve and Analyze), donde se mostrará una pequeña ventana con el mensaje "El problema ha sido resuelto. La solución óptima ha sido lograda" (The problem has been solved. Optimal solution is achieved). Pulsamos el botón ACEPTAR y automáticamente el programa generará la solución óptima. La interpretación de los resultados es un punto neurálgico al usar este tipo de herramientas; no servirá de nada que se adquiera un conocimiento acerca de las herramientas principales del software si no es posible realizar una correcta interpretación de lo que nos despliega la pantalla. La figura 2.6 muestra la matriz resultante.

14:12:29		Monday	November	10	2008
Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status
1 a	0	1,500.0000	0	-149.3151	at bound
2 b	105.4795	2,100.0000	221,506.9000	0	basic
3 c	243.8356	1,300.0000	316,986.3000	0	basic
Objective	Function	(Max.) =	538,493.2000		
Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price
1 C1	3,289.0410	<=	3,300.0000	10.9589	0
2 C2	3,500.0000	<=	3,500.0000	0	69.8630
3 C3	2,900.0000	<=	2,900.0000	0	101.3699

Figura 2.6

La primera columna corresponde al análisis de las variables definidas (a, b y c). La columna Valores de la solución (Solution Value) presenta los valores óptimos encontrados:

- En este ejemplo se tiene que a= 0 unidades, b= 105.4795 unidades y c= 243.8356 unidades.
- La columna Costo o Utilidad Unitaria (Unit Cost or Profit c(j)) muestra los coeficientes de la función objetivo para cada variable.
- La columna Contribución Total (Total Contribution) representa el costo o utilidad generado por cada variable. Por ejemplo, si el valor de la variable b es 105.4795 unidades y la utilidad



unitaria es \$2100, el beneficio total resultará de la multiplicación de ambos valores dando como resultado \$221,506.9 Justo debajo de la última contribución aparece el valor de Z óptimo (\$538,493.2).

- La columna Costo Reducido (Reduced Cost) identifica el costo que genera incrementar una unidad para cada variable no básica.
- La siguiente columna llamada Estatus de la Variable (Basis Status) muestra el estado de las variables a, b y c; es decir si es básica (Basic) o en el límite (at bound).

La parte inferior de la matriz muestra la primera columna (Constraint), presenta las variables de holgura del sistema (C1, C2, C3):

- La columna Lado de la mano izquierda (Left Hand Side) muestra el valor alcanzado al reemplazar los valores de a, b y c en cada restricción (es importante recordar que cada restricción se identifica con su variable de holgura).
- Las dos columnas siguientes (Direction y Right Hand Side) muestran las especificaciones dadas a las restricciones en cuanto al operador de relación (\leq) y los valores originales de las restricciones (3300, 3500 y 2900 minutos).
- La columna Holgura (Déficit) o Superávit (Slack or Surplus) muestran los valores de las variables de holgura
- Finalmente la columna Precios Sombras (Shadow Price) corresponde a cuánto se estaría dispuesto a pagar por una unidad adicional de cada recurso.

Los valores óptimos de las variables duales proporcionan una interpretación e información económica potencial para la toma de decisiones de una empresa, esta es proporcionada por los precios sombra o marginales o de equilibrio, los cuales permiten determinar si resulta ventajoso adquirir recursos adicionales sin necesidad de resolver nuevamente el problema; facilita el estudio del impacto sobre la optimalidad por cambios en los coeficientes del problema (análisis de sensibilidad). El precio sombra



es la cantidad que mejora el valor óptimo de la función objetivo en el problema dual (un incremento de maximización o un decremento en un problema de minimización).

En el renglón anterior se aclara la importancia de la última columna que hace referencia a los precios sombras y que tienen una estrecha relación con las variables duales. El lector interesado puede consultar la bibliografía propuesta en la que se explica de manera exhaustiva el concepto de dualidad, se recomienda en especial Estrada (2009).

WINQSB permite mostrar los resultados óptimos mediante el formato aplicado por el método Simplex. La Figura 2.7. Muestra esta tabla, este tipo de visualización es útil ya que en el aula se nos ha familiarizado más con este tipo de formato.

Para desplegar en el monitor esta ventana se debe (una vez resuelto el problema) seleccionar en el menú Resultados (Results) la opción Tabla final del Simplex (Final Simplex Tableau).

Basis	C(j)	a	b	c	Slack_C1	Slack_C2	Slack_C3	R. H. S.	Ratio
Slack_C1	0	-0.9041	0.0000	0.0000	1.0000	-0.7808	-0.1918	10.9589	
b	2,100.0000	0.5479	1.0000	0.0000	0	0.1096	-0.0959	105.4795	
c	1,300.0000	0.3836	0.0000	1.0000	0	-0.1233	0.2329	243.8356	
	C(j)-Z(j)	-149.3151	0	0	0	-69.8630	-101.3699	538,493.1000	

Figura 2.7

Resulta muy interesante ver como el software va resolviendo el modelo; es decir paso a paso. Esto es bastante útil para el estudiante del Método Simplex, ya que puede servir para cotejar su procedimiento con ayuda de la computadora; para esto necesario regresar nuevamente a la plantilla correspondiente al modelo inicial (sin solucionar); una vez que hemos regresado procedemos a marcar la opción Resolver y mostrar los pasos (Solve and Display Steps). La primera tabla mostrada en la Figura 2.8, corresponde a la tabla inicial del Simplex:



Basis	C(j)	a	b	c	Slack_C1	Slack_C2	Slack_C3	R. H. S.	Ratio
Slack_C1	0	10.0000	15.0000	7.0000	1.0000	0	0	3,300.0000	220.0000
Slack_C2	0	12.0000	17.0000	7.0000	0	1.0000	0	3,500.0000	205.8824
Slack_C3	0	8.0000	9.0000	8.0000	0	0	1.0000	2,900.0000	322.2222
	C(j)-Z(j)	1,500.0000	2,100.0000	1,300.0000	0	0	0	0	

Figura 2.8

WINQSB cuenta con opciones de navegación para pasar de una tabla a otra (este menú se llama Simplex Iteration) y se muestra en la figura 2.9.

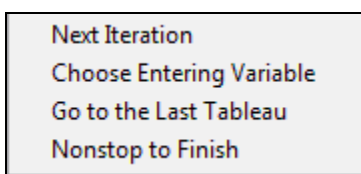


Figura 2.9

Al pulsar sobre la opción Próxima Interacción (Next Iteration) se avanza a la siguiente tabla del Simplex, que es mostrada en la figura 2.10.

Basis	C(j)	a	b	c	Slack_C1	Slack_C2	Slack_C3	R. H. S.	Ratio
Slack_C1	0	-0.5882	0	0.8235	1.0000	-0.8824	0	211.7647	257.1429
b	2,100.0000	0.7059	1.0000	0.4118	0	0.0588	0	205.8824	500.0000
Slack_C3	0	1.6471	0	4.2941	0	-0.5294	1.0000	1,047.0590	243.8356
	C(j)-Z(j)	17.6471	0	435.2941	0	-123.5294	0	432,352.9000	

Figura 2.10

La opción Escoger variable de entrada (Choose Entering Variable) permite seleccionar la variable que entra al sistema de forma manual, al elegir esta opción se abre la ventana que se muestra en la figura 2.11.

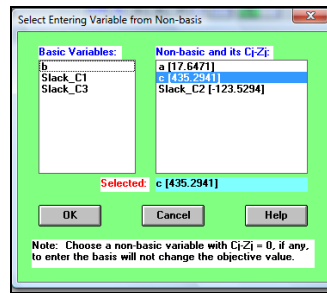


Figura 2.11

Se debe pulsar sobre la variable no básica que desee que entre (en este caso se muestra: a, c y C2 como no básicas).

Para mostrar la última tabla del Simplex directamente podremos optar por seleccionar la opción llamada Ir a la última tabla (Go To The Last Tableau).

La última opción muestra el resultado final completo (Nonstop to Finish), junto al análisis de sensibilidad. Una explicación exhaustiva del análisis de sensibilidad podemos encontrarla en los libros Pinillos (1994) y Estrada (2009).



II-1.2 COMPLICACIONES DEL MÉTODO SIMPLEX

Existen variaciones del modelo matemático con respecto al que se estudia de manera general, las más comunes son diez.

Dos de ellas son: (1) Ausencia de soluciones posibles y (2) Solución óptima sin límite. Es pertinente mencionar que no son analizadas dentro de este apartado, ya que cuando el programa identifica alguna de estas dos opciones muestra un mensaje en pantalla en el que se anuncia que efectivamente "No existen soluciones posibles" o en otro caso "Solución óptima sin límite".

Lo anterior no representa mayores dificultades para el usuario del software, por lo tanto se centra la atención en las siguientes ocho complicaciones:

1. Minimización
2. Desigualdad con sentido invertido
3. Constantes negativas ($b_i < 0$)
4. Igualdades
5. Variables no restringidas en signo
6. Empate para entrar con variables básicas
7. Empate para dejar la base (degeneración)
8. Soluciones múltiples

Estas variaciones pueden presentarse de manera individual o simultánea y una explicación exhaustiva de los métodos de solución *a mano* y *su justificación teórica* podemos encontrarla en el libro Pinillos (1994). *WINQSB* permite la solución a pesar de estas complicaciones, siempre y cuando se introduzcan los datos de manera correcta. Para aclarar lo anterior se proporciona el siguiente ejemplo.



II-1.2.1 MINIMIZACIÓN

Supongamos que tenemos el siguiente modelo matemático tomado del libro Pinillos (1994)².

Minimizar $Z = -X_1 - 2X_2$

Sujeto a: $X_1 \leq 8$

$3X_1 + 2X_2 \leq 30$

$-X_1 + 3X_2 \leq 12$

$(X_1, X_2) \geq 0$ (Restricción de no negatividad)

Los pasos para resolver este tipo de complicaciones son los siguientes:

1. Abrir un nuevo problema en el módulo de Programación Lineal y Entera
2. Introducir los datos como lo muestra la figura 2.12.

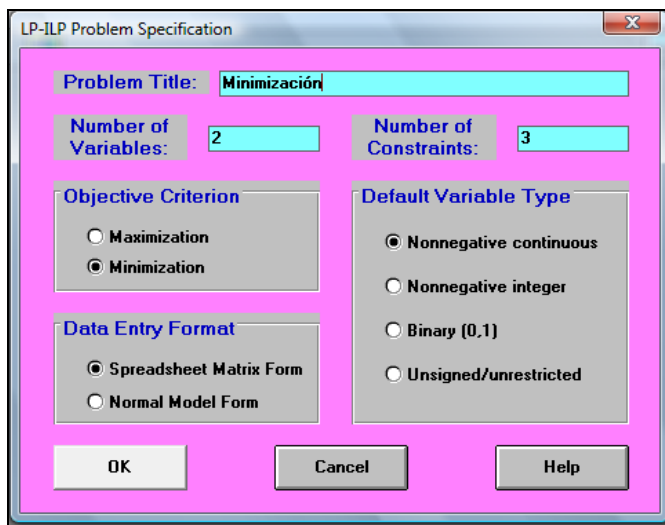


Figura 2.12

² De aquí en adelante se tomarán modelos matemáticos para ejemplificar como WINQSB solventa las complicaciones que presentan, sin entrar en detalles del significado físico de las variables; sin embargo el lector interesado puede consultar la referencia Pinillos (1994) para una explicación del significado de cada una de las variables y del modelo mismo.



3. Damos clic en "OK" e introducimos los datos del problema como lo muestra la Figura 2.13.

Variable ->	X1	X2	Direction	R. H. S.
Minimize	-1	-2		
C1	1		<=	8
C2	3	2	<=	30
C3	-1	3	<=	12
LowerBound	0	0		
UpperBound	M	M		
VariableType	Continuous	Continuous		

Figura 2.13

4. Una vez que se han ingresado los datos, damos clic en resolver. Los resultados se muestran en la figura 2.14.

19:08:05		Wednesday	November	12	2008		
Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)
1 X1	6.0000	-1.0000	-6.0000	0	basic	-3.0000	0.6667
2 X2	6.0000	-2.0000	-12.0000	0	basic	-M	-0.6667
Objective	Function	(Min.) =	-18.0000				
Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS
1 C1	6.0000	<=	8.0000	2.0000	0	6.0000	M
2 C2	30.0000	<=	30.0000	0	-0.4545	8.0000	37.3333
3 C3	12.0000	<=	12.0000	0	-0.3636	1.0000	45.0000

Figura 2.14

Como se puede observar en la tabla de resultados el valor de X1 y X2 que minimiza al máximo la función objetivo es 6 para ambas variables. El valor mínimo de Z es -18. Si resolvemos el problema paso a paso obtendremos las siguientes tablas:

Iteración 1:

Basis	C(j)	X1	X2	Slack_C1	Slack_C2	Slack_C3	R. H. S.	Ratio
Slack_C1	0	1.0000	0	1.0000	0	0	8.0000	M
Slack_C2	0	3.0000	2.0000	0	1.0000	0	30.0000	15.0000
Slack_C3	0	-1.0000	3.0000	0	0	1.0000	12.0000	4.0000
	C(j)-Z(j)	-1.0000	-2.0000	0	0	0	0	

Figura 2.15



Iteración 2:

		X1	X2	Slack_C1	Slack_C2	Slack_C3		
Basis	C(j)	-1.0000	-2.0000	0	0	0	R. H. S.	Ratio
Slack_C1	0	1.0000	0	1.0000	0	0	8.0000	8.0000
Slack_C2	0	3.6667	0	0	1.0000	-0.6667	22.0000	6.0000
X2	-2.0000	-0.3333	1.0000	0	0	0.3333	4.0000	M
	C(i)-Z(i)	-1.6667	0	0	0	0.6667	-8.0000	

Figura 2.16

Iteración 3:

		X1	X2	Slack_C1	Slack_C2	Slack_C3		
Basis	C(j)	-1.0000	-2.0000	0	0	0	R. H. S.	Ratio
Slack_C1	0	0.0000	0.0000	1.0000	-0.2727	0.1818	2.0000	
X1	-1.0000	1.0000	0.0000	0	0.2727	-0.1818	6.0000	
X2	-2.0000	0.0000	1.0000	0	0.0909	0.2727	6.0000	
	C(i)-Z(i)	0	0	0	0.4545	0.3636	-18.0000	

Figura 2.17

El número de iteraciones y resultados es exactamente el mismo que si lo resolvemos manualmente. En la cuarta fila $C(i)-Z(i)$ se encuentra la función objetivo y en la última columna llamada Ratio se muestra el valor de "fi", que nos ayuda a determinar las variables de entrada y salida para las siguientes iteraciones.

WINQSB permite resolver el problema de modo gráfico (sólo muestra la solución). En este caso procedemos de la siguiente manera: A partir de que hemos ingresado los datos como se muestra en la Figura 2.13. Colocamos el cursor del ratón y damos clic en Resolver y analizar, inmediatamente aparecerá un menú en el cual seleccionamos la opción de Método gráfico y aparecerá una ventana como la que se muestra en la figura 2.18.

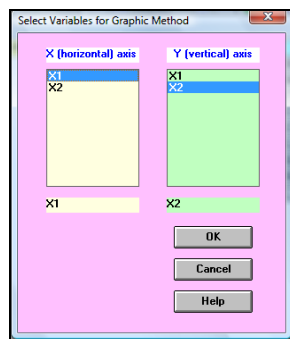


Figura 2.18

En esta ventana podremos elegir cuál de las variables queremos colocar en el eje "X" y cual en el eje "Y". Una vez hecha la elección damos clic en el botón "OK". Aparecerá una ventana como la de la figura 2.19.

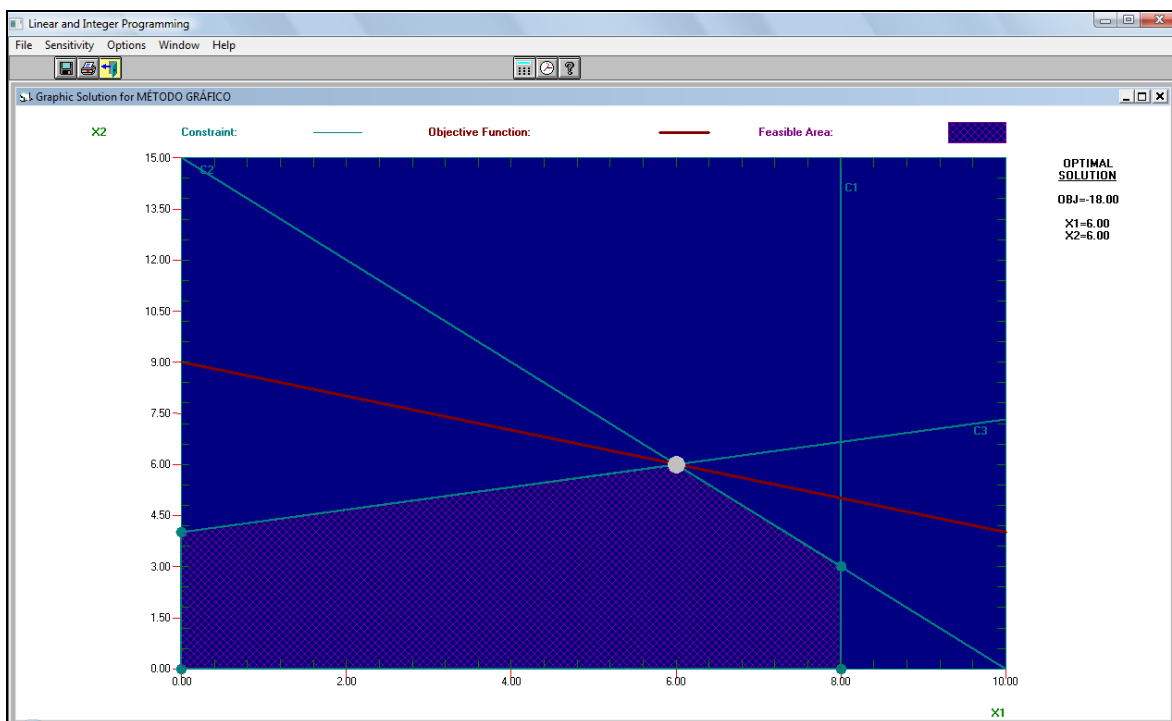


Figura 2.19

Las líneas más delgadas representan las restricciones del problema, la línea en rojo es la función objetivo y el área achurada es la región de soluciones factibles.



Los puntos extremos, son soluciones básicas factibles y recordemos que cualquier punto dentro del polígono (región achurada) satisface el sistema de desigualdades que se plantearon en el modelo matemático, los cuales son también soluciones a nuestro problema.

La función objetivo: $Z = -X_1 - 2X_2$ es una familia de rectas con parámetro Z .

La solución gráfica de este tipo de problemas podemos imaginarlo como el determinar cuales líneas de entre la familia de rectas está más alejada del origen, pero que contiene cuando menos un punto que pertenece a la línea frontera del polígono.

Este método no es recomendable para el caso en el que tenemos más de dos variables; dado que tendríamos un modelo tridimensional (en el caso de tres variables), podríamos intentar resolverlo con geometría analítica espacial; sin embargo se complica demasiado. *Imaginémoslo para más.*



II-1.2.2 DESIGULDAD CON SENTIDO INVERTIDO

Al presentarse esta complicación al plantear nuestro modelo matemático sólo multiplicamos ambos lados de la desigualdad por (-1), para obtener nuestro modelo general; y procedemos entonces a alimentar la plantilla en la forma como ya se ha mostrado.

II-1.2.3 CONSTANTES NEGATIVAS (b i < 0)

En este caso WINQSB realiza los ajustes pertinentes y resuelve la complicación. Para mostrar como lo hace consideremos el siguiente modelo matemático extraído de Pinillos (1994), Pág. 99.

$$\begin{aligned}
 Z - X1 - 2X2 & & + X6 & = 0 \\
 X1 & + X3 & & = 600 \\
 & + X2 & + X4 & = 300 \\
 3X1 + 4X2 & - X5 + X6 & & = 2400
 \end{aligned}$$

Notamos que aún no es posible aplicar la rutina simplex, debido a que Z aún está expresada en función de una variable básica (X6) por lo que eliminándola obtenemos que en la fila C(i)-Z(i) donde se encuentra la función objetivo X6 tiene un coeficiente de cero y aparece con un coeficiente de uno en la fila llamada Artificial_C3.

Iteración 1:

Basis	C(j)	X1	X2	Slack_C1	Slack_C2	Surplus_C3	Artificial_C3	R. H. S.	Ratio
Slack_C1	0	1.0000	2.0000	0	1.0000	0	0	600.0000	M
Slack_C2	0	0	1.0000	0	1.0000	0	0	300.0000	300.0000
Artificial_C3	-M	3.0000	4.0000	0	0	-1.0000	1.0000	2,400.0000	600.0000
C(i)-Z(i)		1.0000	2.0000	0	0	0	0	0	
* Big M		3.0000	4.0000	0	0	-1.0000	0	0	

Figura 2.20



En el programa el renglón de Z esta en el renglón $c(j)-Z(j)$ y además un renglón que dice Big M en donde están los valores de la M en la función objetivo , lo que se tiene regularmente en el algoritmo simplex al hacerlo de manera teórica es el renglón de Z que se obtiene sumando el renglón de $c(j)-Z(j)$ mas el renglón de la Big M , y el valor de Z esta en el renglón de $-M$ en la columna R.H. S.

La variables de holgura son X3 que es la Slack_C1, X4 que es la Slack_C2 , X5 que es la Surplus_C3 y nuestra variable artificial X6 que es Artificial_C3.

Iteración 2:

Basis	C(j)	X1	X2	Slack_C1	Slack_C2	Surplus_C3	Artificial_C3	R. H. S.	Ratio
Slack_C1	0	1.0000	0	1.0000	0	0	0	600.0000	600.0000
X2	2.0000	0	1.0000	0	1.0000	0	0	300.0000	M
Artificial_C3	-M	3.0000	0	0	-4.0000	-1.0000	1.0000	1,200.0000	400.0000
	$C(j)-Z(j)$	1.0000	0	0	-2.0000	0	0	600.0000	
	* Big M	3.0000	0	0	-4.0000	-1.0000	0	0	

Figura 2.21

En esta iteración sucede lo mismo que en la iteración anterior, la matriz del programa tiene los mismos valores que en el procedimiento teórico. Es decir el valor de Z en el procedimiento teórico es de $Z=-600(2M-1)=-1200M+600$ y el programa en el renglón de $-M$ tiene 1200 y en el de $c(j)-Z(j)$ tiene 600 que si los unimos representan el mismo resultado.

Iteración 3:

Basis	C(j)	X1	X2	Slack_C1	Slack_C2	Surplus_C3	Artificial_C3	R. H. S.	Ratio
Slack_C1	0	0	0.0000	1.0000	1.3333	0.3333	-0.3333	200.0000	600.0000
X2	2.0000	0	1.0000	0	1.0000	0	0	300.0000	M
X1	1.0000	1.0000	0.0000	0	-1.3333	-0.3333	0.3333	400.0000	M
	$C(j)-Z(j)$	0	0	0	-0.6667	0.3333	-0.3333	1,000.0000	
	* Big M	0	0	0	0	0	-1.0000	0	

Figura 2.22

En esta iteración los resultados de la matriz en el programa son iguales que en el procedimiento teórico, ya no aparece el renglón $-M$ por que el valor de $Z=1000$.



Iteración 4:

Basis	C(j)	X1	X2	Slack_C1	Slack_C2	Surplus_C3	Artificial_C3	R. H. S.	Ratio
Surplus_C3	0	0	0	3.0000	4.0000	1.0000	-1.0000	600.0000	
X2	2.0000	0	1.0000	0	1.0000	0	0	300.0000	
X1	1.0000	1.0000	0	1.0000	0	0	0	600.0000	
C(j)-Z(j)	0	0	-1.0000	-2.0000	0	0	0	1,200.0000	
* Big M	0	0	0	0	0	0	-1.0000	0	

Figura 2.23

La matriz del programa coincide con nuestra tabla del procedimiento teórico, en donde ya no tenemos pivote seleccionado, ni valores de \emptyset porque esta iteración representa la solución óptima, al igual que en el programa $Z=1200$, $x_5=600$, $x_2=300$ y $x_1=600$

El método que aplica el programa es "el de la gran M" con la variación que la "M" tiene un valor infinitamente grande. WINQSB lo manipula así en cada iteración, hasta que finalmente desaparece de nuestro sistema, junto con las variables no básicas.

El único inconveniente es que el programa no muestra representación gráfica en 3D solo en 2D pero en general es una gran herramienta para la solución de problemas, Este método no es muy conveniente para cuando tenemos más de 2 variables originales, ya que WINQSB no puede graficar tridimensionalmente, sin embargo es muy didáctico el emplear este método para 2 variables. Así pues la solución gráfica permite entender de forma más rápida la solución del modelo, ya que señala tanto a la función optimizada como a las restricciones y región de soluciones factibles como se muestra en la figura 2.24.

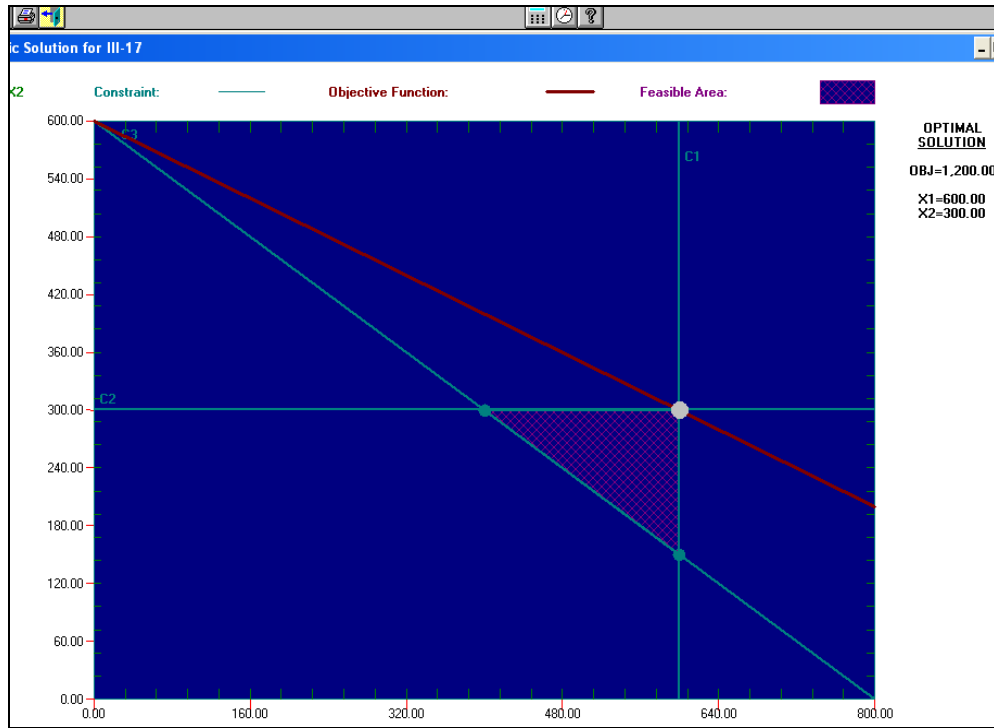


Figura 2.24

II-1.2.4 IGUALDADES

El programa permite resolver esta complicación de una manera muy sencilla, lo único que tenemos que hacer es situar el cursor en la celda que correspondiente al símbolo de igualdad (=) y dar doble clic, esto cambia automáticamente el símbolo.



II-1.2.5 VARIABLES NO RESTRINGIDAS EN SIGNO

En la plantilla de WINQSB se introducen los datos como ya lo hemos hecho con anterioridad; esta complicación se presenta durante el planteamiento del modelo matemático y no durante el proceso, por lo que no compete a este trabajo la explicación exhaustiva del método de solución; sin embargo para dar continuidad a nuestros fines se sugiere que el lector interesado consulte Pinillos (1994), Pág. 107.

II-1.2.6 EMPATE PARA ENTRAR CON VARIABLES BÁSICAS

Esta complicación la resuelve el programa sin mayores dificultades; y se presenta si en alguna iteración de la rutina nos encontramos con que dos o más variables tienen el mismo coeficiente positivo máximo en la función objetivo. Se resuelve seleccionando arbitrariamente, entre las variables en empate, la que deberá ocupar la base. En WINQSB sólo se introducen los datos como ya lo hemos hecho con anterioridad y el propio programa elige por defecto la variable de entrada.

La opción Escoger variable de entrada (Choose Entering Variable) permite seleccionar la variable que entra al sistema de forma manual, al elegir esta opción se abre la ventana que se muestra en la figura 2.11.



II-1.2.7 EMPATE PARA DEJAR LA BASE (DEGENERACIÓN)

A continuación se muestra un ejemplo de cómo el programa resuelve esta complicación. El modelo matemático se ha tomado de Pinillos (1994), Pág. 108-112.

Iteración 1:

Basis	C(j)	X1	X2	Slack_C1	Slack_C2	Slack_C3	R. H. S.	Ratio
Slack_C1	0	1.0000	0	1.0000	0	0	600.0000	M
Slack_C2	0	0	1.0000	0	1.0000	0	300.0000	300.0000
Slack_C3	0	3.0000	4.0000	0	0	1.0000	1,200.0000	300.0000
	C(j)-Z(j)	1.0000	2.0000	0	0	0	0	

Figura 2.25

Se observa que en la primera iteración tenemos un empate entre las variables X4 y X5 (Slack_C2 y Slack_C3) ya que el valor correspondiente a cada una de ellas en la columna ratio, (fi) ó \emptyset , es el mismo, lo cual nos enfrenta al problema de elegir la variable de entrada en el algoritmo del método simplex. El programa elige por defecto la variable que dejará la base (la teoría nos indica que cuando este es el caso, se escoge arbitrariamente cualquiera de las variables en empate para ser la variable de salida). Nuevamente la opción Escoger variable de entrada (Choose Entering Variable) permite seleccionar la variable que entra al sistema de forma manual, al elegir esta opción se abre una ventana similar a la que se muestra en la figura 2.11.

Iteración 2:

Basis	C(j)	X1	X2	Slack_C1	Slack_C2	Slack_C3	R. H. S.	Ratio
Slack_C1	0	1.0000	0	1.0000	0	0	600.0000	600.0000
X2	2.0000	0	1.0000	0	1.0000	0	300.0000	M
Slack_C3	0	3.0000	0	0	-4.0000	1.0000	0	0
	C(j)-Z(j)	1.0000	0	0	-2.0000	0	600.0000	

Figura 2.26

En esta segunda iteración podemos notar que la variable que ha dejado la base por defecto en el programa es X4 (Slack_C2) y la variable X2 ha entrado a la base. Cómo se señala en Pinillos (1994) el efecto de tener un empate en la variable de salida, es que todas las variables en el empate que no



sean escogidas para dejar la base (en este caso X5), quedarán con un valor de cero en la siguiente iteración (tal como se muestra en la iteración 2, donde para la fila Slack_C3 se tiene un valor correspondiente de cero). Por lo tanto obtenemos una solución degenerada (pues X5=0).

Iteración 3:

		X1	X2	Slack_C1	Slack_C2	Slack_C3		
Basis	C(j)	1.0000	2.0000	0	0	0	R. H. S.	Ratio
Slack_C1	0	0	0.0000	1.0000	1.3333	-0.3333	600.0000	
X2	2.0000	0	1.0000	0	1.0000	0	300.0000	
X1	1.0000	1.0000	0.0000	0	-1.3333	0.3333	0	
	C(j)-Z(j)	0	0	0	-0.6667	-0.3333	600.0000	

Figura 2.27

Cómo la solución anterior no es la óptima, el programa selecciona a X1 como la variable de entrada, pero como el valor en la columna ratio \emptyset , para X3 (Slack_C1) es igual a cero debido a que X1 no puede aumentar su valor sin hacer a X5 (Slack_C3) negativa; entonces el programa elige a la variable de salida que es X5 permitiendo la entrada de X1 a la base como se muestra en la iteración3.

Es pertinente mencionar, tal como lo señala Pinillos (1994), que si existe degeneración, es posible tener una secuencia de iteraciones que no aumenten en nada el valor de Z. En la iteración 3 es claro, al observar en la columna RHS para la última fila que representa el valor de la función objetivo, que el valor de Z sigue siendo 600 unidades tal como en la iteración 2.

Los valores de la solución óptima coinciden con la iteración anterior, debido a que X1 no pudo tomar un valor mayor que cero, con lo cual el valor de Z tampoco pudo aumentar.



15:01:19		Wednesday		November		19		2008	
	Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)	
1	X1	0	1.0000	0	0	basic	0	1.5000	
2	X2	300.0000	2.0000	600.0000	0	basic	1.3333	M	
	Objective	Function	(Max.) =	600.0000					
	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS	
1	C1	0	<=	600.0000	600.0000	0	0	M	
2	C2	300.0000	<=	300.0000	0	0.6667	0	300.0000	
3	C3	1,200.0000	<=	1,200.0000	0	0.3333	1,200.0000	3,000.0000	

Figura 2.28

En la figura 2.28 se muestra la solución al problema. En WINQSB sólo se introducen los datos como se mostró al inicio del capítulo y el programa realiza los ajustes pertinentes detallados anteriormente para resolver el problema.

II-1.2.8 SOLUCIONES MÚLTIPLES

Sea el siguiente modelo matemático, extraído de Pinillos (1994):

$$\begin{aligned}
 &\text{Maximizar} && Z = 3X_1 - 4X_2 \\
 &\text{Sujeto a:} && X_1 \leq 600 \\
 &&& X_2 \leq 300 \\
 &&& 3X_1 + 4X_2 \leq 2400 \\
 &&& (X_1, X_2) \geq 0 \text{ (Restricción de no negatividad)}
 \end{aligned}$$



Al introducir los datos en la plantilla de WINQSB y resolver obtenemos la figura 2.29

20:35:44		Wednesday		November		12		2008	
	Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)	
1	X1	400.0000	3.0000	1,200.0000	0	basic	0	3.0000	
2	X2	300.0000	4.0000	1,200.0000	0	basic	4.0000	M	
	Objective	Function	(Max.) =	2,400.0000	(Note:	Alternate	Solution	Exists!!)	
	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS	
1	C1	400.0000	<=	600.0000	200.0000	0	400.0000	M	
2	C2	300.0000	<=	300.0000	0	0	150.0000	600.0000	
3	C3	2,400.0000	<=	2,400.0000	0	1.0000	1,200.0000	3,000.0000	

Figura 2.29

Observe que los valores arrojados por el programa para las variables X1 y X2 son 400 y 300 respectivamente; en realidad se trata de un modelo matemático con soluciones múltiples, y el programa mismo lo señala con un mensaje en la última columna y en la fila llamada Objetivo (Objective), el mensaje nos indica que existe una solución alternativa (Alternate Solution Exists). Al ir al menú de Resultados dar un clic y posteriormente elegir la opción ver la solución alternativa, podemos ver los valores correspondientes a X1 y X2 que son 600 y 150 respectivamente. Si regresamos y resolvemos el problema con el método gráfico, entonces obtendremos lo que se muestra en la figura 2.30.

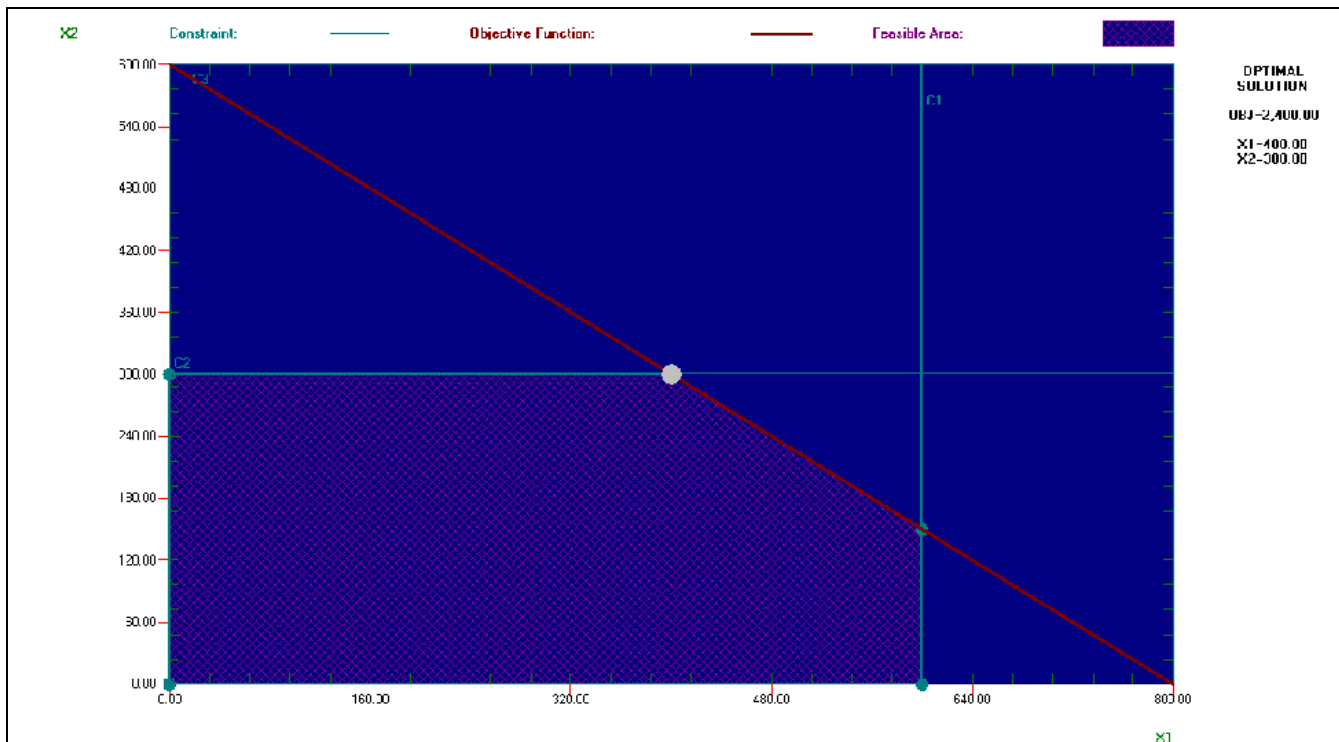


Figura 2.30

Como podemos observar el programa sólo nos muestra un punto como solución a nuestro modelo matemático cuando en realidad es todo un segmento lineal el que maximiza a Z .

Es evidente a lo largo de esta primera parte de la programación lineal la importancia de tener una base teórica bastante sólida, sin ella este trabajo carece de sentido para el que lo consulte.



II-1.3 OBTENCIÓN DEL MODELO DUAL

Entre las herramientas con las que cuenta el programa ofrece la opción de pasar de un modelo Dual a un modelo Primo o viceversa. El lector interesado en la teoría dual puede consultar Pinillos (1994) y Estrada (2009). Supongamos que tenemos el siguiente modelo:

Minimizar $Z = 3X_1 + 4X_2 + 6X_3 + 7X_4 + X_5 + 2X_6 - X_7$

Sujeto a:

$$2X_1 - X_2 + X_3 + 6X_4 - 5X_5 - X_6 \geq 6$$

$$X_1 + X_2 + 2X_3 + X_4 + 2X_5 - X_7 \leq 3$$

$$X_3 - X_4 + 3X_5 + 7X_7 = 9$$

$$8X_1 + 2X_2 - X_3 + 6X_4 + X_6 \geq -1$$

$(X_2, X_4, X_6) \geq 0$ (Restricción de no negatividad)

La captura de los datos se muestra en la siguiente figura 2.31

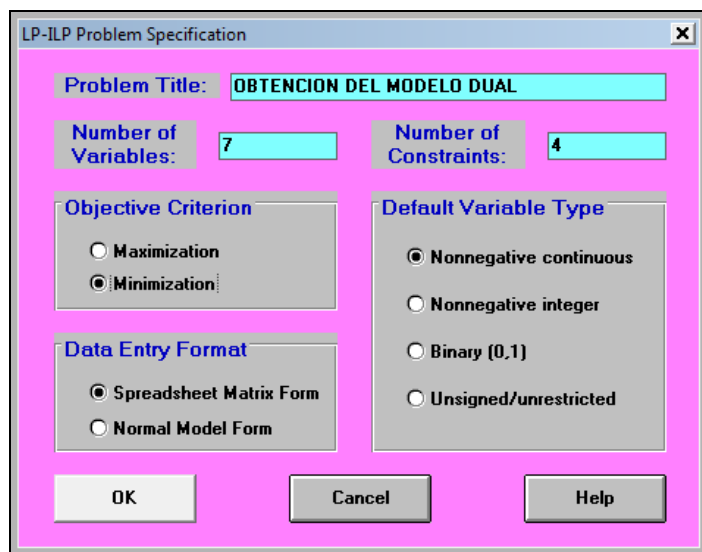


Figura 2.31



Como se puede ver en el modelo original (primo), tenemos siete variables y cuatro restricciones; el modelo original nos dice que se desea minimizar; pulsamos OK y aparecerá la figura 2.32.

Variable -->	X1	X2	X3	X4	X5	X6	X7	Direction	R. H. S.
Minimize									
C1								>=	
C2								>=	
C3								>=	
C4								>=	
LowerBound	0	0	0	0	0	0	0		
UpperBound	M	M	M	M	M	M	M		
VariableType	Continuous	Continuous	Continuous	Continuous	Continuous	Continuous	Continuous		

Figura 2.32

Al ingresar los datos del modelo se debe obtener una ventana como la mostrada en la figura 2.33.

Variable -->	X1	X2	X3	X4	X5	X6	X7	Direction	R. H. S.
Minimize	3	4	6	7	1	2	-7		
C1	2	-1	1	6	-5	-1		>=	6
C2	1	1	2	1	2		-1	<=	3
C3			1	-1	3		7	=	9
C4	8	2	-1	6		1		>=	-1
LowerBound	0	0	0	0	0	0	0		
UpperBound	M	M	M	M	M	M	M		
VariableType	Continuous	Continuous	Continuous	Continuous	Continuous	Continuous	Continuous		

Figura 2.33

Para ver el modelo dual de este modelo primo nos dirigimos al menú formato, tal como se muestra en la figura 2.34. Damos un clic y nos aparece la figura 2.35 que representa el modelo dual de nuestro modelo original.

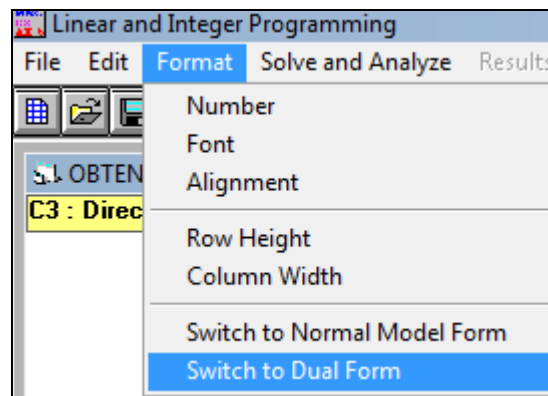


Figura 2.34

Variable -->	C1	C2	C3	C4	Direction	R. H. S.
Maximize	6	3	9	-1		
X1	2	1		8	<=	3
X2	-1	1		2	<=	4
X3	1	2	1	-1	<=	6
X4	6	1	-1	6	<=	7
X5	-5	2	3		<=	1
X6	-1			1	<=	2
X7		1	-7		>=	7
LowerBound	0	-M	-M	0		
UpperBound	M	0	M	M		
VariableType	Continuous	Continuous	Unrestricted	Continuous		

Figura 2.35

Si deseamos resolver este modelo dual, por el método simplex dual; nos dirigimos al menú resolver y analizar, tal como se muestra en la figura 2.36. Como podemos ver, es posible resolver de un solo paso o iteración por iteración.

Al resolver el modelo de una sola vez obtenemos una ventana como la mostrada en la figura 2.37. La cual nos indica que la solución óptima ha sido obtenida. Pulsamos aceptar y aparece una ventana como la mostrada en la figura 2.38.

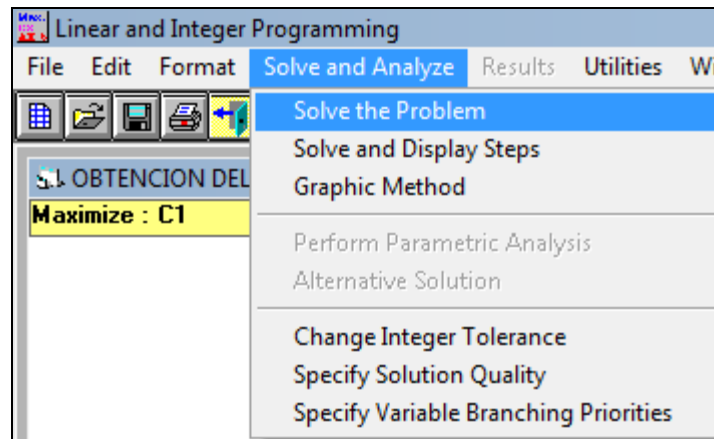


Figura 2.36

Variable -->	C1	C2	C3	C4	Direction	R. H. S.
Maximize	6	3	9	-1		
X1	2	1		8	<=	3
X2	-1	1		2	<=	4
X3	1	2	1	-1	<=	6
X4	6	1	-1	6	<=	7
X5	-5	2	3		<=	1
X6	-1			1	<=	2
X7		1	-7		>=	7
LowerBound	0	-M	-M	0		
UpperBound	M	0	M	M		
VariableType	Continuous	Continuous	Unrestricted	Continuous		

Linear and Integer Programming ✕

The problem has been solved.
Optimal solution is achieved.

Aceptar

Figura 2.37



02:03:39		Thursday		November		12		2009	
	Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)	
1	C1	1,0000	6,0000	6,0000	0	basic	0	30,0000	
2	C2	0	3,0000	0	-3,4286	at bound	-M	0,4286	
3	C3	-1,0000	9,0000	-9,0000	0	at bound	-M	9,0000	
4	C4	0	-1,0000	0	-7,0000	at bound	-M	6,0000	
	Objective	Function	(Max.) =	-3,0000	(Note:	Alternate	Solution	Exists!!)	
	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS	
1	X1	2,0000	<=	3,0000	1,0000	0	2,0000	M	
2	X2	-1,0000	<=	4,0000	5,0000	0	-1,0000	M	
3	X3	0	<=	6,0000	6,0000	0	0	M	
4	X4	7,0000	<=	7,0000	0	1,0000	1,0000	10,0000	
5	X5	-8,0000	<=	1,0000	9,0000	0	-8,0000	M	
6	X6	-1,0000	<=	2,0000	3,0000	0	-1,0000	M	
7	X7	7,0000	>=	7,0000	0	-1,4286	0	49,0000	

Figura 2.38

La primera fila señala la hora y la fecha en la que se ha utilizado la aplicación y la ventana de la figura 2.38 es similar en cuanto a características a la de nuestro primer ejemplo en la sección II-1 por lo que no se entrará en más detalles.



II-2 MODELO DE REDES

Existe una parte importante de modelos de programación lineal que pueden ser analizados en términos de redes que están compuestos por puntos, comúnmente llamados nodos y que se conectan a través de líneas, llamadas arcos; sobre los cuales se efectúa un transporte (de mercancías, materias primas, autos, información, agua, etcétera) que se le conoce como flujo.

Es importante destacar que este tipo de modelos tienen la particularidad de que son aplicados en aquellos casos en que los requerimientos y recursos han sido expresados en términos de un solo tipo de unidad.

Al seleccionar el programa Modelo de redes (Network Modeling) del menú de módulos de WINQSB aparece en pantalla la figura 2.39.

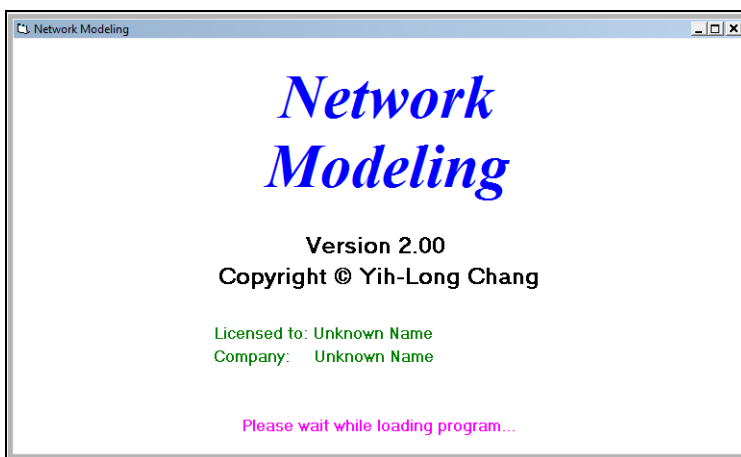


Figura 2.39

Este módulo de WINQSB es nombrado como Modelo de redes (Network modeling). Al seleccionar la aplicación y elegir la opción Nuevo Problema (New Problem) aparecerá la siguiente ventana (figura 2.40).

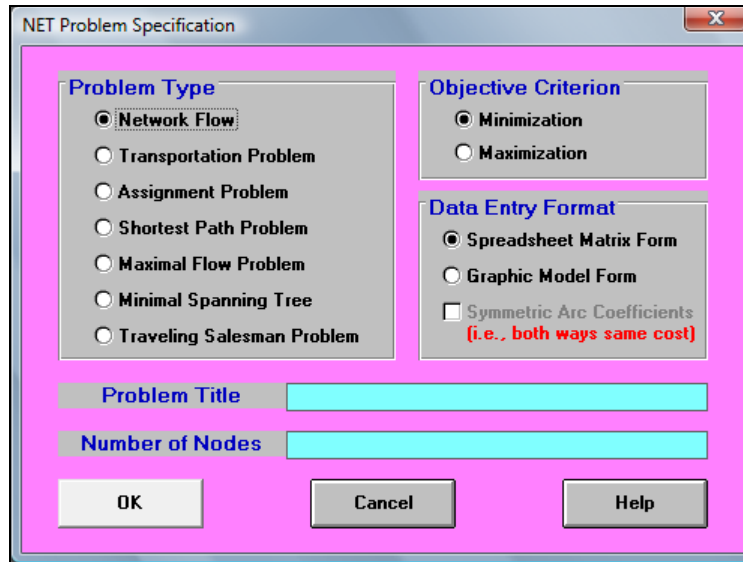


Figura 2.40

El programa nos permite elegir entre 7 modelos para el tratamiento de los problemas que involucran redes con el fin de optimizar el uso de algún recurso, generalmente tratándose de la minimización de costos, tiempo, distancias, o la maximización del flujo a través de una red:

1. Flujo en redes o modelo de trasbordo (Network Flow)
2. Problema de transporte (Transportation Problem)
3. Problema de asignación (Assignment Problem)
4. Problema de la ruta más corta (Shortest Path Problem)
5. Problema de flujo máximo (Maximal Flow Problem)
6. Árbol de mínima expansión (Minimal Spanning Tree)
7. Problema del agente viajero (Traveling Salesman Problem)

Se tratarán cada uno de los modelos con ejemplos sencillos, con la finalidad de ilustrar los alcances de este software; se deben tener las bases teóricas para la adecuada comprensión de lo que se ha desarrollado, la mayor parte de los ejemplos tienen como base bibliográfica a Quesada *et al.*, (2008), Alonso (2008) y Colmenero (2009).



Los pasos para la aplicación de este módulo de acuerdo a Colmenero (2009) son:

1. Analizar el problema.
2. Dibujar la red y numerar cada uno de los nodos apropiadamente.
3. Ingresar el problema dentro de WINQSB.
4. Resolver el problema.

II-2.1 MODELO DE TRASBORDO

Este tipo de modelo es útil sobre todo en logística, cuando se trata minimizar el costo de los envíos. Nos encontramos con 3 elementos: La planta de producción, los centros de distribución (CEDIS) y los clientes; para ilustrar el uso del programa en este tipo de análisis se muestra el siguiente ejemplo. En la plantilla de la figura 2.40 elegimos la primera opción que es flujo en redes o modelo de trasbordo (Network flow); como función objetivo seleccionamos la opción minimizar; si ingresemos la información de un modelo de red que enlaza 2 plantas de producción, 4 almacenes y 3 clientes principales; es claro que tendremos 9 nodos en total, dado que así se ha definido el problema; por lo tanto en el campo de nodos ponemos 9. La figura 2.41 ilustra lo anterior

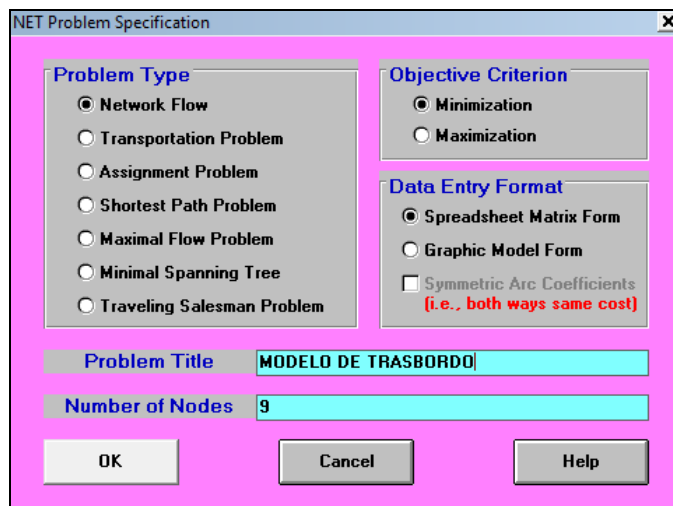


Figura 2.41



La tabla inicial para este modelo se muestra a continuación (figura 2.42):

From \ To	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8	Node9	Supply
Node1										0
Node2										0
Node3										0
Node4										0
Node5										0
Node6										0
Node7										0
Node8										0
Node9										0
Demand	0	0	0	0	0	0	0	0	0	0

Figura 2.42

Para modificar los nombres de los nodos pulsamos sobre Node Name en el menú Editar (Edit) figura 2.43.

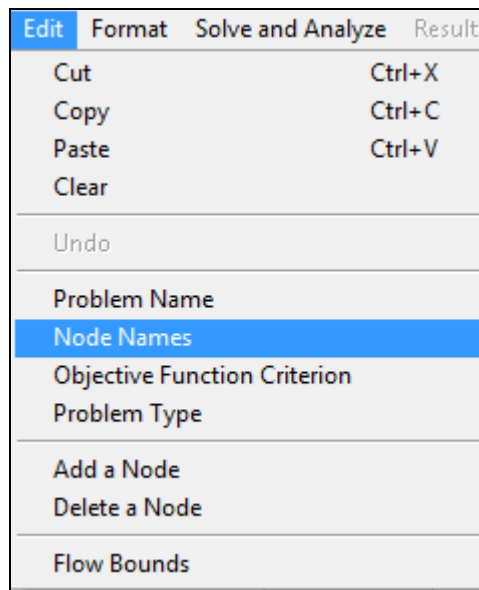


Figura 2.43

Para ilustrar adecuadamente nuestro ejemplo modifiquemos dichos nombres como se muestra a continuación en la figura 2.44



Figura 2.44

La tabla muestra dos fuentes (plantas de producción S1 y S2) que cuentan con capacidades de producción de 600 y 800 unidades para un período dado y un producto que llamaremos "x."

Para alinear los datos en las celdas de la hoja de cálculo se usa la opción del menú format>alignment>center, como se muestra en la figura 2.45

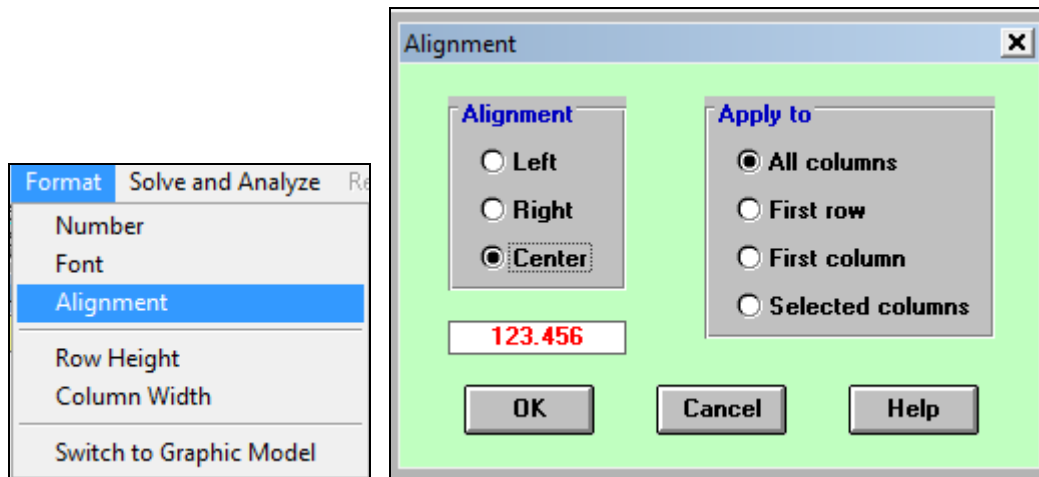


Figura 2.45



Hay 4 almacenes intermedios, T1, T2, T3 y T4, de los cuales T2 Y T3 poseen actualmente en inventario 350 y 200 unidades, respectivamente.

Las demandas de los clientes son: D1, D2 y D3 cada uno requiere un suministro de 500, 350 y 900 unidades, respectivamente. Y los almacenes T1 y T4 no tienen producto "x" actualmente en sus inventarios, ellos han solicitado un suministro de 200 y 100 unidades, respectivamente. Los costos de transportar una unidad de producto "x" desde cada fuente y punto de trasbordo hasta cada sitio de demanda se encuentran en la tabla (figura 2.46).

From \ To	S1	S2	T1	T2	T3	T4	D1	D2	D3	Supply
S1			4	2	3					600
S2					4	8				800
T1				3	1					0
T2					3	5	2	4		350
T3						2	4	7	1	200
T4								3	9	0
D1										0
D2										0
D3										0
Demand	0	0	200	0	0	100	500	350	900	

Figura 2.46

Para ver el modelo en modo gráfico nos dirigimos al menú formato (format) y seleccionamos la opción mostrar el modelo gráfico. Figura 2.47

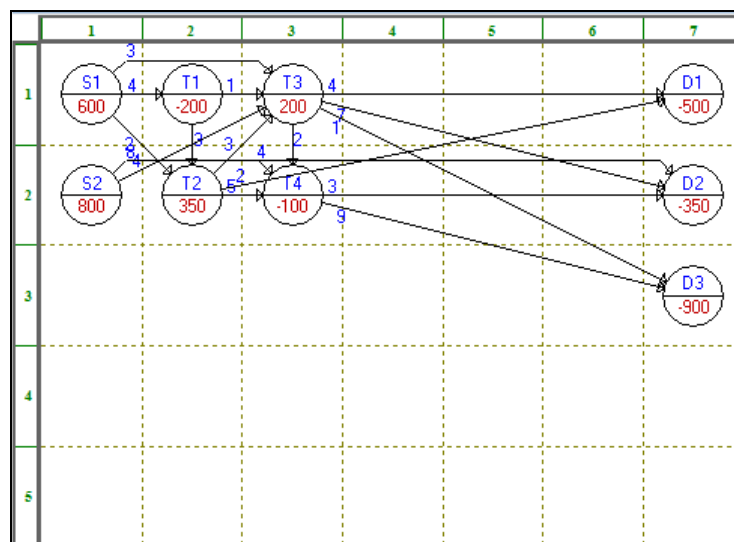


Figura 2.47



Observe que el programa automáticamente antepone un signo negativo para los nodos que requieren cierta cantidad de suministro y un signo positivo para aquellos que cuentan con mercancía en inventarios.

Para los problemas de transporte éste módulo de WINQSB proporciona diferentes alternativas para la solución inicial factible, incluye también la opción de mostrar paso por paso la solución.

Como paso previo a la solución debe escogerse el método mediante el cual se determina la solución básica inicial. Esta opción se despliega seleccionando el menú resolver y analizar. Figura 2.48

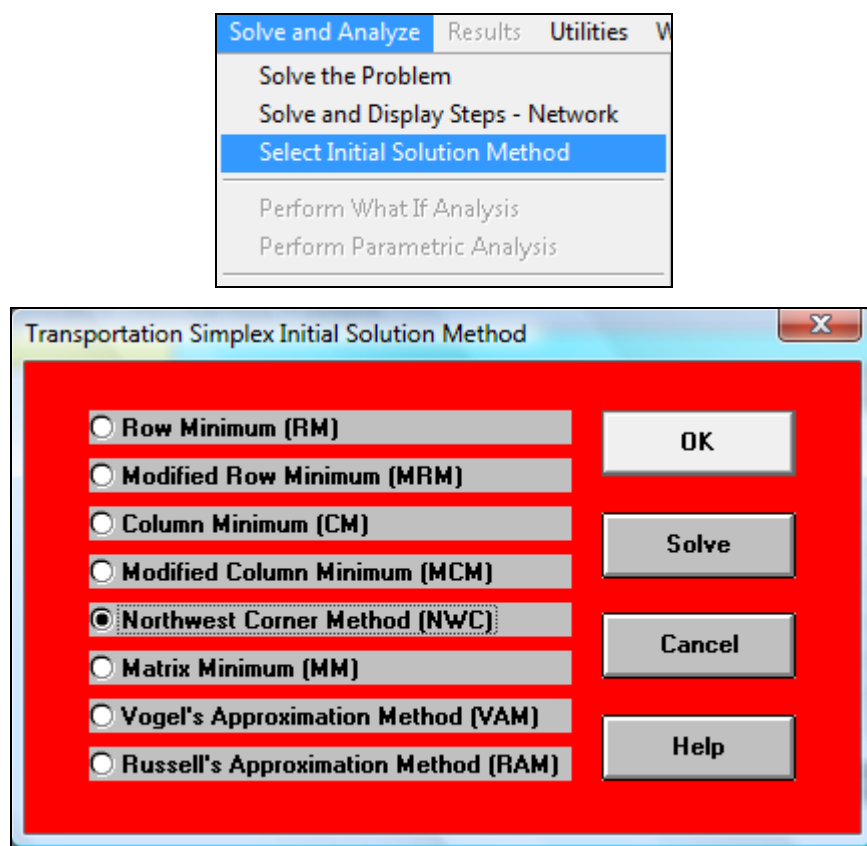


Figura 2.48

En este caso se ha escogido el método de La esquina noroeste y Pulsamos "ok". Aunque podríamos seleccionar cualquier otro como puede ser El método de aproximación de Vogel (VAM), Método de aproximación de Rusell (RAM), etc. Es importante recordar que los métodos asociados con el transporte sólo se diferencian en la forma como se obtiene la solución básica inicial.



La manera de resolver el problema es idéntica a la del simplex, pudiéndose resolver directamente o por pasos. La tabla de resultados finales (figura 2.49) muestra cómo se da el flujo de productos "x" desde a las fuentes iniciales (S) a los puntos de transbordo (T) y de estas a los destinos finales, con un costo total final de 7900 unidades monetarias, que es el mínimo usando como solución básica inicial el método de la esquina noroeste.

11-11-2009	From	To	Flow	Unit Cost	Total Cost	Reduced Cost
1	S1	T1	200	4	800	0
2	S1	T2	400	2	800	0
3	S2	T3	800	4	3200	0
4	T2	D1	500	2	1000	0
5	T2	D2	250	4	1000	0
6	T3	D3	900	1	900	0
7	T3	T4	100	2	200	0
8	Unfilled_Demand	D2	100	0	0	0
	Total	Objective	Function	Value =	7900	

Figura 2.49

En la figura 2.50 observamos la solución gráfica del problema, es importante notar que el programa ha agregado un nodo más que nos indica que la demanda no se satisface (unfilled_Demand); ¿por qué? La explicación es que para este tipo de problemas el origen del cual se enviarán los productos i , debe suministrar específicamente la cantidad a_i , mientras que el destino debe recibir exactamente una cantidad b_j ; es decir la oferta debe ser igual a la demanda. En este caso la demanda excede a la oferta, observe la siguiente suma algebraica:

$$600-200+200+800+350-100= 1650 \text{ artículos ofertados } y,$$

$$-500-350-900= -1750; \text{ es decir } 1750 \text{ artículos demandados}$$

$$\text{Oferta} < \text{Demanda}$$



Luego entonces el programa introduce un origen figurado, para que tome el faltante y satisfaga esta condición.

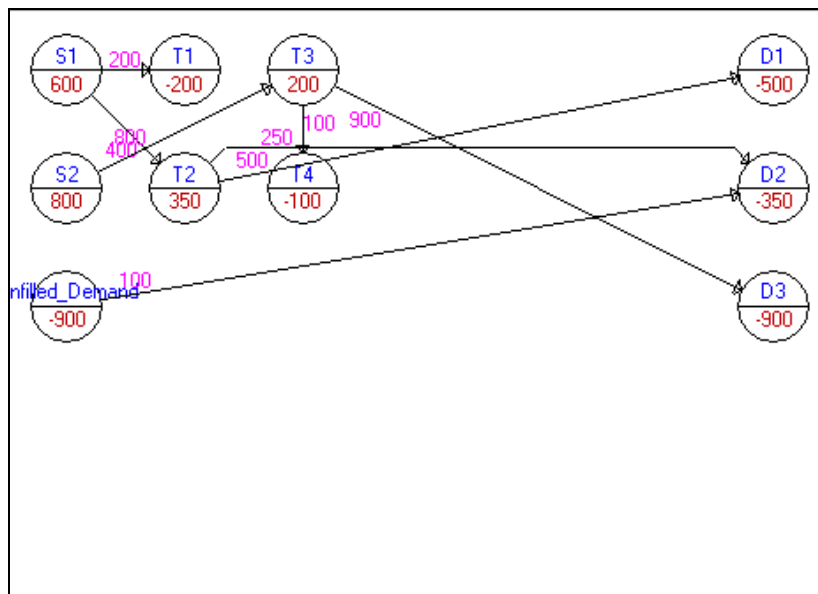


Figura 2.50

Es importante notar que en este modelo no se tiene restricción en cuanto a la capacidad del volumen de productos que se puede transportar, sólo nos muestra cual debe ser el flujo con el mínimo costo para hacer llegar un producto "x".



II-2.2 PROBLEMA DE TRANSPORTE

Como en el caso anterior, aquí se trata de encontrar el programa que minimiza los costos de envío de los productos desde un origen hasta un destino. La diferencia es que aquí se suministran los artículos directamente sin pasar por puntos de trasbordo. Colmenero (2009) sugiere las siguientes notas a tomar en cuenta:

- El envío o embarque (por medio de un arco) de un nodo al mismo nodo (nodo absorbente) no se permite.
- Cuando la demanda total no es igual al abastecimiento, automáticamente el programa agregará un nodo de abastecimiento o demanda artificial (variables de holgura).
- Se pueden seleccionar uno de los ocho métodos para encontrar la solución inicial. El método por defecto es el Mínimo Renglón (RM) (Row Minimum). Los ocho métodos de solución aparecen en la figura 2.51

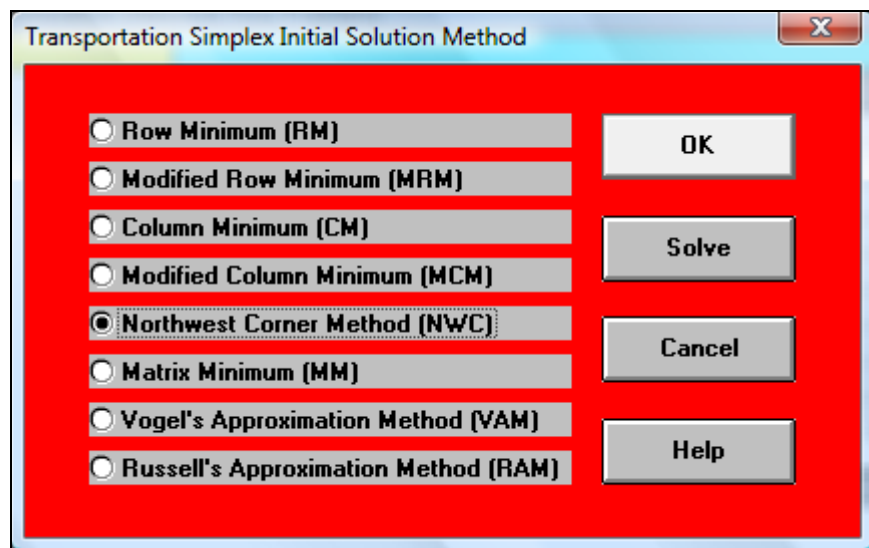


Figura 2.51

Esta opción se despliega seleccionando el menú resolver y analizar, seleccionar solución inicial. La figura 2.52 ilustra el procedimiento.

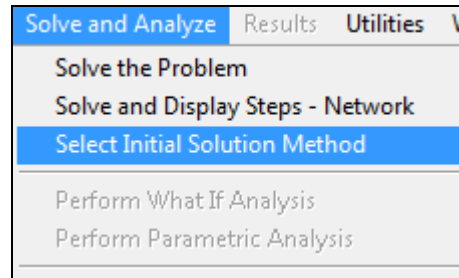


Figura 2.52

Para ingresar un nuevo modelo de la plantilla mostrada en la figura 2.53.

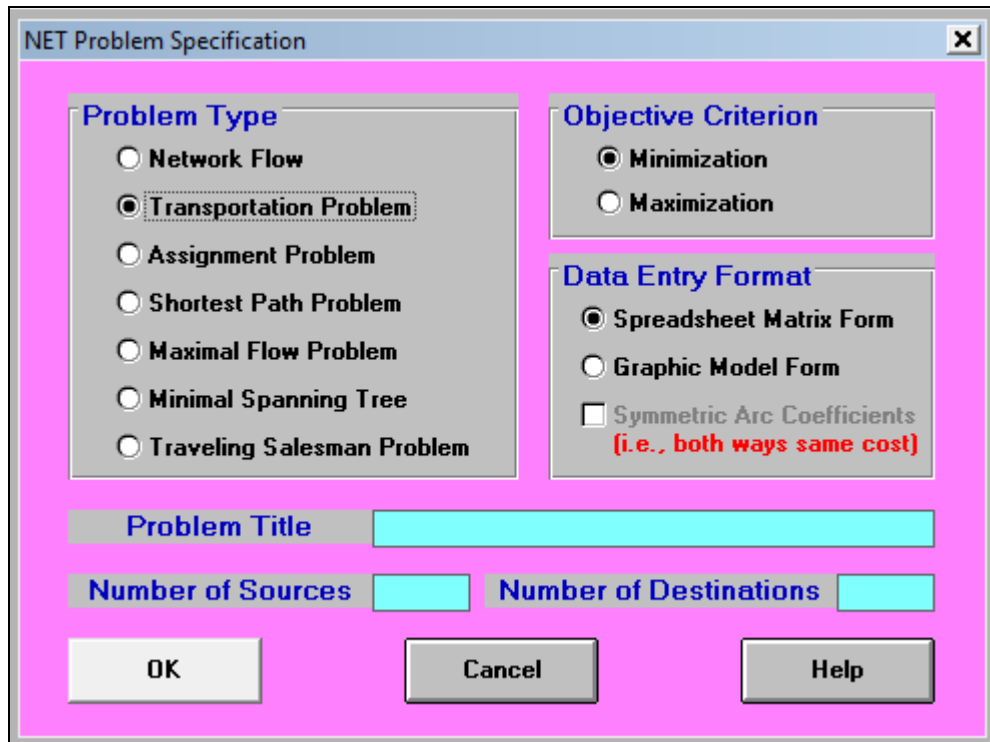


Figura 2.53



Seleccionamos la opción Problema de Transporte (Transportation Problem). Se pueden agregar fuentes y destinos extras, después que el modelo se ha ingresado o leído de una unidad de memoria. Para ilustrar mejor la aplicación se muestran dos ejemplos.

Como primer ejemplo consideremos la figura 2.54 en donde los datos representan la cantidad de material disponible por cada uno de los orígenes (San Ángel, Lomas de San Isidro y Cuautitlán), lo requerido por cada uno de los clientes (Tlalpan, Antara, Reforma y Palmas); observamos dentro de la matriz los costos de transporte entre los distintos orígenes y destinos; los datos que se presentan son figurados.

Origen \ Destino	Tlalpan	Antara	Reforma	Palmas	Disponible
San Ángel	3	2	7	6	5000
Lomas de San Isidro	7	5	2	3	6000
Cuautitlán	2	5	4	5	2500
Requerido	6000	4000	2000	1500	

Figura 2.54

Al elegir la opción Problema de transporte (Transportation Problem) y resolver directamente el problema se obtiene la siguiente tabla que nos muestra la solución (figura 2.55):

12-04-2008	From	To	Shipment	Unit Cost	Total Cost	Reduced Cost
1	San Ángel	Tlalpan	3500	3	10500	0
2	San Ángel	Antara	1500	2	3000	0
3	Lomas de Sn.	Antara	2500	5	12500	0
4	Lomas de Sn.	Reforma	2000	2	4000	0
5	Lomas de Sn.	Palmas	1500	3	4500	0
6	Cuautitlán	Tlalpan	2500	2	5000	0
	Total	Objective	Function	Value =	39500	

Figura 2.55

La solución gráfica se presenta en la figura 2.56

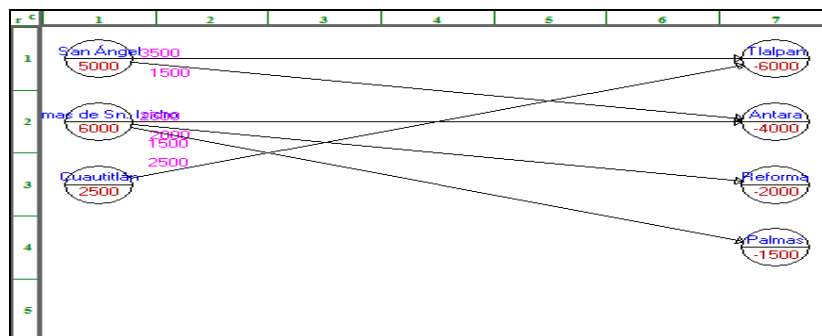


Figura 2.56

Es importante notar que no se ha agregado ninguna variable de holgura, ya que para este caso la oferta es igual a la demanda, por lo que no fue necesario introducir un destino u origen figurado como en el caso anterior. El programa WINQSB agrega automáticamente las variables de holgura en caso de ser necesarias; más adelante se presentan ejemplos que ilustran estos casos. Colmenero (2009) sugiere la siguiente metodología para usar esta plantilla del módulo:

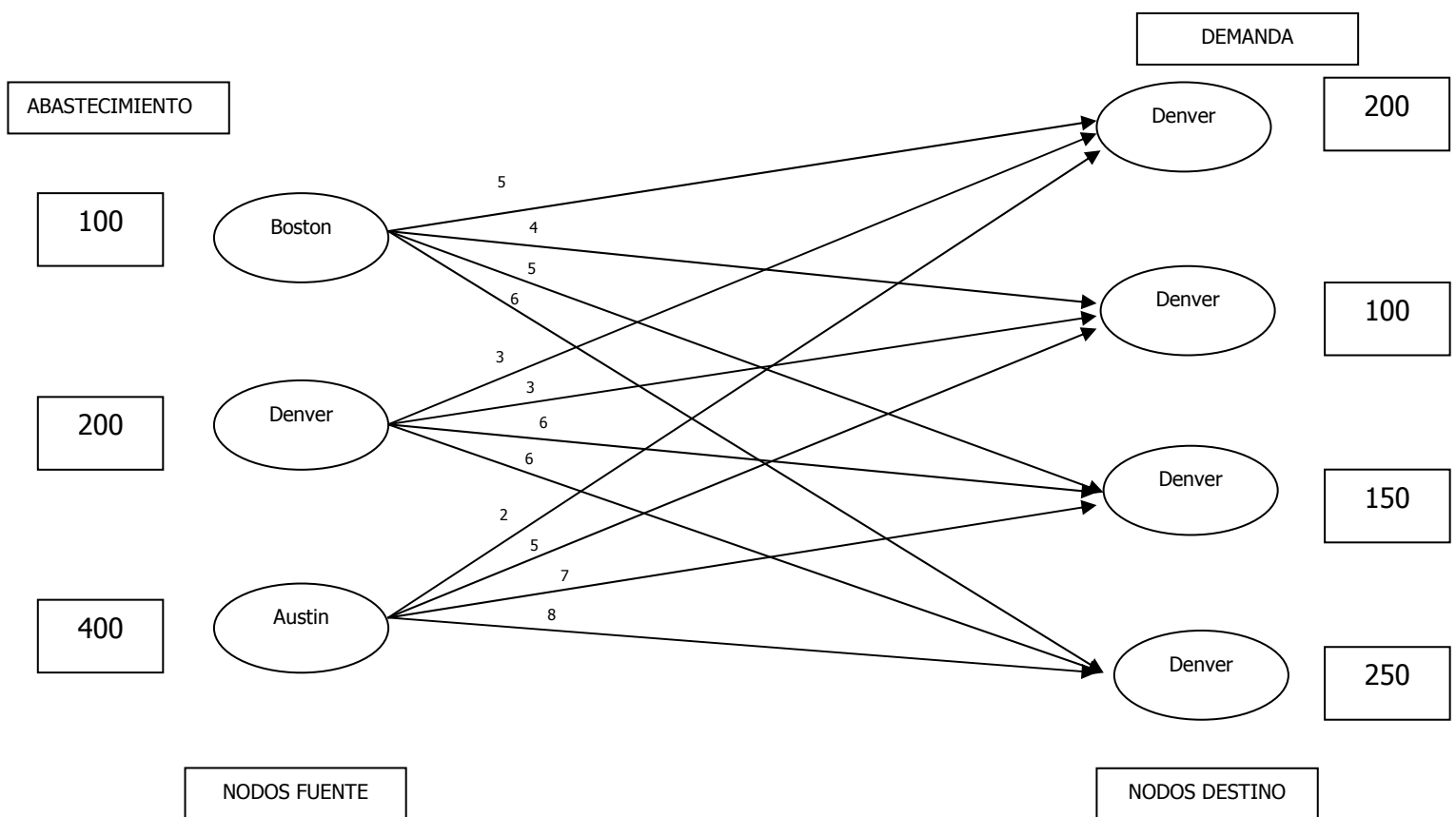
1. Analizar el problema. Definir las fuentes, los destinos y sus capacidades.
2. Determinar los costos/beneficios unitarios por envío entre los nodos.
3. Ingresar el problema dentro de WINQSB en la forma de matriz de hoja de cálculo o en la forma gráfica del modelo, así como el título del problema y el número nodos fuente (sources) y el número de nodos destino (destinations).
4. Revisar el problema si quiere verificar los datos que ingreso.
5. Modificar el problema si es necesario.
6. Guardar el problema ya que lo puede necesitar en consultas posteriores.

Como un segundo ejemplo tenemos el siguiente caso tomado de Colmenero (2009). Considere el siguiente problema; la compañía de petróleo ABC, cuenta con tres puntos para el almacenamiento del



petróleo y debe abastecer cuatro puntos de demanda. La gerencia necesita un programa para su envío de tal forma que se minimice el costo total por transporte sin que se excedan las capacidades de almacenamiento, se supone que las capacidades y los costos unitarios son los que se muestran en la tabla de la figura. La representación gráfica de la tabla anterior se muestra en la figura ¿Cuál es el programa óptimo para la distribución del petróleo desde los puntos de almacenamiento hasta los de demanda, que minimice el costo total por transporte?

Almacenamiento	Puntos de demanda				Capacidad de abastecimiento
	DALLAS	KANSAS	TAMPA	MIAMI	
BOSTON	5	4	5	6	100
DENVER	3	3	6	6	200
AUSTIN	2	5	7	8	400
DEMANDA	200	100	150	250	



Siguiendo la metodología los pasos 1 y 2 que indican que se analicen y determinen los costos unitarios de transporte tal como se muestra en la tabla y en la red de datos del problema.



El Paso 3. Ingrese el problema con el nombre "ABC" y la información que se muestra en la figura 2.57 para definir el problema, para aceptar esta información de un clic en el botón OK. Al hacer esto aparece la figura 2.58 que muestra la entrada de datos en Forma de Matriz de la Hoja de Cálculo (Spreadsheet Matrix Form).

Figura 2.57

From \ To	Destination 1	Destination 2	Destination 3	Destination 4	Supply
Source 1					0
Source 2					0
Source 3					0
Demand	0	0	0	0	

Figura 2.58

Los nombres de los nodos fuente y destino pueden ser modificados al ingresar al menú Editar > Nombres de los nodos (Edit > node names) como se muestra en la figura 2.59.

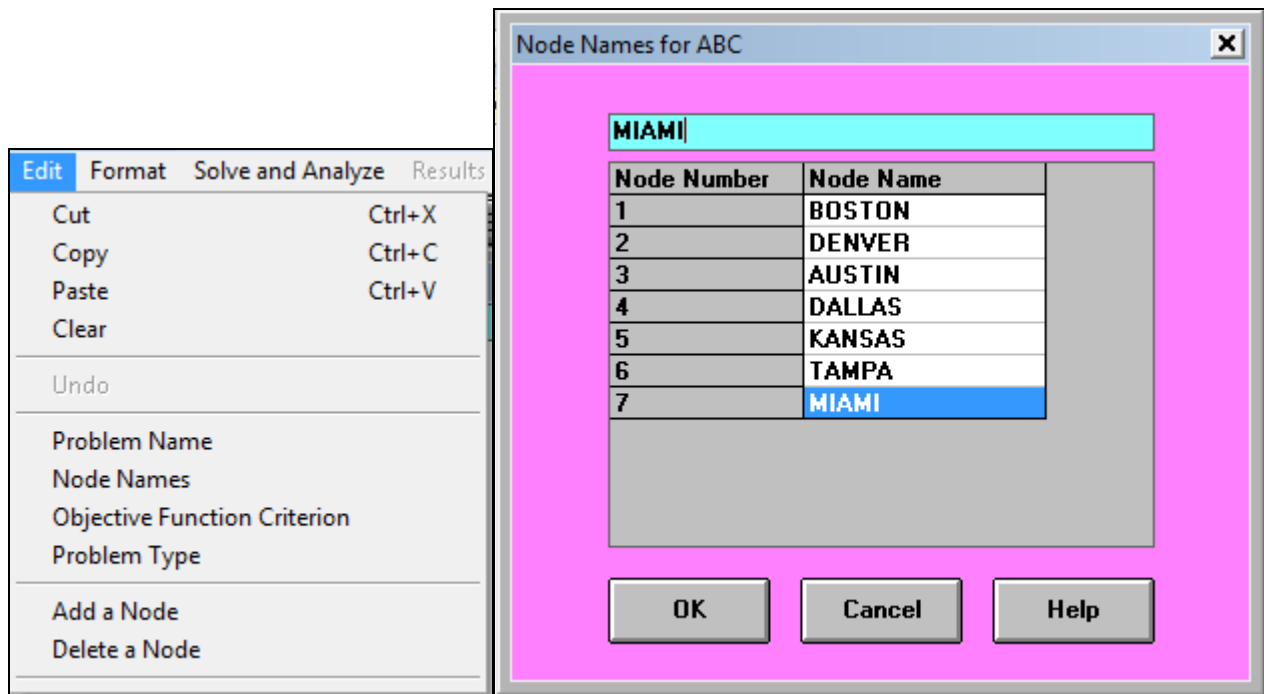


Figura 2.59

En la figura 2.60 aparece la entrada de datos: nombres de las fuentes y destinos, coeficientes de costos, demandas y capacidades de almacenamiento, tal y como aparecen en la tabla de la figura que son los datos del problema original.

From \ To	DALLAS	KANSAS	TAMPA	MIAMI	Supply
BOSTON	5	4	5	6	100
DENVER	3	3	6	6	200
AUSTIN	2	5	7	8	400
Demand	200	100	150	250	

Figura 2.60

El paso 4. El problema ahora puede ser resuelto, por lo que nos dirigimos al menú resolver y analizar y elegimos el método de solución inicial tal como se muestra en la figura 2.61.

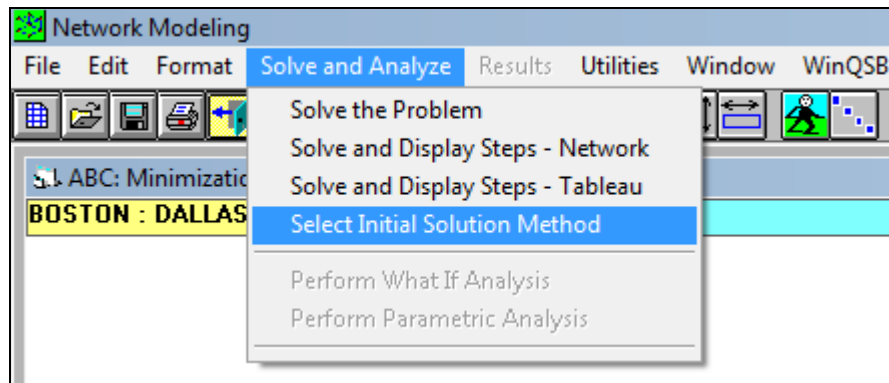


Figura 2.61

Las opciones disponibles para la solución inicial se muestran en la figura 2.62.

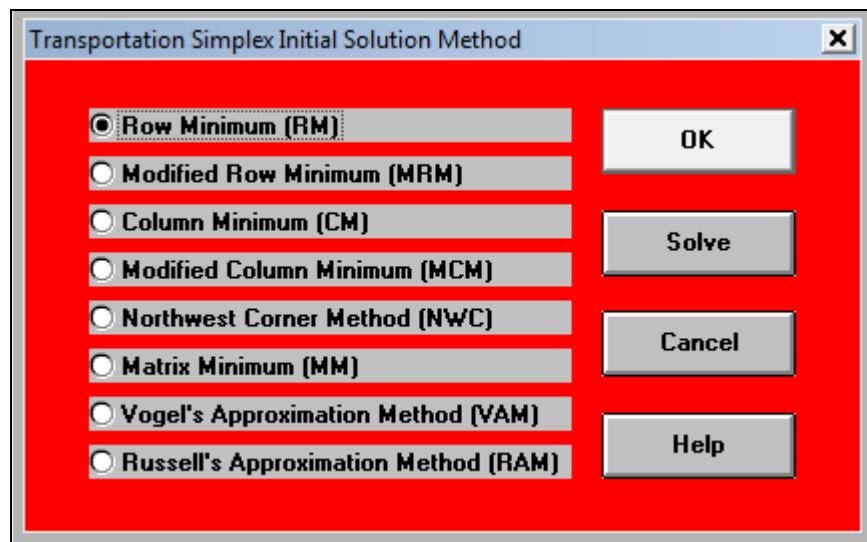


Figura 2.62

La opción por defecto es la primera que se refiere al Método de Mínimo Renglón. Se elige la opción para que el método inicial sea el de la esquina noroeste (Northwest Corner Method (NWC), tal como se muestra en la figura 2.63 y pulsamos Ok.

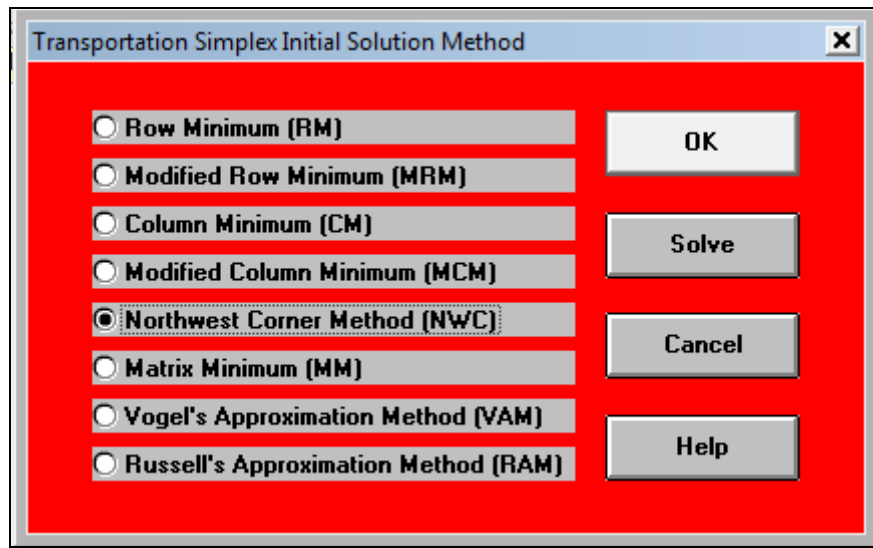


Figura 2.63

Seleccionamos la opción Resolver y mostrar los pasos – tabla. (Solve and Display Steps- tableau). Tal como se muestra en la figura 2.64.

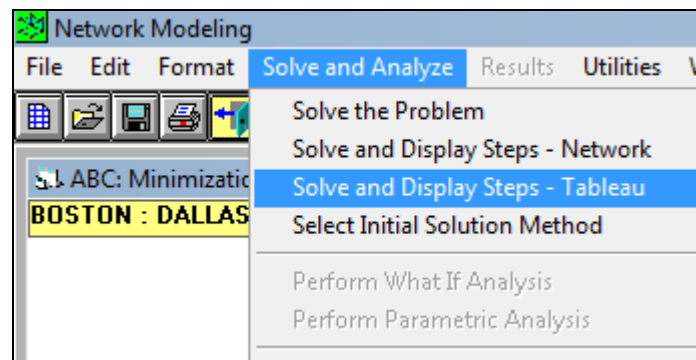


Figura 2.64

Las figuras de la 2.65 a la 2.69, muestran las iteraciones hasta llegar a la solución final.



Transportation Tableau for ABC - Iteration 1						
	DALLAS	KANSAS	TAMPA	MIAMI	Supply	Dual P(i)
BOSTON	5 100	4	5 Cij=-3 ^{***}	6	100	0
DENVER	3 100	3 100	6 0*	6	200	-2
AUSTIN	2	5	7 150	8 250	400	-1
Demand	200	100	150	250		
Dual P(j)	5	5	8	9		
Objective Value = 4150 (Minimization)						
*** Entering: BOSTON to TAMPA * Leaving: DENVER to TAMPA						

Figura 2.65

Para que el programa nos muestre la siguiente iteración basta con ir al menú iteración (iteration), tal como se muestra en la figura 2.65-1.

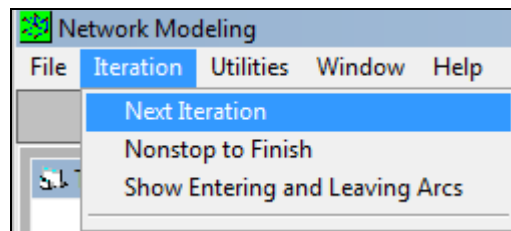


Figura 2.65-1



Transportation Tableau for ABC - Iteration 2						
From \ To	DALLAS	KANSAS	TAMPA	MIAMI	Supply	Dual P(i)
BOSTON	5 100*	4	5 0	6	100	0
DENVER	3 100	3 100	6	6	200	-2
AUSTIN	2 Cij=-5**	5	7 150	8 250	400	2
Demand	200	100	150	250		
Dual P(j)	5	5	5	6		
Objective Value = 4150 (Minimization)						
** Entering: AUSTIN to DALLAS * Leaving: BOSTON to DALLAS						

Figura 2.66

Transportation Tableau for ABC - Iteration 3						
From \ To	DALLAS	KANSAS	TAMPA	MIAMI	Supply	Dual P(i)
BOSTON	5	4	5 100	6	100	0
DENVER	3 100*	3 100	6	6 Cij=-3**	200	3
AUSTIN	2 100	5	7 50	8 250	400	2
Demand	200	100	150	250		
Dual P(j)	0	0	5	6		
Objective Value = 3650 (Minimization)						
** Entering: DENVER to MIAMI * Leaving: DENVER to DALLAS						

Figura 2.67



S.L. Transportation Tableau for ABC - Iteration 4 (Final)						
From \ To	DALLAS	KANSAS	TAMPA	MIAMI	Supply	Dual P(i)
BOSTON	5	4	5	6	100	0
			100			
DENVER	3	3	6	6	200	0
		100		100		
AUSTIN	2	5	7	8	400	2
	200		50	150		
Demand	200	100	150	250		
Dual P(j)	0	3	5	6		
	Objective Value = 3350 (Minimization)					

Figura 2.68

11-11-2009	From	To	Shipment	Unit Cost	Total Cost	Reduced Cost
1	BOSTON	TAMPA	100	5	500	0
2	DENVER	KANSAS	100	3	300	0
3	DENVER	MIAMI	100	6	600	0
4	AUSTIN	DALLAS	200	2	400	0
5	AUSTIN	TAMPA	50	7	350	0
6	AUSTIN	MIAMI	150	8	1200	0
	Total	Objective	Function	Value =	3350	

Figura 2.69

El paso 5 señala que después de resolver el problema es posible ir al menú y seleccionar archivo (file) para imprimir o guardar la solución final. El menú de opciones disponible se muestra en la figura 2.70.

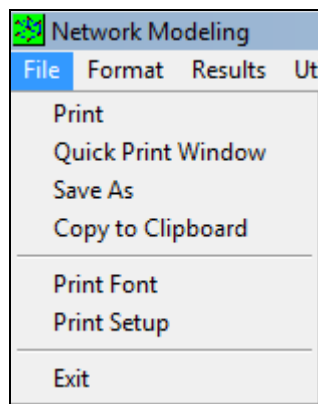


Figura 2.70

La opción guardar como (save as) guarda la solución en formato de imagen .BMP. Al acceder al menú Resultados (Results) podemos seleccionar la opción Solución gráfica (Graphic Solution) tal como se muestra en la figura 2.71.

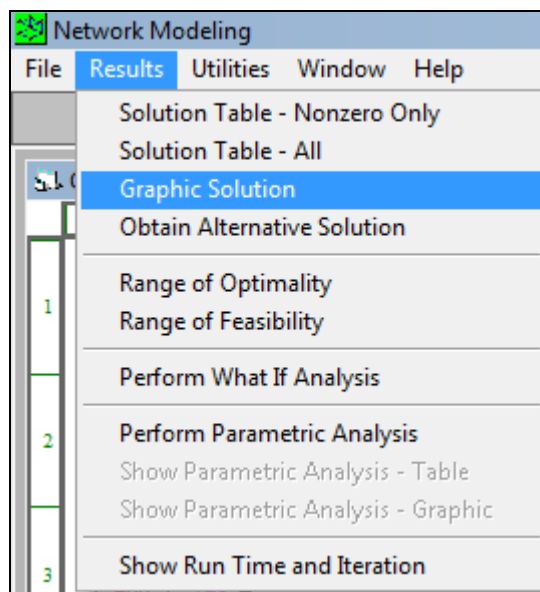


Figura 2.71

Solución gráfica figura 2.72

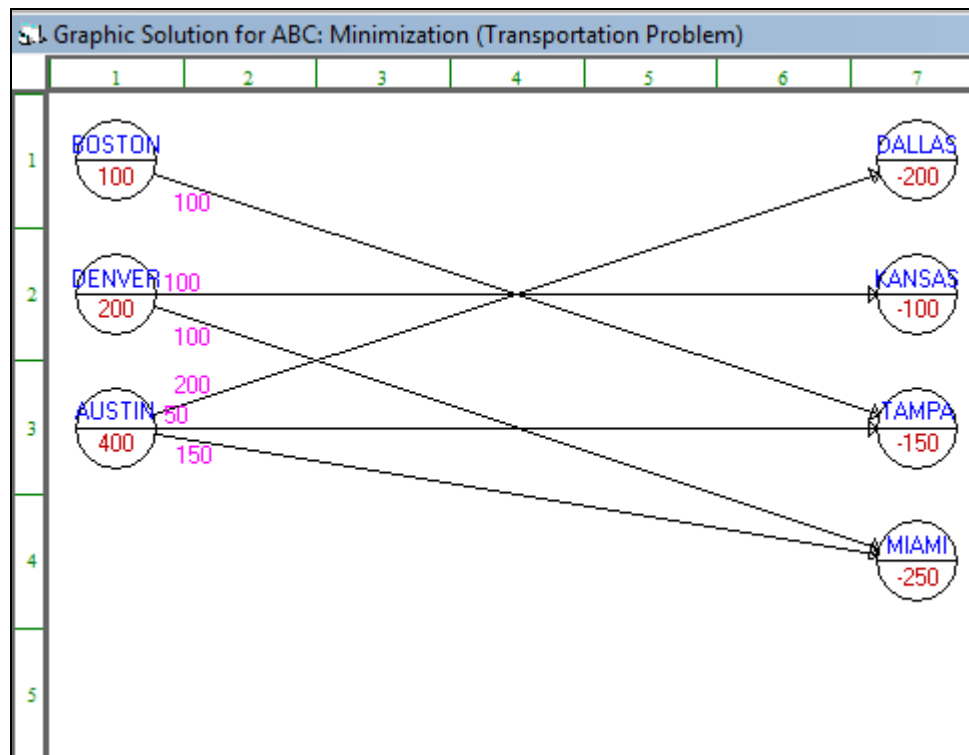


Figura 2.72

II-2.3 PROBLEMA DE ASIGNACIÓN

Este modelo es un caso particular del problema de transporte, aquí existe una matriz cuadrada y cada restricción de disponibilidad, y cada uno de los requerimientos son iguales a la unidad. Con este modelo se busca resolver de la manera más eficiente, la asignación de una serie de tareas a máquinas u operarios, a un costo dado. El programa resuelve el algoritmo de asignación por el Método Húngaro, para una explicación exhaustiva el lector puede consultar en la bibliografía de referencia.

Para ilustrar la aplicación se muestra el siguiente ejemplo:

Origen \ Destino	Restaurante A	Restaurante B	Restaurante C
Chef Roberto	10	15	9
Chef Efraín	9	18	5
Chef Aldo	6	14	3



Se trata de una cadena de restaurantes, la cual ha recibido solicitud de tres de sus sucursales demandando la asistencia de un chef corporativo con la finalidad de que ayude a capacitar a sus cocineros en el proceso de elaboración de un nuevo platillo.

Los números en las celdas son las horas (también podrían representar el costo) que invertiría cada chef en llevar a cabo cada asesoría; las diferencias en horas se deben a las habilidades de cada uno de los chefs y a su ubicación actual. El corporativo desea minimizar el tiempo total dedicado a los proyectos. ¿Cómo asignar los chefs a cada una de las sucursales?

El primer paso para resolver este tipo de problema es elegir la opción Problema de asignación (Assignment Problem) de la figura y seleccionar minimizar, así como el número de los objetos y las asignaciones. La figura 2.73 muestra como se introducen los datos para este problema.

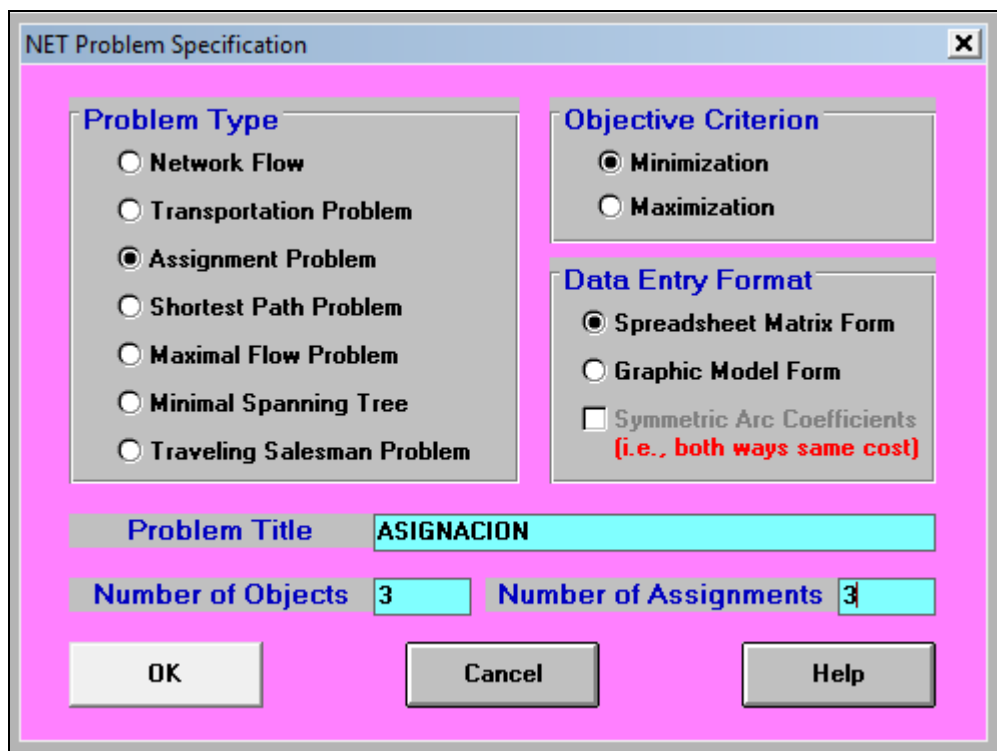


Figura 2.73

Pulsamos OK si estamos de acuerdo. Y aparece una ventana como la de la figura 2.74



From \ To	Assignee 1	Assignee 2	Assignee 3
Assignment 1			
Assignment 2			
Assignment 3			

Figura 2.74

Es posible cambiar los nombres de las asignaciones tal como se muestra en la figura 2.75.

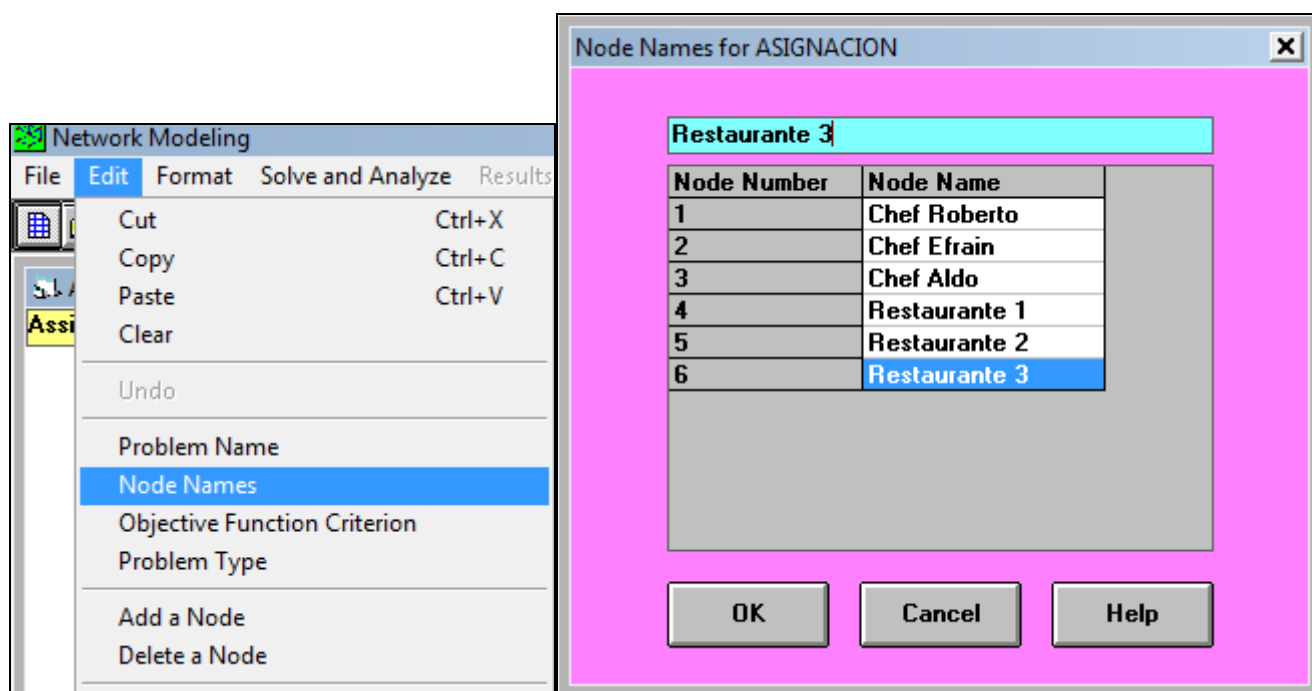


Figura 2.75

Una vez que hemos cambiado los nombres de las celdas pulsamos OK. Y aparece una ventana como la mostrada en la figura 2.76.

From \ To	Restaurante	Restaurante	Restaurante
Chef Roberto			
Chef Efrain			
Chef Aldo			

Figura 2.76



Al ingresar los datos obtenemos la ventana siguiente (figura 2.77)

From \ To	Restaurante	Restaurante	Restaurante
Chef Roberto	10	15	9
Chef Efrain	9	18	5
Chef Aldo	6	14	3

Figura 2.77

Nos dirigimos al menú resolver y analizar, y seleccionamos la opción que nos permita ver las iteraciones paso a paso en un formato de tabla, tal como lo muestra la figura 2.77-1.

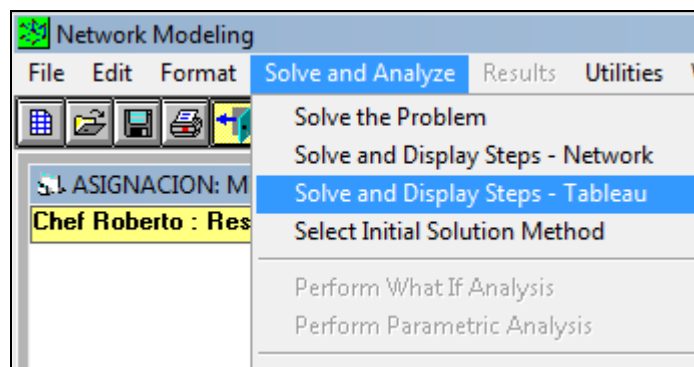


Figura 2.77-1

Las iteraciones se muestran en la secuencia de figuras 2.78- 2.79.

Hungarian Method for ASIGNACION - Iteration			
	Restaurante	Restaurante	Restaurante
Chef Robert	0	0	0
Chef Efrain	3	7	0
Chef Aldo	2	5	0

Figura 2.78



From \ To	Restaurante 1	Restaurante 2	Restaurante 3
Chef Roberto	0	0	2
Chef Efraín	7	5	0
Chef Aldo	0	3	0

Figura 2.79

Finalmente obtenemos la tabla de la figura 2.80 que nos indica que el Chef Roberto se le asigna el Restaurante 2, el Chef Efraín se le asigna el Restaurante 3 y al Chef Aldo se le asigna el Restaurante 1, con un total de 26 horas.

12-04-2008	From	To	Assignment	Unit Cost	Total Cost	Reduced Cost
1	Chef Roberto	Restaurante 2	1	15	15	0
2	Chef Efraín	Restaurante 3	1	5	5	0
3	Chef Aldo	Restaurante 1	1	6	6	0
	Total	Objective	Function	Value =	26	

Figura 2.80.

Cualquier otra asignación de los Chefs corporativos a cada sucursal, resulta económicamente más costoso, retamos al lector a que lo intente asignando de diferente forma a cada uno de los Chefs.



II-2.4 EL PROBLEMA DE LA RUTA MÁS CORTA

El problema de la ruta más corta incluye un juego de nodos conectados donde sólo un nodo es considerado como el origen y sólo un nodo es considerado como el nodo destino. El objetivo es determinar un camino de conexiones que minimicen la distancia total del origen al destino. La limitación del programa WINQSB es que cuando se tiene la situación en la que tenemos más de un origen o más de un destino, el software no es capaz de solventar esta complicación; sin embargo el lector interesado en ahondar más en el tema puede consultar las referencias bibliográficas.

Consideremos el siguiente problema extraído de las notas de clase de Colmenero (2009). *Muebles y Mudanzas* fue contratada para mudar los muebles de oficina y equipo de la empresa de bienes raíces "Propiedades, S. A." a sus nuevas oficinas. ¿Qué ruta recomienda? Las distancias están dadas en kilómetros. La red de carreteras se muestra en la figura 2.81.

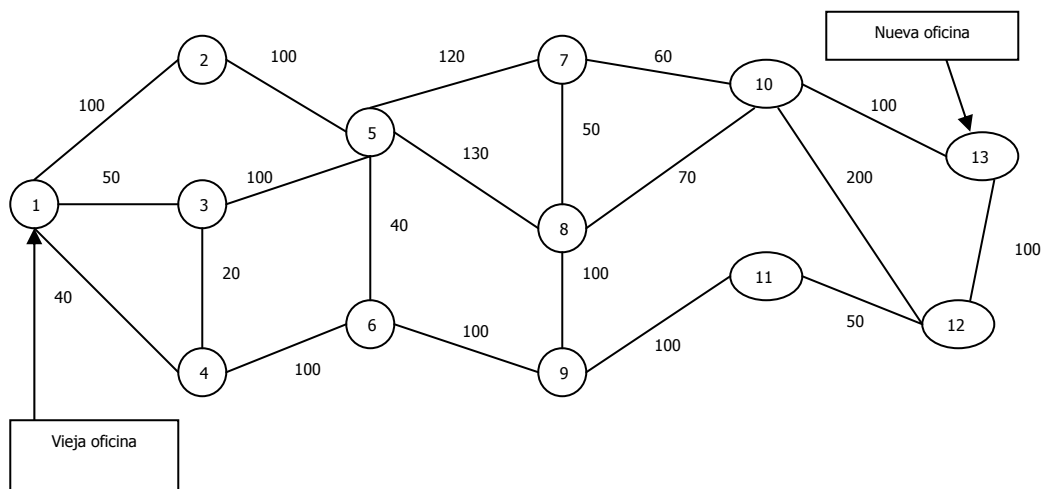


Figura 2.81

Para el uso de esta plantilla seleccionamos el problema de la ruta más corta (shortest path problem), escribimos el nombre del problema y el número de nodos tal como se muestra en la figura 2.82.

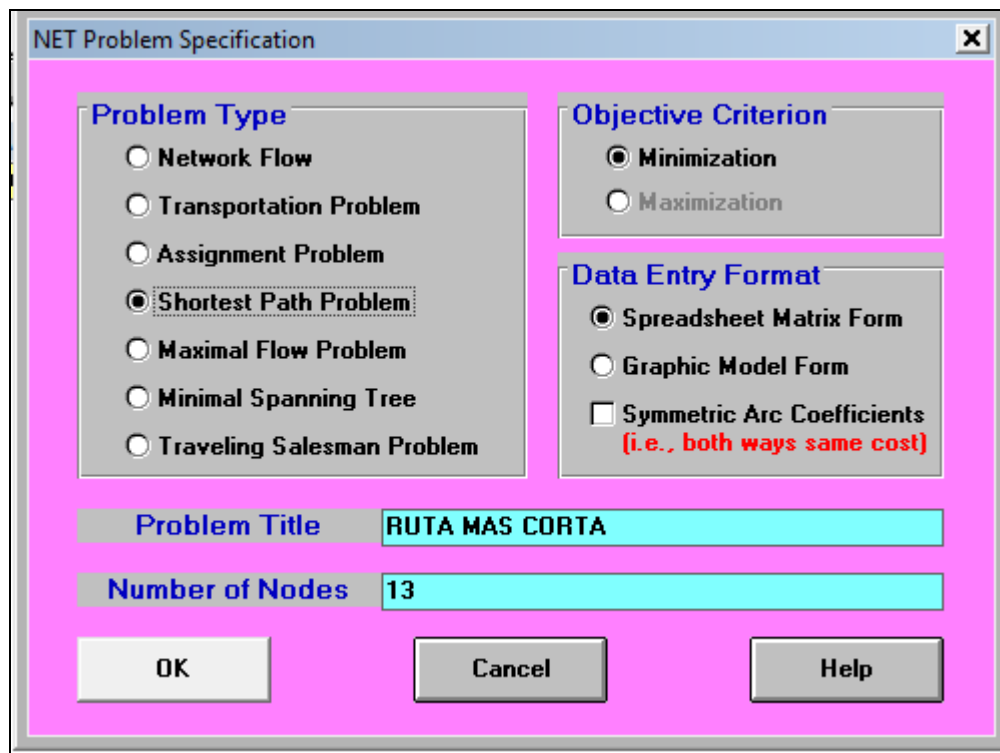


Figura 2.82

Si se está conforme con los datos que se han ingresado pulsamos Ok. Y aparecerá una ventana como la siguiente (figura 2.83):

From \ To	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8	Node9	Node10	Node11	Node12	Node13
Node1													
Node2													
Node3													
Node4													
Node5													
Node6													
Node7													
Node8													
Node9													
Node10													
Node11													
Node12													
Node13													

Figura 2.83

Se cambian los nombres de los nodos, como se ha hecho en los casos anteriores; la tabla resultante se muestra a continuación. (Figura 2.84).



From \ To	1	2	3	4	5	6	7	8	9	10	11	12	13
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													

Figura 2.84

Se ingresan los datos de la forma siguiente (Figura 2.85).

From \ To	1	2	3	4	5	6	7	8	9	10	11	12	13
1		100	50	40									
2					100								
3				20	100								
4						100							
5						40	120	130					
6									100				
7								50		60			
8									100	70			
9											100		
10												200	100
11												50	
12													100
13													

Figura 2.85

Para visualizarlo en modo grafico nos dirigimos al menú formato y seleccionamos, mostrar el modo gráfico, tal como se muestra en la figura 2.86.

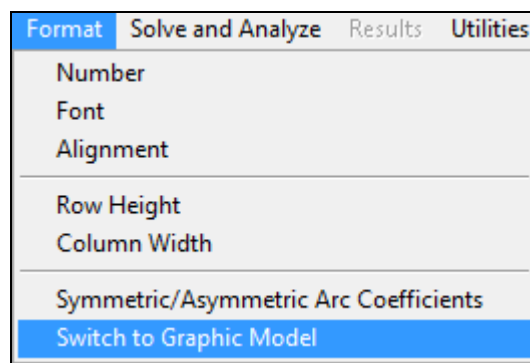


Figura 2.86

A continuación aparecerá una figura como la siguiente (Figura 2.87).

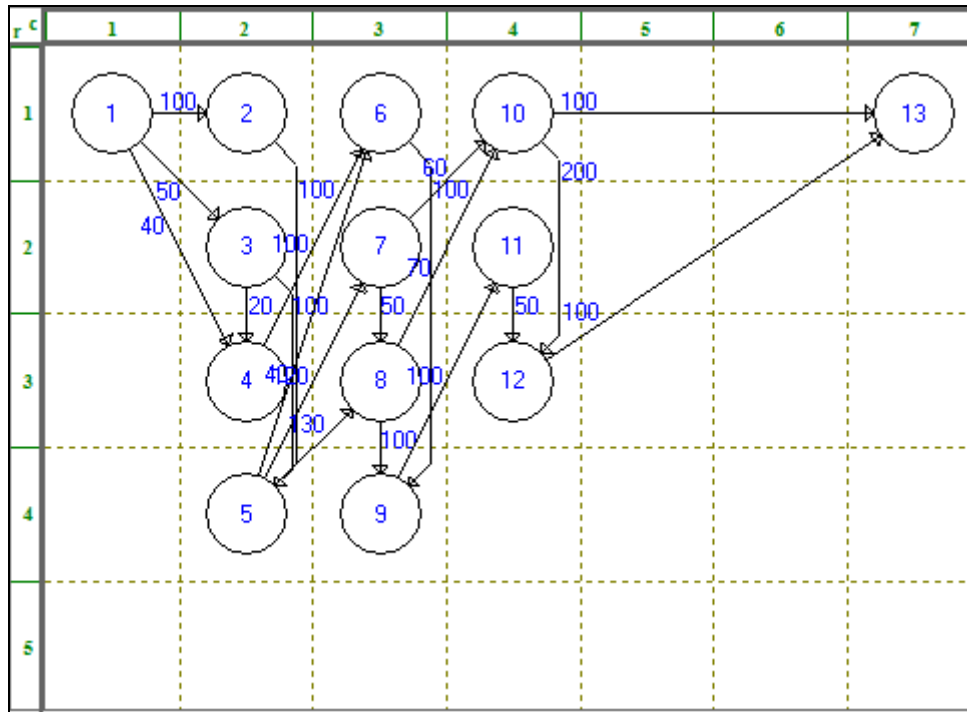


Figura 2.87

Esta figura no es muy clara y puede llegar a confundir, por lo que es posible editarla desde el menú editar, tal como se muestra en la figura 2.88; esta opción despliega un segundo tablero que se muestra en la figura 2.89.

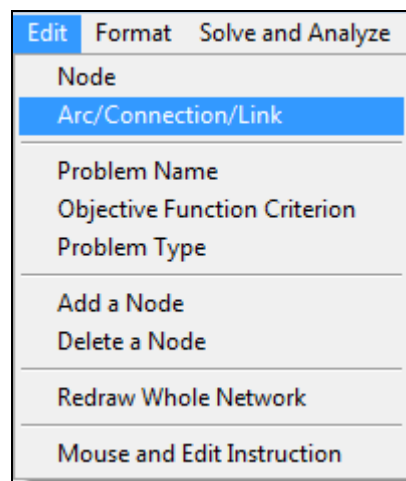


Figura 2.88

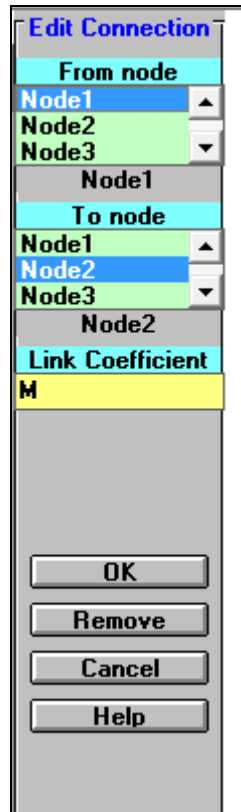


Figura 2.89

Aún así es difícil reacomodar los nodos y crear una red con presentación; por lo que se recomienda salir completamente de la aplicación y entrar de nuevo, pero esta vez seleccionar como formato de entrada de datos la forma grafica; la figura 2.90 muestra como se realiza este procedimiento.

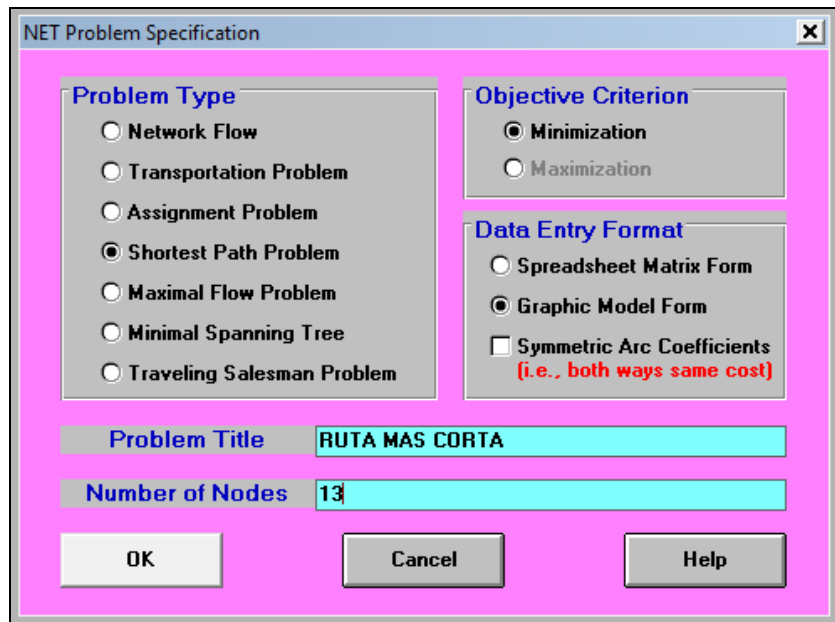


Figura 2.90

Al pulsar OK. Aparece la Figura 2.91 mostrada a continuación:

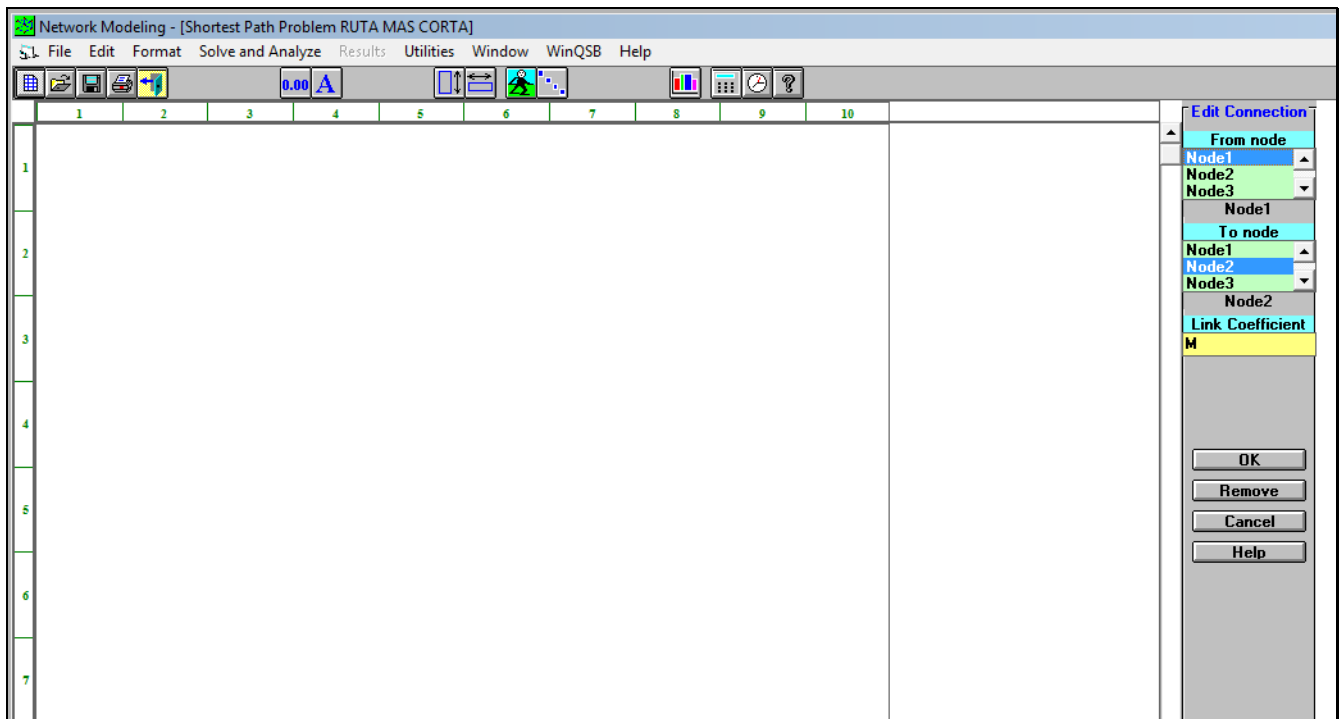


Figura 2.91



Para comenzar a dibujar la red de nuestro ejemplo, damos un clic en Nodo 1 (Node 1) del menú editar conexión (edit connection); una vez que hemos hecho esto, nos dirigimos al cuadrante 1:1 y damos un doble clic; al hacer esto aparece un nodo 1 tal como se muestra en la figura 2.92.

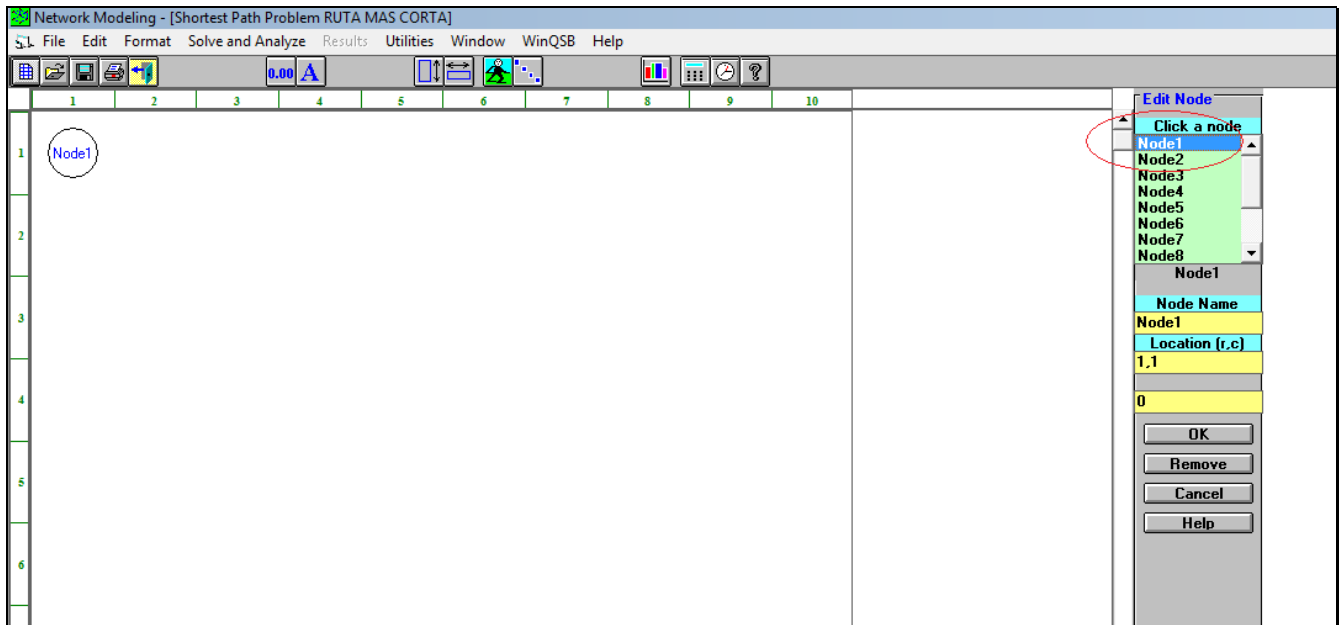


Figura 2.92

De esta forma hemos creado el nodo inicio de nuestra red; es posible cambiar el nombre de los nodos, desde el menú de edición de nodos, en el campo nombre de los nodos (node name); al cambiar el nombre de los nodos tendremos la siguiente apariencia Figura 2.93.

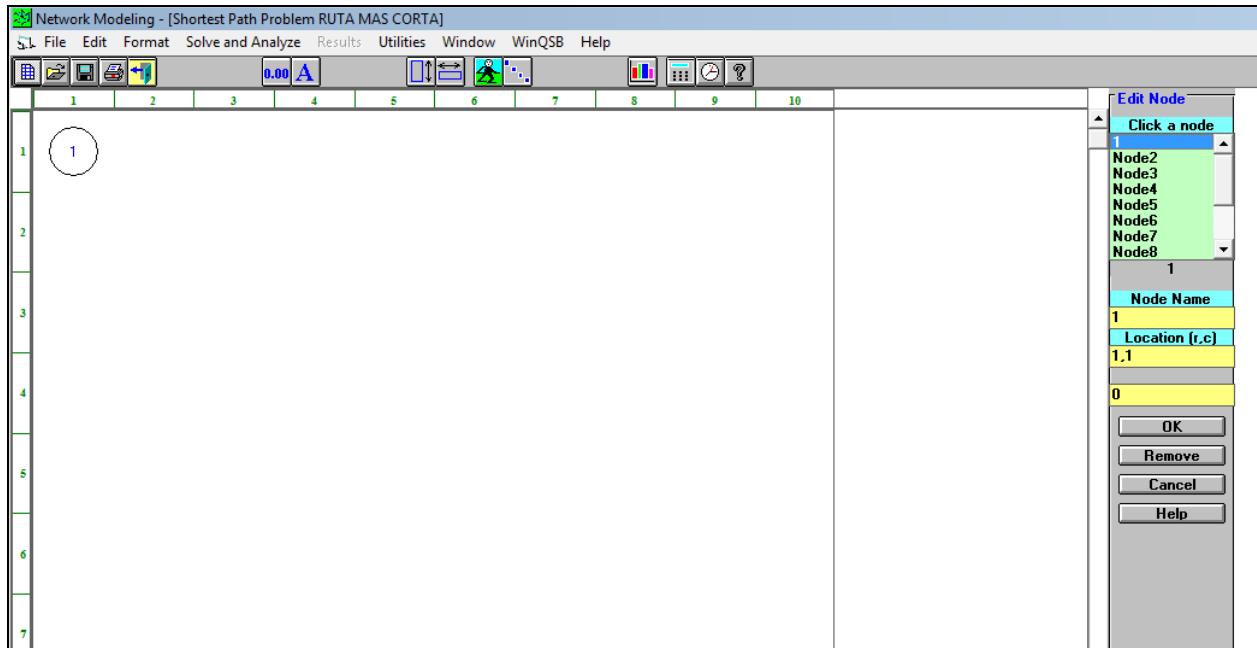


Figura 2.93

De esta forma se procede a nombrar y a insertar los trece nodos que conforman nuestra red.

La figura 2.94 ilustra la apariencia de la red hasta este punto.

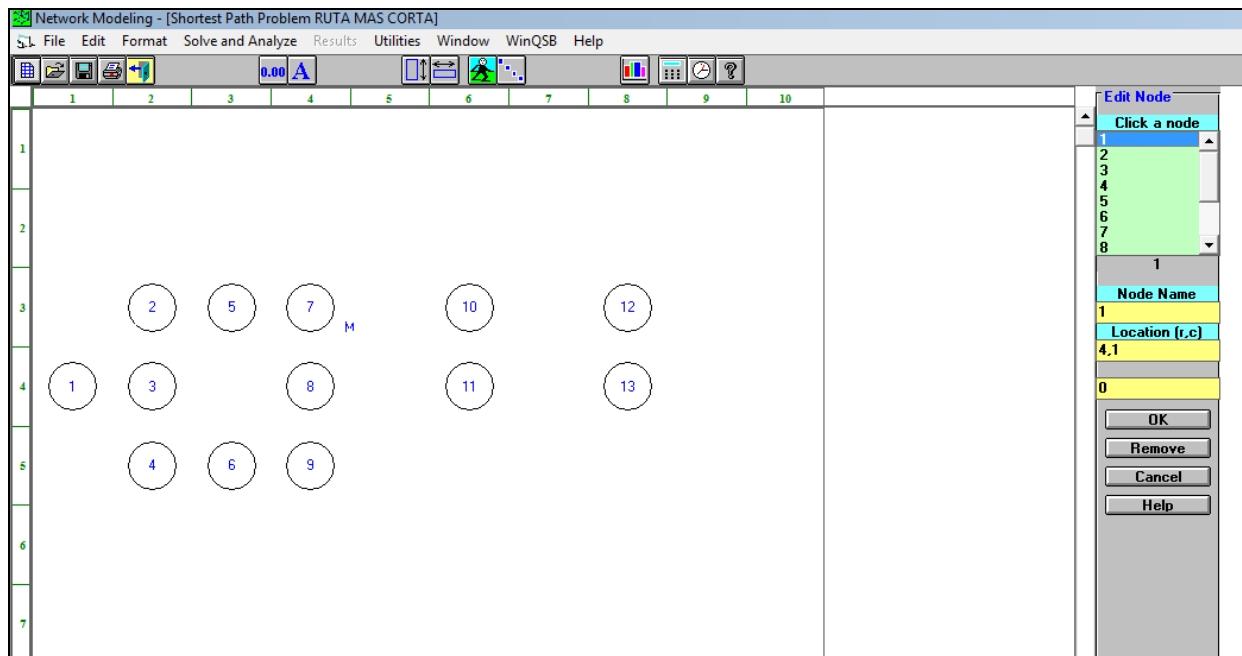


Figura 2.94



Del nodo 1 al nodo 2 tenemos un coeficiente de 100 como lo muestra la figura 2.95. Por lo que en el sub menú que nos sirve para editar la conexión, debemos cerciorarnos de que el nodo origen sea el nodo 1 (from node) y el nodo destino (to node) sea el nodo 2; el valor del arco debe ser cien en el campo (link coefficient). Pulsamos OK. Y obtenemos la figura 2.96 que ilustra el valor del arco. De esta manera procedemos a construir nuestra red del ejemplo. La figura 2.97 muestra la red final.

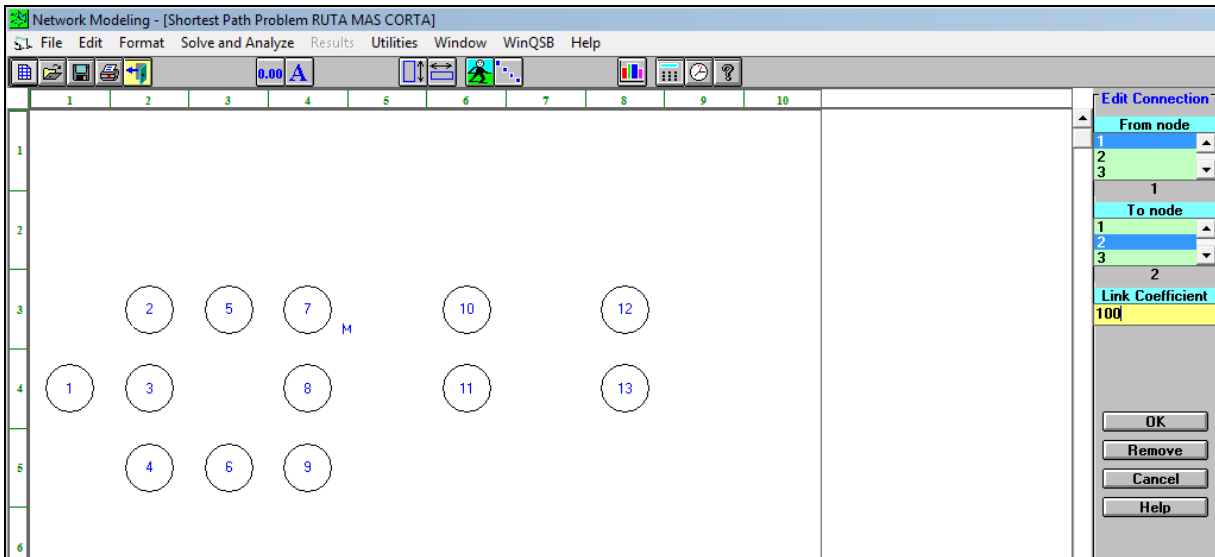


Figura 2.95

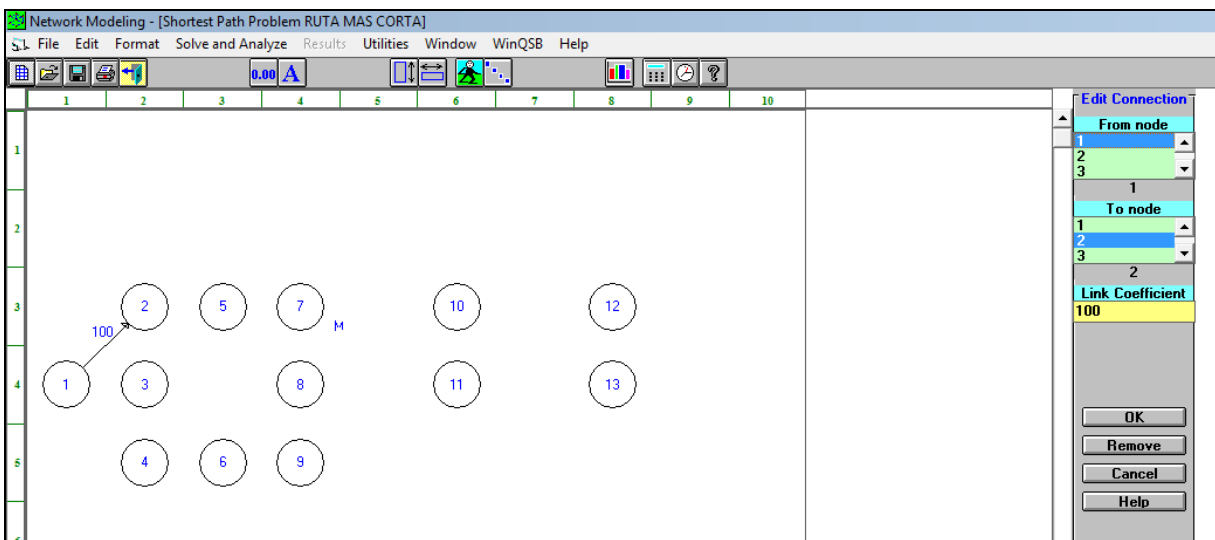


Figura 2.96

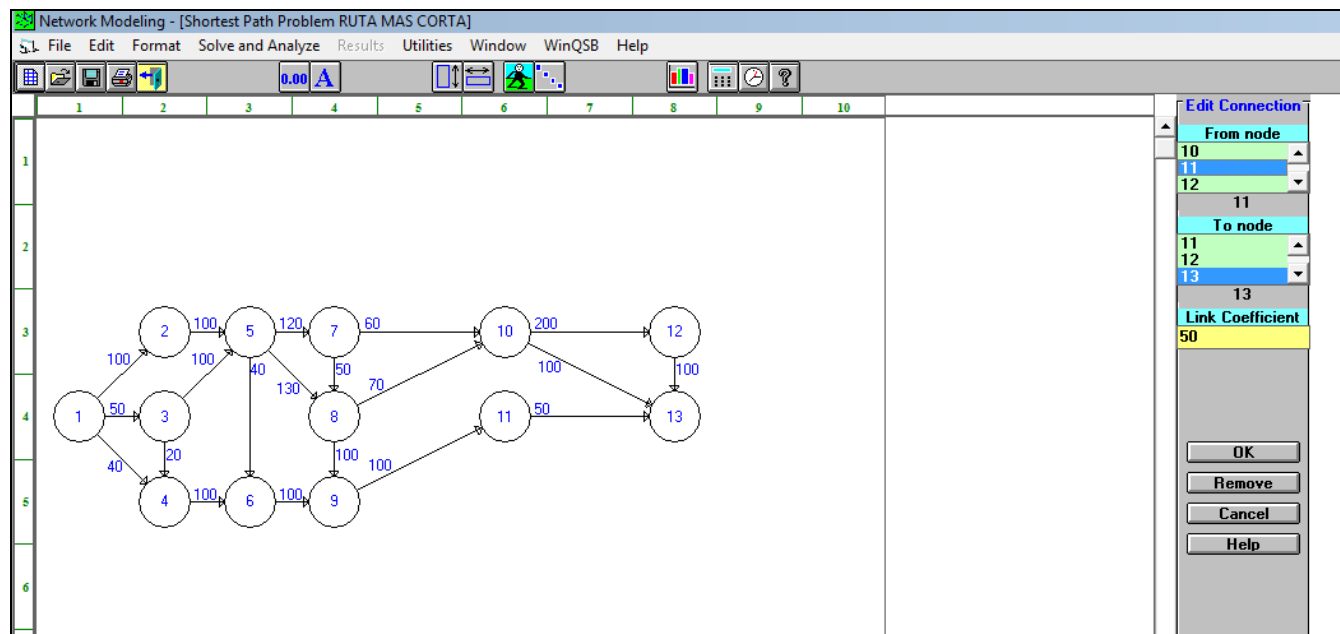


Figura 2.97

Nos dirigimos al menú resolver y analizar y posteriormente elegimos la opción resolver por pasos, tal como se muestra en la figura 2.98. Una explicación exhaustiva del algoritmo de la ruta más corta podemos encontrarlo en Pinillos (1994).

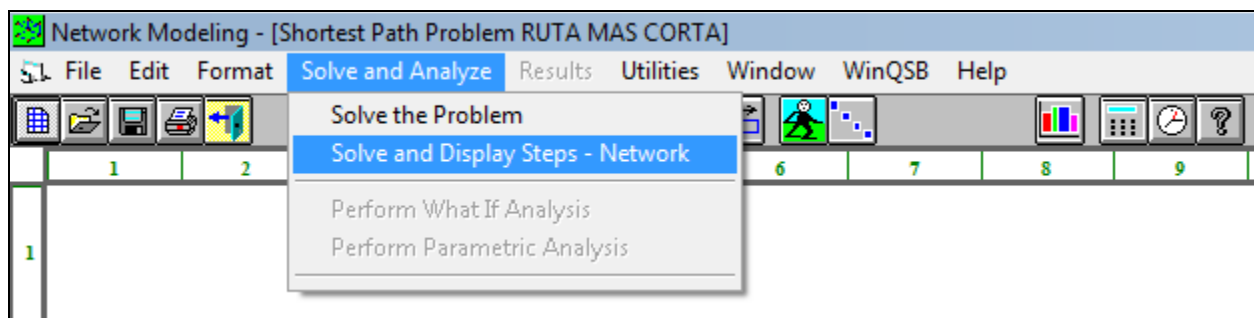


Figura 2.98

Enseguida emerge una ventana como la siguiente (Figura 2.91) la cual nos pide que elijamos cuál es el nodo inicio y cuál es el nodo final; como nodo inicio elegimos el nodo 1 y como nodo final el nodo número 13; damos un clic en resuelve y muestra los pasos (solve and display steps), o en resolver si lo queremos de manera directa.

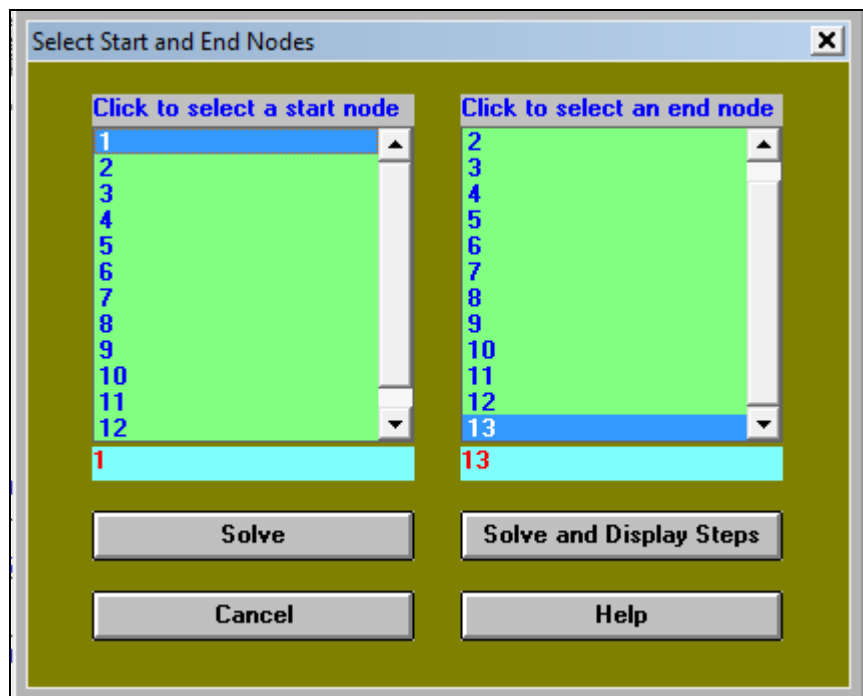


Figura 2.99

Las figuras de la 2.100 a la 2.110 muestran el procedimiento iteración por iteración.

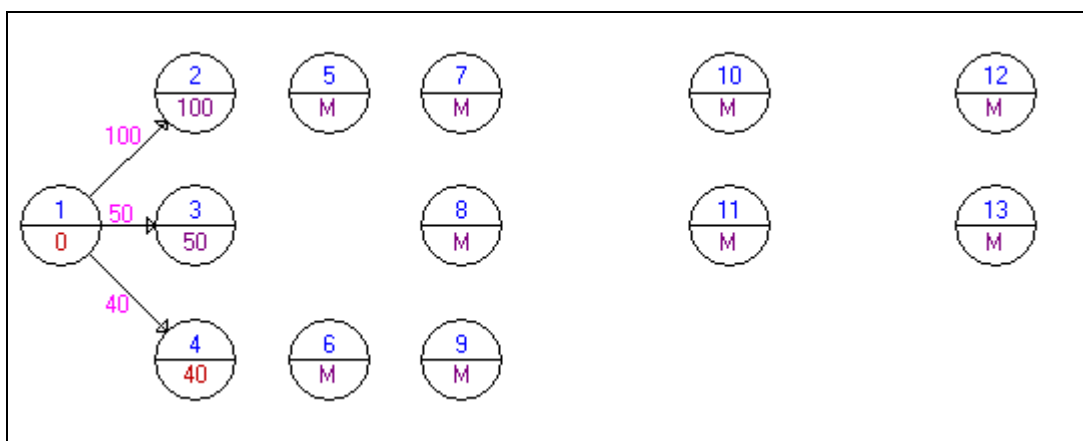


Figura 2.100

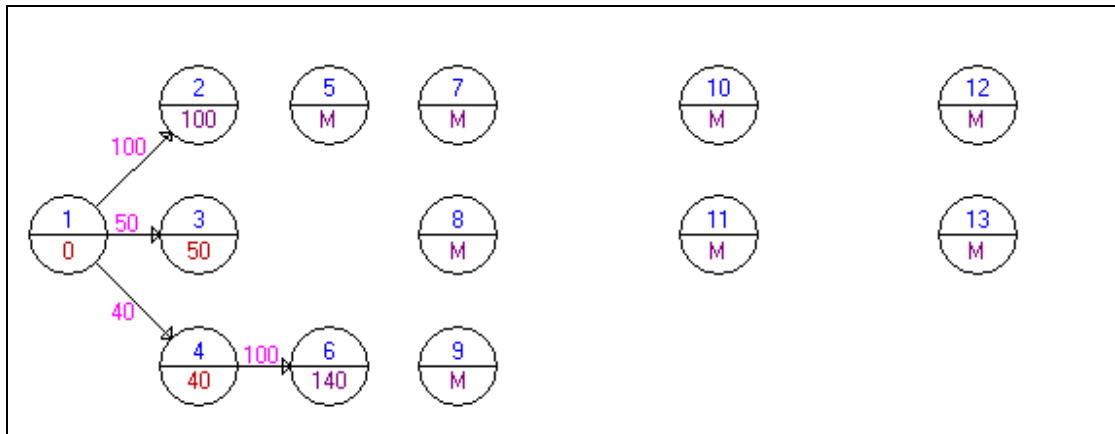


Figura 2.101

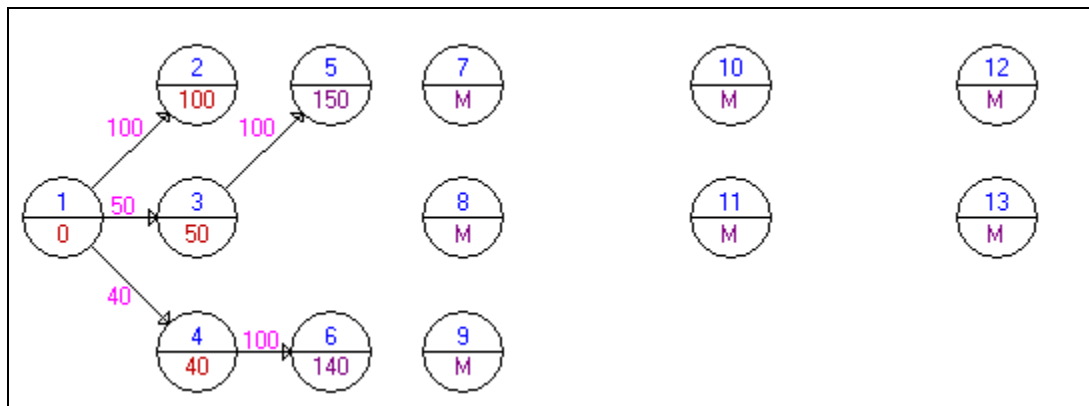


Figura 2.102

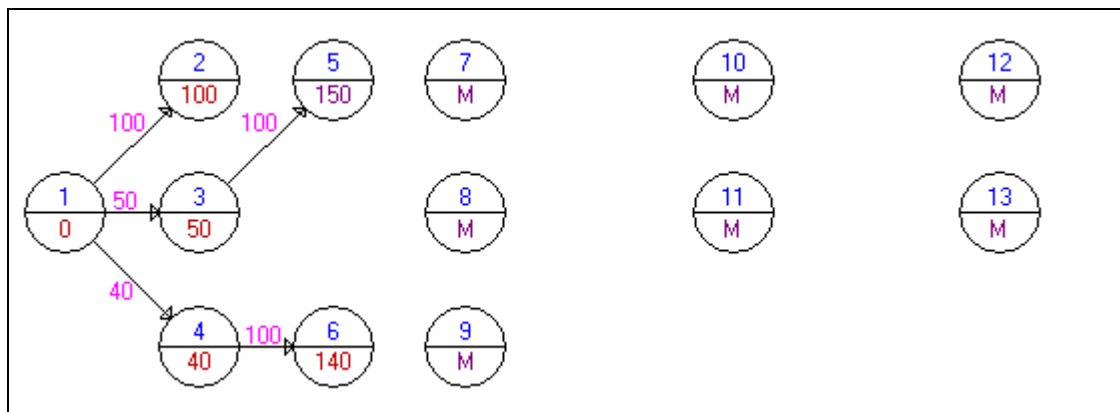


Figura 2.103

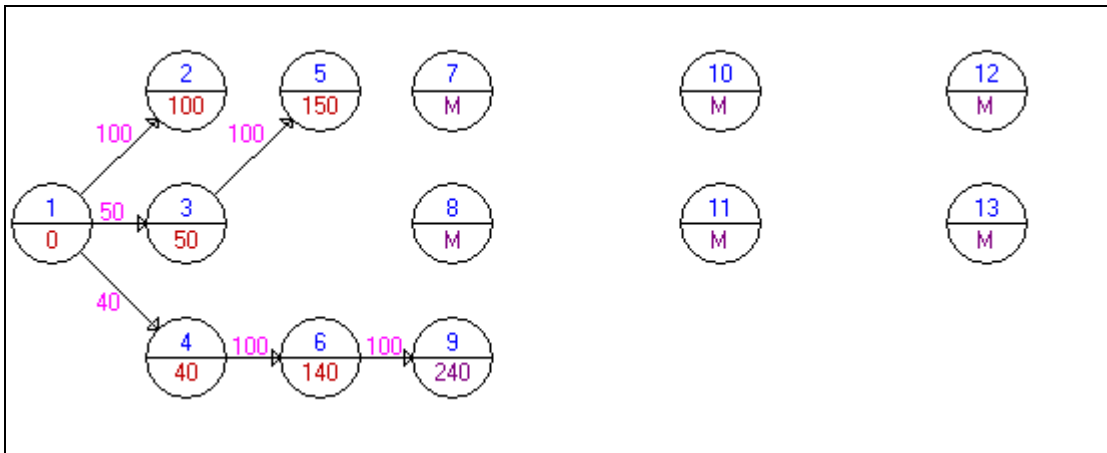


Figura 2.104

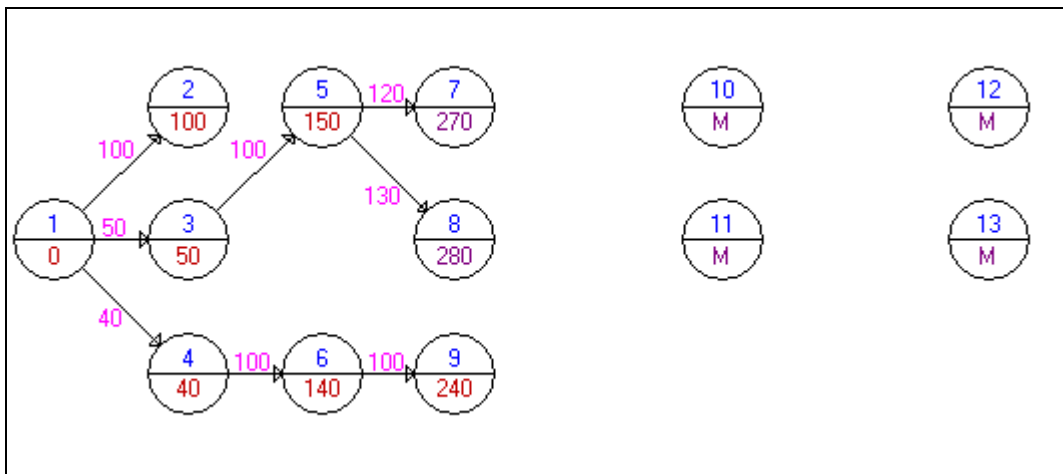


Figura 2.105

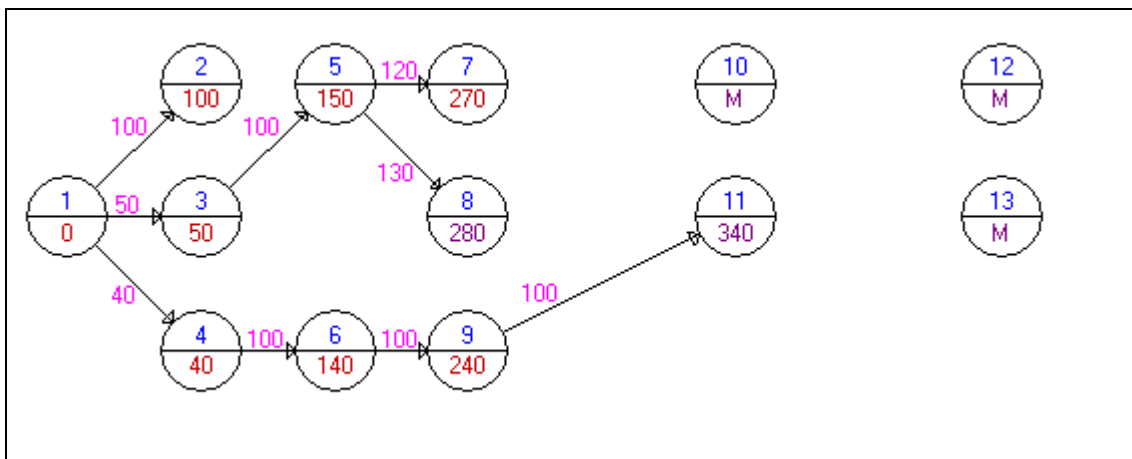


Figura 2.106

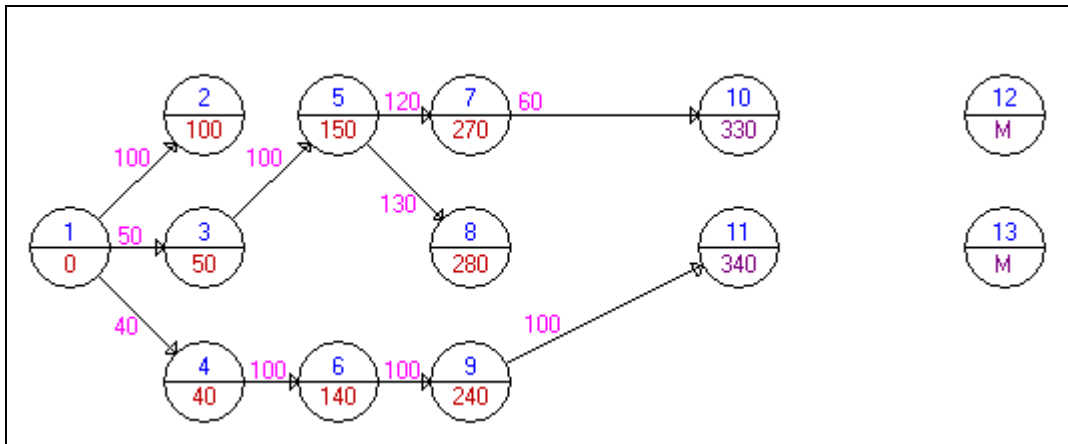


Figura 2.107

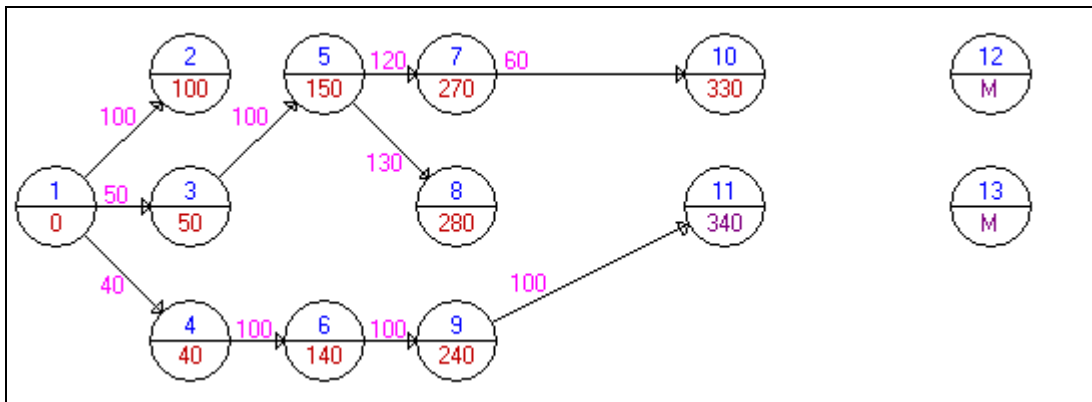


Figura 2.108

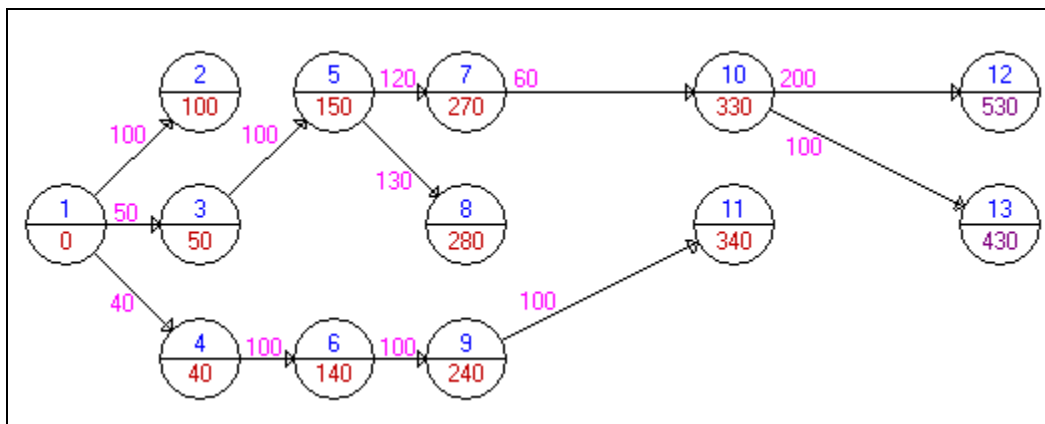


Figura 2.109

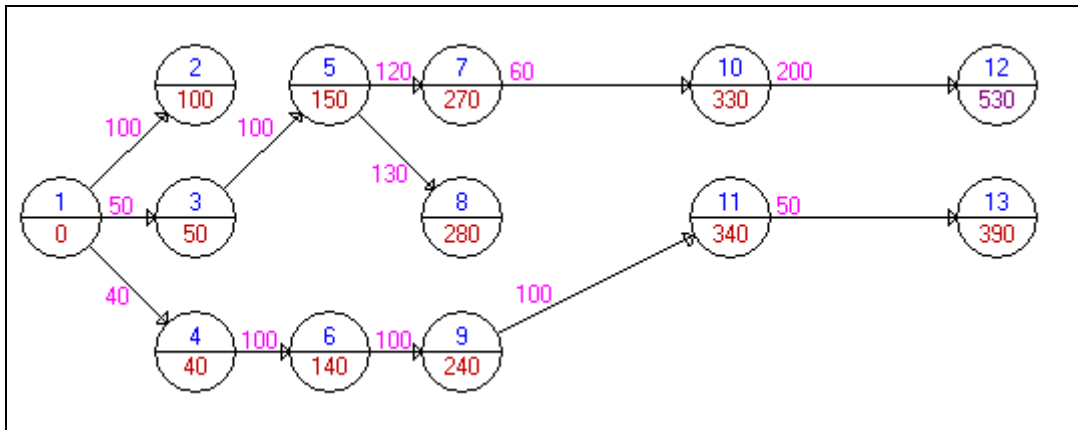


Figura 2.110

La red muestra la ruta más corta (Figura 2.111)

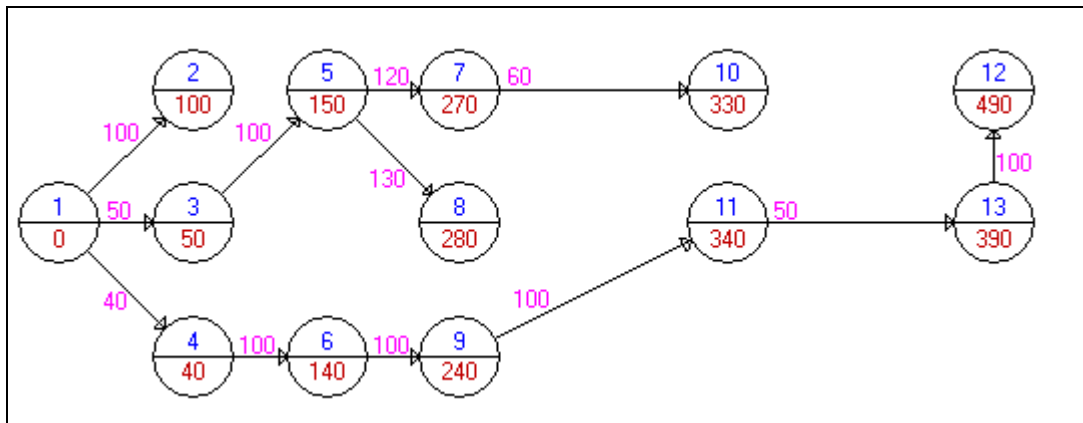


Figura 2.111

La figura 2.112 muestra cual es la ruta más corta en forma tabular con un costo/distancia acumulada de 390, y la secuencia de los nodos es dirigirnos del nodo 1-4-6-9-11-13.



11-12-2009	From	To	Distance/Cost	Cumulative Distance/Cost
1	1	4	40	40
2	4	6	100	140
3	6	9	100	240
4	9	11	100	340
5	11	13	50	390
	From 1	To 13	=	390
	From 1	To 2	=	100
	From 1	To 3	=	50
	From 1	To 4	=	40
	From 1	To 5	=	150
	From 1	To 6	=	140
	From 1	To 7	=	270
	From 1	To 8	=	280
	From 1	To 9	=	240
	From 1	To 10	=	330
	From 1	To 11	=	340
	From 1	To 12	=	490

Figura 2.112

Como un segundo ejemplo tomado de Quesada (2008). Consideremos la siguiente red (figura 2.113).

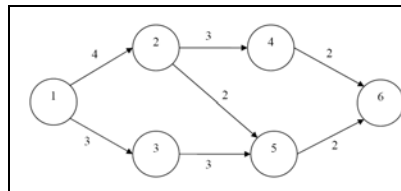


Figura 2.113

Para comenzar el análisis seleccionamos la aplicación Problema de la ruta más corta (Shortest Path Problem) e introducimos los datos como se muestra a continuación (Figura 2.114).

From \ To	INICIO	ESTACION2	ESTACION3	ESTACION4	ESTACION5	FIN
INICIO		4	3			
ESTACION2				3	2	
ESTACION3					3	
ESTACION4						2
ESTACION5						2
FIN						

Figura 2.114



El nodo 1 identificado como INICIO representa una central termoeléctrica y el nodo 6 representa la ciudad a donde debe llevarse el cableado, pasando por algunos de los otros nodos que son subestaciones que conectan la central con la ciudad.

Los números sobre los arcos representan distancias en kilómetros. Se trata de llevar a cabo la interconexión con el menor consumo de cable, lo cual se traduce a un menor costo del material utilizado.

Una vez que se han capturado los datos como lo muestra la Figura 2.115, le damos la instrucción de resolver y aparecerá en pantalla una ventana como la siguiente (Figura 2.115):

12-04-2008	From	To	Distance/Cost	Cumulative Distance/Cost
1	CENTRAL	Estación 3	3	3
2	Estación 3	Estación 5	3	6
3	Estación 5	CIUDAD	2	8
	From CENTRAL	To CIUDAD	=	8
	From CENTRAL	To Estación 2	=	4
	From CENTRAL	To Estación 3	=	3
	From CENTRAL	To Estación 4	=	7
	From CENTRAL	To Estación 5	=	6

Figura 2.115

En la figura 2.31 se muestra la solución al problema, el programa nos sugiere ir de la central a la estación 3 y de ahí a la estación 5, para finalmente dirigirse a la ciudad a un costo de 8 kilómetros de cable. En la segunda parte de la tabla se muestran los costos acumulados de ir de la central a cada uno de los nodos.



II-2.5 PROBLEMA DE FLUJO MÁXIMO

Muchos problemas pueden ser modelados mediante una red en la cual se considera que los arcos tienen la capacidad de limitar la cantidad de un producto que se puede enviar a través del arco. En estas situaciones, frecuentemente se desea transportar la máxima cantidad de flujo desde un punto de partida llamado fuente hacia un punto final denominado destino. Obsérvese que este modelo tiene aplicación en la planificación de transporte vehicular donde la capacidad de flujo puede ser limitada por la cantidad de personas o productos que se pueden llevar en un camión, avión, automóvil, etc.; transporte de líquidos mediante tuberías y otros problemas de estructura similar.

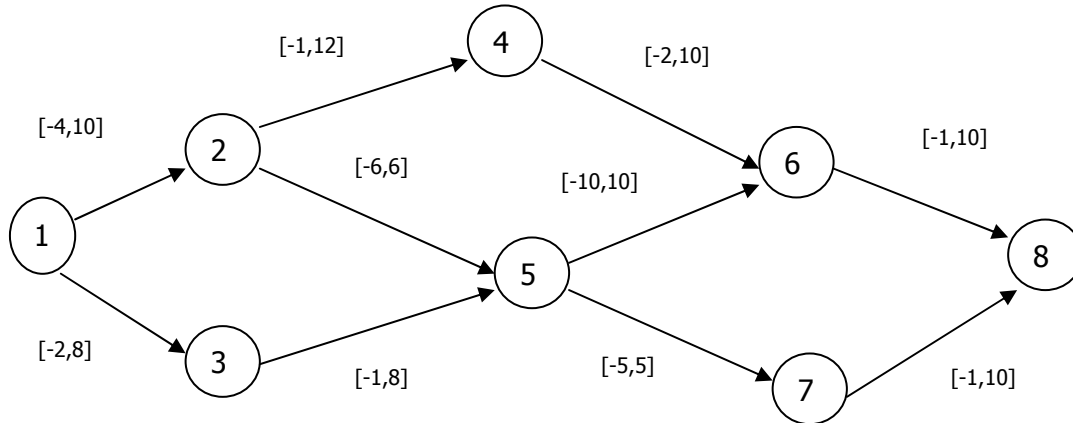
El siguiente ejemplo ha sido tomado de Colmenero (2009) e ilustra la aplicación de esta plantilla.

South Side Oil and Gas, una nueva empresa de Texas, ha desarrollado una red de oleoductos para transportar petróleo de los campos de exploración a la refinería y a otros lugares. Existen 10 oleoductos (ramales) en la red. El flujo de petróleo en cientos de galones y los datos de la red de oleoductos se presentan en la siguiente tabla. ¿Cuál es el máximo flujo que puede ir a través de la red?

Ramal	Nodo inicial	Nodo final	Capacidad (cientos de galones)	Capacidad en sentido inverso (cientos de galones)
R1	1	2	10	4
R2	1	3	8	2
R3	2	4	12	1
R4	2	5	6	6
R5	3	5	8	1
R6	4	6	10	2
R7	5	6	10	10
R8	5	7	5	5
R9	6	8	10	1
R10	7	8	10	1



La red de flujo que representa la problemática es como sigue:



Donde el par $[a, b]$ asociado a cada arco indica la capacidad mínima y máxima a cada arco. Si el valor mínimo es negativo, indica el flujo inverso que puede circular a través del arco.

Para iniciar la aplicación se debe dar clic en Problema de flujo máximo (Maximal Flow Problem). Tal como se muestra en la Figura 2.116.

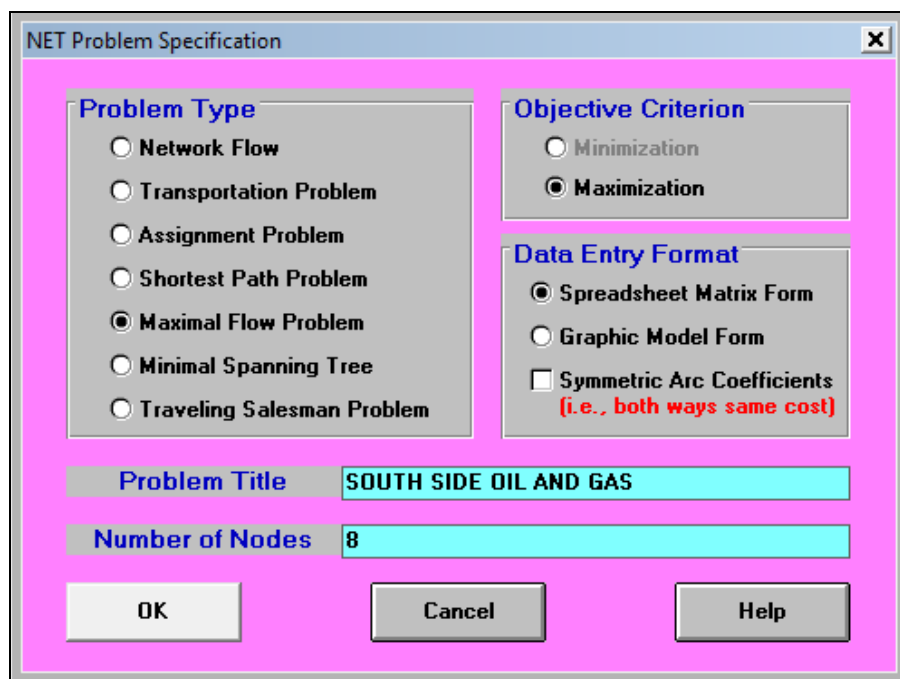


Figura 2.116



La entrada de los datos se hace como se muestra en la figura 2.117

From \ To	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8
Node1		10	8					
Node2	4			12	6			
Node3	2				8			
Node4		1				10		
Node5		6	1			10	5	
Node6				2	10			10
Node7					5			10
Node8						10	10	

Figura 2.117

Al seleccionar resolver el problema, aparece la ventana de la figura 2.118

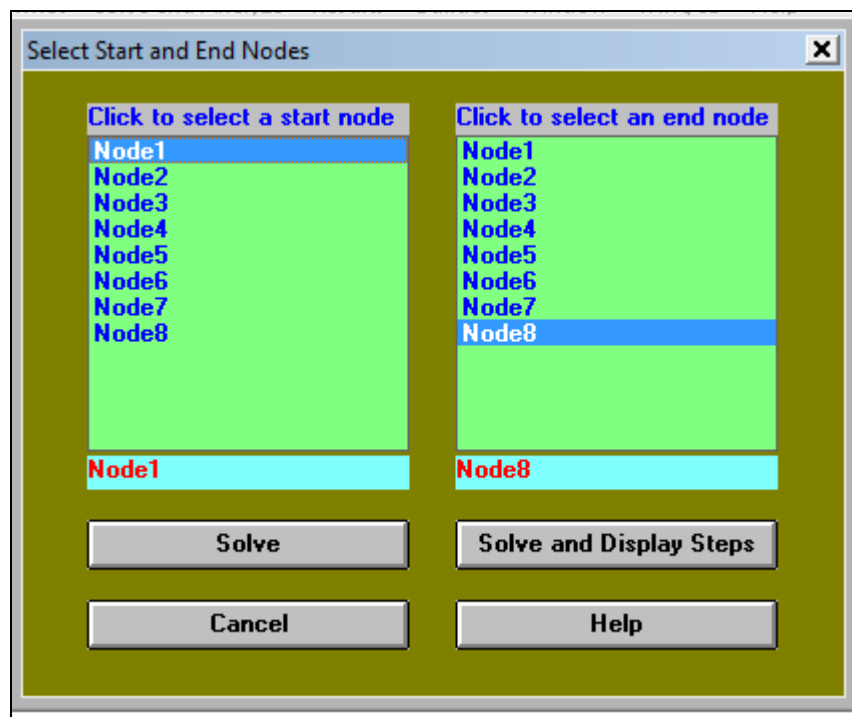


Figura 2.118

En la cual debemos señalar cuál es el nodo inicial y cuál es el nodo final. Posteriormente damos clic en Resolver. Aparece la figura 2.119 que nos señala el flujo máximo que pasa por la red de oleoductos de la compañía. El flujo máximo que puede ir a través de la red es 15 ó 1, 500 galones.



11-12-2009	From	To	Net Flow		From	To	Net Flow
1	Node1	Node2	10	5	Node4	Node6	10
2	Node1	Node3	5	6	Node5	Node7	5
3	Node2	Node4	10	7	Node6	Node8	10
4	Node3	Node5	5	8	Node7	Node8	5
Total	Net Flow	From	Node1	To	Node8	=	15

Figura 2.119

La red correspondiente a este caso se muestra en la figura 2.120

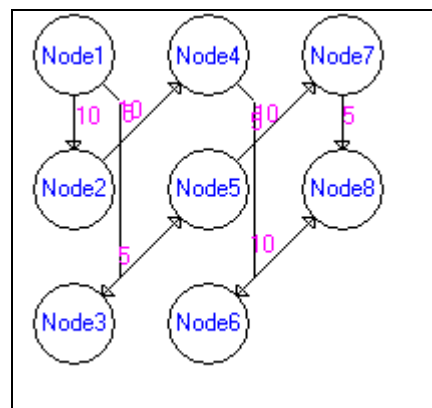


Figura 2.120



II-2.6 MÍNIMO ÁRBOL DE EXPANSIÓN

Es un problema clásico de optimización combinatoria y es una variación del problema de la ruta más corta, formulado en 1926 por Boruvka quien lo planteó para resolver el problema de hallar la forma más económica de distribuir energía eléctrica en el sur de Moravia. La formulación de este problema ha sido útil para la realización de muchas investigaciones en varios campos como el transporte, electrónica, telecomunicaciones e investigación de operaciones.

El modelo contempla un conjunto de arcos que conectan todos los nodos de la red sin crear un solo ciclo o vuelta. El problema consiste en determinar el árbol que minimiza la suma de las longitudes de las ramas que conectan todos los nodos. En cuanto a la introducción de datos y el proceso de solución, es similar a los modelos anteriores de este módulo. Para ilustrar la utilidad de esta aplicación seleccionamos el modelo Árbol de mínima expansión (Minimal Spanning Tree), mostraremos dos ejemplos el primero de ellos tomado de Colmenero (2009) y el segundo de Quesada (2008).

Una compañía constructora está desarrollando un lujoso proyecto residencial campestre en la Delegación Tlalpan. El dueño de la compañía, debe determinar la forma más barata de suministrar los servicios (agua, teléfono, luz, pavimentación) a cada residencia. La red de casas se muestra en la figura 2.121

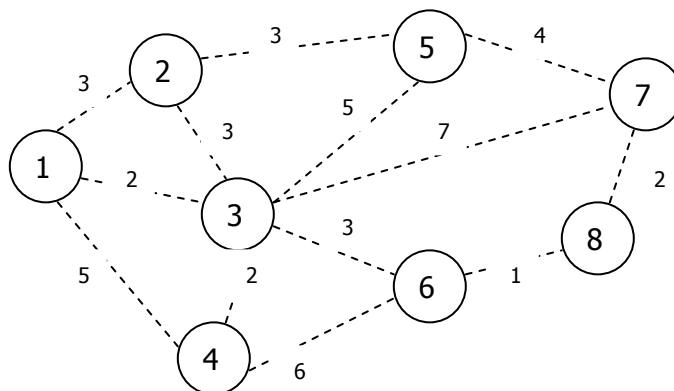


Figura 2.121



Como se ve en la figura 2.121 hay ocho casas. La distancia entre cada una de ellas en decenas de metros se muestra en la red. La distancia entre las casas uno y dos, por ejemplo, es de 30 metros. Utilice la técnica del árbol de mínima expansión para determinar la mínima suma de todas las longitudes que puede ser utilizada para conectar todos los nodos. Los datos deben ingresarse como se muestra a continuación. Figura 2.122.

Figura 2.122

Pulsamos OK. Y obtenemos la siguiente ventana. Figura 2.123.

From \ To	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8
Node1								
Node2								
Node3								
Node4								
Node5								
Node6								
Node7								
Node8								

Figura 2.123

Los datos se ingresan como se muestra a continuación. Figura 2.124.



From \ To	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8
Node1		3	2	5				
Node2	3		3		3			
Node3	2	3		2	5	3	7	
Node4	5		2			6		
Node5		3	5				4	
Node6			3	6				1
Node7			7		4			2
Node8						1	2	

Figura 2.124

Al ir al menú y pulsar resolver como se muestra en la figura 2.125.

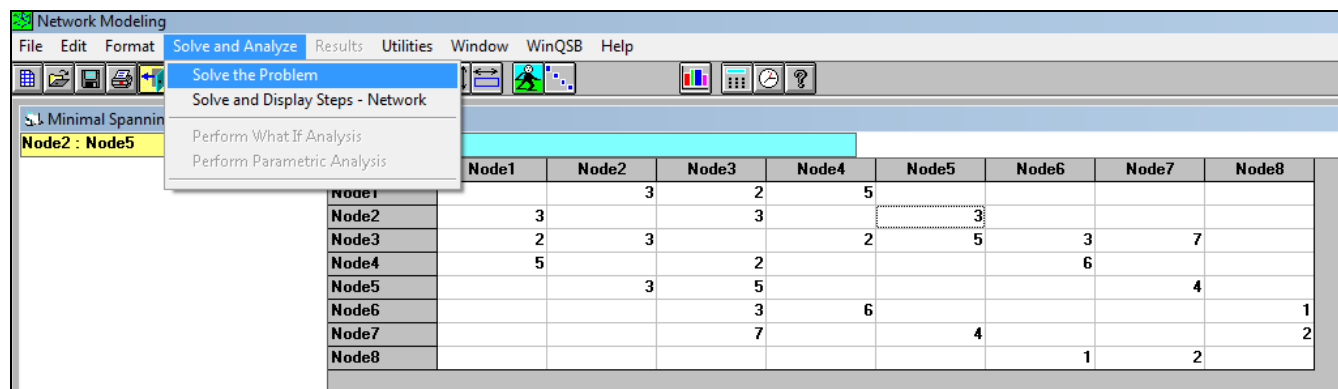


Figura 2.125

Obtenemos la siguiente solución al problema. Figura 2.126.

Solution for Minimal Spanning Tree Problem UNIDAD HABITACIONAL								
11-12-2009	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost	
1	Node1	Node2	3	5	Node3	Node6	3	
2	Node1	Node3	2	6	Node8	Node7	2	
3	Node3	Node4	2	7	Node6	Node8	1	
4	Node2	Node5	3					
	Total	Minimal	Connected	Distance	or Cost	=		16

Figura 2.126

Y el árbol de expansión mínima se muestra al seleccionar la opción ver en modo gráfico del menú resultados. Figura 2.127.

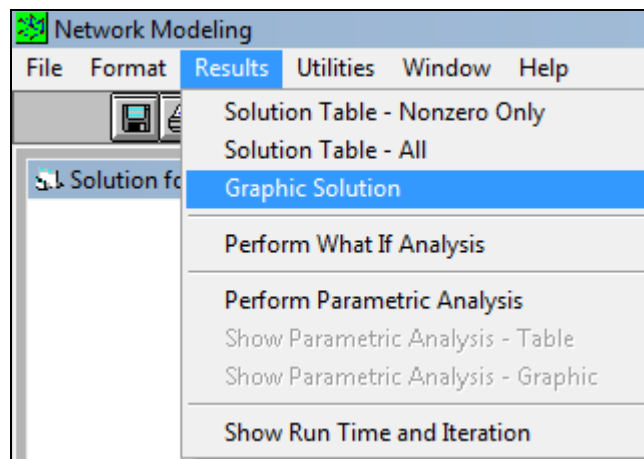


Figura 2.127

La red del árbol de expansión mínima se muestra en la figura 2.128.

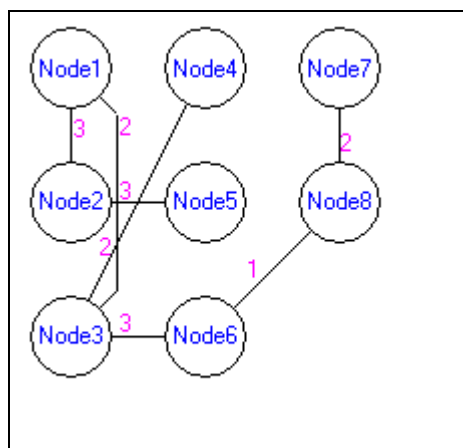


Figura 2.128

A continuación se muestra un segundo ejemplo. En la figura 2.129 se puede observar cómo es que se han introducido los datos. Los números registrados representan las distancias entre cada uno de los nodos; es decir, del nodo 1 al nodo 2 se tiene una distancia de 1 y las celdas en las que no se tiene ningún valor significa que no se puede ir de un nodo a otro por que posiblemente no haya camino o el costo sea muy alto, un ejemplo de esto último, es intentar ir del nodo 1 al nodo 8, cuyo valor es nulo.



From \ To	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8	Node9	Node10
Node1		1	2	3			4			
Node2	1		1			3				4
Node3	2	1			3					
Node4	3				2		7			
Node5			3	2						
Node6		3						1		1
Node7	4			7					5	
Node8						1			3	4
Node9							5	3		
Node10	3	4				1		4		

Figura 2.129

La solución para la plantilla anterior es la que se muestra en la figura 2.130.

12-05-2008	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Node1	Node2	1	6	Node1	Node7	4
2	Node2	Node3	1	7	Node6	Node8	1
3	Node1	Node4	3	8	Node9	Node8	3
4	Node4	Node5	2	9	Node10	Node1	3
5	Node10	Node6	1				
	Total	Minimal	Connected	Distance	or Cost	=	19

Figura 2.130

Si damos la instrucción de mostrar la solución de modo gráfico desde el menú de resultados (results). Obtenemos una ventana en la que aparece una red como la mostrada en la figura 2.37, en la que podemos observar el mínimo árbol de expansión determinado por el programa y que presenta las mismas conexiones que las que nos muestra la tabla de resultados de la figura 2.137.

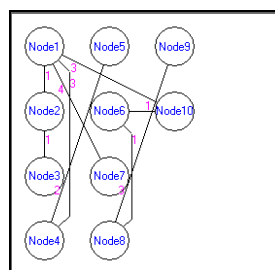


Figura 2.137



II-2.7 EL PROBLEMA DEL AGENTE VIAJERO

El problema del agente viajero, involucra un conjunto de nodos y arcos que se conectan entre sí y forman una red. El objetivo es encontrar la forma de realizar una gira completa que conecte todos los nodos visitando sólo una vez cada nodo y minimizar o maximizar la distancia de la gira total.

El problema del viajante es un ejemplo que muestra y analiza la problemática que subyace tras algunos tipos de problemas matemáticos que a priori parecen tener una solución relativamente fácil, y en la práctica presentan un gran problema. La respuesta al problema es conocida, es decir se conoce la forma de resolverlo, pero sólo en teoría, en la práctica la solución no es aplicable debido al tiempo que computacionalmente se precisa para obtener su resultado. El problema del viajante (también conocido como problema del viajante de comercio o por sus siglas en inglés: TSP) es uno de los problemas más famosos (y quizás el mejor estudiado) en el campo de la optimización combinatoria computacional. A pesar de la aparente sencillez de su planteamiento, el TSP es uno de los más complejos de resolver y existen demostraciones que equiparan la complejidad de su solución a la de otros problemas aparentemente mucho más complejos que han retado a los matemáticos desde hace siglos.

Desde el punto de vista práctico, el problema no está resuelto y desde el punto de vista teórico, las técnicas empleadas son sólo aproximaciones. No suponen una resolución real del TSP y sólo ofrecen soluciones aproximadas suficientemente aceptables.

Los algoritmos clásicos no son capaces de resolver el problema general, debido a la explosión combinatoria de las posibles soluciones. Hay algoritmos que se basan en una configuración concreta del problema. Por ejemplo, algunos algoritmos de ramificación y consolidación se pueden utilizar para resolver problemas de entre 40 a 60 ciudades. Otros han mejorado a éstos con técnicas reminiscentes de la programación lineal que permiten resolver el TSP para valores de N entre 120 y 200 ciudades. En el año 2001 se utilizó una red de 110 ordenadores para resolver el TSP para las 15.112 poblaciones de Alemania y utilizando el equivalente computacional a 22,5 años de un PC. En mayo del 2004 se aplicaron algunas de estas técnicas para la resolución del problema aplicado a las 24.978 poblaciones suecas en un ciclo de unos 72.500 km (probándose además que no se podía encontrar un ciclo más corto).



Los algoritmos genéticos basados en heurísticas³ no encuentran soluciones exactas, pero permiten encontrar aproximaciones suficientemente buenas (un 97% de optimización) y se pueden aplicar a conjuntos de ciudades muy grandes (redes con millones de nodos) con tiempos de ejecución razonables en un superordenador (semanas o meses).

Para ilustrar la utilidad de la aplicación; consideremos la figura 2.138 en la que cada uno de los nodos representa un lugar en el sur de la ciudad y los valores en los arcos son las distancias que las separan (los datos no son reales).

From \ To	Copilco	Universidad	San Ángel	Miguel Ángel	Xochimilco	Tlalpan
Copilco		100	150		300	500
Universidad	100		160	150	300	
San Ángel	150	160		100	260	290
Miguel Ángel		150	100		240	360
Xochimilco	300	300	260	240		200
Tlalpan	500		290	360	200	

Figura 2.138

Al pulsar en resolver nos aparece el menú mostrado en la figura 2.139.

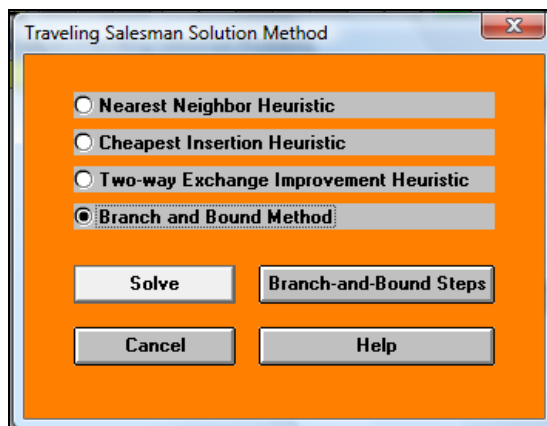


Figura 2.139

³ Se denomina **heurística** a la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos, desde cuyo punto de vista puede describirse como *el arte y la ciencia del descubrimiento y de la invención* o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente.



Elegimos la última opción y pulsamos resolver (solve). La solución al problema aparece en la figura 2.140.

12-05-2008	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Copilco	Universidad	100	4	Xochimilco	Tlalpan	200
2	Universidad	Miguel Ángel	150	5	Tlalpan	San Ángel	290
3	Miguel Ángel	Xochimilco	240	6	San Ángel	Copilco	150
	Total	Minimal	Traveling	Distance	or Cost	=	1130
	(Result	from	Branch	and	Bound	Method)	

Figura 2.140

La solución gráfica al problema se muestra en la figura 2.141.

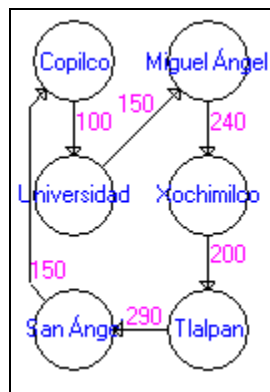


Figura 2.141

Que es la minimiza el recorrido entre los distintos lugares del sur de la ciudad.



II-3 PERT – CPM

Es necesario ingresar al módulo PERT y CPM (PERT_CPM) para acceder a esta aplicación. El método de la ruta crítica, CPM, es una herramienta de tipo determinístico para el análisis de redes de proyectos. La opción Nuevo Problema (New Problem) genera una plantilla en el cual se introducirá las características de nuestro problema. Figura 2.142.

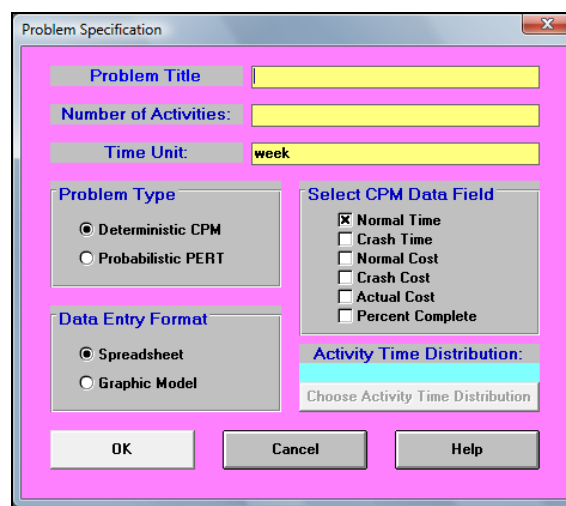


Figura 2.142

A continuación se describirán cada una de las casillas de esta ventana:

- Título del problema (Problem Title): Se escribe el título con que identificamos el problema.
- Número de actividades (Number of Activities): Se escribe la cantidad de actividades (nodos) presentes en la red del proyecto.
- Unidad de tiempo (Time Unit): En este campo se especifica la unidad de tiempo trabajada en la red (Ejemplo: hora, día, mes, año...).



- Tipo de problema (Problem Type): Los problemas representados por redes de proyectos pueden ser analizados mediante dos métodos: CPM Determinístico (Deterministic CPM) y PERT Probabilístico (Probabilistic PERT).

- Formato de entrada de datos (Data Entry Format): Permite elegir entre dos plantillas distintas para introducir los datos del modelo al programa. La primera alternativa se asemeja a una hoja de cálculo, mientras que la segunda, permite diseñar las redes en modo gráfico.

- Campos de datos seleccionados para el CPM (Select CPM Data Field): Esta área que aparece cuando pulsamos en la opción CPM Determinístico (Deterministic CPM) permitiendo seleccionar las variables de análisis que desarrollará WINQSB para el estudio de este tipo de redes:
 - Tiempo normal (Normal Time): En este campo se especifica el tiempo normal de cada actividad.
 - Tiempo de quiebre (Crash Time): Tiempo mínimo al cual se podría reducir una actividad.
 - Costo normal (Normal Cost): Costo de realizar una actividad ejecutada en un tiempo normal (este costo es presupuestado).
 - Costo de quiebre (Crash Cost): Costo incurrido al realizar una actividad en su tiempo de quiebre o crítico.
 - Costo actual (Actual Cost): Costo de una actividad real.
 - Porcentaje completo (Percent Complete): Permite realizar un análisis de costos y tiempos de forma parcial (o la totalidad) a un proyecto que ha sido ejecutado.

- Distribución del tiempo de cada actividad (Activity Time Distribution): Esta opción se activa cuando se pulsa sobre la opción PERT Probabilístico (Probabilistic PERT). El método PERT (Program Evaluation and Review Technique) trabaja bajo incertidumbre, donde los tiempos de la actividad tienen posibilidad



de variar de acuerdo a una distribución probabilística. Al pulsar sobre el botón Escoger distribución del tiempo de cada actividad (Choose Activity Time Distribution), se desplegará una nueva ventana con diferentes distribuciones probabilísticas (Figura 2.143):

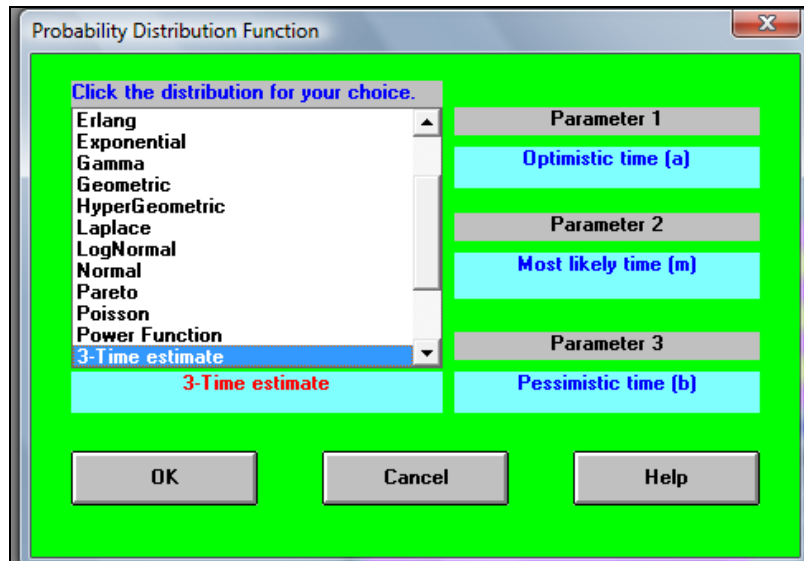


Figura 2.143

Para escoger una distribución, simplemente seleccionamos la más adecuada a nuestro modelo y oprimiremos el botón OK. El usuario debe tener el criterio para saber elegir entre cada una de las distribuciones; en caso contrario se recomienda que consulte algunos de los siguientes libros en los cuales puede encontrar una descripción detallada de las Distribuciones de Probabilidad: Walpole, R y Myers (1992). Y Hines, W y Montgomery, D. (1993).



II-3.1 CPM

Mediante el siguiente ejemplo tomado de Quesada (2008) se mostrará cómo puede crearse un problema de tipo CPM para un proyecto.

La empresa ICAMEX S.A. programó las siguientes actividades para la construcción de una avenida en concreto asfáltico (proyecto resumido – tiempo dado en días). El problema ha sido inventado para ilustrar la aplicación:

No	Actividad	Predecesores inmediatos	Tiempo normal	Tiempo de quiebre	Costo Normal (\$)	Costo de Quiebre
1	Excavación	-	15	10	1000	1200
2	Sub-Base	1	7	6	3000	3500
3	Compactación	2	2	2	700	700
4	Base	3	4	2	1200	2400
5	Compactación	4	1	1	700	700
6	Canaletes	3	6	3	1500	2700
7	Capa asfalto	5, 10	1	1	1100	1100
8	Capa asfalto	6, 7	3	2	4700	5200
9	Compactación	8	1	1	800	800
10	Pruebas Base	5	2	1	400	1100
11	Pruebas ASf.	9	2	1	900	1300

Una vez analizado el enunciado se sigue con la creación del modelo de redes. Procedemos a llenar la ventana Especificaciones del problema (Problem Specification) con los datos del ejercicio. Figura 2.144



Figura 2.144

Marcamos todas las opciones disponibles para CPM (excepto los dos últimos) con el fin de realizar un análisis integral. La ventana siguiente permite ingresar la información disponible de cada actividad (Figura 2.145):

Activity Number	Activity Name	Immediate Predecessor (list)	Normal Time	Crash Time	Normal Cost	Crash Cost
1	Excavación		15	10	\$1000	\$1200
2	Sub-base	1	7	6	\$3000	\$3500
3	Compactación	2	2	2	\$700	\$700
4	Base	3	4	2	\$1200	\$2400
5	Compactación	4	1	1	\$700	\$700
6	Canaletes	3	6	3	\$1500	\$2700
7	Capa asfalto	5,10	1	1	\$1100	\$1100
8	Capa asfalto	7,6	3	2	\$4700	\$5200
9	Compactación	8	1	1	\$800	\$800
10	Pruebas base	5	2	1	\$400	\$1100
11	Pruebas asf.	9	2	1	\$900	\$1300

Figura 2.145

Los puntos que aparecen en esta zona son:

- Número de la actividad (Activity Number): Número consecutivo de actividades.



- Nombre de la actividad (Activity Name): WINQSB predefine los nombres de las actividades con letras (se cambiaron a los nombres dados por el ejercicio).
- Predecesores (Immediate Predecessor): Se especifica el predecesor de cada actividad. Puede ser por el nombre de la actividad o por el número de la misma. En el caso de que no exista predecesor se debe dejar el espacio en blanco.
- Tiempos normales y de quiebre (Normal Time – Crash Time): Tiempos normales y mínimos estimados por actividad.
- Costos normales y de quiebre (Normal Cost – Crash Cost): Costos normales y de quiebre para cada actividad.

WINQSB permite visualizar el problema en modo gráfico; para acceder a él es necesario dirigirnos al menú formato seleccionamos la opción mostrar en modo gráfico (switch to Graphics model). Debe aparecer una ventana con la red en modo gráfico. Figura 2.146.

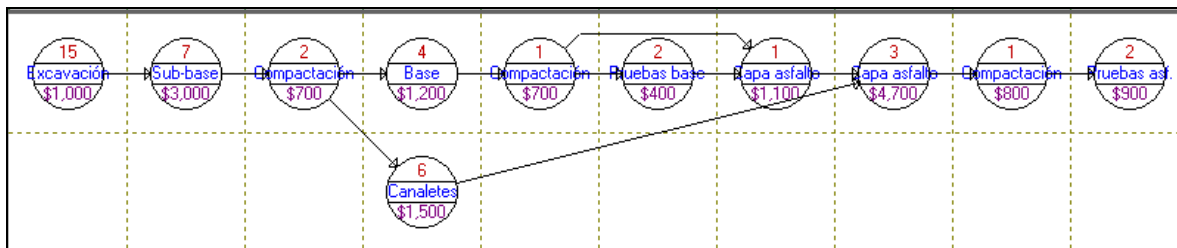


Figura 2.146

WINQSB resuelve las redes de proyectos (ruta críticas) desde el enfoque CPM empleando los tiempos normales o los tiempos de quiebre. En el menú Resolver y analizar (Solve and Analyze) pulsamos sobre Resolver ruta critica usando tiempos normales (Solve Critical Path Using Normal Time). Figura 2.147

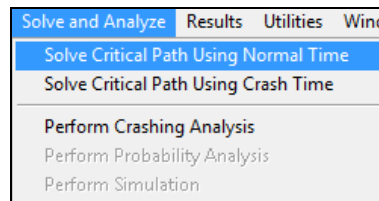


Figura 2.147

La nueva ventana muestra cuales son las actividades críticas de la red. En la columna Sobre la ruta crítica (On Critical Path) se puede observar las actividades críticas (marcadas con Yes). Figura 2.148

12-08-2008 01:35:04	Activity Name	On Critical Path	Activity Time	Earliest Start	Earliest Finish	Latest Start	Latest Finish	Slack (LS-ES)
1	Excavación	Yes	15	0	15	0	15	0
2	Sub-base	Yes	7	15	22	15	22	0
3	Compactación	Yes	2	22	24	22	24	0
4	Base	Yes	4	24	28	24	28	0
5	Compactación	Yes	1	28	29	28	29	0
6	Canaletes	no	6	24	30	26	32	2
7	Capa asfalto	Yes	1	31	32	31	32	0
8	Capa asfalto	Yes	3	32	35	32	35	0
9	Compactación	Yes	1	35	36	35	36	0
10	Pruebas base	Yes	2	29	31	29	31	0
11	Pruebas asf.	Yes	2	36	38	36	38	0
	Project	Completion	Time	=	38	DÍAs		
	Total	Cost of	Project	=	\$16,000	(Cost on CP =	\$14,500)	
	Number of	Critical	Path(s)	=	2			

Figura 2.148

Aparecen los tiempos más próximos de inicio y finalización (Earliest Start y Earliest Finish), junto a los tiempos tardíos (Latest Start y Latest Finish). En la última columna tenemos los tiempos de holgura (Slack). En las tres últimas filas aparecen el tiempo de duración total del proyecto (Project Completion Time), el costo total del proyecto (Total Cost of Project) y el número de rutas críticas (Number of Critical Path), En el ejemplo la duración total es de 38 días, el costo de \$ 16.000 (el costo de la ruta crítica es de \$14.500) y existen dos rutas críticas.



Podemos también observar la ruta crítica en modo gráfico, pulsando sobre el menú Resultados (Results) y en análisis de la actividad gráfica (Graphic Activity Analysis). Figura 2.149

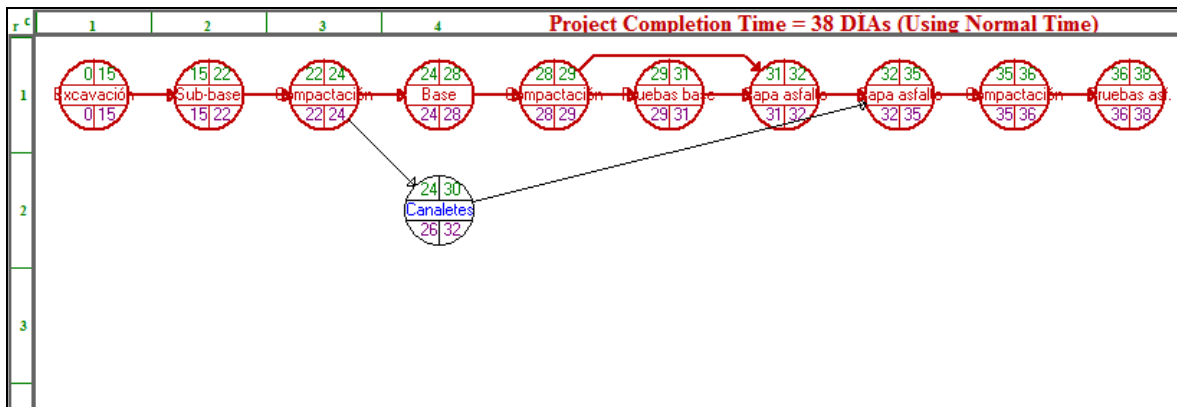


Figura 2.149

Las actividades (nodos) que se encuentran resaltadas forman parte de la ruta crítica. Figura 2.149

Al pulsar sobre el menú Resultados (Results) y en Mostrar ruta crítica (Show Critical Path) aparecerán solo las actividades pertenecientes a la ruta crítica. Figura 2.150

12-08-2008	Critical Path 1	Critical Path 2
1	Excavación	Excavación
2	Sub-base	Sub-base
3	Compactación	Compactación
4	Base	Base
5	Compactación	Compactación
6	Capa asfalto	Pruebas base
7	Capa asfalto	Capa asfalto
8	Compactación	Capa asfalto
9	Pruebas asf.	Compactación
10		Pruebas asf.
Completion Time	38	38

Figura 2.150



Mediante la opción Análisis del estado del proyecto (Project Completion Analysis) ubicado en el menú Resultados (Results) podemos analizar las actividades que debieron ser ejecutadas (o que se encuentran en marcha) una vez pasado cierto periodo de tiempo. Figura 2.51

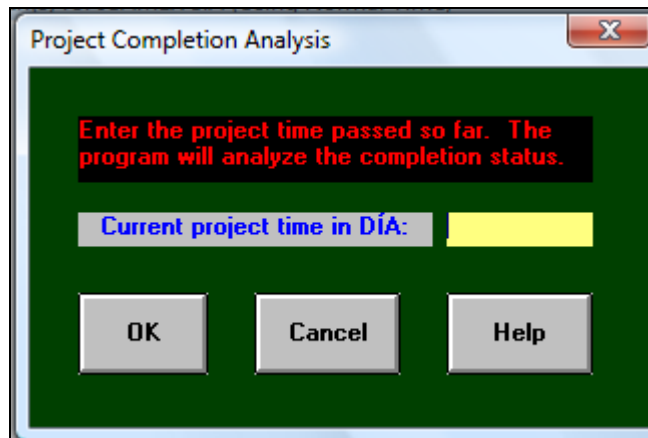


Figura 2.151

Este campo va a permitir introducir el día en el que queremos ver el estado del proyecto; en este mismo menú se encuentra la opción para mostrar la diagrama de Gantt tal y como lo muestra la figura 2.152.

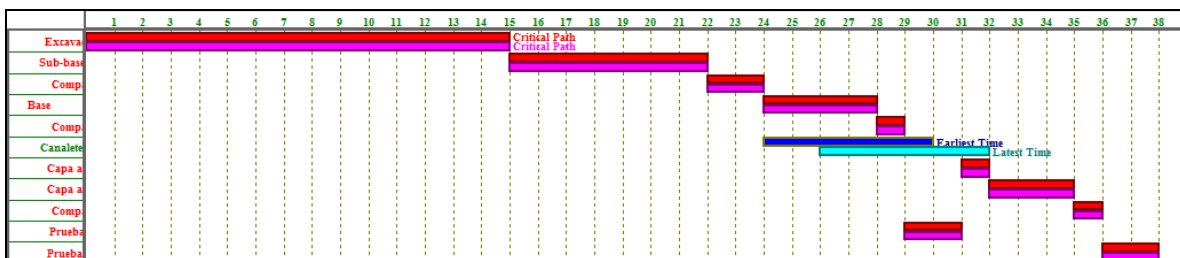


Figura 2.152



II-3.2 PERT

Para mostrar el funcionamiento de esta opción en el WINQSB modificaremos el ejemplo inicial para trabajar con tiempos normales, optimistas y pesimistas para cada actividad (3 time estimate).

No	Actividad	Predecesores inmediatos	Tiempo Optimista	Tiempo Normal	Tiempo pesimista
1	Excavación	-	10	15	17
2	Sub-Base	1	6	7	8
3	Compactación	2	2	2	3
4	Base	3	2	4	5
5	Compactación	4	1	1	2
6	Canaletes	3	3	6	7
7	Capa asfalto	5, 10	1	1	2
8	Capa asfalto	6, 7	2	3	4
9	Compactación	8	1	1	2
10	Pruebas Base	5	1	2	3
11	Pruebas ASf.	9	1	2	3

Indicamos el uso de esta distribución en la ventana Especificación del Problema (Problem Specification). E introducimos los datos como se muestra en la figura 2.153.



Activity Number	Activity Name	Immediate Predecessor (list number/name, separated by ',')	Optimistic time (a)	Most likely time (m)	Pessimistic time (b)
1	XCAVACION		10	15	17
2	SUB-BASE	1	6	7	8
3	IPACTACION	2	2	2	3
4	BASE	3	2	4	5
5	IPACTACION	4	1	1	2
6	CANALES	3	3	6	7
7	PEGANTE	5,10	1	1	2
8	CAPA	6,7	2	3	4
9	IPACTACION	8	1	1	2
10	PRUEBAS	5	1	2	3
11	PRUEBAS	9	1	2	3

Figura 2.153

Los puntos que aparecen en esta zona son:

- Número de la actividad (Activity Number): Número consecutivo de actividades.
- Nombre de la actividad (Activity Name): WINQSB predefine los nombres de las actividades con letras (se cambiaron a los nombres dados por el ejercicio).
- Predecesores (Immediate Predecessor): Se especifica el predecesor de cada actividad. Puede ser por el nombre de la actividad o por el número de la misma. En el caso de que no exista predecesor se debe dejar el espacio en blanco.
- Tiempos optimistas, normales y pesimistas (Optimistic Time - a, Most Likely Time y Pessimistic Time - b): Tiempos normales, pesimistas y optimistas.

En el menú Resolver y analizar (Solve and Analyze) pulsamos sobre Resolver ruta crítica (Solve Critical Path). Figura 2.154



03-25-2009 16:07:25	Activity Name	On Critical Path	Activity Mean Time	Earliest Start	Earliest Finish	Latest Start	Latest Finish	Slack (LS-ES)	Activity Time Distribution	Standard Deviation
1	EXCAVACION	Yes	14.5	0	14.5	0	14.5	0	3-Time estimate	1.1667
2	SUB-BASE	Yes	7	14.5	21.5	14.5	21.5	0	3-Time estimate	0.3333
3	COMPACTACION	Yes	2.1667	21.5	23.6667	21.5	23.6667	0	3-Time estimate	0.1667
4	BASE	Yes	3.8333	23.6667	27.5	23.6667	27.5	0	3-Time estimate	0.5
5	COMPACTACION	Yes	1.1667	27.5	28.6667	27.5	28.6667	0	3-Time estimate	0.1667
6	CANALES	no	5.6667	23.6667	29.3333	26.1667	31.8333	2.5	3-Time estimate	0.6667
7	PEGANTE	Yes	1.1667	30.6667	31.8333	30.6667	31.8333	0	3-Time estimate	0.1667
8	CAPA ASFALTO	Yes	3	31.8333	34.8333	31.8333	34.8333	0	3-Time estimate	0.3333
9	COMPACTACION	Yes	1.1667	34.8333	36	34.8333	36	0	3-Time estimate	0.1667
10	PRUEBAS BASE	Yes	2	28.6667	30.6667	28.6667	30.6667	0	3-Time estimate	0.3333
11	PRUEBAS ASF	Yes	2	36	38	36	38	0	3-Time estimate	0.3333
	Project	Completion	Time	=	38	DIAs				
	Number of	Critical	Path(s)	=	2					

Figura 2.154

La cuarta columna (Activity Mean Time) muestra la duración promedio de cada Actividad. Siguen los tiempos más próximos de inicio y finalización (Earliest Start y Earliest Finish), junto a los tiempos tardíos (Latest Start y Latest Finish) y los tiempos de holgura (Slack).

La última columna representa la desviación estándar para cada actividad.

Dada la información del ejemplo 2, ¿cuál es la probabilidad de concluir el proyecto en 35 días?

Al pulsar sobre el menú Resultados (Results) y en Análisis Probabilístico (Performance Probability Analysis), se podrá determinar la probabilidad de cumplimiento en una red de proyectos. Para nuestro ejemplo, simplemente escribiremos 35 en la casilla Tiempo deseado de ejecución (Desired Completion Time in Día) y luego presionando el botón Compute Probability. Figura 2.155

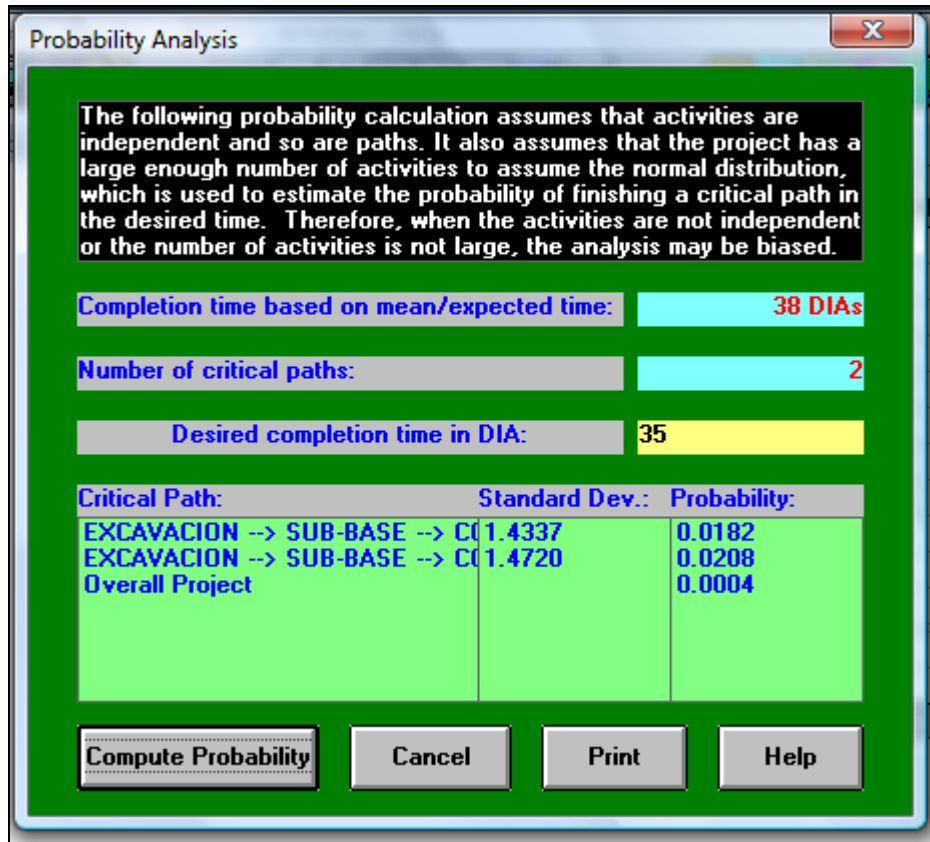


Figura 2.155

La probabilidad se calcula para las dos rutas críticas presentes en el proyecto: 1.8209% y 2.0779%. Existe entonces una probabilidad del 0.04% de terminar el proyecto en 35 días.



CAPÍTULO III- INVESTIGACIÓN DE OPERACIONES II

Lo primero que debe valorar un Hombre es

A sí mismo...

III-1 PROCESO DE MARKOV

Algunas veces nos interesa saber cómo cambia una variable a través del tiempo; por ejemplo, desearíamos saber cómo evoluciona el precio de las acciones de una empresa a través del tiempo. Las cadenas de Markov se han aplicado en áreas tales como la educación, mercadotecnia, servicios de salud, finanzas, contabilidad y producción. La mayor parte de los ejemplos de este módulo están basados en Quesada *et al.*, (2008), Murray (2007) y Bañuelos (2006).

Este módulo de WinQSB sólo resuelve matrices que sean no absorbentes.⁴ Para comenzar la utilización de este módulo nos dirigimos a Proceso de Markov (Markov process). La opción Nuevo Problema (New Problem) genera una plantilla llamada Especificaciones del problema PMK (MKP Problem Specification) en la cual, se introducirán las características de nuestro problema como lo muestra la figura 3.1.

⁴ El lector interesado en este concepto puede revisar la bibliografía de referencia. No es la finalidad de este trabajo el explicar los conceptos teóricos básicos.

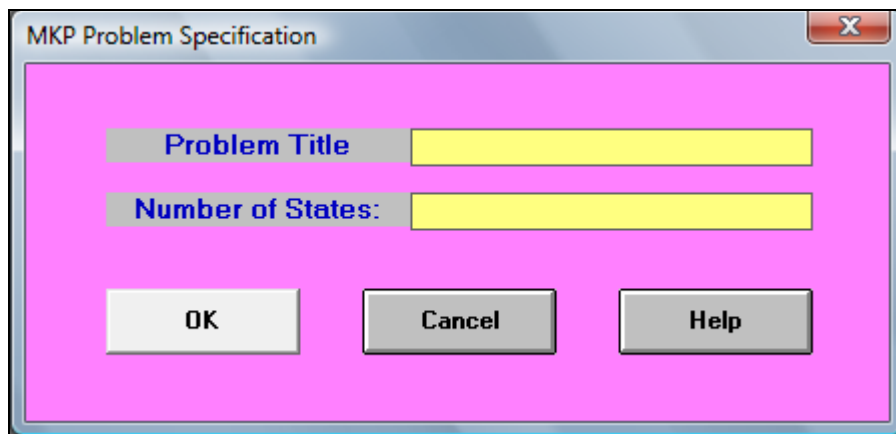


Figura 3.1

Para hacer uso de las herramientas potenciales de este módulo es necesario ingresar los siguientes campos:

- Título del problema (Problem Title)
- Número de estados (Number of States)

Analicemos un problema tomado de las notas de clase de BAÑUELOS (2006)⁵ para ejemplificar el uso del software.

Los analistas de mercado de Hamburguesas Real han realizado estudios de la lealtad de los clientes a su marca, comparándola contra su principal competidor Mac Donaldo. Los resultados de 1000 encuestas aplicadas a los consumidores de hamburguesas revelan que en un mes típico se retienen el 85% de sus clientes, perdiendo un 15% que le compran a Mac Donaldo. Por otro lado Mac Donaldo retiene el 90% de sus clientes y el 10% cambia a Hamburguesa Real. Considerando que se van a abrir locales de ambas Hamburguesas de manera que al iniciar el primer mes cada marca tiene el 50% de los clientes. Determinar la Distribución del mercado un mes después.

⁵ Profesor de carrera de la División de Ciencias Básicas de la Facultad de Ingeniería de la UNAM.



Para resolver el caso anterior iniciamos abriendo la aplicación como se ha señalado.

Ingresamos el nombre del problema y el número de estados. En este caso son 2 los estados de transición. Figura 3.2

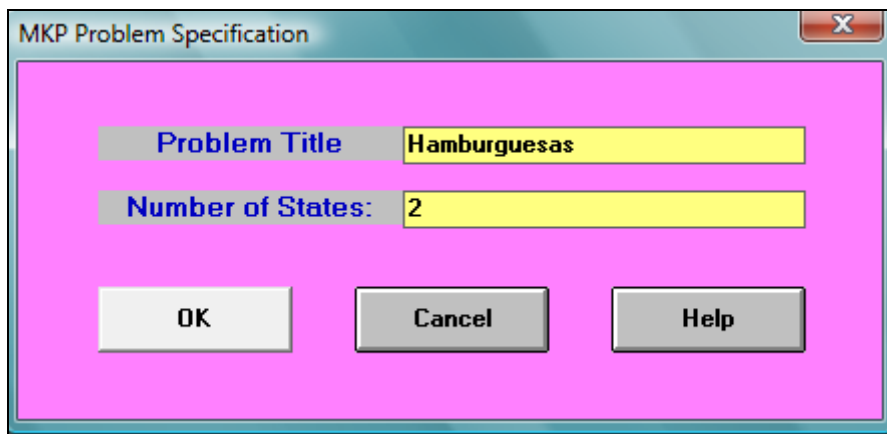


Figura 3.2

Pulsamos "ok". Posteriormente accedemos al menú editar (edit) seleccionamos la opción nombres de los estados (state names). Esto nos va a permitir cambiar los nombres de cada uno de los estados de nuestra matriz. La figura 3.3 muestra como se realiza.

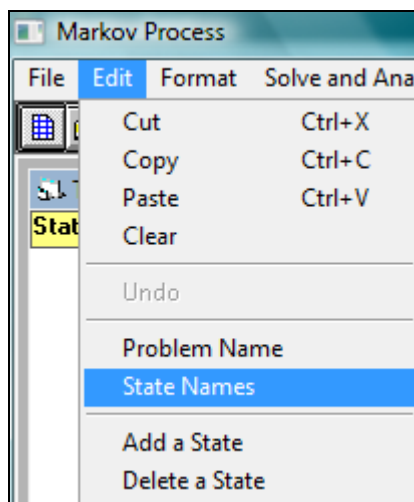


Figura 3.3



Insertamos el nuevo nombre de los estados como se muestra en la figura 3.4.

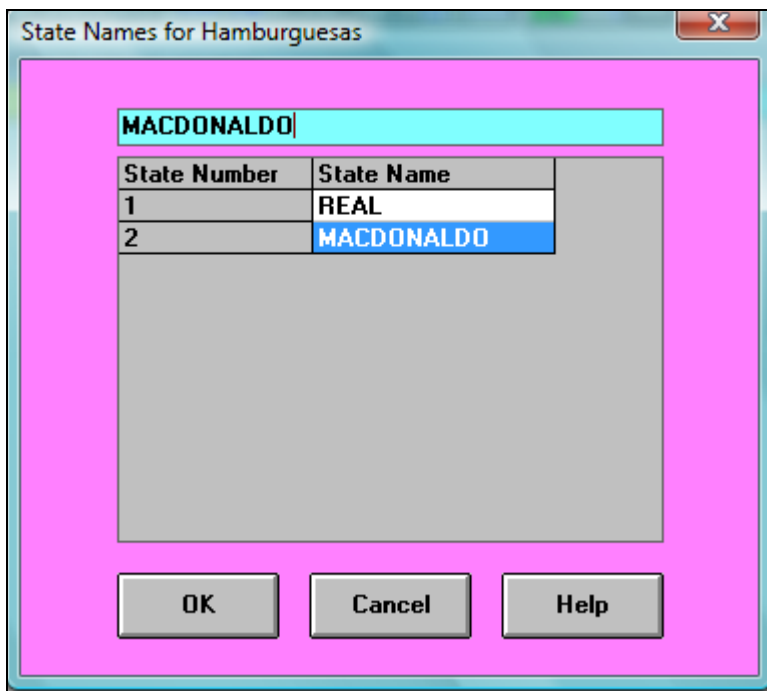


Figura 3.4

Al introducir los datos proporcionados por el problema tenemos lo siguiente. Figura 3.5

From \ To	REAL	MACDONALDO
REAL	.85	.15
MACDONALDO	.1	.9
Initial Prob.	.5	.5
State Cost		

Figura 3.5

La plantilla representa una matriz con las relaciones entre los estados (State), sus probabilidades iniciales (Initial Prob.) y el costo de cada uno de ellos (State Cost). En este caso no se nos ha proporcionado ningún costo, por lo que este campo se deja en blanco.



En el menú Resolver y analizar (Solve and Analyze) tenemos las opciones de Resolver los estados completos (Solve Steady State) o mostrar el Proceso de Markov por pasos (Markov Process Step).

Si seleccionamos la opción Proceso de Markov por pasos se despliega la siguiente ventana. Figura 3.6.



Figura 3.6

Podemos observar el Número de periodos procesados (The Number of Time Periods from Initial). Es en esta casilla donde vamos a capturar el número: 1 y pulsamos "ok". Se registró el número uno dado que lo solicitado por el caso de estudio es determinar la Distribución del mercado un mes después. Al hacerlo obtenemos la siguiente ventana. Figura 3.7

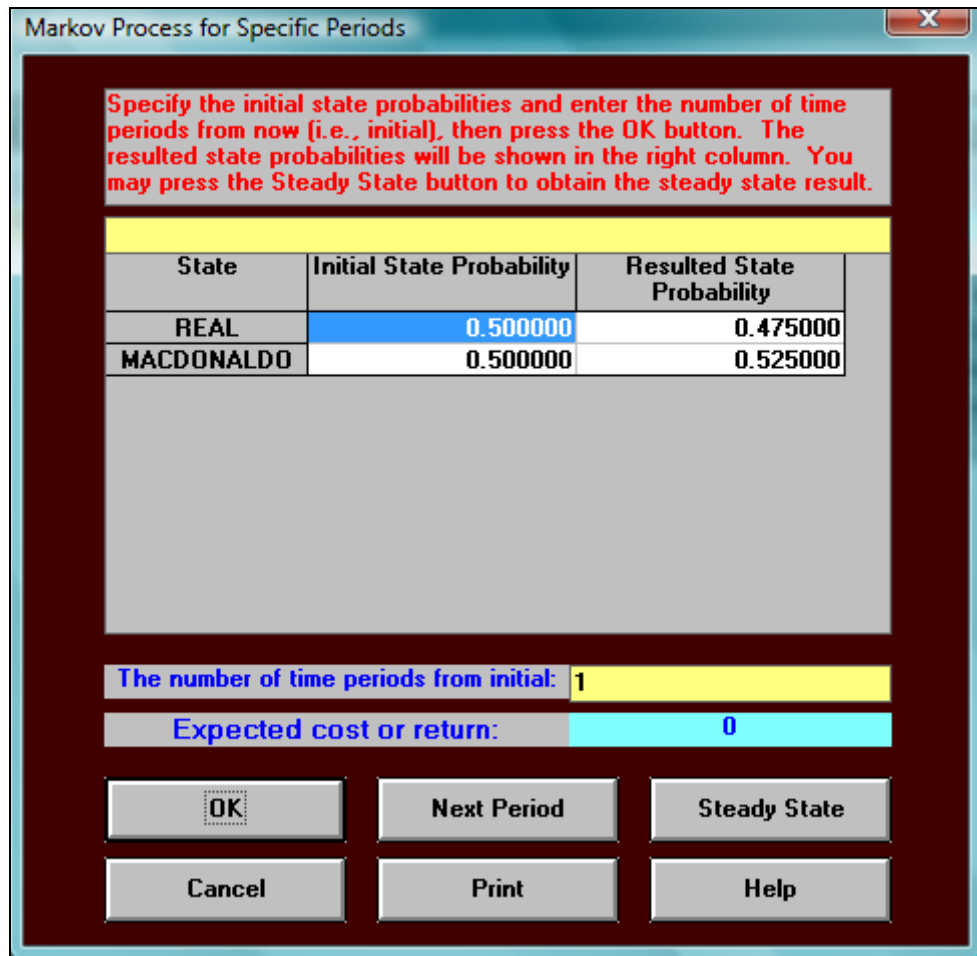


Figura 3.7

Donde la distribución del mercado un mes después es 47.5% para Hamburguesa Real y 52.5% para Mac Donaldo. Si lo que nos interesa es conocer la distribución del mercado 2, 3, 4, 5... n periodos después; debemos introducir el número del periodo que nos interesa en el mismo renglón.

Por otro lado si lo que nos interesa es conocer es la distribución del mercado en el estado estable tenemos dos opciones:

1. Introducir un número muy grande como 1000 en el renglón de número de periodos procesados u oprimir el botón Estado estable (Stady state) de la Figura 3.7.



- En el menú Resolver y analizar (Solve and Analyze) seleccionar la opción Resolver el estado estable. (Solve Steady State)

En las siguientes dos figuras (3.8, 3.9) se muestra que no hay diferencia entre ambas alternativas:

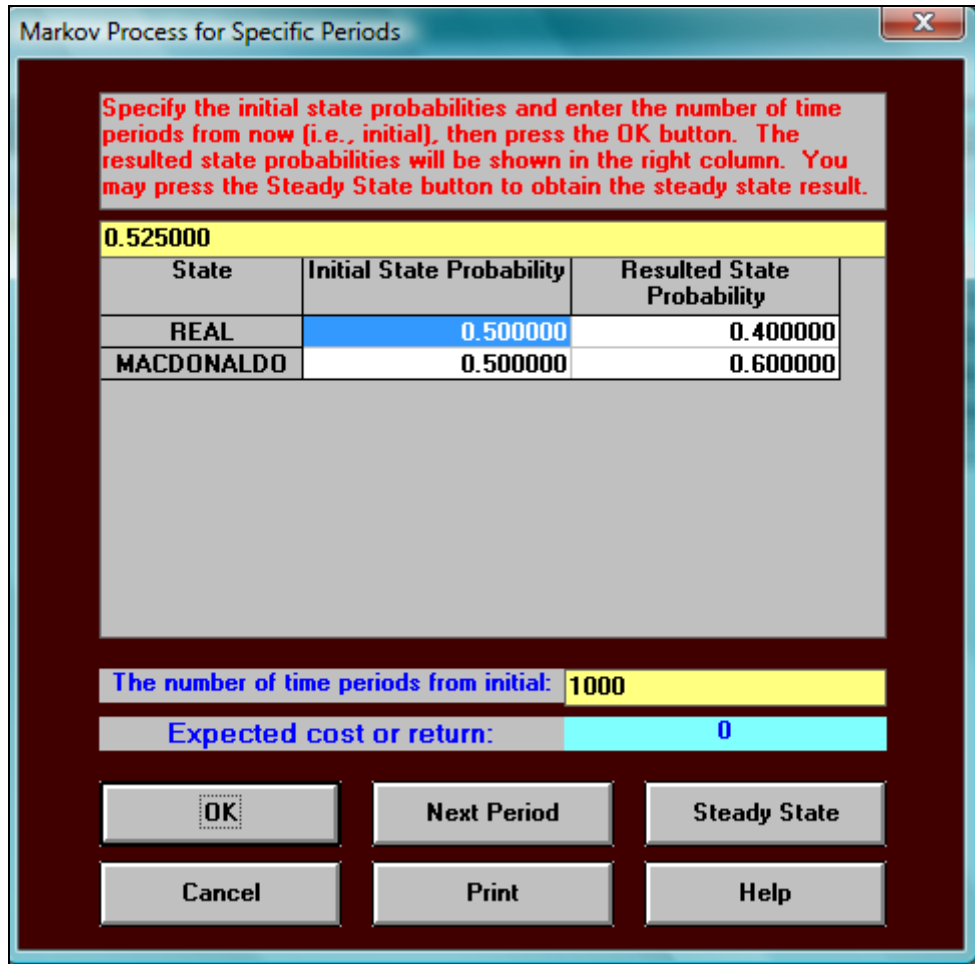


Figura 3.8

06-02-2009	State Name	State Probability	Recurrence Time
1	REAL	0.4000	2.5000
2	MACDONALDO	0.6000	1.6667
	Expected	Cost/Return =	0

Figura 3.9



En el estado estable no importa ya donde empezaste a la larga tiendes a los mismos estados en probabilidades fijas. A continuación se muestra un segundo ejemplo tomado de Quesada *et al.*, (2008).

Ingreseemos un sistema representado por 4 estados. Figura 3.10

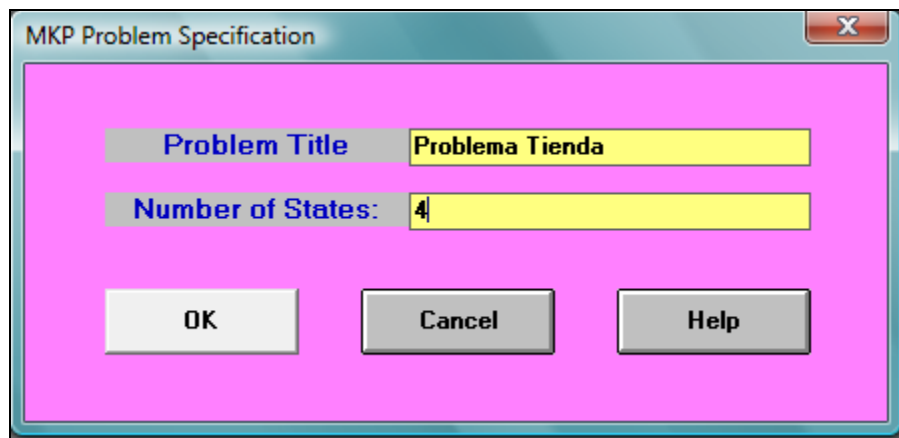


Figura 3.10

La plantilla vacía representa una matriz con las relaciones entre los estados (State), sus probabilidades iniciales (Initial Prob.) y el costo de cada uno de ellos (State Cost). Figura 3.11.

From \ To	State1	State2	State3	State4
State1				
State2				
State3				
State4				
Initial Prob.				
State Cost				

Figura 3.11

En esta tabla que representa nuestra matriz de transición vamos a registrar las probabilidades de nuestro modelo y los costos asociados a cada estado. La tabla llena se muestra a continuación. Figura 3.12.



From \ To	State1	State2	State3	State4
State1	.2	.3	.1	.4
State2	.25	.35	.4	0
State3	.1	.2	.2	.5
State4	.5	.3	.1	.1
Initial Prob.	.2	.1	.35	.35
State Cost	2000	1500	1000	900

Figura 3.12

En el menú Resolver y analizar (Solve and Analyze) tenemos las opciones de Resolver el estado estable (Solve Steady State) o Mostrar el Proceso de Markov por pasos (Markov Process Step). La primera opción da como resultado la siguiente tabla. Figura 3.13

03-25-2009	State Name	State Probability	Recurrence Time
1	State1	0.2638	3.7908
2	State2	0.2938	3.4038
3	State3	0.2090	4.7838
4	State4	0.2334	4.2849
	Expected	Cost/Return =	1387.3530

Figura 3.13

La matriz final indica las probabilidades de estado estable, lo cual significa que en el largo plazo el sistema estará el 26% del tiempo en el estado uno, 29% estará en el estado dos, 21% estará en estado tres y 23% en estado cuatro, lo cual hace que el costo medio en que incurre el proceso es de 1387.3530.

Es posible resolver este ejemplo paso a paso regresando a la matriz inicial y tomando la segunda opción del menú Resolver y analizar (Solve and Analyze) tenemos una ventana que nos permite controlar las iteraciones del proceso. Figura 3.14

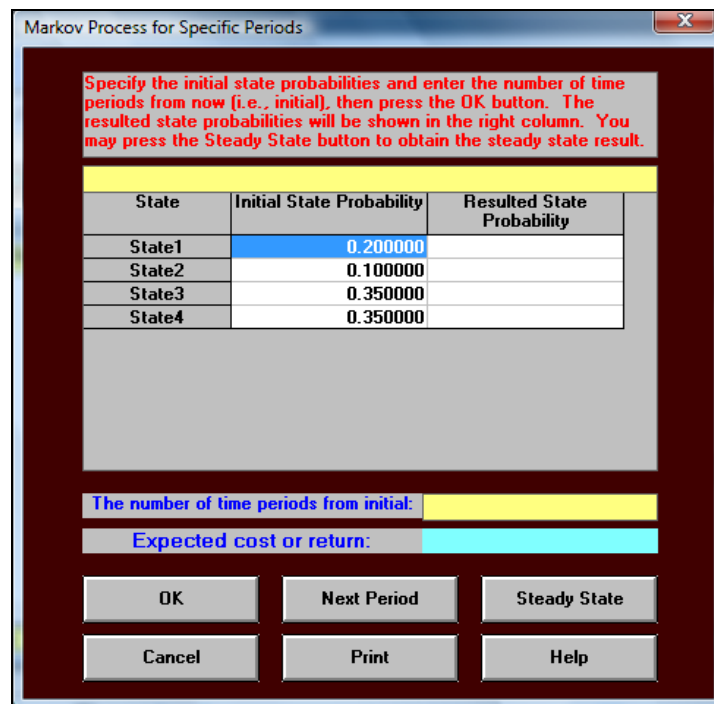


Figura 3.14

Podemos observar el Número de periodos procesados (The Number of Time Periods from Initial). Pulsemos en el botón NEXT PERIOD y luego en el botón "OK". Figura 3.15

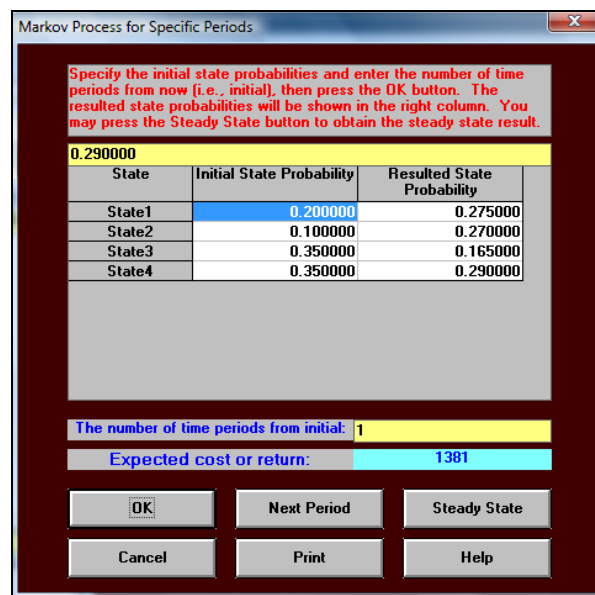


Figura 3.15



Para el periodo dos (recordemos pulsar en NEXT PERIOD seguido del botón "OK") Figura 3.16

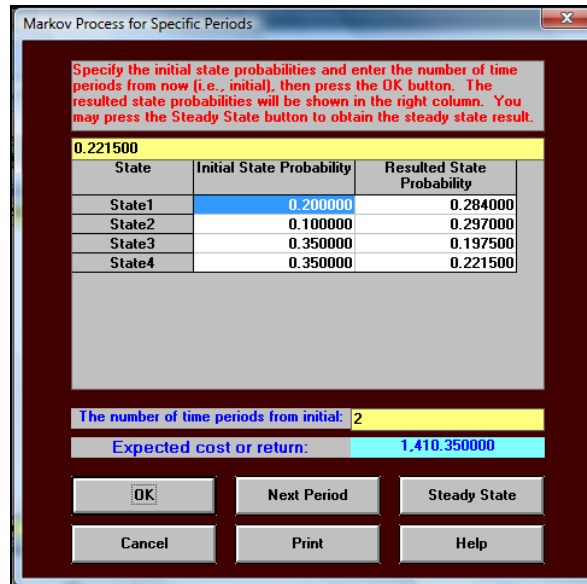


Figura 3.16

En la columna Probabilidad del estado resultante (Resulted State Probability) se muestran las probabilidades para los periodos. Pulsando en el botón STEADY STATE alcanzamos la matriz estable. Figura 3.17.

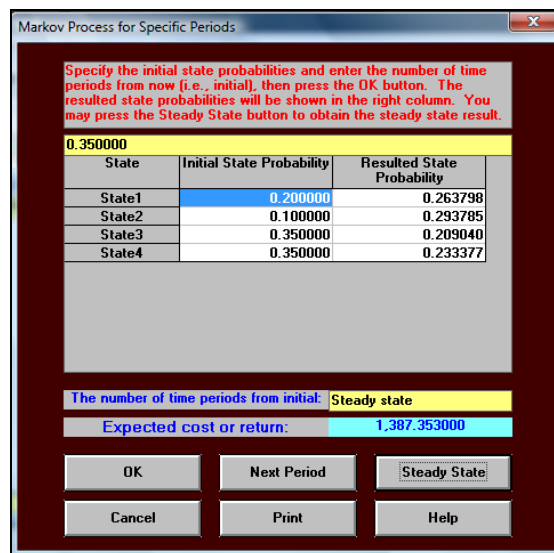


Figura 3.17



Para ver un Análisis paramétrico en el tiempo de los costos y las probabilidades de los estados seleccionamos la opción Resolver y análisis paramétrico en el tiempo:

La nueva ventana contiene. Figura 3.18

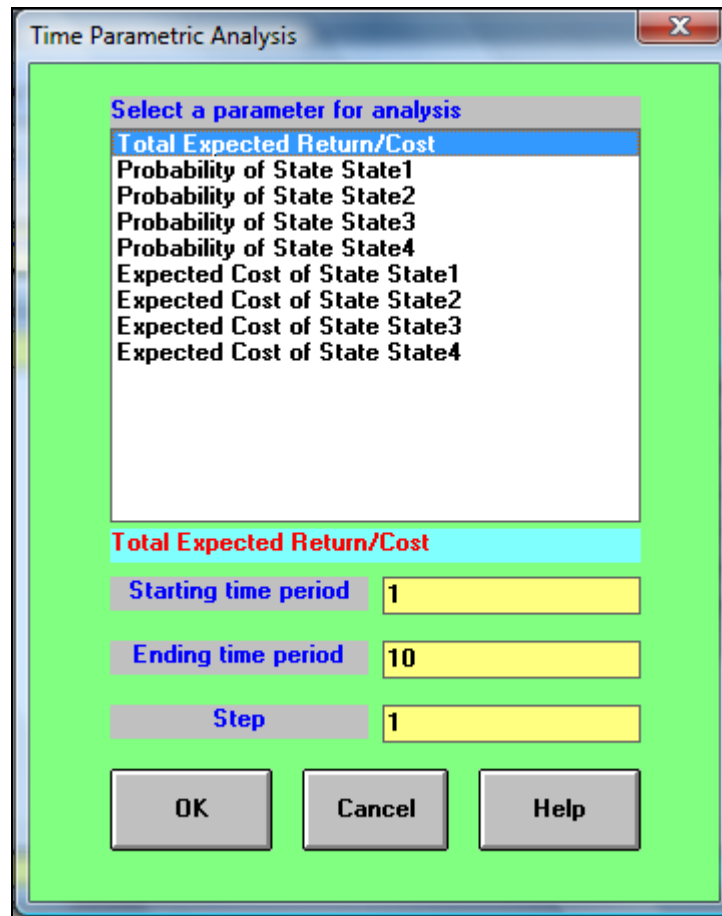


Figura 3.18

- Retorno/Costo total esperado (Total Expected Return/Cost)
- Probabilidad de cada estado (Probability of State State#)
- Costo esperado de cada estado (Expected Cost of State State#)



Pulsemos el botón "OK" para mostrar el Retorno/Costo total esperado (Total Expected Return/Cost) para 10 periodos (1 por periodo – Step = 1). Figura 3.19

03-25-2009	Time Period	Total Expected Return/Cost
1	1	1381
2	2	1410.3500
3	3	1385.6500
4	4	1387.9060
5	5	1387.0120
6	6	1387.4280
7	7	1387.3330
8	8	1387.3610
9	9	1387.3510
10	10	1387.3540

Figura 3.19

Se puede observar como el costo comienza a estabilizarse para los últimos periodos (recuerde que el costo final es de 1987.3530).



III-2 TEORÍA DE COLAS

Todos nosotros hemos esperado mucho tiempo en líneas o colas. En este capítulo se darán algunos ejemplos de cómo *WINQSB* puede ayudar al análisis, solución y simulación de este tipo de modelos; es importante aclarar que las bases teóricas deben haberse adquirido previamente en algún curso de investigación de operaciones. La mayor parte de los ejemplos de este módulo y el siguiente, están basados en Quesada *et al.*, (2008) y Murray (2007).

“El problema de las líneas de espera o de colas se presenta en muchas situaciones de la vida moderna. Ejemplos los tenemos en los servicios gubernamentales, servicios comerciales al pagar en un supermercado, al comprar boletos de cine u otras funciones, al solicitar servicios bancarios, al pagar el teléfono o la luz, esperando en un restaurante que se desocupe una mesa; de transporte como los aviones que tienen que dar vueltas mientras se desocupa una pista de aterrizaje o al despegar mientras se les autoriza el uso de una pista, automóviles esperando que los contrate un pasajero en el aeropuerto o terminal de autobuses, automóviles esperando el siga en un semáforo o para pagar en una caseta de peaje, o esperando en la gasolinera o en el taller para ser atendidos; industriales como piezas defectuosas esperando reparación en el taller de una fábrica, piezas terminadas esperando ser revisadas y empacadas para envío a los distribuidores; servicios profesionales como los pacientes esperando ser atendidos por un médico, clientes esperando ser atendidos por un notario, personas esperando pasar al confesionario en una iglesia; etc.” Murray (2007).

Existen diversos y muy variados modelos de líneas de espera. La notación que usaremos en esta sección se usa para representar un sistema de colas en el que las llegadas esperan en una sola cola, hasta que uno de los servidores idénticos en paralelo queda libre. Entonces el primer cliente de la cola entra al servicio y así sucesivamente.

Un primer paso consiste, como en todos los modelos, en la especificación del problema mediante la cual se establecerá si el modelo a tratar es un *M/M/S* (Simple *M/M* System) o un modelo general (General Queuing System). Es pertinente aclarar que en el caso de que se elija la opción de un



modelo simple en la que la distribución de los tiempos (llegada y servicio) siguen un proceso markoviano debe elegirse esa opción; en caso contrario; es decir, en el que la distribución entre llegadas y tiempos de servicio siga alguna otra distribución como puede ser triangular, uniforme, Erlang, etcétera; se debe elegir la opción de Modelo general de colas. Cada uno de los dos modelos permite incluir a más de un servidor y adoptar las condiciones necesarias para simplificar el análisis, y adecuarlo para que refleje de una manera confiable la realidad del sistema de estudio.

Al abrir el módulo referente a estos modelos y seleccionar la creación de un nuevo problema aparecerá una ventana como la de la figura 3.20

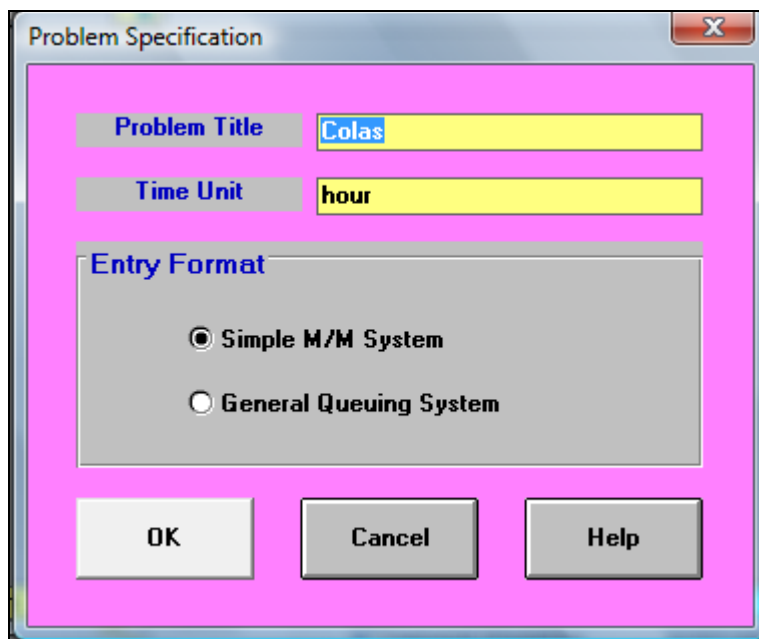


Figura 3.20

Vamos a suponer por ahora un modelo $M/M/S$. Lo que se debe hacer a continuación es el ingreso de los datos de acuerdo con las especificaciones de la ventana. Figura 3.21



Data Description	ENTRY
Number of servers	
Service rate (per server per hour)	
Customer arrival rate (per hour)	
Queue capacity (maximum waiting space)	M
Customer population	M
Busy server cost per hour	
Idle server cost per hour	
Customer waiting cost per hour	
Customer being served cost per hour	
Cost of customer being balked	
Unit queue capacity cost	

Figura 3.21

La ventana anterior consta de los siguientes campos que se describen a continuación:

- Numero de servidores en paralelo (Number of Servers)
- Tasa de servicio (Service Rate)
- Tasa de llegada de clientes (Customer Arrival Rate)
- Capacidad de la cola (Queue Capacity)
- Tamaño de la población de clientes (Customer Population)
- Costo del servidor ocupado por hora (Busy Server Cost per Hour)
- Costo del servidor desocupado por hora (Idle Server Cost per Hour)
- Costo de espera de los clientes por hora (Customer Waiting Cost per Hour)
- Costo de los clientes siendo servidos por hora (Customer Being Served Cost per Hour)



- Costo de los clientes siendo rehusados (Cost of Customer Being Balked)
- Costo de la unidad de capacidad de la cola (Unit Queue Capacity Cost)

Un ejemplo de un modelo de colas es el siguiente (recuerde que las letras M indican un valor infinito o muy grande). Figura 3.22

Data Description	ENTRY
Number of servers	2
Service rate (per server per hour)	15
Customer arrival rate (per hour)	20
Queue capacity (maximum waiting space)	M
Customer population	M
Busy server cost per hour	150
Idle server cost per hour	150
Customer waiting cost per hour	200
Customer being served cost per hour	200
Cost of customer being balked	150
Unit queue capacity cost	pesos

Figura 3.22

Una de las posibilidades de solución es calcular las tradicionales medidas de desempeño (medidas de efectividad), para calcularlas es necesario acceder al menú resolver y analizar (solve and analyze), posteriormente se selecciona el renglón solve the performance como se muestra en la figura 3.23.

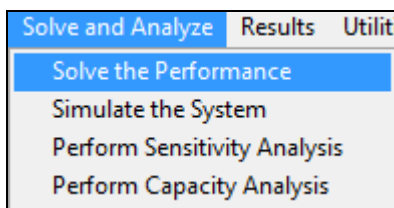



Figura 3.23



Al seleccionar la opción mencionada aparece la siguiente ventana. Figura 3.24

03-25-2009	Performance Measure	Result
1	System: M/M/2	From Formula
2	Customer arrival rate (λ) per hour =	20.0000
3	Service rate per server (μ) per hour =	15.0000
4	Overall system effective arrival rate per hour =	20.0000
5	Overall system effective service rate per hour =	20.0000
6	Overall system utilization =	66.6667 %
7	Average number of customers in the system (L) =	2.4000
8	Average number of customers in the queue (Lq) =	1.0667
9	Average number of customers in the queue for a busy system (Lb) =	2.0000
10	Average time customer spends in the system (W) =	0.1200 hours
11	Average time customer spends in the queue (Wq) =	0.0533 hours
12	Average time customer spends in the queue for a busy system (Wb) =	0.1000 hours
13	The probability that all servers are idle (Po) =	20.0000 %
14	The probability an arriving customer waits (Pw) or system is busy (Pb) =	53.3333 %
15	Average number of customers being balked per hour =	0
16	Total cost of busy server per hour =	\$200.0000
17	Total cost of idle server per hour =	\$100.0000
18	Total cost of customer waiting per hour =	\$213.3334
19	Total cost of customer being served per hour =	\$266.6667
20	Total cost of customer being balked per hour =	\$0
21	Total queue space cost per hour =	\$0

Figura 3.24

En el cual se pueden visualizar las tradicionales medidas de desempeño. Otra opción con la que se cuenta es simular el sistema, para realizar esto es necesario regresar a la pantalla anterior con el botón . Una vez que hemos vuelto; seleccionamos la opción simular el sistema (simulate the system) y aparecerá una ventana como la que se muestra en la figura 3.25.

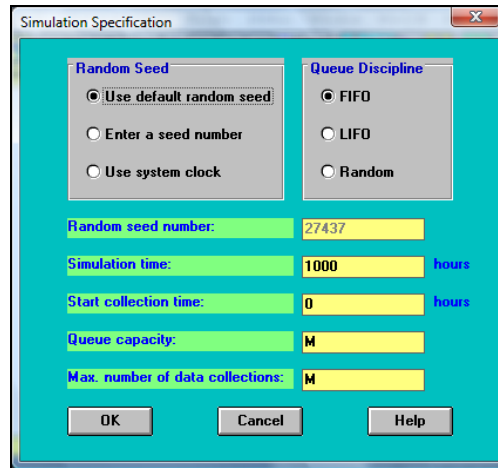


Figura 3.25

En la columna de la derecha podemos modificar la disciplina de la cola, si son Primeras Entradas Primeras Salidas, el Último en Entrar es el Primero en Salir y RANDOM que significa orden aleatorio. Algunas otras notaciones que se encuentran comúnmente en la bibliografía son las siguientes:

FCFS = El primero que llega es el primero en ser atendido (En inglés First Come First Served).

LCFS = El último en entrar es el primero en salir. (En inglés Last Come First Served)

SIRO = Servicio en orden aleatorio. (En inglés Service In Random Order.)

GD = Disciplina general de línea de espera. (En inglés General Discipline)

Usando el sistema de reloj con 1000 horas de simulación del sistema de colas. Figura 3.26

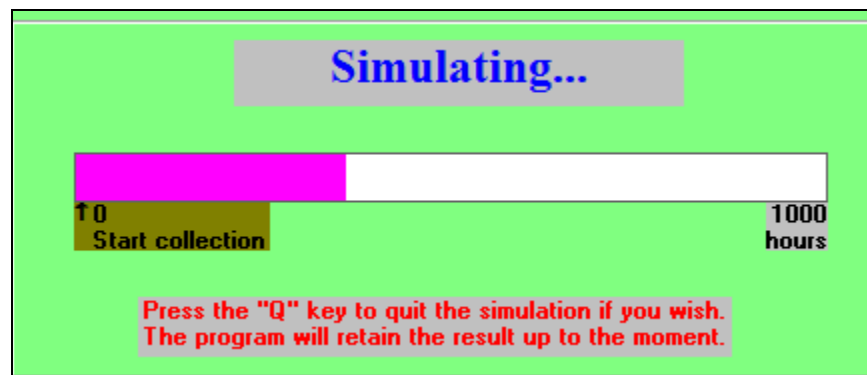


Figura 3.26



El resumen de probabilidades de encontrar n clientes en el sistema es el siguiente. Figura 3.27

03-25-2009 17:15:16 n	Estimated Probability of n Customers in the System	Cumulative Probability
0	0.1946	0.1946
1	0.2601	0.4547
2	0.1726	0.6273
3	0.1186	0.7459
4	0.0787	0.8246
5	0.0555	0.8801
6	0.0354	0.9155
7	0.0248	0.9403
8	0.0155	0.9558
9	0.0119	0.9677
10	0.0100	0.9777
11	0.0069	0.9847
12	0.0052	0.9898
13	0.0037	0.9935
14	0.0025	0.9960
15	0.0016	0.9976
16	0.0009	0.9986
17	0.0008	0.9994
18	0.0005	0.9999
19	0.0001	1.0000
20	0.0000	1.0000


Figura 3.27

En el menú resultados (results) es posible optar por la opción para visualizar las medidas de desempeño (performance measure). Como se muestra en la figura 3.28



06-02-2009	Performance Measure	Result
1	System: M/M/2	From Simulation
2	Customer arrival rate (λ) per hour =	20.0000
3	Service rate per server (μ) per hour =	15.0000
4	Overall system effective arrival rate per hour =	20.0072
5	Overall system effective service rate per hour =	20.0022
6	Overall system utilization =	66.5326 %
7	Average number of customers in the system (L) =	2.4212
8	Average number of customers in the queue (Lq) =	1.0905
9	Average number of customers in the queue for a busy system (Lb) =	2.0441
10	Average time customer spends in the system (W) =	0.1210 hours
11	Average time customer spends in the queue (Wq) =	0.0545 hours
12	Average time customer spends in the queue for a busy system (Wb) =	0.1021 hours
13	The probability that all servers are idle (Po) =	20.2845 %
14	The probability an arriving customer waits (Pw) or system is busy (Pb) =	53.3498 %
15	Average number of customers being balked per hour =	0
16	Total cost of busy server per hour =	\$199.5979
17	Total cost of idle server per hour =	\$100.4021
18	Total cost of customer waiting per hour =	\$218.0354
19	Total cost of customer being served per hour =	\$266.1789
20	Total cost of customer being balked per hour =	\$0
21	Total queue space cost per hour =	\$0
22	Total system cost per hour =	\$784.2143
23	Simulation time in hour =	1000.0000
24	Starting data collection time in hour =	0
25	Number of observations collected =	20003
26	Maximum number of customers in the queue =	18
27	Total simulation CPU time in second =	2.4770

Figura 3.28

Es posible realizar también un Análisis de sensibilidad para algunos parámetros; como por ejemplo a cambios en número de servidores, iniciando en 2 y terminando en 10. Para esto es necesario dirigirnos al menú resultados y seleccionar parámetros para análisis de sensibilidad; para realizar esto es necesario regresar a la pantalla anterior con el botón .

Aparecerá una ventana como la que se muestra a continuación. Figura 3.29.

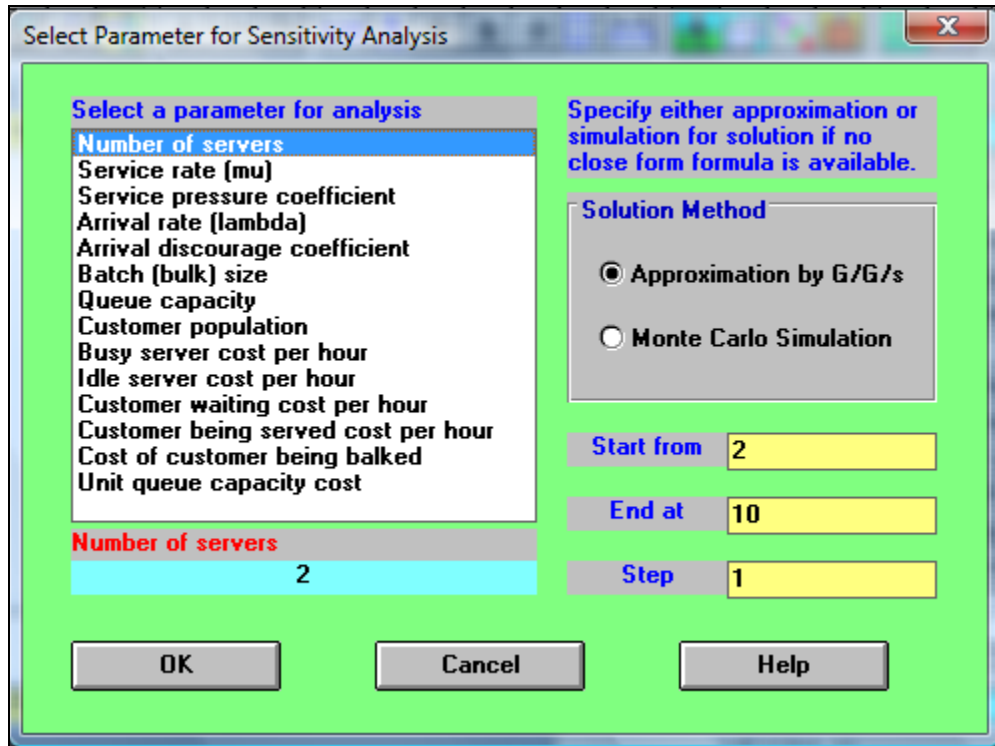


Figura 3.29

Un análisis parecido puede hacerse tomando como base la capacidad del sistema, que puede ir desde una capacidad específica de x clientes (capacidad limitada) hasta infinita. Al pulsar "Ok" Aparece la tabla de la figura 3.30.

03-25-2009 Value	Effective Arrival Rate	System Utilization	L	Lq	Lb	W	Wq	Wb	P0	Pw	Average Balked	Busy Server Cost	Idle Server Cost	Waiting Customer Cost	Served Customer Cost	Balked Customer Cost	Queue Capacity Cost	TOTAL COST
2	20.0000	0.6667	2.4000	1.0667	2.0000	0.1200	0.0533	0.1000	0.2000	0.5333	0	200.0000	100.0000	213.3334	266.6667	0	0	780.0000
3	20.0000	0.4444	1.4780	0.1446	0.8000	0.0739	0.0072	0.0400	0.2542	0.1808	0	200.0000	250.0000	28.9266	266.6667	0	0	745.5933
4	20.0000	0.3333	1.3592	0.0259	0.5000	0.0680	0.0013	0.0250	0.2621	0.0518	0	200.0000	400.0000	5.1780	266.6667	0	0	871.8447
5	20.0000	0.2667	1.3379	0.0046	0.3636	0.0669	0.0002	0.0182	0.2634	0.0126	0	200.0000	550.0000	0.9172	266.6667	0	0	1017.5840
6	20.0000	0.2222	1.3341	0.0008	0.2857	0.0667	0.0000	0.0143	0.2636	0.0026	0	200.0000	700.0000	0.1511	266.6667	0	0	1166.8180
7	20.0000	0.1905	1.3334	0.0001	0.2353	0.0667	0.0000	0.0118	0.2636	0.0005	0	200.0000	850.0000	0.0228	266.6667	0	0	1316.6900
8	20.0000	0.1667	1.3333	0.0000	0.2000	0.0667	0.0000	0.0100	0.2636	0.0001	0	200.0000	1000.0000	0.0031	266.6667	0	0	1466.6700
9	20.0000	0.1481	1.3333	0.0000	0.1739	0.0667	0.0000	0.0087	0.2636	0.0000	0	200.0000	1150.0000	0.0004	266.6667	0	0	1616.6670
10	20.0000	0.1333	1.3333	0.0000	0.1538	0.0667	0.0000	0.0077	0.2636	0.0000	0	200.0000	1300.0000	0.0000	266.6667	0	0	1766.6670

Figura 3.30

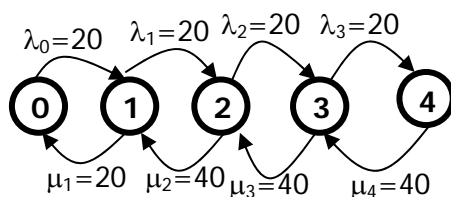


Como hemos visto WinQSB es una herramienta muy útil y práctica. Se tienen muchos otros campos que explorar sin embargo como se estableció al principio del presente trabajo sólo se exponen los principios básicos de la aplicación práctica de este programa.

A continuación se exponer otro ejemplo tomado de Murray (2007), con la finalidad de clarificar y repasar las herramientas anteriormente expuestas.

Se tiene una empresa A que por teléfono vende suscripciones a una revista. La revista hace publicidad por televisión informando a los televidentes el teléfono donde se pueden suscribir. La empresa A tiene un número telefónico con 4 líneas atendidas por dos vendedores. Cuando un suscriptor llama y uno o los dos vendedores están desocupados el suscriptor es atendido de inmediato. Si los dos están ocupados, el sistema telefónico automáticamente avisa al suscriptor que será atendido por el primer vendedor que se desocupe y los pone en una de las líneas desocupadas que tiene un dispositivo para que se atiendan en orden cronológico de llegada de la llamada.. Cuando las 4 líneas están ocupadas y marca un suscriptor, el sistema le da tono de ocupado y se pierde la llamada. Las estadísticas tomadas por la empresa A indican que las llamadas llegan aleatoriamente al conmutador con una tasa media de 20 llamadas por hora con distribución Poisson. La duración media de cada llamada es de 3 minutos distribuida exponencialmente.

Un diagrama de densidades de transición de estados poniendo valores numéricos junto a las densidades en el diagrama señalando cuáles son λ 's y cuáles μ 's.



Al calcular las tradicionales medidas de desempeño, obtenemos el siguiente la tabla mostrada en la figura 3.31.



	System: M/M/2/6	From Formula
2	Customer arrival rate (λ) per hour =	20.0000
3	Service rate per server (μ) per hour =	20.0000
4	Overall system effective arrival rate per hour =	19.7895
5	Overall system effective service rate per hour =	19.7895
6	Overall system utilization =	49.4737 %
7	Average number of customers in the system (L) =	1.2632
8	Average number of customers in the queue (Lq) =	0.2737
9	Average number of customers in the queue for a busy system (Lb) =	0.8387
10	Average time customer spends in the system (W) =	0.0638 hours
11	Average time customer spends in the queue (Wq) =	0.0138 hours
12	Average time customer spends in the queue for a busy system (Wb) =	0.0424 hours
13	The probability that all servers are idle (Po) =	33.6842 %
14	The probability an arriving customer waits (Pw) or system is busy (Pb) =	32.6316 %
15	Average number of customers being balked per hour =	0.2105

Figura 3.31

Y al simularlo durante 1000 horas como se ha mostrado anteriormente. Figura 3.32.

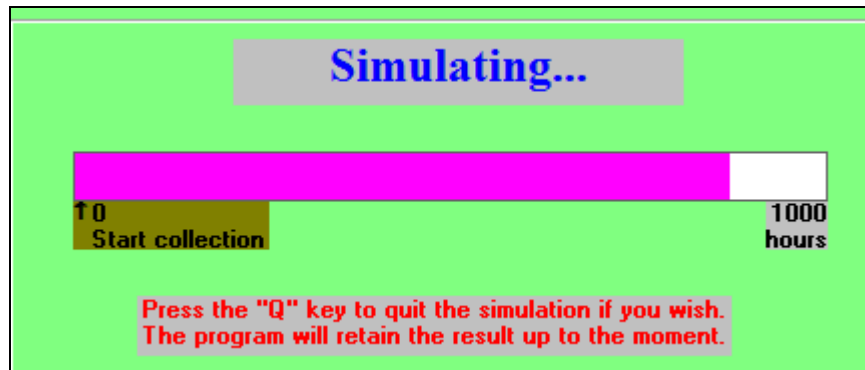


Figura 3.32

Obtenemos la siguiente tabla. Figura 3.33.



06-02-2009	Performance Measure	Result
1	System: M/M/2/6	From Simulation
2	Customer arrival rate (λ) per hour =	20.0000
3	Service rate per server (μ) per hour =	20.0000
4	Overall system effective arrival rate per hour =	19.8006
5	Overall system effective service rate per hour =	19.7986
6	Overall system utilization =	49.3378 %
7	Average number of customers in the system (L) =	1.2523
8	Average number of customers in the queue (Lq) =	0.2656
9	Average number of customers in the queue for a busy system (Lb) =	0.8191
10	Average time customer spends in the system (W) =	0.0633 hours
11	Average time customer spends in the queue (Wq) =	0.0134 hours
12	Average time customer spends in the queue for a busy system (Wb) =	0.0414 hours
13	The probability that all servers are idle (Po) =	33.7461 %
14	The probability an arriving customer waits (Pw) or system is busy (Pb) =	32.4217 %
15	Average number of customers being balked per hour =	0.2110
16	Total cost of busy server per hour =	\$0
17	Total cost of idle server per hour =	\$0
18	Total cost of customer waiting per hour =	\$0
19	Total cost of customer being served per hour =	\$0
20	Total cost of customer being balked per hour =	\$0
21	Total queue space cost per hour =	\$0
22	Total system cost per hour =	\$0
23	Simulation time in hour =	1000.0000
24	Starting data collection time in hour =	0
25	Number of observations collected =	19799
26	Maximum number of customers in the queue =	4
27	Total simulation CPU time in second =	2.8260

Figura 3.33

Y un resumen tenemos las siguientes probabilidades. Figura 3.34

06-02-2009 23:39:57 n	Estimated Probability of n Customers in the System	Cumulative Probability
0	0.3375	0.3375
1	0.3383	0.6758
2	0.1699	0.8457
3	0.0829	0.9287
4	0.0415	0.9702
5	0.0197	0.9899
6	0.0101	1.0000

Figura 3.34



Supongamos que se nos pide calcular la probabilidad de que ambos vendedores estén desocupados. Matemáticamente tenemos la siguiente relación:

$$P_0 = \left[\sum_{n=0}^2 \frac{1}{n!} + \frac{1}{21} \sum_{n=3}^4 (1/2)^{n-2} \right]^{-1} = 8/23 = 0.348$$

Que es igual a la probabilidad que los dos vendedores estén desocupados. Si verificamos en la tabla de medidas de desempeño. Figura 3.33. Sin simular el sistema tenemos directamente que

$$P_0 = 33.4862\%.$$

En la simulación tenemos que para $n=0$ tenemos una probabilidad de $0.3375 = 33.75\%$. Figura 3.34. Obsérvese que a pesar de que estas probabilidades difieren son equivalentes e ilustran el mismo concepto. Calculemos ahora la probabilidad de que a un suscriptor le contesten inmediatamente.

Matemáticamente tenemos la siguiente relación:

$$P_n = \frac{P_0}{n!}, n = 1, 2 ; P_1 = P_0 = 0.348; P_0 + P_1 = 0.606$$

Es igual a la probabilidad que contesten inmediatamente. En la simulación tenemos que la probabilidad acumulada (figura 3.34) para $n=1$ tenemos una probabilidad de $0.6758 = 67.58\%$.

Es importante mencionar en este punto que teóricamente el modelo más exacto es el que se ilustra en los libros como por ejemplo en Hiller *et al.*, (2006).

“Un modelo puede definirse como una representación simplificada de carácter cualitativo o cuantitativo de un sistema.” Gerez *et al.*, (1995). Este sistema puede estar ya funcionando o se puede estar diseñando. En el primer caso se emplea el modelo para obtener u conocimiento de su operación y estado o Futuro. En el segundo caso el modelo se usa para el diseño del sistema.



Calculemos la probabilidad de que a un suscriptor lo pongan en línea de espera.

Para que a un suscriptor lo tengan en línea de espera es necesario que en el sistema tengamos 2 o 3 clientes, por lo tanto la relación matemática es la siguiente:

$$P_2 = \frac{P_0}{2}, \quad P_3 = \frac{P_0}{4}; \quad P_2 + P_3 = 0.75 P_0 = 0.75 (0.348) = 0.261$$

Que es igual a la probabilidad de que lo pongan en la línea de espera. En la simulación (figura 3.34) tenemos que la probabilidad de que pongan a un cliente en espera es $P_2 + P_3$. $P_2 = 0.1699$ y $P_3 = 0.0829$; al sumar estas dos probabilidades obtenemos 0.2528 que es la probabilidad de que lo pongan en la línea de espera con una simulación de 1000 horas para el modelo que introducimos.

Si deseáramos saber en promedio, cuántas llamadas se pierden en un turno de 8 horas

$P_4 = 0.0415$ que es la probabilidad de que haya 4 clientes en el sistema y es igual a la probabilidad que una llamada se pierda por hora; definiendo a $X =$ llamadas perdidas en 8 horas $= 20(8)(0.0435) = 6.96$ llamadas perdidas por día.

En la siguiente y última parte del presente trabajo se profundiza un poco más en el tema de simulación utilizando *WINQSB*. Los conocimientos teóricos en el tema una vez más pueden consultarse en varios libros que se muestran como referencia bibliográfica.



III-3 SIMULACIÓN DE COLAS

WINQSB cuenta con un módulo exclusivo para Simulación de Sistemas de colas; a pesar de que en la sección anterior se ha hecho una simulación muy sencilla, en esta parte se adentra un poco más en el tema; se introducen conceptos básicos sin pretender ahondar en ellos; se ofrece sin embargo la bibliografía de referencia para el lector que esté interesado en este tema.

La simulación digital es una técnica que permite imitar en una computadora el comportamiento de un sistema físico o teórico según ciertas condiciones particulares de operación. "La simulación consiste básicamente en construir modelos informáticos que describen la parte esencial del comportamiento de un sistema de interés, así como en diseñar y realizar experimentos con el modelo y extraer conclusiones de sus resultados para apoyar la toma de decisiones" (Ríos *et al.*, 2000). Esta técnica se utiliza en el análisis de sistemas complejos en los que no es posible el tratamiento analítico o mediante métodos de análisis numérico, y sus orígenes están en los trabajos de Student para aproximar la distribución que lleva su nombre, y los métodos que Von Neumann y Ulam introdujeron para resolver ecuaciones diferenciales.

En ocasiones es posible experimentar con el problema real, e intentar por medio de observaciones y con ayuda de la Estadística llegar a conclusiones válidas que permitan mejorar el funcionamiento del sistema; sin embargo experimentar con el sistema real puede plantear algunos problemas, por lo que puede procederse a construir una versión simplificada, o prototipo del sistema; algunas veces hacer esto no es posible, o simplemente no es económico.

Existe otra opción que es construir un modelo lógico-matemático que describa mediante una serie de ecuaciones y relaciones el comportamiento básico del sistema; sin embargo cuando se requiere estudiar modelos con una mayor aproximación a la realidad, es posible llegar a situaciones en las que no se tendrá la capacidad de afrontar para resolverlas de manera analítica o numérica. En tales casos es posible usar la simulación, que consistirá en construir un programa en un ordenador que describa el comportamiento del sistema de interés, o refleje el modelo que lo representa; y proceder a experimentar con el programa o modelo para llegar a conclusiones que apoyen la toma de decisiones;



es conveniente comparar los resultados de los experimentos arrojados por la simulación *versus* los datos de entrada y salida reales; a esta operación se le llama prueba de bondad del ajuste.

Una vez que se ha detectado que el sistema de interés no opera de manera adecuada se plantea la forma de mejorar el funcionamiento del mismo, se mencionó anteriormente que existen métodos analíticos, numéricos y de simulación para realizar esto, a continuación se da una breve descripción de ellos; para una explicación más exhaustiva puede consultar: Ríos *et al.*, 2000 y Guasch *et al.*, 2002.

Aproximación Analítica: Para hacer uso de la *aproximación analítica* resulta esencial plantear ciertas hipótesis que permitan estudiar el sistema de interés, de manera que alguna distribución de probabilidad describa a las variables de interés (a una o a varias). Esto provoca que el razonamiento para estudiar así al sistema de interés sea poco robusto "...si cambiamos alguna de esas hipótesis, el cálculo exacto puede resultar mucho más complicado e, incluso, imposible de obtener mediante un método analítico" (Ríos *et al.*, 2000).

Aproximación Numérica: La *aproximación numérica* podría ser una solución a la situación anterior la cual puede realizarse mediante una integración numérica en la que el primer paso sería volver a parametrizar el problema, de manera que la región de integración sea acotada, una vez que se ha hecho esto puede resolverse mediante algún método de integración, por ejemplo, la regla del trapecio. La estrategia numérica adoptada es más robusta, en el sentido de que si se sustituye alguna de las hipótesis planteadas en el principio, es posible aún utilizarla. El inconveniente es que cuando se aumenta la región de integración la aproximación será cada vez más ineficaz.

Aproximación Basada en la Simulación: La aproximación basada en la simulación consiste en construir un programa que describa el comportamiento del sistema de forma que sea posible realizar experimentos con él en numerosas repeticiones para analizar los resultados y extraer conclusiones. "La simulación numérica es una recreación matemática de un proceso natural" (Alonso, 2008).

La simulación colabora de manera directa para tener una mejor comprensión del sistema y actualmente se ha convertido en una herramienta imprescindible en disciplinas como la Economía, Estadística, Informática, Medicina, Física y con una gran variedad de aplicaciones industriales y comerciales como los simuladores de vuelo, los juegos de simulación, los pronósticos del clima y las



finanzas, por mencionar algunas. En la siguiente sección se mencionan algunas ventajas y desventajas de esta herramienta.

Algunas ventajas de la simulación son las siguientes: No es necesario interrumpir las operaciones de la compañía; proporciona muchos tipos de alternativas posibles de explorar; proporciona un método más simple de solución cuando los procedimientos matemáticos son complejos y difíciles, proporciona un control total sobre el tiempo debido a que un fenómeno se puede acelerar, auxilia el proceso de innovación ya que permite al experimentador observar y jugar con el sistema; una vez construido el modelo se puede modificar de una manera rápida con el fin de analizar diferentes políticas o escenarios; generalmente es más barato mejorar el sistema vía simulación que hacerlo en el sistema real; es mucho más sencillo visualizar y comprender los métodos de simulación que los métodos puramente analíticos; da un entendimiento profundo del sistema; los métodos analíticos se desarrollan relativamente sencillos donde suele hacerse un gran número de suposiciones y simplificaciones, mientras que en los métodos de simulación es posible analizar sistemas de mayor complejidad o con menor detalle; en algunos casos, la simulación es el único medio para lograr una solución; y proporciona soluciones a problemas *sin* solución analítica.

Por otro lado las desventajas al utilizar la simulación es que es imprecisa, y no se puede medir el grado de esta imprecisión; los resultados de simulación son numéricos, por tanto, surge el peligro de atribuir a los números un grado mayor de validez y precisión. Los modelos de simulación en una computadora suelen ser costosos y requieren mucho tiempo para desarrollarse y validarse ya que se requiere gran cantidad de corridas computacionales para encontrar soluciones y los modelos de simulación no dan soluciones óptimas.

Disponer de un buen generador de números aleatorios es fundamental en la simulación; el uso progresivo de modelos de simulación cada vez más detallados exige generadores de números aleatorios de mayor calidad. Los números aleatorios son series de valores generadas de forma que su distribución se rige por alguna ley de probabilidad. En muchas ocasiones, los números aleatorios se asocian a valores equiprobables de algún intervalo; es decir valores de una variable $U(a, b)$ de hecho esta es la distribución de probabilidad más utilizada cuando no disponemos de información o si deseamos una serie de valores escogidos al azar.



Existen tablas de números aleatorios en algunos de los libros de Simulación y de Investigación de Operaciones (Una tabla de números aleatorios enteros de dos cifras aparece en el libro Winston (1994). Existe otra tabla disponible que es la de la RAND CORPORATION que contiene un millón de dígitos aleatorios, o una página Web: random.org, donde es posible encontrar generadores de números aleatorios. Algunas de estas listas se han calculado mediante experimentos físicos como midiendo el nivel de ruido ambiental. Pero otras se han calculado mediante métodos aritméticos instalados en un ordenador y se denominan pseudo aleatorios, cuyas características son: tener una distribución uniforme; ser estadísticamente independientes; ser reproducibles y tener un ciclo no repetitivo tan largo como se desee.

Actualmente la generación de números aleatorios está en auge, especialmente por su utilidad en criptografía, como generador de claves para sistemas informáticos o de seguridad. Estas claves tienen que parecer totalmente arbitrarias para agentes externos, pero predecibles para quienes las producen. También los programas de cálculo más utilizados incorporan funciones que generan números aleatorios. Por ejemplo la hoja de cálculo de EXCEL tiene la función ALEATORIO que devuelve valores de la uniforme continua en el intervalo cerrado de cero a uno. Estos números aleatorios se generan en la mayor parte de los casos por métodos llamados congruenciales lineales, multiplicativos y mixtos. Para estar seguros que se tiene un generador de números pseudo aleatorios confiable es necesario aplicarle extensas pruebas estadísticas, las cuales se describen de manera general en la sección 1.6. Para una explicación más exhaustiva de estos métodos existen diversas bibliografías una de ellas es Knuth (1969), Ríos *et al.*, (2000) y Guasch *et al.*, (2002).

Si se está analizando un sistema en general se emplea su modelo para obtener la respuesta del sistema a diferentes señales de entrada. Si al cambiar en determinada dirección la variable de entrada se empieza a observar el efecto deseado sobre las variables de salida, se continúa variando en la misma dirección la entrada. Hasta encontrar que ya empieza a disminuir el efecto deseado sobre la salida.

Por otra parte, si se está diseñando un sistema en general se han especificado las acciones o variables independientes a que estará sometido. Si se conoce la respuesta que se desea obtener o sea las variables de salida, es necesario determinar el modelo.



Uno de los mayores irritantes de la vida moderna, es el tener que hacer cola. Nadie desea tener su cuenta en un banco donde las colas son largas en general.

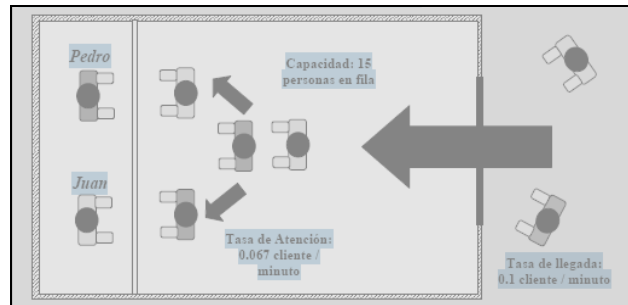
No necesariamente tienen que ser personas las que forman una cola. Pueden ser equipos a veces muy costosos en su operación, los que hacen colas como los aviones, autobuses, barcos, etc. Lo importante es que hacer cola siempre implica un gasto; es decir, se pierde dinero. Por otra parte evitar las colas totalmente también es caro.

WINQSB cuenta con un módulo exclusivo para los sistemas de colas; es mucho más completo que lo que se mostró en la parte anterior. La simulación manejada por WINQSB permite la participación de cuatro actores dentro del ambiente simulado:

1. Tasa de llega de clientes (Customer Arriving Source).
2. Colas (Queue). Líneas de espera.
3. Servidores (Server): Se especifica la cantidad de servidores en el sistema.
4. Colectores de Basura (Garbage Collector): Indica la posibilidad que el cliente abandone el proceso sin terminarlo. Puede ser considerado como un defecto en el sistema.

Para que WINQSB comprenda esta designación la tasa de llegada de los consumidores, las colas, los servidores y los colectores de basura serán identificados con las letras C, Q, S y G respectivamente.

Para ilustrar la manera en la que podemos hacer uso de las herramientas ofrecidas por el software mostraremos un ejemplo tomado de Quesada *et al.*, (2008). Un banco posee dos cajeros (pedro y Juan) los cuales atienden a un cliente en un promedio de 15 minutos con una desviación de 0.01. Los clientes llegan a una tasa de uno cada 10 minutos y hacen una sola cola cuya capacidad es de máximo 15 clientes. Se considera que la llegada de los clientes se comporta de forma muy similar a una distribución tipo Poisson y los cajeros con una distribución normal.



Si deseamos por ejemplo simular con 100 minutos de tiempo el modelo anterior. Es necesario considerar lo siguiente:

- Tasa de llegada: 0.1 cliente / minuto
- Servidores: Son 2 Pedro y Juan
- Tasa de Atención: 0.067 cliente / minuto
- Capacidad: 15 personas en fila

Por otro lado podemos observar también que existen tres actores principales:

1. Dos cajeros, los cuales serán considerados como servidores.
2. Los clientes, representados por una tasa de llegada.
3. La cola o línea de espera, a donde los clientes llegan para ser atendidos.

Para la solución a este problema vamos a considerar que los bancos emplean un sistema de espera de tipo PEPS (FIFO – First In First Out); es decir, los primeros clientes en entrar serán los primeros en ser atendidos.

Para ingresar esta información es necesario abrir el módulo Simulación de Sistemas de Colas (Queuing System Simulation) registramos la cantidad de actores participantes en la ventana Especificaciones del Problema (Problem Specification). Figura 3.35.

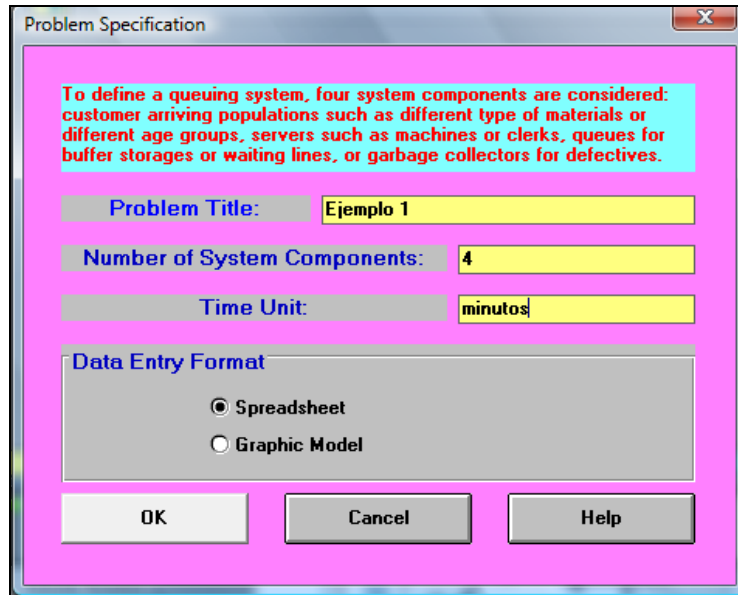


Figura 3.35

Pulsamos "OK" y aparecerá una ventana como la que se muestra en la figura 3.36. Es recomendable darle nombres a cada uno de los actores para evitar confusiones futuras.

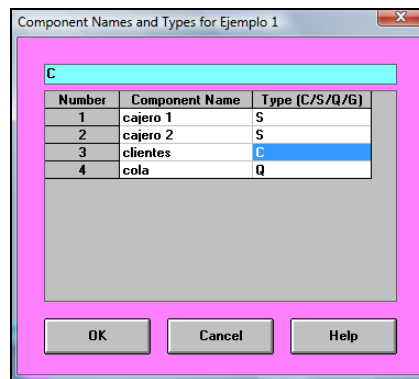


Figura 3.36

Los cajeros se denotan con la S (Server), los clientes con la C (Customer) y la cola con Q (Queue). Al pulsar "OK", aparecerá una plantilla donde ingresaremos la información primaria del problema. Figura 3.37.



Component Name	Type (C/S/Q/G)	Immediate Follower (Name / Prob / TransferTime, separated by ',')	Input Rule	Output Rule	Queue Discipline	Queue Capacity	Attribute Value	Interarrival Time Distribution	Batch Size Distribution	Service Time Distribution
cajero 1	S									
cajero 2	S									
clientes	C									
cola	Q									

Figura 3.37

Comencemos llenando los datos para los cajeros. Para programarlos es necesario introducir la información de que los cajeros dependen de los clientes. Para que WINQSB entienda esto en la columna Distribución de tiempos de servicio (Service Time Distribution) se ingresa la siguiente notación:

Clientes/Normal/0.06667/0.01

La notación completa es:

Nombre predecesor/Distribución/Parámetro 1/Parámetro 2/Parámetro 3

La primera corresponde a la conexión con los clientes, la segunda a la distribución de probabilidad de los servidores y los siguientes datos (parámetros) son utilizados de acuerdo a la información requerida por la distribución (por ejemplo, la distribución Normal requiere de dos parámetros: la media y la desviación estándar). Las distribuciones disponibles son las siguientes:

- Beta (Beta)
- Binomial (Binomial)
- Constante (Constant)
- Discreta (Discrete)
- Erlang (Erlang)
- Exponencial (Exponential)
- Gamma (Gamma)
- Hypergeométrica (Hypergeometric)
- Laplace (Laplace)
- Normal (Normal)
- Pareto (Pareto)



- Poisson (Poisson)
- Función de potencia (Power Function)
- Triangular (Triangular)
- Uniforme (Uniform)
- Weibull (Weibull)

De igual forma debemos completar los parámetros para los clientes. Lo primero es indicar la dependencia de una de las colas en la columna Sucesor inmediato (Immediate Follower). Luego, en la columna Distribución del tiempo entre llegada (Interarrival Time Distribution) con el siguiente formato:

Distribución/Parámetro 1/Parámetro 2/Parámetro 3

Para nuestro caso, la distribución quedaría:

Poisson/0.1

Recordemos que la tasa de llegada es 0.1 clientes por minuto y sigue una distribución de Poisson donde λ es el parámetro necesario. Los parámetros 2 y 3 no son requeridos para esta distribución. El resultado debe lucir como se muestra en la figura 3.38.

Component Name	Type (C/S/Q/G)	Immediate Follower (Name / Prob / TransferTime, separated by ',')	Input Rule	Output Rule	Queue Discipline	Queue Capacity	Attribute Value	Interarrival Time Distribution	Batch Size Distribution	Service Time Distribution
cajero 1	S									entes/normal/0.06670/0.01
cajero 2	S									entes/normal/0.06670/0.01
clientes	C	cola						Poisson/.1		
cola	Q									

Figura 3.38

La columna Distribución de los tamaños de los lotes (Batch Size Distribution), indica si los clientes llegan de forma agrupada o individual. En nuestro caso omitiremos llenar esta columna indicando que los clientes llegan de a uno a uno al banco.



Para programar la cola, debemos indicar que los dos cajeros se alimentarán de ella colocando los nombres en las casillas correspondientes a la columna Sucesor inmediato (Immediate Follower). Figura 3.39.

Component Name	Type (C/S/Q/G)	Immediate Follower (Name / Prob / TransferTime, separated by ',')
cajero 1	S	
cajero 2	S	
clientes	C	cola
cola	Q	cajero 1, cajero 2

Figura 3.39

En la columna Disciplina de la cola (Queue Discipline) marcamos FIFO y en la columna Capacidad de la cola (Queue Capacity) su capacidad (máximo 15 personas en espera). Figura 3.40

Component Name	Type (C/S/Q/G)	Immediate Follower (Name / Prob / TransferTime, separated by ',')	Input Rule	Output Rule	Queue Discipline	Queue Capacity	Attribute Value
cajero 1	S						
cajero 2	S						
clientes	C	cola					Poisson/.1
cola	Q	cajero 1, cajero 2			FIFO	15	

Figura 3.40

Para resolver el problema pulsamos sobre el botón Realizar simulación (Perform Simulation) en el menú Resolver y analizar (Solve and Analyze). Figura 3.41

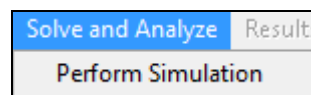


Figura 3.41

En la nueva ventana podremos indicar la cantidad de minutos a simular y que tipo de base (seed) para la generación de números aleatorios. Figura 3.42

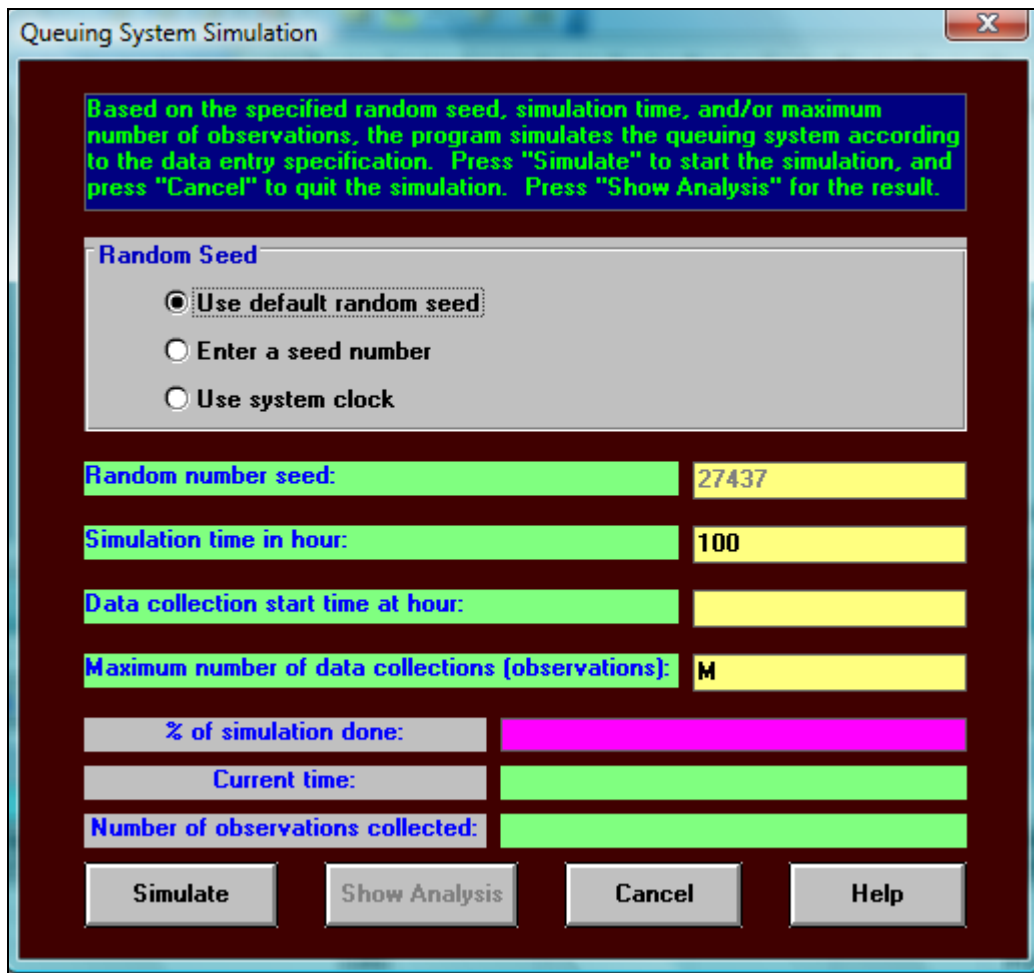


Figura 3.42

Pulsemos en el botón SIMULATE. WINQSB tomará el tiempo y mostrará las observaciones recolectadas durante ese tiempo: El botón SHOW ANALYSIS nos mostrará los resultados de la simulación. Figura 3.43



Result	Cientes
Total Number of Arrival	1123
Total Number of Balking	260
Average Number in the System (L)	2,2144
Maximum Number in the System	17
Current Number in the System	1
Number Finished	871
Average Process Time	0,0663
Std. Dev. of Process Time	0,0071
Average Waiting Time (Wq)	0,1879
Std. Dev. of Waiting Time	0,1560
Average Transfer Time	0
Std. Dev. of Transfer Time	0
Average Flow Time (W)	0,2542
Std. Dev. of Flow Time	0,1562
Maximum Flow Time	0,6007
Data Collection: 0 to	100 Minutoss
CPU Seconds =	1,6250

Figura 3.43

Se puede observar que en los 10 minutos llegaron 1123 clientes (Total Number of Arrival). El tiempo de espera promedio fue de 0.1879 (Average Waiting Time). El número máximo de personas en el sistema fue de 17: 15 en espera y 2 siendo entendidos (Maximun Number in the System). En promedio permanecieron 2,2144 personas en el sistema (Average Number in the System).

El software permite visualizar en modo gráfico. Para lograrlo es necesario ingresar el problema mediante el modo gráfico que provee WINQSB en la ventana inicial. Figura 3.44

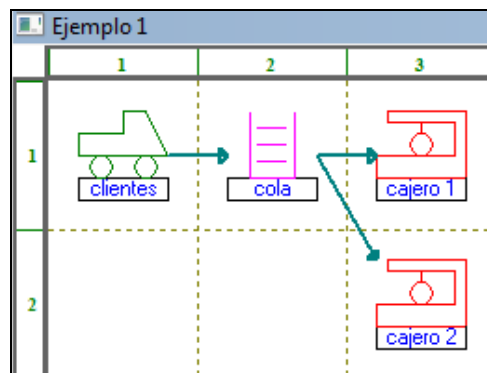


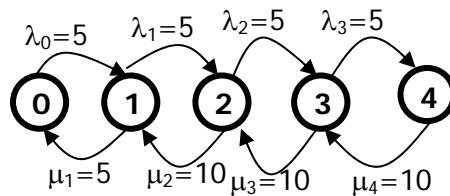
Figura 3.44



Para intercambiar los modos pulsamos en Pasar a formato matriz (Switch to Matrix Form) en el menú Formato (Format). A continuación simularemos un segundo ejemplo tomado de Murray (2007).

Se tiene un negocio de lavado de coches que cuenta con dos estaciones de lavado y espacio para dos carros que esperan ser lavados cuando se desocupe alguna de las estaciones de lavado. Cualquier coche adicional que llegue cuando están las dos estaciones y los dos lugares de espera ocupados se le rehúsa el servicio. La llegada de los automóviles tiene distribución Poisson con tasa media de 5 coches por hora. La duración del servicio de lavado tiene distribución de Poisson con una tasa media de 5 coches por hora.

El diagrama de densidades de transición de estados del problema es el siguiente:



Para simular el modelo es necesario abrir el módulo de Simulación de Sistemas de Colas. Figura 3.45.

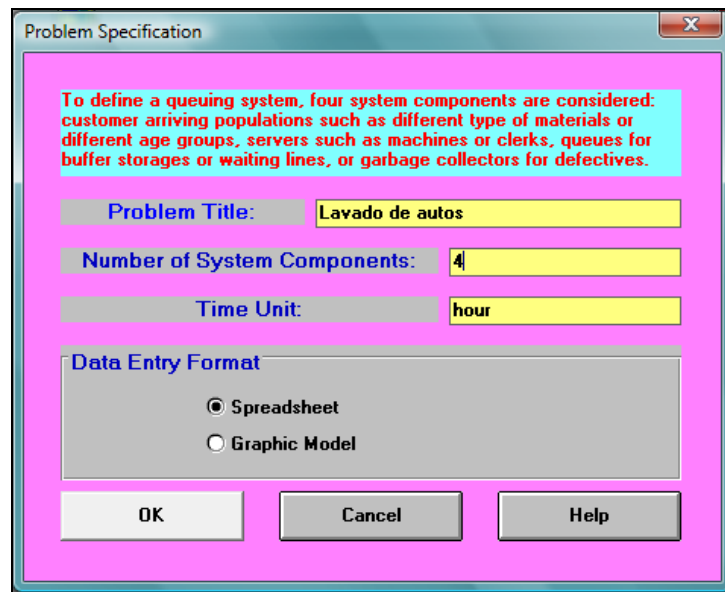


Figura 3.45



Asignamos nombres a cada uno de los actores como siguiendo el procedimiento mostrado en el ejemplo anterior. Figura 3.46

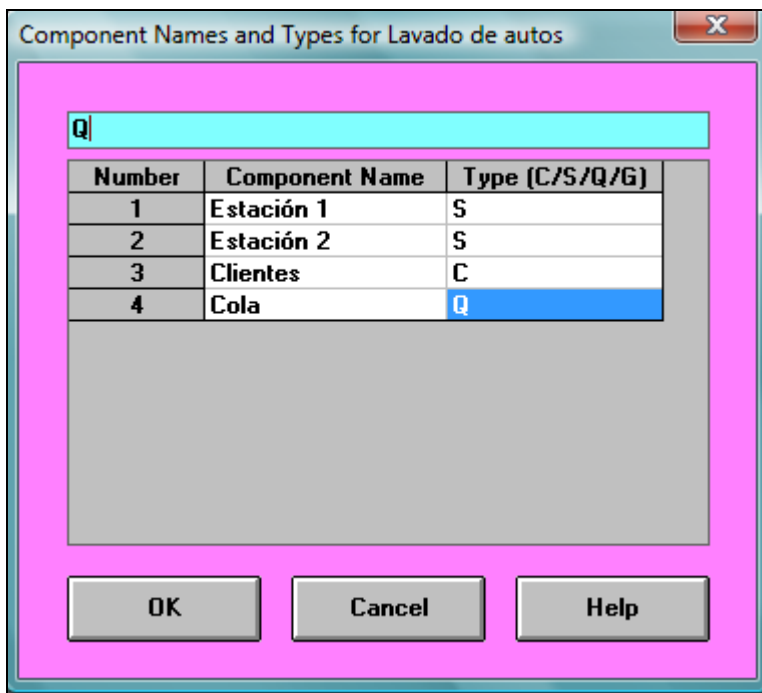


Figura 3.46

El llenado de la matriz es como se muestra en la figura 3.47

Component Name	Type (C/S/Q/G)	Immediate Follower (Name / Prob / TransferTime, separated by ',')	Input Rule	Output Rule	Queue Discipline	Queue Capacity	Attribute Value	Interarrival Time Distribution	Batch Size Distribution	Service Time Distribution
Estación 1	S									Poisson/5
Estación 2	S									Poisson/5
Clientes	C	cola						Poisson/5		
Cola	Q	Estación 1, Estación 2			FIFO	4				

Figura 3.47

Para este modelo tanto la tasa de llegadas como la tasa de servicio, siguen una distribución de Poisson con parámetro de 5. En el menú formato podemos acceder a la opción para visualizarlo gráficamente. Figura 2.48.

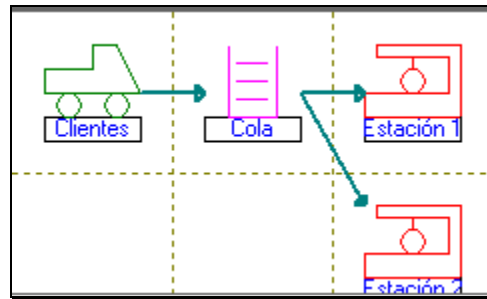


Figura 3.48

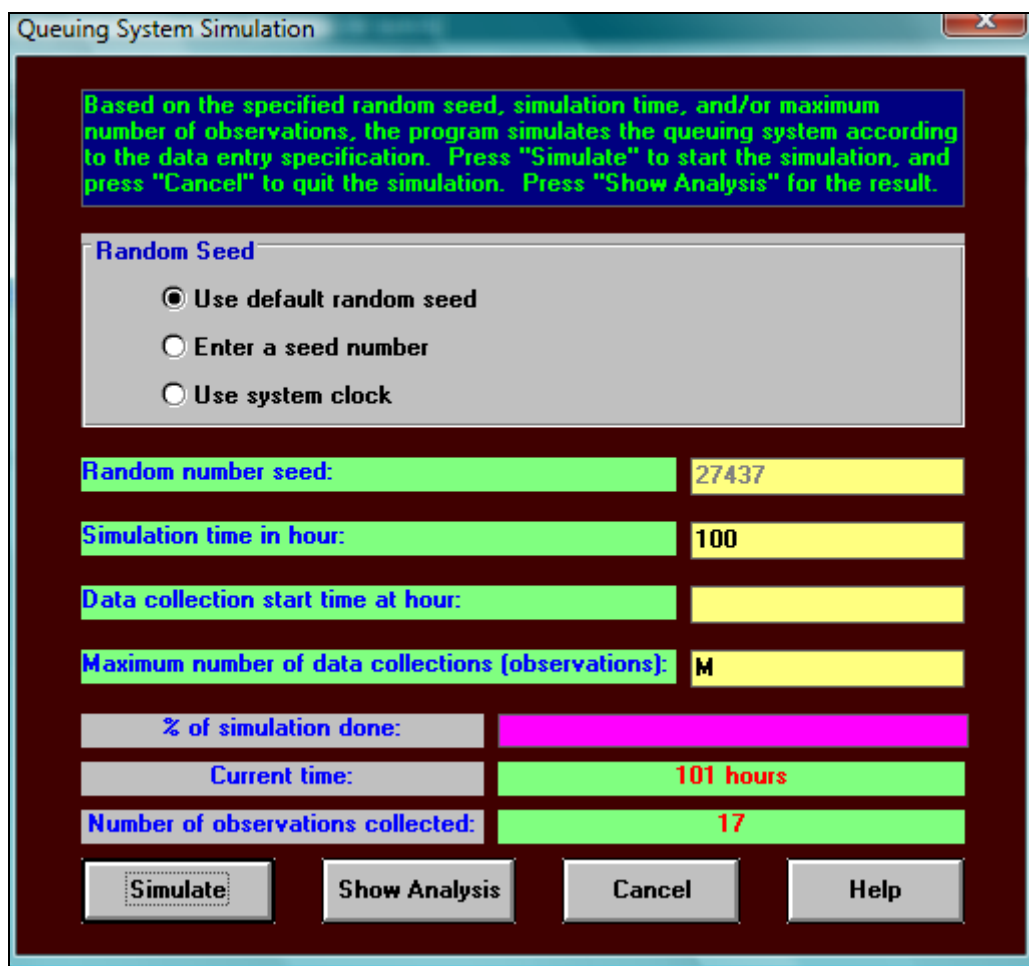


Figura 3.49

Pulsemos en el botón SIMULATE. WINQSB tomará el tiempo y mostrará las observaciones recolectadas durante ese tiempo. Figura 3.49 El botón SHOW ANALYSIS nos mostrará los resultados de la simulación. Figura 3.50.



06-03-2009	Result	Cientes
1	Total Number of Arrival	19
2	Total Number of Balking	0
3	Average Number in the System (L)	0.9700
4	Maximum Number in the System	2
5	Current Number in the System	2
6	Number Finished	17
7	Average Process Time	5.3529
8	Std. Dev. of Process Time	2.1677
9	Average Waiting Time (Wq)	0
10	Std. Dev. of Waiting Time	0
11	Average Transfer Time	0
12	Std. Dev. of Transfer Time	0
13	Average Flow Time (W)	5.3529
14	Std. Dev. of Flow Time	2.1677
15	Maximum Flow Time	8
	Data Collection: 0 to	100 hours
	CPU Seconds =	0.1720

Figura 3.50

Se puede observar que en los 100 minutos llegaron 19 clientes (Total Number of Arrival). El tiempo de espera promedio fue de 0 (Average Waiting Time). El número máximo de autos en el sistema fue de 2 en espera y 2 siendo atendidos (Maximum Number in the System). En promedio permanecieron 0.9700 autos en el sistema (Average Number in the System).

Un análisis desde el punto de vista de las estaciones de lavado nos muestra más información de la simulación; para acceder a él es necesario dirigirnos al menú Resultados (Results). Figura 3.51

06-03-2009	Server Name	Server Utilization	Average Process Time	Std. Dev. Process Time	Maximum Process Time	Blocked Percentage	# Customers Processed
1	Estación 1	41.00%	5.1250	1.9645	8	0.00%	8
2	Estación 2	50.00%	5.5556	2.3147	8	0.00%	9
	Overall	45.50%	5.3529	2.1677	8	0.00%	17
Data	Collection:	0 to	100	hours	CPU	Seconds =	0.1720

Figura 3.51



Las estaciones de lavado tuvieron un promedio de utilización (Server Utilization) del 41% y 50% para la estación 1 y 2 respectivamente.

La estación 1 atendió 8 autos y la estación 2 a 9 autos para un total de 17 autos. (Customer Processed). Desde el punto de vista de la cola tenemos. Figura 3.52

06-03-2009	Queue Name	Average Q. Length (Lq)	Current Q. Length	Maximum Q. Length	Average Waiting (Wq)	Std. Dev. of Wq	Maximum of Wq
1	Cola	0	0	1	0	0	0
Data	Collection:	0 to	100	hours	CPU	Seconds =	0.1720

Figura 3.52

El promedio de autos en la cola fue de 0 (Average Q. Length). El máximo de autos en la cola es de 1 (Maximum Q. Length).



CONCLUSIONES

Se ha señalado durante el desarrollo de cada uno de los capítulos la importancia de tener una base teórica por parte del lector. Es fundamental para el MÁXIMO aprovechamiento del software, que el alumno no sólo esté *familiarizado* con los algoritmos, métodos y procesos referentes a los temas tratados; sino que tenga una comprensión de los **conceptos básicos** referentes a la Investigación de Operaciones.

Por otro lado se debe tener presente la validez de la información con la que se alimenta el programa en cada uno de los módulos; ya que si se registran datos incorrectos, el programa arrojará resultados inválidos.

WINQSB se presenta como una herramienta capaz de solucionar muy diversos algoritmos relacionados con la Investigación de Operaciones. El lector ha encontrado un valor agregado ya que el uso del software le permite agilizar el estudio de algunas técnicas de la Investigación de Operaciones, ahorrándole tiempo apreciable en el manejo de datos y permitiéndole analizar de una mejor manera la información para la toma de decisiones. El abanico de opciones que muestran los 19 módulos amerita que se estudien los más posibles (en este trabajo sólo se han estudiado seis), ya que como se mencionó anteriormente cada uno de ellos ofrece herramientas potenciales para el ahorro de cálculos y desarrollos matemáticos bastante largos.

No es fácil encontrar un programa que sea lo suficientemente flexible para modelar cualquier situación. Una de las deficiencias del programa es que no permite usar el método de las dos fases para problemas de programación lineal. Un programa flexible puede resultar muy complejo y requiere de un programador especializado.

El factor de costos es un punto a favor; ya que este software está disponible en varios sitios de internet y además es gratuito. Otro factor a tomar en cuenta son los gráficos que como se ha visto en el desarrollo de los ejemplos son de gran ayuda y bastante claros.



En general el programa cuenta con gráficos bien diseñados, es accesible, robusto y se encuentra disponible en la red y en el laboratorio de Ingeniería Industrial de la UNAM, por lo que su uso es imprescindible en estas disciplinas.

Considero que aprender el uso de un software de esta índole debe ser un elemento integral en la formación profesional de los futuros Ingenieros Industriales, Actuarios y todas aquellas disciplinas relacionadas con la Administración Científica y el Análisis Cuantitativo; pues son herramientas necesarias en el ámbito laboral que aportan un valor agregado al egresado. Cabe mencionar que *WINQSB* y los ejemplos desarrollados en este trabajo ya se han utilizado en la docencia en las sesiones de laboratorio computacional con alumnos de la carrera de Ingeniería Industrial; y ha resultado ser un programa "muy amigable", "fácil de usar" y "de entender" como los propios alumnos señalan. Así el presente material realmente cumple con el objetivo señalado en el principio:

Explicar las herramientas básicas que contempla el software WINQSB incluyendo aplicaciones prácticas en la Investigación de Operaciones, que puedan servir como guía para estas asignaturas en el Laboratorio, contribuyendo a la formación profesional del futuro Ingeniero Industrial.

Por lo cual me llena de orgullo contribuir al desarrollo profesional de los futuros Ingenieros Industriales, aportar un poco a la docencia de la Ingeniería Industrial y a lo mucho que me da la Universidad Nacional Autónoma de México.



BIBLIOGRAFÍA

- Ackoff, R. L. , Sasieni, M.W., Fundamentos de Investigación de Operaciones. Limusa, México, 1975.
- Alonso Revenga Juana M. Flujo en redes y gestión de proyectos, teoría y ejercicios resueltos. GESBIBLIO, S L. España 2008.
- Colmenero Rivera José Antonio. Modelación en Redes con WINQSB. México, 2009.
- Estrada Juan Manuel. Programación Lineal. Universidad Nacional Autónoma de México, México, 2009.
- Gerez Greiser Víctor. , Cziltróm de Gerez. Introducción al análisis de Sistemas e Investigación de Operaciones. Representación y servicios de Ingeniería. S. A. 1978.
- Hiller, F. S. & Lieberman, G. Investigación de Operaciones, Mc Graw Hill. 7ª edición, México, 2001.
- Hiller, F. S., Lieberman, G.J: Investigación de Operaciones. McGraw-Hill. 8ª Ed. 2006.
- Hines, W y Montgomery, D. "Probabilidad y Estadística para Ingeniería y Administración" CECSA, 1993.
- Marín Pinillos Benito. Investigación de operaciones I. México, año 1994.
- Murray Lasso Marco Antonio. Apuntes de clase de investigación de operaciones II, 2007.
- Prawda, J. Métodos y Modelos de Investigación de Operaciones. Volumen II. Modelos Estocásticos. Editorial Limusa. México. 1980
- Quesada Irbanguen Víctor Manuel. , Juan Carlos Vergara Schmalbach. Métodos cuantitativos con WINQSB.



Ríos Insua David, Sixto Ríos Insua, Jacinto Martín Jiménez. Simulación Métodos y Aplicaciones. Alfaomega. Colombia, 2000.

Runyon, Richard y Haber Autrey. Estadística para las ciencias sociales. USA: Addison-Wesley Iberoamericana, 1992.

Sierra Bravo, R. Técnicas de investigación social: teoría y ejercicios, Madrid: Paraninfo, 1983.

Taha, Hamdy A. Investigación de Operaciones. Pearson Educación, 7ª edición, México, 2004.

Walpole, R y Myers (1992) "Probabilidad y Estadística". McGraw-Hill.

Winston, Wayne L. (1994): Investigación de operaciones. Aplicaciones y Algoritmos. Segunda Edición. Grupo Editorial Iberoamérica

Winston, W. L. Investigación de Operaciones, aplicaciones y algoritmos, Thomson, cuarta edición, 2005.

Yih-Long Chang. Win QSB Versión 2.0. Ed. Wiley and Sons. 2003



ANEXO

EJERCICIO 1

Una compañía de Pintura produce pinturas tanto para interiores como para exteriores, a partir de dos materias primas, M1 y M2. Por cada tonelada de pintura para interiores se requiere 4 toneladas de M1 y 2 toneladas de M2 y para cada tonelada de pintura para exteriores se requiere de 6 toneladas de M1 y una de M2. Se dispone de 24 toneladas de M1 y 6 de M2 diariamente. La utilidad que arroja una tonelada de pintura para exteriores es de 5000 dólares y de una tonelada de pintura para interiores es de 4000 dólares. La demanda máxima diaria de pintura para interiores es de 2 toneladas. Además la demanda diaria de pintura para interiores no puede exceder a la de pintura para exteriores por más de una tonelada. La compañía quiere determinar la mezcla de producción óptima de pinturas para interiores y exteriores que maximice las utilidades diarias y satisfaga las limitaciones.

a) Variables de Decisiones:

x_1 = n° de toneladas diarias producidas de pintura para exteriores.

x_2 = n° de toneladas diarias producidas de pintura para interiores.

b) Función Objetivo:

Cada tonelada de pintura para exteriores contribuye en la utilidad en 5000 dólares y cada tonelada de pintura para interiores en 4000 dólares, luego el objetivo es maximizar las utilidades diarias, entonces esto se expresa como:

$$\text{Máx } Z = 5000x_1 + 4000x_2 \text{ (Utilidad diaria expresada en dólares)}$$

c) Restricciones del modelo:

Cada tonelada de pintura para exteriores utiliza 6 toneladas de M1 y cada tonelada de pintura para interiores utiliza 4 toneladas de M1 y solo se dispone al día de 24 toneladas de M1, luego la expresión algebraica será:

$$6x_1 + 4x_2 \leq 24 \text{ (disponibilidad máxima de M1 en toneladas).}$$



Cada tonelada de pintura para exteriores utiliza una tonelada de M2 y cada tonelada de pintura para interiores utiliza 2 toneladas de M2 y solo se dispone al día de 6 toneladas de M2, luego la expresión algebraica será:

$$1x_1 + 2x_2 \leq 6 \text{ (disponibilidad máxima de M2 en toneladas).}$$

Hay además dos tipos de restricciones de la demanda: Una de demanda máxima diaria de pintura para interiores que se limita a dos toneladas. Y la otra el exceso de producción diaria de pintura para interiores sobre la pintura para exteriores es cuando mucho de una tonelada. Luego las expresiones algebraicas serán de la siguiente manera:

$$1x_2 \leq 2 \text{ (demanda máxima de pintura para interiores en toneladas).}$$

$$1x_2 - 1x_1 \leq 1 \text{ (diferencia máxima entre la producción de los dos tipos de pinturas).}$$

d) No negatividad de las variables:

Como le que se produce es pinturas y esto tiene que tener un valor positivo ó cero, entonces se cumple: $x_1, x_2 \geq 0$

Luego el modelo de Programación Lineal será:

Maximizar $Z = 5000x_1 + 4000x_2$ (Utilidad diaria expresada en dólares)

Sujeto a:

$$6x_1 + 4x_2 \leq 24 \text{ (disponibilidad máxima de M1 en toneladas).}$$

$$1x_1 + 2x_2 \leq 6 \text{ (disponibilidad máxima de M2 en toneladas).}$$

$$1x_2 \leq 2 \text{ (demanda máxima de pintura para interiores en toneladas).}$$

$$-1x_1 + 1x_2 \leq 1 \text{ (diferencia máxima entre la producción de los dos tipos de pinturas).}$$

$$(x_1, x_2) \geq 0. \text{ (la no negatividad de las variables)}$$

En base al modelo obtenido anteriormente y usando el software WIN QSB.



Muestre lo siguiente:

- La solución iteración por iteración
- La solución final que devuelve el programa y explique que significa cada renglón y columna
- La solución gráfica y compare con la solución obtenida anteriormente.

EJERCICIO 2

Un herrero con 80 kg. de acero y 120 kg. de aluminio quiere hacer bicicletas de paseo y de montaña para su venta, respectivamente a \$20.000 y \$15.000 pesos cada una para sacar el máximo beneficio. Para la bicicleta de paseo empleará 1 kg. De acero y 3 kg de aluminio, y para la de montaña 2 kg. de ambos metales. Formule un modelo de programación lineal.

EJERCICIO 3

Considere los siguientes modelos de programación lineal y usando WINQSB resuelva lo que se pide.

1.- Función Objetivo: $\text{Max. } Z = \$20.000X + \$15.000Y$

Restricciones (Sujeto a):

$$3X+2Y \leq 120$$

$$X+2Y \leq 80$$

$$(X, Y) \geq 0 \text{ (Restricción de no negatividad)}$$

a) Represente gráficamente las restricciones y señale la región de soluciones factibles, las soluciones básicas factibles y las soluciones básicas no factibles.

b) Encuentre la solución usando el método gráfico.



c) Resuelva el problema usando el método simplex.

2.- Encontrar la solución óptima de:

Función Objetivo: $\text{Max. } Z = \$1500X_1 + \$2100X_2 + \$1300X_3$

Restricciones (Sujeto a):

$$10X_1 + 15X_2 + 7X_3 \leq 3300$$

$$12X_1 + 17X_2 + 7X_3 \leq 3500$$

$$8X_1 + 9X_2 + 8X_3 \leq 2900$$

$$(X_1, X_2, X_3) \geq 0 \text{ (Restricción de no negatividad)}$$

EJERCICIO 4

Utilice WINQSB para solucionar los siguientes problemas:

1.- MAXIMIZAR $Z = 4 X_1 + 8 X_2$

Sujeto a: $X_1 + 2 X_2 \leq 2$

$$2 X_1 + X_2 \leq 3$$

$$(X_1, X_2) \geq 0 \text{ (Restricción de no negatividad)}$$



2.- MAXIMIZAR $Z = X_1 + X_2$

Sujeto a: $X_1 + X_2 \geq 4$

$$X_1 + 2X_2 \leq 2$$

$$(X_1, X_2) \geq 0 \text{ (Restricción de no negatividad)}$$

3.- Minimizar $Z = 3X_1 + X_2 + 5X_3$

Sujeto a: $6X_1 + 3X_2 + 5X_3 \leq 45$

$$3X_1 + 4X_2 + 5X_3 \leq 30$$

$$(X_1, X_2) \geq 0 \text{ (Restricción de no negatividad)}$$

EJERCICIO 5

Con el módulo de Modelación de Redes de WINQSB. Resuelva los siguientes problemas.

1.- Un contratista tiene que acarrear grava a tres construcciones. Puede comprar hasta 18 toneladas en un foso de grava al norte de la ciudad y 14 toneladas en uno al sur. Necesita 10, 5 y 10 toneladas en las construcciones 1, 2 y 3 respectivamente. El precio de compra por tonelada en cada foso y los costos de acarreo se dan en la siguiente tabla:



				PRECIO POR TONELADA
COSTO POR TONELADA				
CONSTRUCCION	1	2	3	
FOSO				
NORTE	\$300	\$600	\$500	\$1000
SUR	\$600	\$300	\$400	\$1200

El contratista desea determinar cuánto acarrear de cada foso a cada construcción de manera que se minimice el costo total de compra y acarreo de grava. Formule este problema como un problema de transporte construyendo la tabla de costos y requerimientos apropiada y resuelva el modelo para contestar la pregunta del contratista.

2.- Resuelva los siguientes problemas usando el algoritmo de asignación:

A) Una compañía ha sido contratada para realizar 5 trabajos. Estos trabajos se pueden efectuar en seis de sus plantas de manufactura. Debido a la magnitud de los trabajos no es factible asignar más de un trabajo a una planta en particular. Así mismo el segundo trabajo no se puede asignar a la tercera planta. A continuación se dan los costos estimados en millones de pesos para efectuar los trabajos en las distintas plantas:

TRABAJO	PLANTA					
	1	2	3	4	5	6
1	50	55	42	57	48	52
2	66	70	M	68	75	63
3	81	78	72	80	85	78
4	40	42	38	45	46	42
5	62	55	58	60	56	65



B) Resuelva la siguiente matriz de asignación:

	MAQUINA		
TRABAJADOR	A	B	C
1	5	13	19
2	13	10	15
3	11	15	27
4	15	9	6

EJERCICIO 6

Use WINQSB para los siguientes problemas:

1.- Obtenga el modelo dual para los siguientes problemas de programación lineal:

A) Minimizar $Z = 6X_1 + 8X_2 + 3X_3 - 4X_4$

Sujeto a: $3X_1 + X_2 - X_3 = 4$

$5X_1 + 2X_2 - X_4 = 7$

$(X_1, X_2, X_3, X_4) \geq 0$ (Restricción de no negatividad)



B) Minimizar $Z = 3X_1 + 4X_2 + 6X_3 + 7X_4 + X_5 + 2X_6 - X_7$

Sujeto a:

$$\begin{aligned} 2X_1 - X_2 + X_3 + 6X_4 - 5X_5 - X_6 &\geq 6 \\ X_1 + X_2 + 2X_3 + X_4 + 2X_5 - X_7 &\leq 3 \\ X_3 - X_4 + 3X_5 + 7X_7 &= 9 \\ 8X_1 + 2X_2 - X_3 + 6X_4 + X_6 &\geq -1 \\ (X_2, X_4, X_6) &\geq 0 \text{ (Restricción de no negatividad)} \end{aligned}$$

C) Maximizar $Z = 2X_1 + 3X_2 - 5X_3$

Sujeto a:

$$\begin{aligned} X_1 + X_2 - X_3 + X_4 &\geq 5 \\ 2X_1 + X_3 &\leq 4 \\ X_2 + X_3 + X_4 &= 6 \\ (X_2, X_3) &\geq 0 \text{ (Restricción de no negatividad)} \end{aligned}$$

EJERCICIO 7

Resuelva cada uno de los siguientes modelos de programación lineal usando el método simplex dual y con ayuda de WINQSB.

A) Minimizar $Z = 4X_1 + 10X_2 + 12X_3 + 15X_4$

Sujeto a:

$$\begin{aligned} 2X_1 + X_3 + 3X_4 &\geq 15 \\ 2X_2 + X_3 + 2X_4 &\geq 8 \\ X_1 + X_3 + 2X_4 &\leq 13 \end{aligned}$$

$(X_1, X_2, X_3, X_4) \geq 0$ (Restricción de no negatividad)



B) Maximizar $Z = X_1 + X_2$

Sujeto a:

$$\begin{aligned} -X_1 + X_2 &\leq 4 \\ -2X_1 + X_2 &\geq -5 \\ X_1 + X_2 &\leq 10 \\ (X_1, X_2) &\geq 0 \text{ (Restricción de no negatividad)} \end{aligned}$$

EJERCICIO 8

Se tiene un negocio de lavado de coches que cuenta con dos estaciones de lavado y espacio para dos carros que esperan ser lavados cuando se desocupe alguna de las estaciones de lavado. Cualquier coche adicional que llegue cuando están las dos estaciones y los dos lugares de espera ocupados se le rehusa el servicio. La llegada de los automóviles tiene distribución Poisson con tasa media de 5 coches por hora. La duración del servicio de lavado tiene distribución de Poisson con una tasa media de 5 coches por hora.

A. ¿Cuál es la probabilidad de que no haya ningún coche ni esperando ni siendo lavado en una hora?

B. ¿Cuál es la probabilidad de que a un coche se le niegue el servicio por estar lleno el espacio de espera en una hora?

El lector interesado en realizar más ejercicios puede encontrar un gran variedad en: Hiller (2006), Pinillos (1994), Prawda (1980), Taha (2004) y Winston (2005).

