2 ej 45

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

METODOS PARA LA TOMA DE DECISIONES BAJO CRITERIOS MULTIPLES

T E S I S

QUE PARA OBTENER EL TITULO DE

A C T U A R I O

P R E S E N T A :

CLAUDIA VAZQUEZ MUÑOZ

México, D. F. 1989

I N D I C E

Introducción 1

Capítulo I 3

1.1 Descripción del problema, 4
1.2 Modelos multiatributos, 8

1.2.1 Modelos de dominación, 10
1.2.2 Modelos de satisfacción, 11
1.2.3 Modelos lexicográfico, 13
1.2.4 Modelo maximin, 15
1.2.5 Modelo maximax, 17
1.2.6 Modelo aditivo, 19
1.2.7 Modelos de utilidad configural, 21
1.2.8 Curvas de indiferencia, 23
1.2.9 Escalamiento multidimensional, 24
1.2.10 Programación lineal o no lineal, 26

Capítulo II 32
Programación de metas

2.1 Modelo de metas múltiples y sin prioridades, 38
2.2 Modelo de metas múltiples con prioridades, 43
2.3 Modelo de metas múltiples con prioridades y ponderaciones, 49
2.4 Método gráfico, 55
2.5 Algoritmo de programación de metas, 64

Capítulo III 73

3.1 Método Electra I, 74
3.2 Algoritmo Electra I, 81
3.3 Método Electra II, 96
3.4 Algoritmo Electra II, 101
3.5 Método Electra III, 119
3.6 Algoritmo Electra III, 127

Capitulo IV 138

4.1 Programa Electra I, 139
4.2 Estructura del programa, 150
4.3 Programa Electra II, 153
4.4 Estructura del programa, 163
4.5 Programa Electra III, 168
4.6 Estructura del programa, 181

Conclusiones 185

Bibliografía 188

Apéndice 190

listado de los programas

INTRODUCCION

En la vida diaria se nos presentan situaciones de decisión las cuales son más complicadas cuando se tiene disponible un gran número de alternativas y cada una de ellas presenta varias características por el cual ser seleccionadas.

En una toma de decisiones no sólo es indispensable el conocimiento del conjunto de alternativas o de opciones del cual se va a elegir la mejor de ellas sino que también es muy importante definir los intereses del tomador de decisiones, ya que con ello se conocen los atributos mediante los cuales se califica cada alternativa reflejando la importancia que tiene cada una para el interesado.

Existen varios métodos para asesorar al tomador de decisiones en la toma de decisiones bajo criterios múltiples. Cada uno tiene su utilidad de acuerdo a los intereses de éste y de la información que se tiene.

Así, si se requiere de un intercambio entre criterios o atributos existen los modelos compensatorios; en caso contrario, se encuentran los modelos no compensatorios.

Con los modelos de representación espacial, como su nombre lo indica, se manejan a las alternativas por medio de una representación espacial.

A través de los modelos de programación matemática, el problema de la toma de decisiones bajo criterios múltiples se plantea por medio de un objetivo sujeto a un conjunto de restricciones.

Con el método Electra I, II y III se representa el problema por medio de una matriz en la que el elemento (i,j) indica el valor o grado de satisfacción de la alternativa i bajo el punto de vista del criterio j.

El objetivo de esta tésis es el de describir los métodos citados anteriormente, la forma de aplicarlos, así como las ventajas y desventajas de cada uno.

La presentación del material en este trabajo es el siguiente:

En el capítulo I se clasifica a los métodos de criterios múltiples en compensatorios y no compensatorios. Se describen algunos de ellos y se ilustran a través de ejemplos. Así mismo, se exponen otros métodos cuya aplicación es más compleja por la información inicial que se necesita pero que guardan cierta similitud con los anteriores.

En el capítulo II se describe el método de programación de metas con el cual se resuelven problemas de toma de decisiones que tienen metas múltiples (y que a menudo están en conflicto) a través de ecuaciones lineales de metas (restricciones) y que pueden asignarse prioridades a estas metas (jerarquizaciones) en términos de importancia.

En el capítulo III se exponen tres métodos: Electra I, Electra II y Electra III para resolver problemas de decisión con criterios múltiples en donde intervienen ampliamente los intereses del tomador de decisiones permitiendo un análisis de sensibilidad.

En el capítulo IV se presenta la automatización, a través de programas de cómputo, de estos tres métodos para facilitar el manejo de los mismos.

Cabe aclarar que el origen del término Electra no tiene relación alguna con el título de la célebre tragedia griega ni con otra obra literaria. Este término fue propuesto por los autores de un algoritmo de solución de un problema de decisión: Electre = ELimination Et Choix Traduisant la REalité.

Finalmente, se presentan corridas de los programas de cómputo que incluyen ejemplos realizados en el capítulo III.

CAPITULO I

En la mayoria de situaciones de decisión real el elegir la mejor acción debe basarse en aspectos múltiples o criterios; es decir, cada acción lleva asociada con ella a un conjunto o vector de varias consecuencias.

En términor burdos, el problema es " ¿ Cómo puede un tomador de decisiones elegir la mejor de diversas acciones alternas cuando cada acción produce un conjunto o vector de resultados y donde cada resultado puede en el mejor de los casos ser descrito solamente en términos de sus caracteristicas de desempeño o atributos diversos ? "

Debido a la extensión de este tipo de problemas, existen varias técnicas o modelos para resolverlos. El objetivo de este capitulo es el de describir varios de ellos iniciando con la descripción general del problema y proporcionando un ejemplo para su ilustración. Posteriormente, se introduce un esquema de clasificación de los modelos a tratar.

Muchos de estos modelos son sencillos y fáciles de aplicar; otros son de mayor complejidad pues necesitan datos iniciales más detallados pero a pesar de esto existen similitudes, lo que permite una clasificación.

Esta clasificación se basa en el manejo de los criterios que deben tomarse en cuenta para la elección de la mejor alternativa. Si al aplicar el método se considera un intercambio entre los criterios o atributos se les llama modelos compensatorios, en caso contrario, se les llama no compensatorios.

Para comprender mejor la aplicación y planteamiento de los modelos se ilustran numéricamente y se discuten los puntos de similaridad y diferencia.

1.1 DESCRIPCION DEL PROBLEMA:

Supongamos que se tiene un conjunto finito de alternativas A . Cada alternativa $A_i \in A$, está caracterizada por un conjunto de atributos y por lo tanto puede representarse la alternativa A_i mediante la m-ada $(A_{1i}, A_{2i}, \ldots, A_{mi})$ en donde A_{ji} es el valor del j-ésimo atributo de la alternativa A_i ; dicho valor no es necesariamente numérico.

Supongamos además, un orden de preferencia, por ejemplo : si el valor del j-ésimo atributo de la alternativa 2 es mejor que el de la alternativa 1, entonces decimos que $A_{1j} < A_{2j}$.

El problema es elegir la mejor alternativa del conjunto de alternativas A. Si existe una alternativa A_i del conjunto A tal que es mejor que todas las demás "estrictamente"; es decir, bajo todos los criterios, el problema está resuelto, A_i es la solución. Si tal alternativa no existe, se tratará de elegir la mejor bajo casi todos los puntos de vista.

A continuación se ilustra un problema de este tipo.

Ejemplo 1.

Una compañía desea producir un nuevo producto que aumentará las utilidades de dicha empresa. Se tiene la opción de utilizar tres tipos diferentes de máquinas para fabricar el producto deseado. La compañía podrá escoger solamente una de las máquinas para la elaboración del artículo.

Los atributos o características, es decir los criterios que deben tomarse en cuenta para la elección de la máquina, de cada alternativa son:

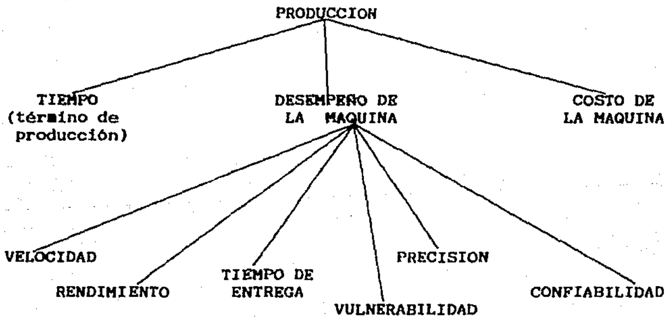
tiempo (terminación del producto), desempeño (de la máquina) y costo (de la máquina).

Estos atributos pueden descomponerse aún más para obtener especificaciones significantes mensurables como por ejemplo: desempeño que se puede dividir en:

velocidad de producción
precisión de producción
tiempo de entrega
confiabilidad o calidad de la producción
vulnerabilidad o defectuosidad de la producción
y rendimiento

El tomador de decisiones tiene la opción de descomponer cada atributo en niveles de tal manera que sean medibles y describan la esencia del problema. Sin embargo, no todos los atributos generados necesitarán utilizarse pues algunos pudieran ser redundantes como sería el caso de la velocidad y tiempo de producción que reflejan los mismos resultados; otros no sirven para hacer una discriminación entre alternativas porque producen la misma satisfacción independientemente de cual se elija bajo ese criterio como es el caso de la precisión de las máquinas si se tiene que todas las que intervienen en la elección son igualmente precisas.

Supongamos que ya se llevó a cabo un análisis llegando a que el tiempo de entrega, costo, vulnerabilidad y rendimiento son los atributos del problema. El siguiente esquema proporciona una visión de la jerarquía de nuestros objetivos.



ATRIBUTO	MAQUINARIA		
	A1	A2	A3
1) TIEMPO DE ENTREGA (hrs.)	2	1	1/2
2) COSTO ($ x 100,000)	12	15	18
3) VULNERABILIDAD	media	alta	baja
4) RENDIMIENTO (prod. x día)	6,000	7,000	5,000

El atributo vulnerabilidad, se expresa cualitativamente como alta, media y baja. Los atributos cuantitativos como es en el caso de los otros atributos, poseen una medida métrica. Los atributos cualitativos aunque también se expresan por medio de valores numéricos, requieren algún tipo de juicio o medida subjetiva.

Este cuadro nos proporciona, por columnas, los valores de los respectivos criterios para cada alternativa; por ejemplo, la alternativa 1 (máquina A1) se caracteriza por un tiempo de entrega de 2 hrs. por cada producto, un costo de $1,200,000, una vulnerabilidad media y por último, un rendimiento de 6,000 productos.

Si el tomador de decisiones quisiera un tiempo mínimo de producción escogería la máquina A3, pero si le interesara más el rendimiento escogería A2, en caso de un menor costo se tendría que decidir por la máquina A1. Pero como la efectividad de la producción no solo está afectada por un atributo, hay que considerar los cuatro atributos; es decir, se desea elegir la mejor alternativa bajo casi todos los puntos de vista.

Ya que se tienen los atributos bien definidos, se puede proceder de dos maneras para asignar valores a las alternativas con objeto de poder compararlas tomando en cuenta todos los criterios. Estas son:

1. Asociar a cada alternativa un solo número de utilidad representando así su valor global.

2. Llegar a una decisión por procedimientos que retiren la individualidad de diversos atributos; es decir, utilizar procedimientos que no requieran comparación entre atributos.

Para explicar el primer punto contamos con que la utilidad de una alternativa se puede medir a través de

$$U(A) = \sum_{1<=j<=m} w_j \; U(A_{ji}) \qquad i=1,\ldots,n, \; j=1,\ldots m$$

donde

w_j es el peso de importancia para el atributo j.

en el ejemplo 1 n=3 y m=4.

Hemos transformado cada configuración multiatributo en una sola utilidad total.

Así, considerando un peso de 80 para el atributo tiempo de entrega, 85 para el costo, para vulnerabilidad 50 y para el rendimiento 70, se tiene que la utilidad para la alternativa 1 es:

$$U(A1) = 80(2)+85(12)+50(35)+70(6000)$$

nótese que para medir numéricamente el atributo vulnerabilidad, se le asoció al nivel bajo el valor de 25, al medio el valor de 35 y al alto el valor de 50.

En el punto 2, hay que asignar niveles satisfactorios-insatisfactorios (grado máximo y mínimo de satisfacción) a cada atributo separadamente y ver qué máquina cumple satisfactoriamente los criterios en todos los niveles.

Ejemplo:

Se acepta un tiempo máximo de entrega de 1 hr., un costo máximo de $1,500,000.- , una vulnerabilidad máxima media y un rendimiento mínimo de 6,000 productos.

Si ninguna máquina cumple con dichas especificaciones tendríamos que ajustar nuevamente las mismas para obtener una máquina que sea aceptable.

Si en cambio, dos o más máquinas cumplen con las especificaciones habria que ponerse más exigentes y cambiar los niveles satisfactorios-insatisfactorios para volverlos más estrechos.

Ya explicada la notación a utilizar, describiremos los modelos para elegir la mejor alternativa tomando en cuenta casi todos los puntos de vista o criterios.

1.2 MODELOS MULTIATRIBUTOS

Los modelos se clasifican en: modelos compensatorios , modelos no compensatorios, modelos de programación matemática y modelos de representación espacial.

Los modelos compensatorios permiten intercambios entre atributos, es decir, se asigna una sola utilidad a cada caracterización multidimensional que representa una alternativa. Por el contrario, los modelos no compensatorios no permiten intercambios entre los atributos, las comparaciones se hacen atributo por atributo y la caracterización general de una alternativa no es solo un número de utilidad.

Estos tipos de modelos a su vez se clasifican en otros que se encuentran resumidos en el siguiente cuadro:

MODELOS DE UTILIDAD MULTIATRIBUTO (UMA)			
MODELOS NO COMPENSATORIOS	MODELOS COMPENSATORIOS	MODELOS DE PROGRAMACION MATEMATICA	MODELOS DE REPRESENTACION ESPACIAL
-DOMINACION	-UTILIDAD ADITIVA	-PROGRAMACION LINEAL	-CURVAS DE INDIFERENCIA
-SATISFACCION	-UTILIDAD CONFIGURAL	-PROGRAMACION NO LINEAL	-ESCALAMIENTO MULTIDIMENSIONAL
-LEXICOGRAFICO		-PROGRAMACION META	
-MAXIMIN			
-MAXIMAX			

El cuadro anterior categoriza y muestra los modelos y técnicas que se examinarán en los capitulos I y II. La programación matemática y los modelos de representación espacial son tratados como categorias especiales de modelos UMA. Estas clases de modelos han sido listados por separado porque 1) su alcance teórico requiere mayor explicación 2) no caen puramente en las clasificaciones de modelos compensatorios y no-compensatorios. Por ejemplo, la programación de metas puede considerarse como un modelo no compensatorio porque no contempla intercambios entre atributos, sin embargo, este modelo puede auxiliar al tomador de decisiones en situaciones más complejas como se verá posteriormente. Algo similar sucede para el caso de programación lineal que podria verse como un modelo compensatorio.

En este trabajo se omite el análisis de las similitudes de los modelos de programación matemática y los de representación espacial con los compensatorios y no compensatorios (Keeney y Raiffa, 1973; MacCrimmon, 1973).

A continuación se describe cada uno de estos modelos.

MODELOS NO COMPENSATORIOS

1.2.1 MODELOS DE DOMINACION

Este modelo nos lleva a un problema más sencillo que el original ya que reduce el número de alternativas.

Para ello se dividen las alternativas en dominadas y no dominadas. Una alternativa se considera no dominada si es mejor o no es peor que las demás bajo todos los puntos de vista o atributos. Obteniendo dicha división, se eliminan las alternativas dominadas y se toma en cuenta unicamente el conjunto de alternativas no dominadas.

En general, si se tiene el problema de elegir la mejor alternativa entre un conjunto de alternativas A1, A2, A3,...,An en donde los atributos para la alternativa Ai son A_{1i}, A_{2i}, ... A_{mi} el problema se reduce a través del modelo de dominación considerando lo siguiente:

Se dice que la alternativa A1 domina a la alternativa A2 si:

$$A_{j1} >= A_{j2} \quad \text{para todo atributo } j$$

y además

$$A_{j1} > A_{j2} \quad \text{por lo menos para el atributo } j \text{ con } j \in (1,...m)$$

Por lo tanto, la eliminación nos deja unicamente con alternativas no dominadas.

Ejemplo:

Considerando el ejemplo 1, ninguna de las alternativas es dominada ya que si comparamos la alternativa A1 con A2 se tiene que el tiempo de entrega para A2 es mejor que A1 (1 hr. vs. 2 hrs.) pero el costo es peor (15 vs. 12), la vulnerabilidad es mejor para la alternativa A1(media vs. alta) y por último, el rendimiento es peor (6,000 vs. 7,000). Pero si ambas alternativas tuvieran el mismo tiempo de entrega y el mismo rendimiento entonces A1 sería mejor que A2.

Observación:

Este modelo no es muy bueno para tomar una decisión final ya que podría suceder que un gran número de alternativas permanezcan después de ser aplicado el método y más aún cuando están involucrados varios atributos.

Cuando se emplea este modelo, no es necesario conocer el valor numérico de los atributos y no se requiere que el tomador de decisiones dé importancia relativa a cada uno de los atributos, ambas características son una ventaja ya que conllevan a la sencilléz del modelo.

1.2.2 MODELOS DE SATISFACCION

Este modelo clasifica a las alternativas en aceptables y no aceptables, para ello, el tomador de decisiones define una cota para los valores de las alternativas en cada atributo; si el valor de la alternativa es mayor o igual que dicha cota, se dice que es una alternativa aceptable. Así, el número de alternativas se reduce al eliminar las alternativas no aceptables.

En términos generales, si se tiene un conjunto de alternativas A y se quiere escoger la mejor opción, el modelo de satisfacción divide a dicho conjunto en alternativas aceptables y no aceptables considerando que la alternativa Aj es aceptable si es mejor o no es peor que $g_1, g_2, \ldots, g_m$

en donde

g_j es una constante definida por el tomador de decisiones para el j-ésimo atributo con $j= 1,2,,,m$.

Ejemplo:

Supongamos que en el ejemplo 1 se establecen los siguientes requisitos:

1.- tiempo de entrega no mayor de 2 hrs.

2.- costo no mayor de $1,500,000.-

3.- vulnerabilidad no mayor que la media y un rendimiento no menor que 5,000 artículos.

Por lo tanto, A1 es la única alternativa aceptable.

Ejemplo:

Supongamos que se tienen las constantes g_1 y g_2:

1.- rendimiento no menor que 6,000 artículos y

2.- vulnerabilidad media.

Nuevamente A1 es la única alternativa aceptable.

Observación:

Cabe señalar que la diferencia entre este último modelo y el de dominación es que, a pesar de que al aplicar el modelo nos contínúan quedando varias alternativas, es un método que puede aplicarse iterativamente; es decir, se pueden ir elevando o bajando los niveles de aspiración para llegar a una sola alternativa.

Sin embargo el inconveniente es que se requiere conocimiento de los niveles de aspiración y si se utilizan valores inadecuados para cada atributo ninguna de las alternativas recibe crédito por valores de atributos especialmente buenos o malos.

Como en el modelo de dominación, no se requiere de comparación entre atributos ni que la información entre atributos se encuentre en forma numérica; tampoco se necesita información de la importancia relativa de los atributos.

1.2.3 MODELO LEXICOGRAFICO

En este modelo se clasifica a las alternativas de acuerdo a los atributos más importantes. Si todas las alternativas se pueden ordenar de acuerdo al atributo mas importante entonces los atributos restantes no se toman en cuenta; por el contrario, si dos o mas alternativas están empatadas con respecto al mejor atributo entonces se considera el siguiente atributo de mayor importancia, y así sucesivamente hasta lograr un orden en las alternativas.

Si el criterio 1 es el más importante, el criterio m el de menor importancia y

$$A1=(A_{11},A_{21},\ldots,A_{m1})$$

y $$A2=(A_{12},A_{22},\ldots,A_{m2})$$

son dos alternativas con sus niveles de atributos entonces A1 $\succ$A2 (A1 se prefiere a A2) si $A_{11} \succ A_{12}$. Si $A_{11} = A_{12}$ vemos la relación A_{21} y A_{22}, en caso de que $A_{21} \succ A_{22}$, se prefiere A1 y así sucesivamente.

Ejemplo:

En el ejemplo 1, supongamos que el mejor atributo es el rendimiento, entonces escogeríamos A2 . Pero si A1 y A2 tuvieran el mismo rendimiento y el costo fuera el siguiente atributo más importante entonces escogeríamos A1 ya que $A_{21} < A_{22}$.

Observación:

Estos tres métodos son no compensatorios pues las alternativas pueden evaluarse sobre la base de atributo por atributo y no sobre todos los atributos. No existen comparaciones inter-atributos por lo que la escala de un atributo no necesita compararse con la escala de otro y así un atributo puede expresarse en horas mientras que otro se expresa en pesos.

Otra ventaja es que son modelos sencillos y es posible aplicar primero el modelo de satisfacción después el de dominación y por último el lexicográfico para realizar una mejor elección.

1.2.4 MODELO MAXIMIN

Para utilizar este modelo es necesario que los valores de los atributos sean valores numéricos.

Así, se escoge el valor mínimo de cada alternativa, es decir, una alternativa será representada por el peor valor de los atributos. Teniendo los valores representantes, se toma el máximo de dichos valores o

$$\max_j \{Ai\} \min_j A_{ji}$$

y se elige la alternativa correspondiente a este valor.

Ejemplo 2.

Para el ejemplo 1, se tomará en cuenta una escala de 100 puntos; además, se considerará que el mayor tiempo de término de producción que se tiene es 2 hrs. y el mínimo 1/2 hr, con ello, a 1/2 hr. se le asignará un valor de 85, a 2 hrs. un valor de 65 y a 1 hr. uno de 75; a una vulnerabilidad alta un valor de 30, para una media 55 y para una baja 80.

Cada tomador de decisiones puede dar valores a los atributos de la manera deseada siempre y cuando proponga una escala.

Además, hay que fijarse en la diferencia entre el valor para una alternativa y el valor para otra desde un mismo punto de vista o atributo.

Se propone la siguiente tabla:

	A1	A2	A3
TIEMPO DE ENTREGA	65	75	85
COSTO	80	70	60
VULNERABILIDAD	55	30	80
RENDIMIENTO	60	70	50

En el ejemplo se tiene que 55 es el mínimo para la alternativa A1, 30 para A2 y 50 para A3. El máximo de estos tres valores es 55, por lo tanto, se escoge la alternativa A1.

Observación:

Los valores interatributos son comparables pero tienen inconvenientes como es el de utilizar sólo una pequeña parte de la información para tomar la decisión final. Así, podria presentarse el caso de una alternativa superior claramente en todos los atributos excepto en una y otra alternativa con valores promedio en todos sus atributos, se escogería ésta última.

1.2.5 MODELO MAXIMAX

En este modelo al igual que el anterior, se requiere de valores numéricos para los atributos.

El procedimiento del modelo maximax consiste en seleccionar la alternativa con el valor alto más grande del atributo.

En símbolos

$$\max_j \{Ai\} \max_{ji} A \; .$$

Ejemplo:

En el ejemplo 2, el valor máximo para A1 es 80, para A2 75 y para A3 85; el máximo es 85, por lo tanto escogemos la alternativa A3.

Observación:

Al igual que en el método anterior la decisión final se realiza considerando solamente un atributo por alternativa y se suponen todos los atributos intercambiables sin importar cual sea el que se maximice, por lo que este método es aplicable cuando el tomador de decisiones no tiene requerimientos a priori sobre cual atributo es éste.

MODELOS COMPENSATORIOS

En este tipo de modelos un valor bajo en algún atributo puede ser intercambiado o compensado por un valor alto en algún otro atributo. Una de las diferencias de los modelos de este tipo es la forma de compensar estos valores.

Cabe señalar que un intercambio es la declaración de la cantidad de un atributo que una persona estaría dispuesta a sacrificar para obtener una ganancia especifica en algún otro atributo y viceversa.

Un enfoque para realizar un intercambio seria pedirle al tomador de decisiones que realizara intercambios hasta que le sea indiferente entre los resultados que describen una alternativa real (O_1) y un conjunto de referencia de resultados (O'_1), en este caso $O_1 \sim O'_1$.

Si suponemos que tenemos dos alternativas, entonces O'_1 y O'_2 tendrian el mismo valor para todos los atributos excepto para uno. Si O'_1 se prefiere a O'_2, ésto implica que O_1 se prefiere pues $O_1 \sim O'_1 \succ O'_2 \sim O_2$.

Se ilustrará la realización de intercambios tomando en cuenta el ejemplo 1 y considerando unicamente las alternativas A3 y A2 con la siguiente referencia :

(1,15,media,6000).

Supóngase que si queremos aumentar el tiempo de término de producción 1/2 hr. el costo se reduce $300,000 entonces (1/2,18,baja,5000)~(1,15,baja,5000); y si queremos aumentar el rendimiento en 1,000 articulos entonces la vulnerabilidad aumenta por lo que:

$$O_1 = (1/2,18,\text{baja},5000) \sim (1,15,\text{baja},5000) \sim (1,15,\text{media},6000) = O'_1 .$$

Con respecto a la aternativa A2, si queremos disminuir la vulnerabilidad entonces el rendimiento hay que disminuirlo en 3000 unidades, ésto implica que :

$$(1,15,\text{alta},7000)\sim(1,15,\text{media},4000)$$

Si aumentamos el rendimiento en 2000 articulos, entonces el costo aumenta $400,000, por lo que:

$$O_2 \approx (1,15,\text{alta},7000)\sim(1,15,\text{media},4000)\sim(1,19,\text{media},6000) = O'_2$$

Comparando O'_1 con O'_2 es decir,

$$(1,15,\text{media},6000) \text{ con } (1,19,\text{media},6000),$$

llegamos a la conclusión de que nos conviene la alternativa A3.

Con este método es fácil el intercambio entre valores de atributos y no se requiere de ninguna suposición particular sobre las preferencias del tomador de decisiones. Pero dicho proceso puede llegar a ser tedioso y se requiere mucho tiempo si existen varios atributos y alternativas , por ello se han desarrollado otros modelos.

1.2.6 MODELO ADITIVO

En este modelo las alternativas son representadas por su utilidad; la alternativa con mayor utilidad es la mejor opción.

Para obtener la utilidad de una alternativa el tomador de decisiones tiene que asignarles pesos (valores numéricos) a los atributos representando su importancia relativa ; multiplicando el valor o utilidad de la alternativa para cada atributo por el peso de tal criterio y sumando dichos productos se llega a la utilidad de la alternativa.

La forma básica para este modelo es

$$U(Ai)= \sum_{1<=j<=m} w_j U(A_{ji}) \quad (i=1,2,...,n;\ j=1,2,...,m)$$

w_j denota la importancia relativa o peso para el j-ésimo atributo

Para ejemplificar este modelo consideremos el atributo de tiempo como el más importante, en seguida el rendimiento, el costo y la vulnerabilidad, con ésto tenemos que $w_1=50$, $w_2=20$, $w_3=15$ y $w_4=35$.

La utilidad total para A1 es:

U(A1)= 50(65)+20(80)+15(55)+35(60)= 7,775.

U(A2)= 8,050 y U(A3)= 8,400.

Como la utilidad más grande es la de la alternativa A3, se decide escoger dicha alternativa.

1.2.7 MODELOS DE UTILIDAD CONFIGURAL (NO LINEAL)

Estos modelos permiten manejar funciones más complejas que el modelo aditivo lineal; por ejemplo, si consideramos dos atributos y los valores de los atributos interactúan, tendríamos el siguiente modelo configural.

$$U(Ai) = w_1 U(A_{1i}) + w_2 U(A_{2i}) + w_3 U(A_{1i}, A_{2i})$$

donde

$U(A_{ji})$ es la utilidad de la alternativa i bajo el criterio j.

entonces

$$U(Ai) = U_{1i} + U_{2i} + U_{1i,2i}$$

donde

U_{1i} y U_{2i} son las componentes de utilidad del primero y segundo valores del atributo y $U_{1i,2i}$ es la componente de utilidad que tiene en cuenta el efecto de interacción. Dicho efecto implica que las utilidades no pueden separarse para cada atributo individual. Por ejemplo, el tiempo de entrega más rápido podría no ser valorado independientemente del costo, rendimiento y vulnerabilidad. Con ello, un modelo aditivo lineal podría llegar a ser no realista y por lo tanto, conducir a conclusiones confusas.

Es así como los atributos a menudo no pueden considerarse separadamente y sumarse como en $U(Ai) = \sum_{1 \le j \le m} w_j U(A_{ji})$

Otro modelo de utilidad configural podría ser un modelo donde la utilidad total se determine por el producto de todos los valores de los atributos.

Considerando la tabla del ejemplo 2 se tiene que:

U(A1) = 65 x 80 x 55 x 60 = 17,160,000
U(A2) = 75 x 70 x 30 x 70 = 11,025,000
U(A3) = 85 x 60 x 80 x 50 = 20,400,000

Así, la alternativa A3 se escogería por tener la utilidad más alta.

MODELOS DE REPRESENTACION ESPACIAL

Como se mencionó anteriormente, la teoría de estos modelos es compleja en relación con la de los modelos compensatorios y no compensatorios es por esto que no se encuentran incluídos en dicha clasificación, sin embargo, no dejan de existir algunas similitudes como las que se mencionan en Keeney y Raiffa, 1973; MacCrimmon, 1973.

Existen dos tipos de estos modelos:

1) Curvas de indiferencia y

2) Escalamiento multidimensional.

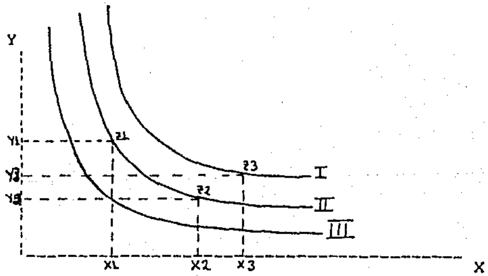
1.2.8 CURVAS DE INDIFERENCIA

Uno de los modelos que muestran las combinaciones de los valores de las alternativas que son igualmente preferidas es el de las curvas de indiferencia.

El proceso para construir estas curvas de indiferencia es el que tradicionalmente se sigue en los modelos de Economía por lo que los párrafos siguientes solo se ocuparán de la aplicación de éstas en la toma de decisiones.

Las alternativas que deben evaluarse son puntos de estas curvas de indiferencia e identificando la curva de indiferencia en la cual ellas descansan, se puede generar un ordenamiento completo entre las alternativas.

Para ilustrar su uso, supongamos que se cuenta con las curvas de indiferencia para un modelo y se tiene el problema de la elección entre tres alternativas z1, z2 y z3 bajo los criterios X y Y, se grafican sobre éstas los puntos z1=(x1,y1), z2=(x2,y2), z3=(x3,y3) donde xi y yi son los valores de la alternativa i con respecto al atributo X y Y respectivamente como se muestra en la siguiente gráfica:



Para tomar una decisión, sabemos que una curva de indiferencia superior se prefiere a una inferior. En la gráfica anterior, se observa que I>II>III donde ">" significa "se prefiere a ". Asi la alternativa z3 es la que el tomador de decisiones prefiere. Como z1 y z2 se encuentran en la misma curva de indiferencia, entonces decimos que dichas alternativas son indiferentes para el tomador de decisiones.

En teoría, si hay n atributos se pueden utilizar hiperplanos de indiferencia de dimensiones (n-1) para representar preferencias. En la práctica usualmente sólo dos atributos son los que puede manejar el tomador de decisiones.

En esencia, este procedimiento es simplemente una forma gráfica de realizar intercambios.

1.2.9 ESCALAMIENTO MULTIDIMENSIONAL

Este modelo además de ocuparse en conjuntos de alternativas que en ciertos casos tienen valores de atributos vagos o desconocidos, se caracteriza por tener especificamente:

1. Un procedimiento para obtener juicios entre atributos o un juicio agregado.

2. La construcción de una representación espacial.

3. La identificación de configuraciones ideales y la regla de elección basados en las alternativas de proximidad de estas configuraciones ideales.

La representación espacial multidimensional se construye a través de la clasificación de proximidades de pares de alternativas. Dichas alternativas se representan por puntos. Los puntos que estan más cercanos son aproximadamente de igual preferencia. El tomador de decisiones localiza su alternativa ideal en este espacio y la distancia del punto ideal al punto real es medida para clasificar asi las alternativas en orden de preferencia, utilizando la medida euclidiana o cualquier otra que se defina.

Debido a que las técnicas utilizadas en este modelo requieren de mayor atención por su gran extensión (Green and Wind (1973) se omiten en este trabajo.

MODELOS DE PROGRAMACION MATEMATICA

La teoría de estos modelos es compleja en relación con la de los modelos compensatorios y no compensatorios es por esto que no se encuentran incluidos en dicha clasificación, sin embargo, no dejan de existir algunas similitudes como las que se mencionan en Keeney y Raiffa, 1973; MacCrimmon, 1973.

Estos modelos se caracterizan por tener:

1. Un conjunto grande o infinito de alternativas.
2. Un conjunto de restricciones de preferencia.
3. Una función objetivo la cual es compensatoria.
4. Un algoritmo que converge a una solución óptima (por ejemplo el método simplex de programación lineal).

Los tres tipos básicos de modelos de programación matemática, utilizados en la toma de decisiones bajo criterios múltiples, son:

1. Programación lineal (entera o no entera).
2. Programación no lineal.
3. Programación de metas.

1.2.10 PROGRAMACION LINEAL O NO LINEAL

Un modelo de programación lineal (no lineal) puede mirarse como un modelo de decisión de atributos múltiples en el siguiente sentido:

Las variables están definidas por los atributos, las restricciones están definidas por combinaciones de atributos y hay una funcion objetivo compensatoria lineal (no lineal).

El objetivo es maximizar o minimizar dicha función sujeta a las restricciones.

Este modelo utiliza un conjunto infinito y factible de alternativas implicitamente definido por el conjunto de restricciones mientras que en los modelos anteriores hay una lista pequeña explícita de alternativas.

Dentro de la programación lineal existen métodos eficientes para determinar una decisión óptima, (una estrategia óptima) escogida de un gran número de decisiones posibles. La decisión óptima es la que satisface un objetivo sujeto a varias restricciones. Por ejemplo, una decisión óptima podría ser la decisión que produzca la más alta o máxima utilidad o el más bajo o mínimo costo.

A continuación se expone un problema:

Ejemplo 1.

La Fábrica de Muebles Tennessee, es especialista en la producción de dos clases de comedores muy de moda en Norteamérica. Cada comedor requiere de una cantidad de tiempo diferente para la construcción y para la pintura. El dueño de dicha fábrica desea determinar el número de unidades de cada tipo de comedor a producir diariamente, de tal manera que las utilidades producidas sean máximas considerando que se logra una utilidad de $200 y $240 de la venta de un comedor Virginia y uno Massachusetts respectivamente. Además, se ha experimentado una alta demanda de ambos comedores; en consecuencia, el gerente general cree que puede vender todos los comedores que produzca. Los comedores requieren tiempo de proceso en construcción y de pintura. Los requerimientos y capacidades de producción diarios están resumidos en la siguiente tabla:

Recursos requeridos para producir 1 unidad	PRODUCTO Virginia	Massachusetts	Recursos disponibles (capacidad)
Tiempo de construcción (hrs.)	6	12	120
Tiempo de pintura (hrs.)	8	4	64
Utilidad unitaria	\$200	\$240	

Para formular un problema como uno de programación lineal es necesario:

1. Función objetivo. Debe haber un objetivo (o meta) que se desee alcanzar. Por ejemplo, maximizar las utilidades, minimizar el costo, maximizar el potencial de clientes esperados, minimizar el tiempo total, etc.

2. Restricciones y decisiones. Debe haber un conjunto de alternativas o decisiones que permitan alcanzar el objetivo.

3. La función objetivo y las restricciones son lineales. Debemos estar en posibilidad de expresar las decisiones del problema, incorporándolas a la función objetivo y a las restricciones sobre decisiones, usando solamente ecuaciones lineales o desigualdades lineales. Es decir, debemos estar en posibilidad de formular el problema como un modelo de programación lineal.

Para construir el modelo de decisión del ejemplo 1, primero tenemos que definir las variables de decisión. Estas son:

$x1$ = número de comedores Virginia a producir diariamente.
$x2$ = número de comedores Massachusetts a producir diariamente.

Luego tenemos que definir el objetivo o meta de la fábrica de muebles considerada, en términos de sus variables de decisión. Para hacer esto introducimos el concepto de función objetivo. Esta función muestra la relación existente entre la producción total, $x1$ y $x2$ y la utilidad. Sea z = utilidad. Entonces:

$\$200x1$ = utilidad total por la venta de $x1$ comedores Virginia.
$\$240x2$ = utilidad total por la venta de $x2$ comedores Massachusetts.

por lo tanto,

utilidad = z = $200x1 + $240x2 (función objetivo)

El objetivo es escoger los valores de x1 y x2 tales que maximicen la utilidad, es decir:

max z = 200x1 + 240x2

En seguida, se definen las restricciones de capacidad usando x1 y x2. El tiempo usado por día fabricando los dos productos no puede exceder el tiempo total disponible en el procesamiento de los departamentos de construcción y pintura. Por lo tanto:

Para el departamento de construcción se tiene la siguiente restricción:

$$6x1 + 12x2 <= 120$$

y para el departamento de pintura

$$8x1 + 4x2 <= 64.$$

Por último, es necesario restringir todas las variables para que sean no negativas, lo que implica que:

$$x1 >= 0 \quad y \quad x2 >= 0.$$

De esta manera, el modelo de programación lineal para la fábrica de muebles Tennessee puede ser resumido como sigue:

max z = 200x1 + 240x2 (función objetivo)
sujeto a:

$$\begin{aligned} 6x1 + 12x2 &<= 120 \\ 8x1 + 4x2 &<= 64 \qquad \text{(restricciones)} \\ x1,x2 &>= 0 \end{aligned}$$

Aplicando el método simplex a este modelo, se obtiene la siguiente solución óptima:

$$x1 = 4 \quad y \quad x2 = 8$$

Esta solución implica que si se producen cuatro comedores Virginia y ocho Massachusetts diariamente, se alcanza una utilidad de $2720 la cual es la máxima.

Si vemos este problema como un modelo de decisión de atributos múltiples, los criterios son dos:

Criterio 1 = no. de comedores Virginia a producir diariamente.
Criterio 2 = no. de comedores Massachusetts a producir diariamente.

Las diferentes combinaciones de número de unidades de cada tipo de comedor a fabricar que cumplen con las siguientes restricciones:

$$6x1 + 12x2 <= 20$$
$$8x1 + 4x2 <= 64$$
$$x1, x2 >= 0$$

donde

x1 = no. de comedores Virginia a producir diariamente.
x2 = no. de comedores Massachusetts a producir diariamente.

forman el conjunto de alternativas; así, una alternativa sería producir únicamente 10 comedores Massachusetts ya que cumple con las restricciones anteriores.

El objetivo es encontrar la mejor alternativa, en este caso la mejor combinación de los valores de x1 y x2, de tal manera que maximicen la utilidad.

Ejemplo 2.

El departamento de publicidad de Almacenes Hawaii y Cia. tiene que planear para el próximo mes una estrategia de publicidad para el lanzamiento de una línea de televisores a color. Tienen en consideración dos medios diferentes:

HDAL TV-Honolulu
El Honolulu Times.

Los estudios de mercado han mostrado que:

1. La publicidad por TV llega al 2% de las familias de ingresos altos y al 3% de las familias de ingresos medios por comercial en Hawaii.

2. La publicidad en el periódico llega al 3% de las familias de ingresos altos y al 6% de las familias de ingresos medios por anuncio en Hawaii.

La publicidad en el periódico tiene un costo de $500 por anuncio y la publicidad por TV tiene un costo de $2,000 por comercial. La meta del departamento de publicidad de Almacenes Hawaii es obtener al menos una presentación como mínimo al 36% de las familias de ingresos altos y al 60% de las familias de ingresos medios minimizando los costos en publicidad.

Definido el problema se construye un problema de programación lineal posteriormente:

Las variables de decisión son:

x1 = número de anuncios en el periódico
x2 = número de anuncios comerciales en TV

El objetivo es el de escoger x1 y x2 de tal manera que minimicen los costos totales en publicidad, es decir:

$$\min Z = \$500x1 + \$2000x2$$

Las restricciones o las metas son:

$$3\%x1 + 2\%x2 \geq 36 \quad \text{(familias de ingresos altos)}$$
$$6\%x1 + 3\%x2 \geq 60 \quad \text{(familias de ingresos medios)}$$

y por último:

$$x1, x2 \geq 0$$

Resumiendo se tiene el siguiente problema de programación lineal:

$$\begin{aligned} \min z &= 500x1 + 2000x2 \\ \text{s.a.} & \\ 3x1 + 2x2 &\geq 36 \\ 6x1 + 3x2 &\geq 60 \\ x1, x2 &\geq 0 \end{aligned}$$

Aplicando el método simplex se llega a: x1 = 12 y x2 = 0. Lo que implica que para lograr el objetivo del departamento de publicidad de Almacenes Hawaii, es necesario únicamente emitir 12 anuncios en el periódico.

Observando el problema como uno de decisión bajo criterios múltiples, los criterios son:

Criterio 1 = no. de anuncios en el periódico.
Criterio 2 = no. de anuncios comerciales en T.V.

Las estrategias o alternativas de publicidad dependen de las diferentes combinaciones del número de anuncios en el periódico y en la T.V. que cumplen:

$$3x1 + 2x2 \geq 36$$
$$6x1 + 3x2 \geq 60$$
$$x1, x2 \geq 0$$

donde

x1 = no. de anuncios en el periódico.
x2 = no. de anuncios comerciales en T.V.

Por lo tanto, una alternativa es el presentar únicamente 12 anuncios en el periódico, otra alternativa sería el presentar 18 anuncios por T.V. y uno en el periódico.

La mejor alternativa es aquella que minimiza los costos.

En el siguiente capítulo se hablará sobre la descripción y las características del modelo programación de metas.

CAPITULO II

El modelo de programación de metas está muy relacionado con el modelo de programación lineal, de hecho el método utilizado es un método simplex modificado.

A través de él, se pueden solucionar problemas de programación lineal pero no reciprocamente, es decir, a través de la programación lineal no se pueden resolver problemas de programación de metas; ésto se debe a que el primero consta de una función objetivo ya sea minimizar costos o maximizar utilidades y su propósito es encontrar una solución que optimice el objetivo, en cambio, el segundo consta de una o varias funciones metas tales como: minimizar costos, maximizar utilidades, minimizar el tiempo extra de trabajo y maximizar la capacidad de trabajo. El propósito de este método y en general el de los modelos de criterios múltiples es proporcionar una solución que satisfaga los objetivos múltiples. Pero, ¿ cómo lograr ésto teniendo varias metas ?

Esto se lleva a cabo manejando los objetivos o metas como restricciones e introduciendo en cada una de ellas una variable de desviación que mide la cercania o lejania a nuestra meta. Por tanto, la función objetivo será minimizar dichas desviaciones las cuales tienen un peso o prioridad. Estos pesos sirven para encontrar una solución que satisfaga lo más cercanamente posible la meta con mayor peso antes de considerar las demás metas de menor peso. Puede notarse una gran diferencia con el modelo de programación lineal donde se busca una solución óptima de un conjunto de soluciones factibles.

Como se dijo anteriormente, en un problema de programación lineal se busca minimizar o maximizar la función objetivo mientras que en uno de programación de metas siempre será minimizar la función objetivo ya que con ello se minimizan las desviaciones de las metas de acuerdo a la jerarquia de pesos.

Además, la programación de metas nos proporciona mayor información que la programación lineal, como veremos durante el desarrollo de este capitulo.

Ejemplo:

Supongamos que una empresa ha utilizado la programación lineal para maximizar las utilidades en la fabricación de sus productos. Además, la compañía tiene el compromiso de surtir un pedido de 50 unidades de un determinado articulo y su meta es cumplir con ese compromiso.

En un planteamiento de programación lineal se especificaria la meta como una restricción y si existiera suficiente capacidad de producción el modelo de programación lineal obtendría una solución. Sin embargo, si la capacidad fuera insuficiente, se obtendría una solución no factible.

Al aplicar el modelo de programación de metas a este problema, obtendriamos una solución independiente de la capacidad. Si existiera suficiente capacidad, la solución de programación de metas sería la misma que la de programación lineal, pero si la capacidad fuera insuficiente, las soluciones serian diferentes. La solución de la programación de metas dependeria del orden de prioridades, el costo del tiempo extra y las utilidades, pero se proporcionaria una solución. A partir de esta solución, se determinaria el tiempo extra requerido y las utilidades a las que se renunciaria para lograr la meta de satisfacer al cliente.

PROGRAMACION DE METAS

El modelo de programación de metas se caracteriza por:

1.- Un conjunto grande o infinito de alternativas.

2.- Un conjunto de objetivos o metas que serán manejados como restricciones con sus respectivas variables de desviación.

3.- Una función objetivo a minimizar formada por la suma de las variables de desviación con sus respectivos coeficientes de prioridad y pesos relativos.

4.- Un algoritmo que proporciona una solución y satisface los objetivos múltiples.

Suponiendo que existen m metas, p restricciones en donde son involucradas variables de desviación , n variables de decisión y K niveles de prioridad, el modelo general puede expresarse de la siguiente manera:

$$\min z = \sum_{1 <= k <= K} P_k \sum_{1 <= i <= m} (W^+_{i,k} d^+_i + W^-_{i,k} d^-_i)$$

$$\text{s.a.} \quad \sum_{1 <= j <= n} a_{ij} x_j + d^-_i - d^+_i = b_i, \quad i = 1,,,m$$

$$\sum_{1 <= j <= n} a_{ij} x_j \quad (<= \ = \ >=) \ b_i, \quad i = m + 1,,,m + p$$

$$x_j, d^+_i, d^-_i >= 0, \quad j = 1,,,n; \ i = 1,,,m$$

en donde

P_k = coeficiente de prioridad para la k-ésima prioridad

$W^+_{i,k}$ = peso relativo de la variable d^+_i en el k-ésimo nivel de prioridad

$W^-_{i,k}$ = peso relativo de la variable d^-_i en el k-ésimo nivel de prioridad

Dentro de la programación de metas son considerados varios modelos que cumplen con estas características. Las diferencias que esencialmente existen entre ellos es en la definición de la función objetivo. Unos contemplan a las metas con o sin prioridades, otros de ellos con prioridades y ponderaciones.

En los párrafos posteriores nos ocuparemos de la descripción de estos modelos y para comprender mejor el planteamiento anterior iniciaremos con un problema que tiene una sola meta; se planteará como problema de programación lineal y a continuación como uno de programación de metas.

Ejemplo 1.

Una compañía fabrica y comercializa aparatos novedosos de video. La compañía está organizada con base en centros de utilidades, es decir, se determina el desempeño de cada centro operativo de la compañía a través de las utilidades semanales que genera.

El centro de utilidades de tableros de circuitos los fabrica de dos clases que se utilizan en diversos productos finales que manufactura la compañía. Se requieren 15 minutos para fabricar el tablero de circuito número 1; 24 minutos para el número 2. Las horas normales de operación para el centro son 240 horas semanales. Las utilidades para los tableros son \$4 para el tablero número 1 y \$5 para el 2.

Además se tienen que cumplir ciertos pedidos ya comprometidos que es el de producir 100 unidades del tablero 1 y 150 unidades del 2. El gerente del centro de utilidades desea saber cuántos tableros debe producir a la semana para maximizar las utilidades.

El planteamiento del problema como un problema de programación lineal es el siguiente:

$$
\begin{aligned}
&\max z = 4x1 + 5x2 \\
&\text{s.a.} \quad 15x1 + 24x2 \le 240(60) = 14400 \\
&\qquad x1 \ge 100 \\
&\qquad x2 \ge 150 \\
&\qquad x1, x2 \ge 0
\end{aligned}
$$

donde

x1 = # de tableros del no. 1 a producir en la semana.

x2 = # de tableros del no. 2 a producir en la semana.

En seguida se plantea el problema como uno de programación de metas:

$$\min z = d^-$$

s.a.

$$
\begin{array}{lcl}
4x1 + 5x2 + d^- - d^+ & = & 4000 \\
15x1 + 24x2 & <= & 14400 \\
x1 & >= & 100 \\
x2 & >= & 150 \\
x1,\ x2,\ d^-,\ d^+ & >= & 0
\end{array}
$$

donde

x1 = # de tableros del no. 1 a producir a la semana.

x2 = # de tableros del no. 2 a producir a la semana.

d^- = cantidad en la que no se alcanza una utilidad de $4000

d^+ = cantidad en la que se supera una utilidad de $4000

Hay que hacer notar que se escogió arbitrariamente una utilidad de $4000 pues se considera una utilidad alta.

La primera restricción es en realidad la función meta; las dos variables d^- y d^+ son las variables de desviación; el fin es minimizar d^- , o dicho de otra manera, tratar de que d^- sea cero para que la utilidad sea máxima ya que $4000 es una utilidad suficientemente alta para este problema.

Si se quisiera una utilidad exactamente igual a $4000 entonces la función meta cambiaría a minimizar la suma d^- y d^+ , es decir, tratar de que ambas fueran cero, en otro caso alguna de las dos debe ser distinta de cero como en el ejemplo 1 ya que es imposible que al mismo tiempo se supere y no se alcance la utilidad de $4000.

En caso de que por alguna causa se quisiera una utilidad menor de $4000 entonces la ecuación meta sería minimizar d^+ .

Este problema de programación de metas se puede resolver como un problema de programación lineal y obtener una solución con el método simplex.

Se resolverá el ejemplo 1 a través del método de doble fase.

C_b	x_b	x1	x2	x3	x4	x5	d⁻	d⁺	w1	w2	ctes.	coc.
	c_i	0	0	0	0	0	0	0	1	1		
0	d⁻	4	5	0	0	0	1	-1	0	0	4000	1000
0	x3	15	24	1	0	0	0	0	0	0	14400	960
1	w1	1	0	0	-1	0	0	0	1	0	100	100
1	w2	0	1	0	0	-1	0	0	0	1	150	-
	c	-1	-1	0	1	1	0	0	0	0	-z=-250	
0	d⁻	0	5	0	4	0	1	-1	-4	0	3600	720
0	x3	0	24	1	15	0	0	0	-15	0	12900	537.5
0	x1	1	0	0	-1	0	0	0	1	0	100	-
1	w2	0	1	0	0	-1	0	0	0	1	150	150
	c	0	-1	0	0	1	0	0	1	0	-z=-150	
0	d⁻	0	0	0	4	5	1	-1	-4	-5	2850	
0	x3	0	0	1	15	24	0	0	-15	-24	9300	
0	x1	1	0	0	-1	0	0	0	1	0	100	
0	x2	0	1	0	0	-1	0	0	0	1	150	
	c	0	0	0	0	0	0	0	0	0	-z = 0	

Fase Dos

C_b	x_b	x1	x2	x3	x4	x5	d⁻	d⁺	ctes.	coc.
	c_i	0	0	0	0	0	1	0		
1	d⁻	0	0	0	4	5	1	-1	2850	712.5
0	x3	0	0	1	15	24	0	0	9300	620
0	x1	1	0	0	-1	0	0	0	100	-
0	x2	0	1	0	0	-1	0	0	150	-
	c	0	0	0	-4	-5	0	1	-z =-2850	
1	d⁻	0	0	-4/15	0	-7/5	1	-1	370	
0	x4	0	0	1/15	1	8/5	0	0	620	
0	x1	1	0	1/15	0	8/5	0	0	720	
0	x2	0	1	0	0	-1	0	0	150	-
	c	0	0	4/5	0	7/5	0	1	-z =-370	

$x1=720, x2=150, d^-=370, d^+=0, x3=0, x4=620$ y $x5=0$, por lo tanto se tienen que fabricar 720 tableros no. 1, 150 tableros no. 2 y la tilidad máxima es: 4000-370=$3630 semanales.

Veamos pues los modelos de programación de metas:

Existen tres tipos de modelos

a) de metas múltiples y sin prioridades

b) de metas múltiples y con prioridades

c) de metas múltiples con prioridades y ponderaciones

El más útil de estos planteamientos es el tercero; sin embargo, puede comprenderse mejor el concepto de las prioridades y las ponderaciones examinando los tres tipos de modelos.

2.1 MODELO DE METAS MULTIPLES Y SIN PRIORIDADES

El tipo de problemas que soluciona este modelo son aquellos problemas de programación lineal que constan de varias funciones metas las cuales se manejan como restricciones con variables de desviación.

Al minimizar la función objetivo de dichos problemas se minimiza la suma de las variables de desviación considerando a las metas sin prioridades, es decir, todas las metas tienen la misma importancia. Esto hace que este modelo sea el más sencillo.

El fin del modelo es encontrar una solución que satisfaga los objetivos lmúltiples y como las variables de desviación en la función objetivo carecen de prioridades se puede utilizar el método simplex para encontrar la solución.

El modelo de metas múltiples y sin prioridades se caracteriza por:

1. Un conjunto grande o infinito de alternativas.
2. Un conjunto de metas con las mismas prioridades.
3. Una función objetivo a minimizar.
4. Un algoritmo que proporcione una solución la cual satisfará los objetivos múltiples. El algoritmo simplex o el algoritmo simplex modificado pueden proporcionar dicha solución.

Suponiendo que existen m metas, p restricciones (en donde se involucran variables de desviación) y n variables de decisión, el modelo general puede expresarse de la siguiente manera:

$$\min z = \sum_{1 <= i <= m} (d^+_i + d^-_i)$$

$$\text{s.a.} \quad \sum_{1 <= j <= n} a_{ij}x_j + d^-_i - d^+_i = b_i, \quad i = 1,,,m$$

$$\sum_{1 <= j <= n} a_{ij}x_j \quad (<= \; = \; >=) \; b_i, \quad i = m + 1,,,m + p$$

$$x_j, d^+_i, d^-_i >= 0, \quad j = 1,,,n; \; i = 1,,,m$$

A continuación se muestra un ejemplo que se plantea como un modelo de programación de metas múltiples sin prioridades y que por la solución obtenida hace notar la importancia de las prioridades para las metas.

Ejemplo 2.

Cierta compañía de pintura se especializa en la fabricación de dos tipos de pinturas de exteriores: una soluble al agua y otra de tipo barniz. La fabricación de cada 100 galones de pintura soluble al agua, requiere 10 hrs. de mano de obra mientras que cada 100 galones de barniz requieren 15 hrs. de mano de obra.

Existen disponibles 40 hrs. de mano de obra por semana. No existe disponible ayuda adicional y no se utiliza tiempo extra. Ambas pinturas producen utilidades de $1.00 por galón.

El propietario de la compañia desea evitar las operaciones en tiempo extra y a la vez alcanzar utilidades semanales de $1000.

Para plantear el problema fijémonos en las metas.

Para que no exista tiempo extra nuestra restricción es:

$$10x1 + 15x2 + d_1^- - d_1^+ = 40, \quad \text{se quiere que } d_1^+ = 0$$

donde

x1= cientos de galones que deben fabricarse de pintura soluble al agua.

x2= cientos de galones que deben fabricarse de pintura tipo barniz.

d_1^+ = número de horas en las que el tiempo extra excede 40 hrs.

d_1^- = número de horas en las que el tiempo extra no al alcanza las 40 hrs.

Para obtener una utilidad de \$1000 semanales la restricción es:

$$100x1 + 100x2 + d^-_2 - d^+_2 = 1000 \quad \text{se quiere que } d^-_2 = 0$$

donde

d^+_2 = cantidad en pesos que sobrepasan una utilidad de \$1000 semanales.

d^-_2 = cantidad en pesos que no alcanzan una utilidad de \$1000 semanales.

El problema es:

$$\min z = d^-_1 + d^+_2$$

$$\begin{aligned} \text{s.a.} \quad & 10x1 + 15x2 + d^-_1 - d^+_1 = 40 \\ & 100x1 + 100x2 + d^-_2 - d^+_2 = 1000 \\ & x1, x2, d^-_1, d^+_1, d^-_2, d^+_2 \geq 0 \end{aligned}$$

Como las desviaciones de la función objetivo carecen de prioridades podemos utilizar el método simplex para encontrar una solución.

Observemos la última tabla resultante de aplicar el método simplex:

		$x1$	$x2$	d^-_1	d^+_1	d^-_2	d^+_2	
0	$x1$	1	1	0	0	1/100	-1/100	10
1	d^+_1	0	-5	-1	1	1/10	-1/10	60
	c	0	5	1	0	9/10	1/10	-z=-60

Obtenemos entonces que:

$$x1=10,\ x2=0,\ d^+_1=60,\ d^-_1=0,\ d^+_2=0,\ d^-_2=0;$$

Dicha solución implica que unicamente se puede alcanzar la utilidad de $1000 semanales con un tiempo extra de 100 hrs. y produciendo 10 galones de pintura soluble al agua.

Como puede observarse no se logró una de las metas que fue el evitar el tiempo extra, esto se debe a que no existen prioridades para las metas y por el contrario sólo se intentó minimizar la suma de las desviaciones para todas las metas.

2.2 MODELO DE METAS MULTIPLES CON PRIORIDADES

Cuando se tienen metas múltiples, lo más probable es que se cuente con alguna escala de prioridades para ellas. La programación de metas contempla el orden preferencial de las metas a través del uso de coeficientes de prioridad denotados como P.

A todas las variables de desviación involucradas en la meta de prioridad más alta, la primera, se le asigna un valor P1 en la función objetivo, a las metas que tienen la segunda prioridad se les asigna un valor P2 y se continúa este proceso hasta jerarquizar todas las metas.

Los coeficientes P1,P2,P3,,,etc. no toman valores numéricos, simplemente representan niveles de prioridad ($P1>P2>P3>..$). Dado que los coeficientes de prioridad aparecen en la función objetivo, no puede utilizarse el algoritmo simplex usual para resolver estos problemas. Sin embargo, puede aplicarse ese algoritmo para que maneje los coeficientes de prioridad y asegurar que se minimizan las desviaciones de las metas de más alta prioridad antes de examinar las de prioridad más baja en forma sucesiva.

Las características de un modelo de programación de metas con prioridades son:

1. Un conjunto grande o infinito de alternativas.
2. Un conjunto de metas, cada una de ellas con su prioridad o peso y con sus respectivas variables de desviación para ser manejadas como restricciones.
3. Una función objetivo en donde se involucran las prioridades o pesos de las metas y al minimizar dicha función se minimizan las variables de desviación de las restricciones.
4. Un algoritmo que proporcione una solución la cual satisfaga los objetivos múltiples.

Suponiendo que existen m metas, p restricciones en donde son involucradas variables de desviación , n variables de decisión y K niveles de prioridad, el modelo general puede expresarse de la siguiente manera:

$$\min z = \sum_{1 \le k \le K} P_k \sum_{1 \le i \le m} (d^+_i + d^-_i)$$

$$\text{s.a.} \quad \sum_{1 \le j \le n} a_{ij} x_j + d^-_i - d^+_i = b_i, \quad i = 1,..,m$$

$$\sum_{1 \le j \le n} a_{ij} x_j \quad (\le \; = \; \ge) \; b_i, \quad i = m+1,..,m+p$$

$$x_j, d^+_i, d^-_i \ge 0, \quad j = 1,..,n; \; i = 1,..,m$$

en donde

P_k = coeficiente de prioridad para la k-ésima prioridad

En el siguiente ejemplo, el problema se plantea como uno de programación de metas múltiples con prioridades.

Ejemplo 3.

Supóngase que la compañia T & L fabrica tres tipos diferentes de baleros que se utilizan en equipo textil. Todos los baleros se fabrican en una operación de prensado. El tiempo de fabricación que se requiere para elaborar un balero básico es de 5 hrs. en tanto que uno de alta precisión requiere 12 hrs. de tiempo de producción. El balero de aplicación general requiere 8 hrs. de tiempo de producción.

La compañia dispone de 340 hrs. semanales de producción. Las utilidades unitarias que se obtienen de la venta de baleros son: \$1000 por balero básico; \$1450 por balero de aplicación general y \$2500 por el de alta precisión.

El departamento de mercado técnico de T & L ha señalado que el comportamiento de la demanda de los baleros implica que la compañia puede vender todos los que fabrica.

Los administradores de la T & L han listado las siguientes metas (en orden de importancia):

Meta 1. Utilizar toda la capacidad de producción existente.

Meta 2. Alcanzar las metas semanales de ventas para cada tipo de balero: 20 básicos, 24 de aplicación general y 15 de alta precisión.

Meta 3. Limitar el tiempo extra a 40 hrs. por semana.

Meta 4. Maximizar las utilidades.

Planteamiento del problema:

restriccion para la meta 1:

$$5x1 + 12x2 + 8x3 + d^-_1 - d^+_1 = 340$$

donde

x1 = # de baleros básicos a fabricarsemanalmente.

x2 = # de baleros de alta precisión a fabricar semanalmente.

x3 = # de baleros de aplicación general a fabricar semanalmente.

d^+_1 = # de hrs. que sobrepasan a las 340 hrs. de capacidad de trabajo.

d^-_1 = # de hrs. que no alcanzan a las 340 hrs. de capacidad de trabajo.

Para la meta 2:

$$x1 + d^-_2 - d^+_2 = 20$$
$$x2 + d^-_3 - d^+_3 = 15$$
$$x3 + d^-_4 - d^+_4 = 24$$

donde

d^+_2 = # de baleros básicos cuya venta sobrepasa a las 20 unidades.

d_2^- = # de baleros básicos cuya venta no alcanza 20 unidades.

d_3^+ = # de baleros de alta precisión cuya venta excede 15 baleros.

d_3^- = # de baleros de alta precisión cuya venta no alcanza 15 baleros.

d_4^+ = # de baleros de aplicación general cuya venta excede 24 baleros.

d_4^- = # de baleros de aplicación general cuya venta no alcanza 24 unidades.

Para la meta 3:

$$d_1^+ + d_5^- - d_5^+ = 40$$

donde

d_5^+ = # de hrs. de tiempo extra que exceden a 40 hrs.

d_5^- = # de hrs. de tiempo extra que no alcanzan 40 hrs.

Para la meta 4:

$$1000x1 + 1450x3 + 2500x2 + d^-_6 - d^+_6 = 100000$$

donde

d^+_6 = cantidad en pesos que sobrepasan a los $100000 en utilidades.

d^-_6 = cantidad en pesos que no alcanzan una utilidad de $100000.

Por lo tanto el problema es:

$$\min z = P1d^-_1 + P1d^+_1 + P2d^-_2 + P2d^-_3 + P2d^-_4 + P3d^+_5 + P4d^-_6$$

$$\begin{array}{lrl} \text{s.a.} & 5x1 + 12x2 + 8x3 + d^-_1 - d^+_1 & = 340 \\ & x1 + d^-_2 - d^+_2 & = 20 \\ & x2 + d^-_3 - d^+_3 & = 15 \\ & x3 + d^-_4 - d^+_4 & = 24 \\ & d^+_1 + d^-_5 - d^+_5 & = 40 \\ & 1000x1 + 2500x2 + 1450x3 + d^-_6 - d^+_6 & = 100000 \\ & x1, x2, x3, d^-_1, d^+_1, d^-_2, d^+_2, d^-_3, d^+_3, d^-_4, d^+_4, d^-_5, d^+_5, d^-_6, d^+_6 & >= 0 \end{array}$$

Este tipo de problemas se resuelve a través del algoritmo simplex de programación de metas que se verá posteriormente; sin embargo, también se puede resolver por medio del algoritmo simplex de programación lineal proporcionando un valor numérico a los coeficientes de prioridad; dichos valores dependen de la jerarquía de metas. A P1 le corresponde un valor mayor que a P2 , a P2 uno mayor que a P3 y así sucesivamente.

2.3 MODELO DE METAS MULTIPLES CON PRIORIDADES Y PONDERACIONES

Los problemas planteados anteriormente en este capitulo, tienen metas múltiples y el mismo índice de prioridades. En ocasiones, es deseable asignar a las metas que tienen la misma prioridad mayor importancia que a otras. Para reflejar la diferencia en importancia dentro del mismo nivel de prioridad se utiliza un peso diferencial.

Suponga por ejemplo, que las utilidades y el tiempo de un problema determinado tienen el mismo nivel de prioridad. Si no se asignan pesos o ponderaciones quien toma las decisiones está planteando que una desviación de $1 de la meta de utilidades tiene igual importancia que una hora de tiempo extra. Si esto no es veradero, entonces pueden asignarse pesos y así si el tomador de decisiones decide que 6 horas de tiempo extra equivalen a $1 de utilidades, entonces se usaría una razón de 6 a 1 para los pesos.

Las características del modelo de metas múltiples con prioridades y ponderaciones se describen a continuación:

1. Un conjunto grande o infinito de alternativas.

2. Un conjunto de metas con sus respectivas variables de desviación y con prioridades.

3. Una función objetivo en donde además de especificar la prioridad de la meta correspondiente, también se especifique si alguna meta tiene mayor importancia a pesar de tener la misma prioridad que otra u otras metas.

4. Un algoritmo que proporcione una solución que satisfaga las metas múltiples.

Suponiendo que existen m metas, p restricciones en donde son involucradas variables de desviación , n variables de decisión y K niveles de prioridad, el modelo general puede expresarse de la siguiente manera:

$$\min z = \sum_{1<=k<=K} P_k \sum_{1<=i<=m} (w^{+}_{i,k} d^{+}_{i} + w^{-}_{i,k} d^{-}_{i})$$

$$\text{s.a.} \quad \sum_{1<=j<=n} a_{ij} x_j + d^{-}_{i} - d^{+}_{i} = b_i, \quad i = 1,,,m$$

$$\sum_{1<=j<=n} a_{ij} x_j \quad (<= \; = \; >=) \; b_i, \quad i = m + 1,,,m + p$$

$$x_j, d^{+}_{i}, d^{-}_{i} >= 0, \quad j = 1,,,n; \; i = 1,,,m$$

en donde

P_k = coeficiente de prioridad para la k-ésima prioridad

$W^+_{i,k}$ = peso relativo de la variable d^+_i en el k-ésimo nivel de prioridad

$W^-_{i,k}$ = peso relativo de la variable d^-_i en el k-ésimo nivel de prioridad

Para comprender mejor lo dicho anteriormente veamos un ejemplo.

Ejemplo 4.

Modifiquemos el problema anterior (ejemplo 3) de metas múltiples con prioridades a metas múltiples con prioridades y ponderaciones considerando la siguiente meta como la meta 2.

Meta 2: Alcanzar las metas semanales de ventas para cada tipo de balero: 20 básicos, 24 de aplicación general y 15 de alta precisión. Asignar pesos diferenciales de acuerdo con la utilidad relativa de cada tablero.

El planteamiento del problema es:

$$\min z = P1d_1^- + P1d_1^+ + P2d_2^- + 3P2d_3^- + 2P2d_4^- + P3d_5^- + P4d_6^-$$

$$
\begin{array}{lrrrrcl}
\text{s.a.} & 5x1 + & 12x2 + & 8x3 + & d_1^- - d_1^+ & = & 340 \\
 & x1 + & & & d_2^- - d_2^+ & = & 20 \\
 & & x2 + & & d_3^- - d_3^+ & = & 15 \\
 & & & x3 + & d_4 - d_4 & = & 24 \\
 & & & & d_1^+ + d_5^- - d_5^+ & = & 40 \\
 & 1000x1 + & 2500x2 + & 1450x3 + & d_6^- - d_6^+ & = & 100000
\end{array}
$$

$$x1, x2, x3, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+, d_4^-, d_4^+, d_5^-, d_5^+, d_6^-, d_6^+ \geq 0$$

En este ejemplo como los pesos diferenciales se asignan de acuerdo a la utilidad relativa la cual es de \$1000 por balero básico, \$1450 por balero de aplicación general y \$2500 por uno de alta precisión lo que indica que la variable de desviación d_3^- debe tener la ponderación más alta; en este caso, se le asignará el coeficiente 3 en la función objetivo. A la variable de desviación d_4^- se le asignará un coeficiente 2 pues es de más baja ponderación y a d_2^- el coeficiente 1 por tener la ponderación más baja.

Ejemplo 5.

Suponga que una persona debe limitarse a una dieta de leche, carne de res y huevos. A esa persona no se le restringe la cantidad de cualquiera de esos artículos que elija pero es importante satisfacer ciertos requerimientos mínimos y minimizar el consumo de colesterol.

La siguiente tabla refleja la cantidad de vitamina A, B y C que contiene cada uno de los productos alimenticios, así como su nivel de colesterol.

La tabla también incluye los requerimientos mínimos diarios de vitaminas y el costo de cada uno de los productos. Suponga que se han establecido las siguientes metas, listadas en orden de importancia:

Meta 1. Satisfacer los requerimientos vitaminicos diarios minimos Tiene el doble de importancia satisfacer el requerimiento de vitamina A que los requerimientos de las vitaminas B y C.

Meta 2. Minimizar el consumo de colesterol.

Meta 3. Minimizar los costos asociados con la dieta.

Componentes de los alimentos	Producto alimenticio			
	Leche (mg/gal)	Carne de res (mg/lb)	Huevos (mg/docena)	Requerimiento diario minimo (mg)
Vitamina A	2	2	20	2
Vitamina B	200	20	20	60
Vitamina C	20	200	20	10
Colesterol	140 unidades/gal	100 unidades/lb	240 unidades/docena	
Costo	\$2.00/gal	\$2.75/lb	\$1.20/docena	

Planteamiento del problema:

$$\min z = 2P1d_1^- + P1d_2^- + P1d_3^- + P2d_4^+ + P3d_5^+$$

$$\begin{aligned}
\text{s.a.}\quad 2x1 + 2x2 + 20x3 + d_1^- - d_1^+ &= 2\\
200x1 + 20x2 + 20x3 + d_2^- - d_2^+ &= 60\\
20x1 + 200x2 + 20x3 + d_3^- - d_3^+ &= 10\\
140x1 + 100x2 + 240x3 + d_4^- - d_4^+ &= 50\\
2x1 + 2.75x2 + 1.20x3 + d_5^- - d_5^+ &= 0.5\\
x1, x2, x3, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+, d_4^-, d_4^+, d_5^-, d_5^+ &>= 0
\end{aligned}$$

donde

$x1$= # de mg/gal de leche a consumir diariamente

$x2$= # de mg/lb de carne de res a consumir diariamente

$x3$= # de mg/docenas de huevos a consumir diarios.

d^+_1= cantidad (en mg.) que sobrepasa el requerimiento minimo de 2 mg de vitamina A.

d^-_1= cantidad (en mg.) faltante a los 2 mg de requerimiento minimo de vitamina A.

d^+_2= cantidad (en mg.) que sobrepasa el requerimiento minimo de 60 mg de vitamina B.

d^-_2= cantidad (en mg.) faltante a los 60 mg de requerimiento minimo de vitamina B.

d^+_3= cantidad (en mg.) que sobrepasa el requerimiento minimo de 10 mg de vitamina C.

d^-_3= cantidad (en mg.) faltante a los 10 mg de requerimiento minimo de vitamina C.

d^+_4= cantidad de colesterol que excede a 50 unidades.

d^-_4= cantidad de colesterol que no alcanźa 50 unidades.

d^+_5= cantidad (en pesos) que excede un costo de \$0.5

d^-_5= cantidad (en pesos) que no alcanza un costo de \$0.5

METODOS DE SOLUCION

Se presentarán dos métodos para resolver los problemas de programación de metas mútiples:

1) método gráfico

2) algoritmo de programación de metas: ampliación del método simplex.

2.4 METODO GRAFICO

Este método sirve para aquellos problemas de programación de metas múltiples con dos variables de decisión pues únicamente se trabaja con dos ejes para graficar las restricciones.

Para ello, hay que tomar en cuenta aquellas restricciones que no constan de variables de desviación y se grafican obteniendo una región factible.

Las restricciones que sí constan de variables de desviación, se grafican después de igualar a cero dichas variables.

Posteriormente, se encuentra la región factible que satisfaga la meta de prioridad más alta sin degradar la región factible que se tenía; es decir, permaneciendo en la región factible anterior.

Se continúa con la siguiente meta de prioridad más alta y se encuentra una solución sin degradar las soluciones anteriores; así sucesivamente hasta que ya se hallan tomado en cuenta todas las metas. De esta manera se obtiene una solución óptima que satisface las metas múltiples.

ALGORITMO PARA OBTENER UNA SOLUCION GRAFICA DE PROBLEMAS DE PROGRAMACION DE METAS.

El método gráfico que se explicará a continuación sirve para aquellos problemas de programación de metas con dos variables de decisión y consiste de cuatro pasos :

1) Graficar todas las restricciones:

Para ello, primero se grafican las restricciones que no tienen variables de desviación encontrando así la región factible, en caso de que todas las restricciones tengan variables de desviación la región factible es el cuadrante no negativo pues x1 y x2 son mayores o iguales a cero.

Para aquellas restricciones que si tienen variables de desviación, igualamos dichas variables a cero y trazamos la ecuación resultante.

2) Determinar la región factible que satisface la meta de prioridad más alta.

3) Pasar a la siguiente meta de prioridad más alta y encontrar la mejor solución sin degradar las soluciones ya alcanzadas.

4) Repetir el paso 3 hasta que se hayan tomado en cuenta todas las metas.

Ejemplo 6.

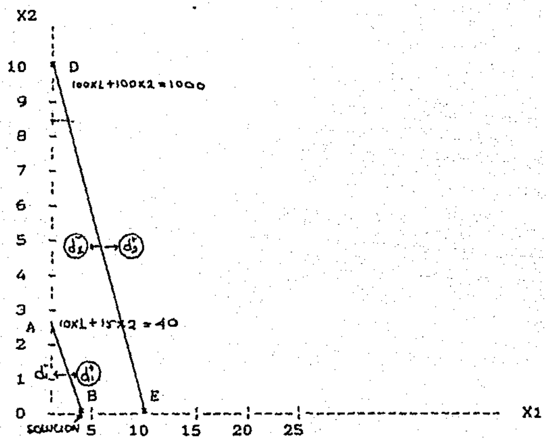
Grafiquemos el problema de la compañía de pinturas (ejemplo 2.) tomando en cuenta como primera meta el evitar el tiempo extra y como segunda meta el alcance de $1000 como utilidad semanal.

$$\min z = P1d^+_1 + P2d^-_2$$

$$\text{s.a} \quad 10x1 + 15x2 + d^-_1 - d^+_1 = 40$$

$$100x1 + 100x2 + d^-_2 - d^+_2 = 1000$$

$$x1, x2, d^-_1, d^+_1, d^-_2, d^+_2 \geq 0$$



Efectuamos el primer paso:

Como todas las restricciones tienen variables de desviación, graficamos x1,x2>=0, es decir, nuestra región es el cuadrante no-negativo.

A continuación graficamos la primera restricción haciendo $d^-_1 = d^+_1 = 0$. Así mismo graficamos la segunda restricción con

$d^-_2 = d^+_2 = 0.$

Pasamos al segundo paso:

La prioridad más alta es aquella que se refiere a la primera restricción; como nuestro fin es minimizar d^+_1 la región factible es aquella delimitada por los puntos ABO. Hemos logrado que $d^+_1 = 0$.

Pasamos a la siguiente meta de prioridad más alta que sería la segunda restricción. Si minimizamos d^-_2 entonces la región factible es aquella que está por arriba de la recta DE y delimitada por las rectas $x1 \geq 0$ y $x2 \geq 0$, esta última región no se intersecta con la región que teníamos anteriormente.

Para obtener una solución aceptable necesitamos que d^-_2 sea tan pequeña como se pueda para ello trazaremos varias rectas paralelas a la recta DE hasta tocar la región ABO, notamos que el punto de intersección es $x1=4$ y $x2=0$.

Para obtener los valores de las desviaciones se tiene que recurrir a las restricciones; sabemos que $x1=4$, $x2=0$, $d^-_1 = d^+_1 = 0$ y $d^+_2=0$, tomando la segunda restricción tenemos que

$$100(4) + 100(0) + d^-_2 - 0 = 1000$$

ésto implica que:

$$d^-_2 = 1000-400 = 600$$

Por lo tanto la solución es producir 4 galones de pintura soluble al agua que es distinta a la del ejemplo 2 ya que se están considerando ponderaciones.

Ejemplo 7.

$$\min z = P1(d_1^- + d_1^+) + P2d_2^- + P3d_3^- + 5P4d_3^+ + 3P4d_2^+$$

$$\begin{aligned} \text{s.a.} \quad x1 + x2 + d_1^- + d_1^+ &= 800 \\ 5x1 + d_2^- + d_2^+ &= 2500 \\ 3x2 + d_3^- + d_3^+ &= 1400 \\ x1, x2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+ &>= 0 \end{aligned}$$

El procedimiento para encontrar la solución a través del método gráfico es la siguiente:

Como todas las restricciones tienen variables de desviación iniciamos con la región factible $x1>=0$ y $x2>=0$.

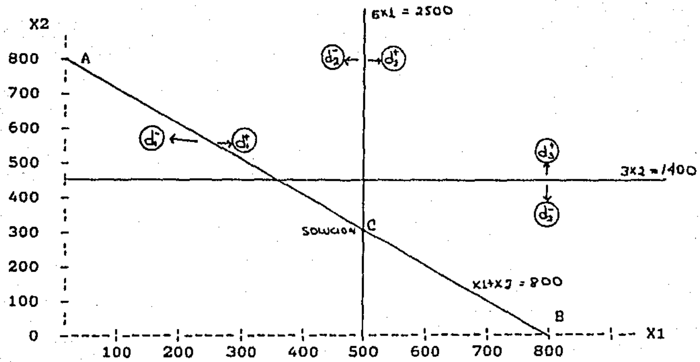
Trazamos tres rectas:

1) $x1+x2=800$

2) $5x1=2500$

3) $3x2=1400$

La primera meta es minimizar tanto d_1^- como d_1^+ lo que implica que la región factible se restringe y es precisamente la recta $x1+x2=800$ ya que es donde $d_1^-=d_1^+=0$.



A continuación nos fijamos en la segunda meta que es minimizar d_2^-, dicha región es aquella que se encuentra a la derecha de la recta 5x1=2500 incluyendo la misma.

Si intersectamos la región calculada anteriormente y ésta última tenemos que la solución buscada se encuentra a lo largo de la recta BC.

La tercera meta es minimizar d_3^-, la región que cumple con ello es el cuadrante positivo a partir de la recta 3x2=1400.

Como no existe intersección entre ésta y la anterior, trazamos paralelas a la recta 3x2=1400 hasta tocar la región que se tenia. Dicha intersección es el punto (500,300).

La cuarta meta es minimizar la suma de d_3^+ con d_2^+ pero como tiene mayor prioridad d_3^+ observemos cual es la región que se tiene tomando en cuenta dicha restricción. Notemos que el punto (500,300) cumple con $d_3^+=0$ y también con $d_2^+=0$ por lo que hemos llegado a la solución:

$x1=500,\ x2=300,\ d_1^- = d_1^+ = d_2^- = d_2^+ = d_3^+ = 0.$

Para obtener d_3^- hay que substituir los valores ya encontrados en la restricción

$$3x2+d_3^- - d_3^+ = 1400 ,$$

despejando d_3^- se tiene que:

$$d_3^- = 1400-3(300)=500.$$

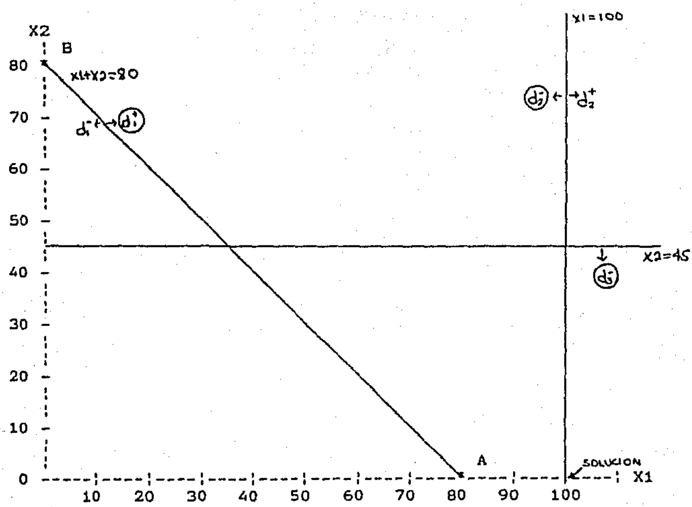
En resúmen, la solución es x1 = 500 y x2 = 300.

Ejemplo 8.

$$\text{Min } z= P1d_2^- + P2d_1^+ + P3d_3^-$$

$$\begin{aligned} \text{s.a.}\quad & x1+x2+d_1^- -d_1^+ = 80 \\ & x1 \quad +d_2^- -d_2^+ = 100 \\ & x2+d_3^- \geq 45 \\ & x1,x2,d_1^-,d_1^+,d_2^-,d_2^+,d_3^- \geq 0 \end{aligned}$$

La siguiente gráfica nos muestra la solución al problema:



La región factible al inicio del problema es x1,x2>=0.

Trazamos las rectas:

x1+x2=80

x1   =100

   x2=45

Nos fijamos en la primera restricción que es el minimizar d_2^-

en la restricción $x1+d^-_2-d^+_2=100$. La solución son aquellos puntos que se encuentran en la recta $x1=100$ y los que están a la derecha de ella.

Pasamos a la siguiente prioridad que es minimizar d^+_1 tomando en cuenta la restricción $x1+x2+d^-_1-d^+_1=80$. La región es la recta AB y la parte inferior de la misma delimitada por $x1,x2>=0$.

No existe intersección entre esta última región y la anterior por lo que trazamos paralelas a la recta AB hasta tocar la región anterior lo que implica que la solución factible es el punto (100,0). Dicha solución es fija ya que consta unicamente de un punto, por lo que la solución óptima es:

$$x1=100, x2=0, d^-_1=d^-_2=d^+_2=0, d^-_3=45 \text{ y } d^+_1=20.$$

OBSERVACION:

Con esta solución no se logra la meta 2 ni la 3. A pesar de que la solución factible $x1=100, x2=45, d^-_3=d^-_2=d^+_2=d^-_1=0$ y $d^+_1=65$ alcanza las metas 1 y 3, no es mejor que la anterior ya que por tener prioridades se tiene que minimizar d^+_1 y después alcanzar la meta 3 lo que provoca que la primera solución sea la solución óptima.

2.5 ALGORITMO DE PROGRAMACION DE METAS (método simplex modificado).

El método que se explica a continuación es una pequeña variante del simplex. El propósito de un problema con metas múltiples es minimizar las desviaciones de las metas propuestas. En este tipo de programación no existen funciones objetivos donde se minimicen costos o tiempo y se maximicen ganancias o beneficios. Los coeficientes de prioridad P1 › P2 ›...› Pn toman el lugar de los coeficientes de costos. Estos coeficientes son ordinales (no cardinales) y por lo tanto, pueden ser multidimensionales; por esta razón, no se puede utilizar el criterio del método simplex para elegir los vectores que entran a la base en cada iteración.

La tabla que se utiliza para la programación con metas múltiples tiene la siguiente estructura:

			C	
			Nombre de las variables continuas y de desviación.	
Variables en la base (vectores unitarios de A) $\rightarrow$	C_b		A	b
Coeficientes de prioridad $\rightarrow$			Matriz de valores de $z_j - C_j$ modificados para este tipo de programas.	Ø

donde

m = no. de restricciones

n = no. de variables continuas y de desviación

C = vector de coeficientes en la función objetivo de las variables continuas y de desviación

b = vector de disponibilidad de recursos

A = matriz de restricciones

$\emptyset$ = vectores de indicadores de optimalidad

C_b = vector de coeficientes de las variables básicas

Para el cálculo de la matriz $z_j - C_j$ se deba tomar en cuenta que los coeficientes de ponderación no necesariamente tienen las mismas unidades dimensionales y que por lo tanto las leyes aritméticas de suma y resta no se pueden utilizar. Las C_j representan las componentes del vector C, mientras que z_j son productos de la suma de las C_j veces "ciertas constantes o coeficientes". Es difícil dar una regla general y por lo tanto se ilustrará su obtención con un ejemplo.

Ejemplo 9.

$$\min z = P1d_1^+ + P2d_2^-$$

$$\text{s.a.} \quad 10x1 + 15x2 + d_1^- - d_1^+ = 40$$

$$100x1 + 100x2 + d_2^- - d_2^+ = 1000$$

$$x1, x2, d_1^-, d_1^+, d_2^-, d_2^+ \geq 0$$

La tabla inicial se muestra a continuación:

C_j		0	0	0	P1	P2	0		
C_b	X_b	x1	x2	d^-_1	d^+_1	d^-_2	d^+_2		
0	d^-_1	10	15	1	-1	0	0	40	4
P2	d^-_2	100	100	0	0	1	-1	1000	10
	P1	0	0	0	-1	0	0	0	$z_j - C_j$
	P2	100	100	0	0	0	-1	1000	

Sean las componenetes de $A = a_{ij}$, las de $C_b = C_{bj}$ y las de $C = C_j$, $i = 1,2,,,m$; $j = 1,2,,,n$. Entonces la columna $j = 1$ se asocia a la variable x1 con $C_1 = 0$ y la columna $z_1 - C_1$ será igual a los coeficientes del siguiente vector:

$C_{b1} \times a_{11} = 0 \times 10 = 0$
$C_{b2} \times a_{21} = P2 \times 100 = 100P2$

Por lo tanto se pone un 100 en el renglón asociado a P2, columna 1 de la matriz $z_j - C_j$. El resto de los elementos de esa columna son cero.

Para la segunda columna x2, se tiene que $C_2 = 0$ y

$C_{b2} \times a_{12} = 0 \times 15 = 0$
$C_{b2} \times a_{22} = P2 \times 100 = 100P2$

Por lo que se pone un 100 en el renglón P2 columna x2 y el resto de los elementos de esa columna son nulos.

Para la columna d^-_1 con $C_3 = 0$

$C_{b3} \times a_{13} = 0 \times 1 = 0$
$C_{b3} \times a_{23} = P2 \times 0 = 0$

Por lo tanto los elementos de $z_j - C_j$ son nulos.

Para la columna d^+_1 con $C_4 = P1$ se tiene:

$C_{b1} \times a_{14} = 0 \times (-1) = 0$
$C_{b2} \times a_{24} = P2 \times 0 = 0$

Por lo tanto $z_4 - C_4 = 0 - P1 = -P1$

Entonces, se pone un -1 en el renglón P1 columna d^+_1 y un ro en el renglón P2 de la misma columna.

Para la columna d^-_2 con C_3 = P2 se tiene:

$C_{b1} \times a_{13} = 0 \times 0 = 0$
$C_{b2} \times a_{23} = P2 \times 1 = P2$

$z_3 - C_3 = P2 - P2 = 0$

Por lo tanto se pone cero al renglón P1 y P2 de la columna d^-_2.

Para la columna $d^+_2 = 0$ se tiene:

$C_{b1} \times a_{14} = 0 \times 0 = 0$
$C_{b2} \times a_{24} = P2 \times (-1) = -P2$

$z_4 - C_4 = -P2$

por lo que se pone un -1 en el renglón P2 de la columna d^+_2 y un cero al renglón P1 de la misma columna.

Posteriormente, para determinar la variable que entra a la base, se examina el coeficiente de prioridad más alto que tenga un valor distinto de cero en el vector de indicadores de optimidad (Ø) y se escoge el mayor coeficiente de las $z_j - C_j$.

Para el ejemplo en cuestión, la tabla inicial nos indica que se ha logrado la primera meta con la solución: $x1=x2=d^+_1$, $d^+_2=0$ y $d^-_1=40$, $d^-_2=1000$.

Como en el renglón $z_j - C_j$ correspondiente a P2 y la columna tanto de x1 como de x2 existe un valor mayor que cero, esto implica que puede entrar a la base x1 o x2, escojamos x1.

Para determinar la variable que sale de la base, sean $b_1, b_2, b_3,,, b_m$ las componenetes del vector b y sea a_j la columna j de la matriz A que se ha seleccionado para entrar a la base. Entonces el vector de salida se elige como el

$$\text{Min } (b_j/a_{ij} \mid a_{ij} > 0) \quad i = 1,2,,,m$$

Para el ejemplo anterior, la variable que sale es d^-_1.

Segunda tabla:

		0	0	0	P1	P2	0	
		$x1$	$x2$	d^-_1	d^+_1	d^-_2	d^+_2	
0	$x1$	1	15/10	1/10	-1/10	0	0	4
P2	d^-_2	0	-50	-10	10	1	-1	600
	P1	0	0	0	-1	0	0	0
	P2	0	-50	-10	10	0	-1	600

Esta tabla nos indica que la segunda meta todavia no se logra y si quisieramos que entrara otra variable en la base tendria que entrar d^+_1 lo cual no nos conviene pues la primera meta es minimizar dicha variable. Todo esto implica que nuestra solución final es: $x1=4$, $d^-_2=600$, $x2=d^-_1=d^+_1=d^+_2=0$ que coincide con la determinada gráficamente.

La descripción formal del algoritmo de programación de metas es el siguiente:

ALGORITMO

El método simplex modificado consta de seis pasos:

Sea N= # de niveles de prioridad.

Paso 1. Establecer la tabla simplex inicial sin calcular el renglón de coeficientes de costos reducido (c) y fijar k=1.

Paso 2. Calcular el renglón z_j-C_j para el nivel de prioridad Pk. Si existe un cero en el vector de optimidad (Ø) entonces se ha alcanzado la meta cuya prioridad es Pk, en este caso ir al paso 6. En caso contrario ir al paso 3.

Paso 3. Identificar la variable que entra: se examina el coeficiente de prioridad más alto que tenga un valor distinto a cero en el vector de indicadores de optimalidad (Ø). En esta fila se escoge el mayor coeficiente de la z_j-C_j. Los empates se pueden decidir analizando los coeficientes de las z_j-C_j en el siguiente nivel de prioridad. De persistir el empate, se puede romper éste arbitrariamente.

Si todos los valores del renglón Pk son negativos ir al paso 6.

En caso contrario continuar con el paso 4.

Paso 4. Calcular la variable que sale a través del mismo procedimiento que se utiliza en el método simplex.

Paso 5. Calcular la nueva tabla de la manera como se calcula en el método simplex sin el renglón de coeficientes de costo reducido. Ir al paso 2.

Paso 6. Fijar k=k+1.

Si k>N terminar.

Si k<=N ir al paso 2.

Ejemplo 10.

Resolvamos el siguiente problema por el método simplex modificado y comparemos la solución con la obtenida por el método gráfico (Ejemplo 7).

$$\min z = P1(d_1^- + d_1^+) + P2d_2^- + P3d_3^- + 5P4d_3^+ + 3P4d_4^+$$

$$\begin{aligned} \text{s.a.} \quad x1 + x2 + d_1^- - d_1^+ &= 800 \\ 5x1 \quad\quad + d_2^- - d_2^+ &= 2500 \\ 3x2 + d_3^- - d_3^+ &= 1400 \\ x1, x2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+ &\geq 0 \end{aligned}$$

	0	0	P1	P1	P2	3P4	P3	5P4		
	x1	x2	d_1^-	d_1^+	d_2^-	d_2^+	d_3^-	d_3^+	ctes	coc.
P1 d_1^-	1	1	1	-1	0	0	0	0	800	800
P2 d_2^-	5	0	0	0	1	-1	0	0	2500	500
P3 d_3^-	0	3	0	0	0	0	1	-1	1400	--
P1	1	1	0	-2	0	0	0	0	800	
P1 d_1^-	0	5	5	-5	-1	1	0	0	1500	300
0 x1	1	0	0	0	1/5	-1/5	0	0	500	--
P3 d_3^-	0	3	0	0	0	0	1	-1	1400	466.6
P1	0	5	4	-6	-1	1	0	0	1500	
0 x2	0	1	1	-1	-1/5	1/5	0	0	300	
0 x1	1	0	0	0	1/5	-1/5	0	0	500	
P3 d_3^-	0	0	-3	3	3/5	-3/5	1	-1	500	
P1	0	0	-1	-1	0	0	0	0	0	
P2	0	0	0	0	-1	0	0	0	0	
P3	0	0	-3	3	3/5	-3/5	0	-1	500	
P4	0	0	0	0	0	-3	0	-5		

La última tabla nos dá la solución mas adecuada a pesar de que en el renglón $(z_j - C_j)$ correspondiente a P3 existe un valor mayor que cero tanto para la columna d_1^+ como para la columna d_2^-;

no pueden entrar dichas variables ya que ambas se tienen que minimizar como primera y segunda meta respectivamente por lo que cualquiera de ellas entra a la base estariamos rompiendo con los objetivos ya alcanzados.

La solución de mayor conveniencia que concuerda con la obtenida gráficamente es entonces:

$$x1=500,\ x2=300,\ d_3^-=500,\ d_1^-=d_1^+=d_2^-=d_2^+=d_3^+=0\ .$$

Ejemplo 11.

Para el problema:

$$\min z= P1d_2^- + P2d_1^+ + P3d_3^-$$

$$\begin{array}{ll} \text{s.a.} & x1+x2+d_1^- - d_1^+ = 80 \\ & x1 + d_2^- - d_2^+ = 100 \\ & x2+d_3^- \geq 45 \\ & x1,x2,d_1^-,d_1^+,d_2^-,d_2^+,d_3^- \geq 0 \end{array}$$

se tiene que:

La última tabla representa a la solución:

$x1=100,\ x2=0, d_1^-=d_2^+=d_2^-=0, d_1^+=20$ y $d_3^-=45$.

Dicha solución es la de mayor conveniencia pues a pesar de que se tiene un coeficiente positivo en el renglón $z_j - C_j$ para la prioridad P2 en la columna d_2^-, dicha variable no puede entrar a la base, en caso de que se hiciera lo contrario, d_2^- sería distinto de cero por lo que se rompería con la prioridad P1.

Para el renglón $z_j - C_j$ correspondiente a la prioridad P3 también existe un coeficiente mayor que cero en la columna x2, dicha variable no puede entrar a la base ya que para la prioridad P2 existe un número negativo en la misma columna.

Esto nos indica que se ha llegado a la solución óptima. A continuación se presentan las tablas que nos llevan a dicha solución.

		0	0	0	P2	P1	0	P3		
		x1	x2	d_1^-	d_1^+	d_2^-	d_2^+	d_3^-	ctes.	coc.
0	d_1^-	1	1	1	-1	0	0	0	80	80
P1	d_2^-	1	0	0	0	1	-1	0	100	100
P3	d_3^-	0	1	0	0	0	0	1	45	-
	P1	1	0	0	0	0	-1	0	100	
0	x1	1	1	1	-1	0	0	0	80	-
P1	d_2^-	0	-1	-1	1	1	-1	0	20	20
P3	d_3^-	0	1	0	0	0	0	1	45	-
	P1	0	-1	-1	1	0	-1	0	20	
0	x1	1	0	0	0	1	-1	0	100	100
P2	d_1^+	0	-1	-1	1	1	-1	0	20	-
P3	d_3^-	0	1	0	0	0	0	1	45	45
	P2	0	-1	-1	0	1	-1	0	20	
	P3	0	1	0	0	0	0	0	45	

C A P I T U L O III.

Existen varios métodos que permiten asesorar a un tomador de decisiones en la elección de una alternativa dentro de un conjunto finito de ellas tomando en cuenta su evaluación desde el punto de vista de una serie finita de criterios u objetivos.

Los siguientes capitulos están dedicados al análisis de tres de estos métodos: Electra I, Electra II y Electra III.

Una caracteristica común entre estos métodos es que parten de la misma estructura para representar el problema que nos ocupa.

La estructura consiste en una matríz cuyas columnas representan las alternativas y los renglones los criterios, asi, al elemento (i,j) se le asigna el valor o grado de satisfacción que produce la alternativa j con respecto al criterio i al tomador de decisiones.

Existen otras caracteristicas comunes entre los métodos que se analizarán a lo largo de la descripción de cada uno de éstos.

La relación que guardan estos métodos con los anteriores es que son también modelos para la toma de decisiones bajo criterios múltiples. Son métodos de mayor complejidad que los descritos en el primer capitulo por lo que es posible resolver con ellos problemas más complicados que en muchas ocasiones es dificil encontrar su solución aplicando el modelo de programación de metas debido a su estructura.

3.1 METODO ELECTRA I

Este método pretende reducir el problema presentando al tomador de decisiones un número menor de alternativas de las que inicialmente considera, eliminando aquellas que le producen menor satisfacción.

Para lograr su objetivo, Electra I, divide al conjunto total de las alternativas en dos subconjuntos:

ALTERNATIVAS DOMINADAS: son aquellas en las que existe al menos otra alternativa que produce una mejor satisfacción al tomador de decisiones desde TODOS los puntos de vista o criterios.

ALTERNATIVAS NO-DOMINADAS: son aquellas para las que no existe otra alternativa que produzca mayor satisfacción según TODOS los criterios; es decir, es el complemento de las alternativas dominadas.

El algoritmo Electra I, como se indicará posteriormente, trabaja sobre el segundo subconjunto y busca uno más pequeño.

Lo interesante de este método es que esta reducción es diferente para cada tomador de decisiones pues es él quien asigna el peso de los criterios reflejándose las preferencias de cada persona.

En base a estos valores se definen dos índices para cada alternativa que permiten descartalas o aceptarlas, según sea el caso.

Otra de las características de este método es que en base a los intereses del tomador de decisiones se definen los límites para estos índices. Estos límites pueden variarse, permitiendo una análisis de sensibilidad, propósito esencial para una toma de decisiones.

A continuación se plantea un problema donde se refleja una situación como la que nos ocupa.

Ejemplo 1.

Supongamos que al profesor Ramirez le ofrecen clases en cinco universidades diferentes. El Sr. Ramirez tiene que escoger una de ellas en la cual va a impartir clases.

Para elegir la más conveniente considera que su decisión depende del salario, del tiempo de transporte en minutos, del número de hrs. al semestre, del número de hrs. de asesorías en el semestre y del número de cursos al año en cada universidad.

Cada universidad le ofrece el salario de 9 meses y el salario en vacaciones, que es de gran importancia para él y le ha asignado un peso de cinco puntos a estos dos criterios en una escala del uno al cinco.

Al tiempo de transporte en minutos le ha asignado un peso de cuatro puntos, mientras que al número de hrs. al semestre le dió un peso de tres puntos. Finalmente, son de igual importancia para él, el número de hrs. en asesorías al semestre y el número de cursos al año con dos puntos en la misma escala.

Su problema se representa en la siguiente matriz de valores:

Criterios /Univer.	I	II	III	IV	V
Salario de 9 meses	\$24,000	\$23,000	\$26,500	\$27,000	\$22,000
# hrs. al semestre	9	7.5	10.5	10	6
tiempo de transporte en minutos	20	30	50	20	40
salario en vacaciones	\$4,800	\$4,600	\$2,600	\$5,500	\$6,000
# hrs. al semestre en asesorías	4	4	3	1	5
# cursos al año	3	2	4	5	1

y en el siguiente cuadro:

CRITERIO	PESO
1	5
2	3
3	4
4	5
5	2
6	2

Observemos que se desea el máximo salario, un mínimo de horas al semestre de asesorías, un mínimo tiempo de transporte y un mínimo número de cursos al año.

A través del algoritmo Electra I, el problema se tratará de simplificar, reduciendo el número de alternativas.

DESCRIPCION DEL METODO ELECTRA I.

A continuación se describirá brevemente el método Electra I.

Para reducir el problema de decisión, en este método se define una relación binaria R entre las alternativas no-dominadas que se denota como iRj que significa la preferencia de la alternativa i sobre la alternativa j bajo el punto de vista de la relación R.

Para la construcción de esta relación se definen dos índices: índice de concordancia e índice de discordancia; dichos índices se aplican a cada par de alternativas i,j no-dominadas.

El índice de concordancia mide el grado de satisfacción de la alternativa i sobre la alternativa j; es por ello que para su cálculo se toman en cuenta tanto los pesos de los criterios en que i es mejor que j como aquellos en que i tiene igual preferencia que j pero con menor importancia.

Asi este indice se define como:

$$C(i,j)= \left(\sum_{k\in I^{+}} w_k + \sum_{k\in I^{=}} w_k /2\right)/\left(\sum_{k\in I^{+}} w_k + \sum_{k\in I^{=}} w_k + \sum_{k\in I^{-}} w_k \right)$$

donde

- w_k = peso del criterio k asignado por el tomador de decisiones.
- I^{+} = conjunto de los criterios para los cuales la alternativa i es mejor que la alternativa j.
- $I^{=}$ = conjunto de los criterios para los cuales la alternativa i es igual a la alternativa j.
- I^{-} = conjunto de los criterios para los cuales la alternativa i es peor que la alternativa j.

Cabe señalar que algunos autores le proporcionan igual importancia a los pesos de los criterios donde la alternativa i es preferible a la alternativa j y a aquéllos donde se tiene la misma preferencia, por lo que también se puede definir el indice de concordancia como:

$$C(i,j)= \left(\sum_{k\in I^{+}} w_k + \sum_{k\in I^{=}} w_k \right)/\left(\sum_{k\in I^{+}} w_k + \sum_{k\in I^{=}} w_k + \sum_{k\in I^{-}} w_k \right)$$

Si la alternativa i es mejor que la alternativa j bajo todos los criterios entonces $\sum_{k\in I^{-}} w_k = \sum_{k\in I^{=}} w_k = 0$, lo que implica que $C(i,j) = 1$.

Si la alternativa i es peor que la alternativa j bajo todos los criterios entonces $\sum_{k\in I^{+}} w_k = \sum_{k\in I^{=}} w_k = 0$, lo que implica que

$C(i,j) = 0$.

Estas dos afirmaciones indican que el grado de satisfacción $C(i,j)$ se encuentra en el intervalo [0,1].

El indice de discordancia mide el grado de insatisfacción de la alternativa i sobre la alternativa j . Para la definición de este indice es necesario definir un intervalo de escalas común a todos los criterios y con ello poder comparar la inconformidad causada del nivel k_1 al k_2 del criterio r con la inconformidad causada del nivel k_3 al k_4 del criterio s.

Para llevar a cabo este objetivo, el tomador de decisiones define una escala de evaluación para cada criterio que depende del rango en el que desea que se encuentre el mejor y el peor nivel de cada uno.

Asi, todos los niveles de los criterios pueden ser comparados ya que tienen una puntuación a una escala que se encuentra en un rango común de escalas.

Con ello, el indice de discordancia puede ser definido como:

$$D(i,j) = \frac{\text{máximo intervalo donde j es mejor que i}}{\text{rango máximo de las escalas}}$$

Para que la alternativa i sea mejor que la alternativa j el máximo intervalo donde j es mejor que i debe ser pequeño, es decir, el indice de discordancia debe acercarse a cero. Este indice también se encuentra en un rango de valores [0,1] pues:

rango máximo de las escalas $>=$ máximo intervalo donde j es mejor que i

Un valor de $C(i,j)$ cercano a uno y un valor de $D(i,j)$ cercano a cero implica que la elección de la alternativa i es más satisfactoria para el tomador de decisiones que la elección de la alternativa j.

Con ello, la relación binaria R se define como:

i R j si y sólo si $C(i,j) >= p$ y $D(i,j) <= q$

donde

p y q son la concordancia mínima y la discordancia máxima permitidas por el tomador de decisiones respectivamente.

Ya definida la relación R, se construye una gráfica en donde cada nodo representa una alternativa (no-dominada) y las flechas o arcos entre los nodos representan la relación R, es decir, existe un arco del nodo i al j si y sólo si i R j.

Posteriormente, de la gráfica trazada, se agrupan en un solo nodo aquellos que forman un ciclo; dicho de otra manera, aquellas alternativas que son equivalentes bajo la relación R se agrupan en una sola transformando la gráfica en acíclica (sin ciclos).

Finalmente, se obtiene el subconjunto de alternativas no dominadas bajo la relación R llamado kernel que cumple dos propiedades:

1.- Toda alternativa j fuera del kernel está dominada bajo la relación R por al menos alguna alternativa i dentro del kernel, es decir, se cumple que i R j.

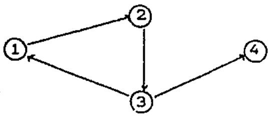
2.- Toda alternativa i dentro del kernel no es dominada por ninguna alternativa dentro del kernel; es decir, si la alternativa i y la j están dentro del kernel entonces no se cumple i R j o j R i.

Debido a la falta de las herramientas necesarias para obtener el kernel, se tomará en cuenta un subconjunto de éste cuya definición se describe a continuación y de aquí en adelante lo llamaremos kernel.

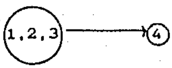
A partir de una gráfica acíclica y al estar de acuerdo con la afirmación de que la alternativa i es preferible a la alternativa j bajo la relación R, entonces se puede eliminar la alternativa j de la gráfica anterior y si se hace esta eliminación para todos los pares de alternativas, el conjunto que se tenía inicialmente de ellas se reducirá obteniendo el subconjunto de alternativas no-dominadas bajo la relación R .

Antes de describir el algoritmo Electra I se ejemplificará lo anterior.

Si tenemos la siguiente gráfica Ga:



la gráfica aciclica Gc es:



por lo tanto el kernel de la gráfica Gc es el nodo 1,2,3 y el problema de decisión se reduce a escoger cualquiera de las alternativas 1, 2 o 3.

3.2 ALGORITMO ELECTRA I

Paso 1. Se define un indice de concordancia para todo par de alternativas:

$$C(i,j) = \left(\sum_{k \in I^{+}} w_k + \sum_{k \in I^{=}} w_k /2 \right) / \left(\sum_{k \in I^{+}} w_k + \sum_{k \in I^{=}} w_k + \sum_{k \in I^{-}} w_k \right)$$

donde

- w_k = peso del criterio k asignado por el tomador de decisiones.
- I^{+} = conjunto de los criterios para los cuales la alternativa i es mejor que la alternativa j.
- $I^{=}$ = conjunto de los criterios para los cuales la alternativa i es igual a la alternativa j.
- I^{-} = conjunto de los criterios para los cuales la alternativa i es peor que la alternativa j.

Paso 2. Se define un indice de discordancia para todo par de alternativas:

$$D(i,j) = \frac{\text{máximo intervalo donde j es mejor que i}}{\text{rango máximo de las escalas}}$$

Paso 3. Se definen dos valores, p y q donde p es la concordancia mínima y q es la discordancia máxima permitidas por el tomador de decisiones.

Paso 4. Se forma una gráfica Ga donde los vértices representan las alternativas; existe el arco del vértice i al vértice j ($i \rightarrow j$) si y solo si $C(i,j) >= p$ y $D(i,j) <= q$.

Paso 5. En la gráfica Ga se agrupan en un solo nodo aquéllos que forman un ciclo; es decir, aquéllos que son equivalentes bajo la relación anterior; dicha gráfica acíclica se denota Gc.

Paso 6. De la gráfica Gc se obtienen todos aquellos nodos que no tienen predecesores, es decir, aquellos vértices o nodos para los que no existen flechas que lleguen a ellos.

Los nodos que se obtienen en el paso anterior forman el kernel de la gráfica Gc y es el subconjunto de alternativas no-dominadas buscado.

Ejemplo 2.

Se resolverá el ejemplo 1 siguiendo el algoritmo:

Paso 1. Cálculo del índice de concordancia:

Para ello se tiene que

$$\sum_{k\in I^+} w_k + \sum_{k\in I^=} w_k + \sum_{k\in I^-} w_k = 5+3+4+5+2+2=21$$

C(I,II)=(5+0+4+5+2/2+0)/21=15/21=0.71

C(II,I)=(0+3+0+0+2/2+2)/21=6/21=0.29

C(II,V)=(5+0+4+0+2+0)/21=11/21=0.52

C(V,II)=(0+3+0+5+0+2)/21=10/21=0.48

C(I,IV)=(0+3+4/2+0+0+2)/21=7/21=0.33

C(IV,I)=(5+0+4/2+5+2+0)/21=14/21=0.67

Siguiendo el mismo procedimiento se tiene la siguiente matriz de concordancia donde la coordenada (i,j) corresponde al índice de concordancia C(i,j).

----	0.71	0.67	0.33	0.52
0.29	----	0.67	0.24	0.52
0.33	0.33	----	0.10	0.33
0.67	0.76	0.90	----	0.52
0.48	0.48	0.67	0.48	----

Paso 2. Cálculo del índice de discordancia

Para ello, se asignarán las siguientes escalas a los criterios:

CRITERIO	ESCALA
1	150
2	130
3	140
4	150
5	100
6	100

El valor de los niveles de cada criterio se calcula de la siguiente manera:

El máximo rango de escalas es 150 y los criterios salario de 9 meses, número de horas al semestre, salario de vacaciones y número de cursos al año constan de cinco niveles por lo que la diferencia entre un nivel y otro es de 150/5 = 30 puntos, mientras que en los demás criterios es de 150/4 = 37.5 puntos por la existencia de cuatro niveles.

Entonces, la puntuación para cada nivel es:

CRITERIO	NIVEL	PUNTOS
1	$27,000	150
	$26,500	120
	$23,000	90
	$24,000	60
	$22,000	30
2	6 hrs.	10
	7.5 hrs.	40
	9 hrs.	70
	10 hrs.	100
	10.5 hrs.	130
3	20 min.	27.5
	30 min.	65
	40 min.	102.5
	50 min.	140
4	$6,000	150
	$5,500	120
	$4,800	90
	$4,600	60
	$2,600	30

CRITERIO	NIVEL	PUNTOS
5	1	-12.5
	3	25
	4	62.5
	5	100
6	1	-20
	2	10
	3	40
	4	70
	5	100

y el rango máximo de las escalas es 150.

Con la información anterior se puede calcular el indice de discordancia:

D(I,II)criterio 1 = 0

D(I,II)criterio 2 = (70-40)/150=0.2

D(I,II)criterio 3 = 0

D(I,II)criterio 4 = 0

D(I,II)criterio 5 = 0

D(I,II)criterio 6 = (40-10)/150=0.02

por lo tanto

D(I,II)=máximo(0,0.2)=0.2

Tomando en cuenta las alternativas V y II el indice de discordancia es el siguiente:

D(V,II)criterio 1 = (60-30)/150=0.2

D(V,II)criterio 2 = 0

D(V,II)criterio 3 = (102.5-65)/150=0.25

D(V,II)criterio 4 = 0

D(V,II)criterio 5 = (100-62.5)/150=0.25

D(V,II)criterio 6 = 0

entonces

D(V,II)=máximo(0,0.2,0.25)=0.25

D(II,V)criterio 1 = 0

D(II,V)criterio 2 = (40-10)/150=0.2

D(II,V)criterio 3 = 0

D(II,V)criterio 4 = (150-60)/150=0.6

D(II,V)criterio 5 = 0

D(II,V)criterio 6 = (10-(-20))/150=0.2

por lo que

D(II,V)=máximo(0,0.2,0.6)=0.6

Siguiendo el mismo procedimiento se tiene la matriz de discordancia donde la coordenada (i,j) corresponde al indice de discordancia D(i,j).

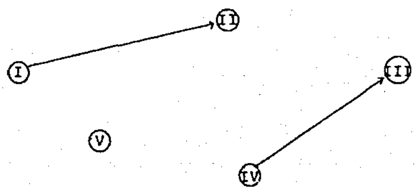
----	0.20	0.25	0.50	0.40
0.25	----	0.40	0.60	0.60
0.75	0.60	----	0.75	0.80
0.40	0.60	0.20	----	0.80
0.50	0.25	0.60	0.80	----

Paso 3. Supongamos que se desea una concordancia minima de 0.6 y una discordancia máxima de 0.2

Pasos 4 y 5. La gráfica Ga está formada por todas aquellas parejas que cumplen con la restricción descrita en el paso 3; dichas parejas son:

(I,II) y (IV,III)

La gráfica Gc es entonces:



Paso 6. Los nodos de la gráfica Gc que forman el kernel, o bien, los nodos que no tienen predecesores son: {1,4,5}

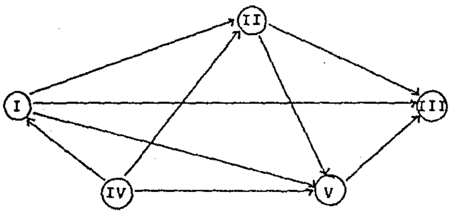
El problema se ha reducido ya que se tiene que elegir entre tres alternativas y no entre cinco.

OBSERVACION:

Si en el paso 3 nos ponemos más estrictos el problema se puede reducir todavía más. Supongamos que la concordancia mínima y la discordancia máxima son 0.5 y 0.8 respectivamente. Las parejas de alternativas que cumplen con dicha restricción son:

(I,II),(I,III),(I,V),(II,III),(II,V),(IV,I),(IV,II),(IV,III),(IV,V) y (V,III).

Por lo tanto, la gráfica Gc es la que a continuación se presenta:



El kernel de la gráfica es {IV} por lo que el problema se redujo a unicamente la alternativa IV.

Obsérvese que variando la concordancia mínima y la discordancia máxima existe un cambio en la solución permitiendo de este modo un análisis de sensibilidad.

Ejemplo III.

Supongamos que se quiere establecer una nueva planta cortadora de piel y para alcanzar los objetivos deseados se han propuesto 5 sistemas como alternativas.

Los criterios usados para evaluar y comparar los 5 sistemas propuestos incluyen la disponibilidad de cuchillas o navajas, energía requerida, agua requerida, modo del dispositivo de desgaste de agua, planta y costos de equipo, costos de operación, desarrollo económico, oportunidades para el desarrollo de programas de investigación y respuesta a la inversión realizada.

Las caracteristicas de cada sistema se encuentran en la siguiente tabla:

CRITERIO/SISTEMA	1	2	3	4	5
1. disponibilidad de cuchillas (por día)	4000	1500	3000	2000	1000
2. energía requerida (1000 watts por cuchilla)	150	272	180	250	200
3. agua requerida (miles de galones/día)	900	500	900	400	200
4. modo del dispositivo de desgaste de agua	bueno	muy bueno	regular	excelente	bueno
5. planta y costos de equipo (millones)	\$10.9	\$4.11	\$8.2	\$2.7	\$1.8
6. costos de operación (millones/año)	\$12.4	\$4.6	\$9.3	\$3.1	\$3.1
7. aprovechamiento económico	excelente	regular	muy bueno	regular	bueno
8. oportunidades para el desarrollo de programas de investigación	ninguna	suficientes	algunas	muchas	algunas
9. respuesta a la inversión	%1.00	%1.09	%1.19	%0.84	%0.64

Así mismo, el peso de cada criterio se establece en la siguiente tabla:

CRITERIO	PESO
1	9
2	4
3	2
4	7
5	6
6	6
7	8
8	10
9	8

Lo que se busca en este problema es elegir un sistema que se encuentre más cerca de las siguientes caracteristicas: mayor número de cuchillas o navajas, menor requerimiento de energía y agua, mayor modo de dispositivo de desgaste de agua, menor costo en planta y equipo así como de operación, mayor aprovechamiento económico, mayor número de oportunidades para el desarrollo de programas de investigación y mayor tasa como respuesta a la inversión.

Para calcular el indice de discordancia, se asignarán las siguientes escalas a cada criterio:

CRITERIO	ESCALA
1	95
2	60
3	55
4	85
5	80
6	80
7	90
8	100
9	85

La puntuación para cada nivel es:

CRITERIO	NIVEL	PUNTOS
1	4,000	95
	3,000	75
	2,000	55
	1,500	35
	1,000	15
2	150	-20
	180	0
	200	20
	250	40
	272	60

CRITERIO	NIVEL	PUNTOS
3	200	-20
	400	5
	500	30
	900	55
4	excelente	85
	muy bueno	60
	bueno	35
	regular	10
5	1.8	0
	2.7	20
	4.11	40
	8.2	60
	10.9	80
6	3.1	5
	4.6	30
	9.3	55
	12.4	80
7	excelente	90
	muy bueno	65
	bueno	40
	regular	15
8	muchas	100
	suficientes	75
	algunas	50
	ninguna	25
9	1.19	85
	1.09	65
	1.00	45
	0.84	25
	0.64	5

y el rango máximo de escalas es 100.

Cálculo de algunos de los indices de concordancia:

$$C(5,3)=\frac{2+7+6+6+10/2}{60}=26/60=0.43$$

$$C(1,4)=\frac{9+4+8+8}{60}=29/60=0.48$$

$$C(3,2)=\frac{9+4+8+8}{60}=29/60=0.48$$

La siguiente matriz es la matriz de concordancia:

----	0.35	0.48	0.48	0.54
0.65	----	0.52	0.20	0.57
0.52	0.48	----	0.48	0.57
0.52	0.80	0.52	----	0.62
0.46	0.43	0.43	0.38	----

Cálculo de algunos indices de discordancia:

D(2,3)criterio 1 = (75-35)/100=0.4

D(2,3)criterio 2 = (60-0)/100=0.6

D(2,3)criterio 7 = (65-15)/100=0.5

D(2,3)criterio 9 = (85-65)/100=0.2

por lo tanto

D(2,3)=máximo(0.4,0.6,0.5,0.2)= 0.6

D(5,2)criterio 1 = (35-15)/100=0.2

D(5,2)criterio 4 = (60-35)/100=0.25

D(5,2)criterio 8 = (75-50)/100=0.25

D(5,2)criterio 9 = (65-5)/100=0.6

entonces

D(5,2)=máximo(0.2,0.25,0.6)= 0.6

D(4,2)criterio 9 = (65-25)/100= 0.4

por lo que

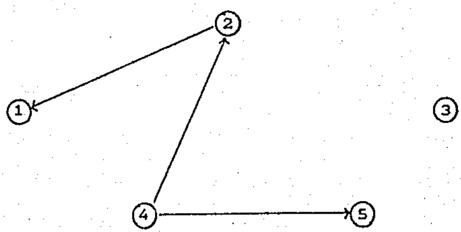
D(4,2)=0.4

La matriz de discordancia es la siguiente:

----	0.50	0.40	0.75	0.80
0.80	----	0.60	0.25	0.50
0.25	0.50	----	0.75	0.75
0.75	0.40	0.60	----	0.25
0.80	0.60	0.80	0.50	----

Supongamos una concordancia mínima de 0.6 y una discordancia máxima de 0.8; así, las parejas de alternativas que cumplen con lo anterior son: (2,1),(4,2) y (4,5).

La gráfica Gc es:



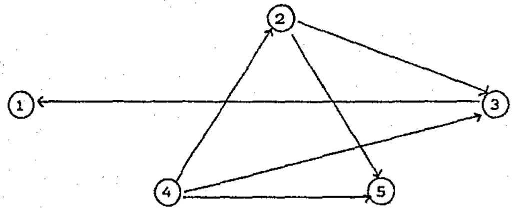
El kernel de la gráfica anterior está formado por las alternativas {3,4}, lo que quiere decir que el problema se ha reducido a tres alternativas en lugar de las cinco que teníamos.

Consideremos 0.5 y 0.7 como la concordancia mínima y la discordancia máxima respectivamente.

Las parejas de alternativas que cumplen lo anterior son:

(2,3),(2,5),(3,1),(4,2),(4,3) y (4,5).

Construyamos entonces la gráfica Gc:



El kernel de dicha gráfica es la alternativa 4 lo que implica que al tomador de decisiones le conviene escoger el sistema número 4.

3.3 METODO ELECTRA II

Este método, al igual que el Electra I, se aplica en la toma de decisiones con criterios múltiples.

La ventaja de este método con respecto al anterior es que en el Electra II se obtiene un ordenamiento completo de las alternativas no-dominadas permitiendo resolver aquellos problemas donde no sólo es interesante obtener una alternativa óptima a elegir sino que se presenta el caso de la elección de varias alternativas de acuerdo a un orden de preferencias.

Un ejemplo de este caso es el de la dentista Guadalupe Gómez que abrirá su propio consultorio y tiene que realizar la compra de varios aparatos. Por falta de recursos económicos no puede llevar a cabo la compra de todos a la vez por lo que en base a la inflación, la necesidad del aparato para iniciar las consultas sencillas, las tasas de interés que le proporciona el banco y otros factores desea establecer el orden óptimo en las compras que debe ir realizando.

El proceso de este método, como es de suponer, es más complejo que el del método Electra I ya que para obtener una solución única es necesario realizar varios ordenamientos, proceso que se describe en las páginas posteriores.

Así como en el Electra I, en el Electra II existen condiciones de concordancia y discordancia pero en este caso existen varios niveles utilizados para la construcción de los ordenamientos citados anteriormente.

DESCRIPCION DEL METODO ELECTRA II.

Para poder establecer un ordenamiento en las alternativas, los intereses del tomador de decisiones se reflejan en varios niveles de concordancia y discordancia.

El método Electra II se basa en estos niveles para definir dos relaciones de sobreordenamiento: una fuerte y otra débil, denotadas como Rf y Rd respectivamente.

Si el grado de satisfacción de la alternativa i sobre la alternativa j es "estrictamente" alto y el de insatisfacción es bajo o ambos niveles son altos entonces i Rf j. Si ambos grados son bajos entonces i Rd j.

Estas relaciones se utilizan para definir dos gráficas que como en el Electra I, son gráficas acíclicas o sin ciclos. La gráfica definida por la relación Rf se reduce a la gráfica Gf y en forma similar se obtiene Gd de la relación Rd.

Dichas gráficas se utilizan en un procedimiento iterativo de ordenamiento de alternativas que consta de tres ordenamientos: el ordenamiento fuerte que establece un orden de mayor a menor en las alternativas tomando en cuenta la relación fuerte, el ordenamiento débil que establece un orden de mayor a menor en las alternativas tomando en cuenta la relación débil y el ordenamiento medio. Este último, obtiene el ordenamiento final de las alternativas.

Para dar inicio al proceso del método Electra II se define el índice de concordancia de la siguiente manera:

$$C(i,j) = \left(\sum_{k \in I^+} w_k + \sum_{k \in I^=} w_k\right) / \left(\sum_{k \in I^+} w_k + \sum_{k \in I^=} w_k + \sum_{k \in I^-} w_k\right)$$

en donde

w_k = peso del criterio k asignado por el tomador de decisiones.

I^+ = conjunto de criterios en donde la alternativa i es mejor que la alternativa j.

$I^=$ = conjunto de criterios en donde la alternativa i y la j son iguales.

I^- = conjunto de criterios en donde la alternativa j es mejor que la alternativa i.

y el índice de discordancia como:

$$D(i,j)= (v_k(j) - v_k(i))/ \theta(k,v_k(i)))$$

en donde la alternativa j es mejor que la alternativa i para el criterio k y además:

$v_k(j)$= valor de la alternativa j tomando en cuenta el criterio k.

$v_k(i)$= valor de la alternativa i tomando en cuenta el criterio k.

$\theta(k,v_k(i))$= parámetro teta.

Existen dos formas del cálculo del parámetro teta

a) Si se quiere una discordancia estática entonces el parámetro teta se fija igual a una constante para cada criterio i:

$\theta_i = c_i$ donde c_i es una constante.

b) Si se desea una medida de discordancia dinámica entonces se calcula de la siguiente manera:

$$\theta(k,v_k(i))= \text{máximo}\{v_k(i),s(k)\}$$

en donde s(k) es un valor que el tomador de decisiones fija y que le permite controlar la importancia de la diferencia $v_k(j)-v_k(i)$; es decir, si ocurre que s(k) es muy grande entonces el índice de discordancia disminuye, en cambio, si s(k) es muy chico el índice de discordancia aumenta.

Los algoritmos de cada ordenamiento se describen a continuación:

Ordenamiento fuerte

Sea Yk un subconjunto de la gráfica Gf donde Y0 = Gf, entonces

1. Fijar k=0.
2. Calcular el kernel de Yk, es decir, aquellas alternativas o nodos de la gráfica Yk que no tienen predecesores. Sea D este conjunto.
3. Identificar los nodos en el conjunto D que tengan arcos en la gráfica Gd. Sea U este conjunto.
4. Seleccionar aquellos nodos en U que no tengan precedentes en la gráfica Gd. Sea B este conjunto.
5. Definir

$$A_k = (D - U) \cup B = (D \cap U^c) \cup B$$

6. Definir el ordenamiento fuerte v1 en las alternativas i de Ak como:

$$v1(i) = k+1.$$

7. Fijar

$$Y_{k+1} = Y_k - A_k = Y_k \cap (A_k)^c$$

8. Si Yk+1 es el conjunto vacio parar; en caso contrario hacer : k= k+1 y regresar a 2.

En el ordenamiento fuerte se toman en cuenta sólo aquellas alternativas que producen un nivel alto de satisfacción y un nivel bajo de insatisfacción para el tomador de decisiones; cabe aclarar que dichos niveles son proporcionados por el tomador de decisiones.

Al final del proceso del ordenamiento fuerte es la función v1 la que almacena este orden y aquella(s) alternativa(s) i que produce(n) mayor satisfacción al tomador de decisiones tiene(n) asignado un valor v1(i) igual a uno, las que producen una menor satisfacción a éstas se les asigna un valor de dos y así sucesivamente.

Ordenamieto débil

1. Invertir la dirección de los arcos de la gráfica Gd y Gf.
2. Seguir el ordenamiento fuerte siguiendo de igual forma cada paso, unicamente hacer un cambio en el paso 6 substituyendo a(i) en lugar de v1(i).
3. Obtener el ordenamiento débil de la siguiente forma:

   $$v2(i) = 1 + \max_{j} \{ a(j) \} - a(i)$$

   para toda alternativa i.

Las alternativas que intervienen en el ordenamiento débil son aquellas que su nivel de satisfacción y de insatisfacción no es tan alto ni tan bajo, respectivamente, como en el ordenamiento fuerte y es en la función v2 donde se almacena dicho orden.

Ordenamiento medio

Para obtener un ordenamiento promedio entre este último ordenamiento y el fuerte, se calcula el ordenamiento medio de la siguiente forma:

$$m(i) = (v1(i) + v2(i)) / 2$$

Ordenamiento final

Se ordenan las alternativas i por valor decreciente de m(i), así a la alternativa o alternativas con valor más bajo les corresponde el primer orden, quedando ordenadas las alternativas de mejor a peor satisfacción para el tomador de decisiones.

3.4 ALGORITMO ELECTRA II

A continuación se describe detalladamente el método:

Paso 1. Se obtiene la matriz de concordancia de la siguiente manera:

Para todo par de alternativas i,j se calcula

$$C(i,j) = \left(\sum_{k\in I^{+}} w_k + \sum_{k\in I^{=}} w_k\right) \Big/ \left(\sum_{k\in I^{+}} w_k + \sum_{k\in I^{=}} w_k + \sum_{k\in I^{-}} w_k\right)$$

en donde

w_k = peso del criterio k asignado por el tomador de decisiones.

I^{+} = conjunto de criterios en donde la alternativa i es mejor que la alternativa j.

$I^{=}$ = conjunto de criterios en donde la alternativa i y la j son iguales.

I^{-} = conjunto de criterios en donde la alternativa j es mejor que la alternativa i.

Paso 2. Se calcula la matriz de discordancia para toda pareja de alternativas i,j tomando en cuenta que:

$$D(i,j) = (v_k(j) - v_k(i)) / \Theta(k, v_k(i))$$

en donde la alternativa j es mejor que la alternativa i para el criterio k y además:

$v_k(j)$= valor de la alternativa j tomando en cuenta el criterio k.

$v_k(i)$= valor de la alternativa i tomando en cuenta el criterio k.

$\Theta(k,v_k(i))$= parámetro teta.

Paso 3. Se definen tres niveles de concordancia p^*, po, p^- tal que

$0 < p^- < po < p^* < 1$

y dos niveles de discordancia qo, q^* tal que

$0 < qo < q^* < 1$.

Paso 4. Se define la relación fuerte Rf para aquellas alternativas i,j que cumplen cualquiera de las siguientes dos restricciones

Restricción 1

1. $C(i,j) >= p^*$
2. $D(i,j) <= q^*$
3. $\sum_{k \in I^+} w_k >= \sum_{k \in I^-} w_k$

Restricción 2

1. $C(i,j) >= po$
2. $D(i,j) <= qo$
3. $\sum_{k \in I^+} w_k >= \sum_{k \in I^-} w_k$

Paso 5. Se define la relación débil Rd para aquellas alternativas i,j que cumplan con la siguiente restricción

1. $C(i,j) >= p-$
2. $D(i,j) <= q^*$
3. $\sum_{k \in I^+} w_k >= \sum_{k \in I^-} w_k$

Paso 6. Se obtiene una gráfica Gf acíclica formada por las parejas de alternativas que cumplen con la relación fuerte; así mismo se construye una gráfica acíclica Gd formada por las parejas de alternativas que cumplen la relación débil.

Paso 7. Se obtiene un ordenamiento fuere v1.

Paso 8. Se obtiene un ordenamiento débil v2.

Paso 9. Se obtiene un ordenamiento medio m.

Paso 10. Se obtiene el ordenamiento final.

Ejemplo 1.

Considérese el Banco X organizado por cinco áreas.

Debido a que existen varios porblemas en dicho Banco es necesario saber que áreas están en mejores condiciones y cuales son las que tienen más dificultades para darles mayor apoyo y mejorarlas.

Se ha otorgado una calificación a cada área de acuerdo a cuatro puntos de vista o criterios. Dichos criterios son:

1. Participación en los problemas generales del Banco.

2. Incremento de utilidades.

3. Número de quejas recibidas (como un porcentaje del personal).

4. Tasa de cambio del personal.

La matriz de calificaciones de las áreas para cada uno de los criterios es:

criterios:	1	2	3	4
alt. 1	0.81	0.50	-0.27	-0.60
alt. 2	0.80	0.52	-0.30	-0.58
alt. 3	0.90	0.47	-0.22	-0.59
alt. 4	0.75	0.60	-0.40	-0.62
alt. 5	0.78	0.41	-0.25	-0.62

A cada criterio se le ha asignado el peso 1,4,1 y 3 respectivamente. Además, se quiere una discordancia dinámica: $s(2)=0.3$, $s(4)=0.2$, para el resto de los criterios $s(k)=0$.

Utilicemos el método Electra II para resolver el problema.

Paso 1. Cálculo del índice de concordancia para algunas parejas de alternativas:

C(1,5)= (1+4+3)/9 = 0.89 lo que implica que

C(5,1)= 1 - 8/9= 0.11

C(4,3)= 4/9 = 0.44 entonces

C(3,4)= 1 - 4/9 = 0.56

Asi la matriz de concordancia es:

----	0.22	0.44	0.56	0.89
0.78	----	0.78	0.56	0.89
0.56	0.22	----	0.56	1.00
0.44	0.44	0.44	----	0.78
0.11	0.11	0.00	0.56	----

Paso 2. Cálculo del índice de discordancia:

D(3,1)criterio 2 = (0.50-0.47)/máximo{0.3,0.47} = 0.03/0.47= 0.06

por lo tanto

D(3,1)=0.06

D(1,4)criterio 2 = (0.60-0.50)/máximo{0.30,0.50}= 0.10/0.50= 0.2

por lo que

D(1,4)= 0.2

D(2,3)criterio 1 = (0.90-0.80)/máximo{0.80,0.00}= 0.1/0.8= 0.13

D(2,3)criterio 3 = (-0.30+0.22)/mínimo(-0.3,0)= -0.08/-0.3= 0.27

entonces

D(2,3)= máximo{0.13,0.27}= 0.27

OBSERVACION:

Ya que la medida de discordancia debe ser positiva, para los criterios con valores negativos se define:

$$D(i,j)= (v_k(i) - v_k(j))/\text{mínimo}\ \{v_k(i), s(k)\}$$

La matriz de discordancia es:

----	0.04	0.19	0.20	0.07
0.10	----	0.27	0.15	0.17
0.06	0.11	----	0.28	0.00
0.32	0.25	0.45	----	0.38
0.22	0.27	0.15	0.46	----

Paso 3. Sean $p^-=0.6$
$p^o=0.65$
$p^*=0.75$

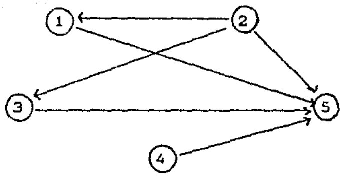
los niveles de concordancia mientras que los de discordancia son:

$q^o=0.2$
$q^*=0.65$

Paso 4. Las parejas de alternativas que cumplen con la relación fuerte son: (2,1),(2,3),(1,5),(2,5),(3,5) y (4,5)

Paso 5. Las parejas de alternativas que cumplen con la relación débil son: (2,1),(2,3),(1,5),(2,5),(3,5) y (4,5)

Paso 6. La gráfica Gf y Gd son iguales y se muestra a continuación

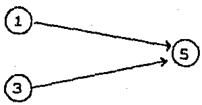


Paso 7. Y(0)= Gf

Primera iteración:

1. k=0
2. D={2,4}
3. U={vacio}

4. B={vacio}
5. A(o)={2,4}
6. v1(2)=1, v1(4)=1
7. La gráfica Y(1) es:



8. k=1

Segunda iteración:

2. D={1,3}
3. U={vacio}
4. B={vacio}
5. A(1)={1,3}
6. v1(1)=2, v1(3)=2
7. La gráfica Y(2) es:

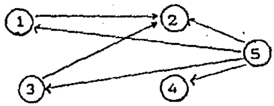
5

8. k=2

Tercera iteración:

2. D={5}
3. U={vacio}
4. B={vacio}
5. A(2)={5}
6. v1(5)=3
7. Y(3) no contiene ningún nodo
8. alto

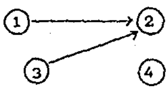
Paso 8. La inversa de la gráfica Gf es:



Primera iteración:

1. k=0
2. D={5}
3. U={vacio}
4. B={vacio}
5. A(0)={5}
6. a(5)=1

7. Se tiene la gráfica Y(1)



8. k=1

Segunda iteración:

2. D={1,3,4}

3. U={vació}

4. B={vacio}

5. A(1)={1,3,4}

6. a(1)=2, a(3)=2, a(4)=2

7. La gráfica Y(2) es:

8. k=2

Tercera iteración:

2. D={2}

3. U={vacio}

4. B={vacio}

5. A(2)={2}

6. a(2)=3

7. Y(3) no tiene ningún nodo

8. alto

v2(5)=3

v2(1)=2

v2(3)=2

v2(4)=2

v2(2)=1

Paso 9.

m(1)=(2+2)/2=2

m(2)=(1+1)/2=1

m(3)=(2+2)/2=2

m(4)=(1+2)/2=1.5

m(5)=(3+3)/2=3

Paso 10.

alternativa 2	rango 1
alternativa 4	rango 2
alternativa 1	rango 3
alternativa 3	rango 4
alternativa 5	rango 5

Este ordenamiento implica que el área 2 es la que se encuentra en mejores condiciones y la 5 necesita mayor apoyo por estar en las peores condiciones. El área 4 está en peores condiciones que el área 2 pero está mejor que la 1. Por último, la situación del área 3 es mejor que la de la 5.

Ejemplo II.

Resolveremos el ejemplo I que se resolvió anteriormente (página 70), por medio del Electra I.

Tenemos la siguiente matriz de valores:

alt./criterio	1	2	3	4	5	6
1	24000	9.0	20	4800	4	3
2	23000	7.5	30	4600	4	2
3	26500	10.5	50	2600	3	4
4	27000	10.0	20	5500	1	5
5	22000	6.0	40	6000	5	1

Para los criterios 2,3,5 y 6 se prefiere la alternativa de menor valor, en cambio, para los criterios 1 y 4 se prefiere la alternativa de mayor valor.

Los pesos de los criterios del 1 al 6 son 5,3,4,5,2 y 2 respectivamente.

Paso 1. Calculo de la matriz de concordancia:

$C(1,4)= (3+2+4)/21= 9/21= 0.43$

$C(4,1)= (5+4+5+2)/21= 16/21= 0.76$

$C(3,2)= (5+2)/21= 0.33$ lo que implica que

$C(2,3)= 1-0.33= 0.67$

Siguiendo el mismo procedimiento para toda pareja de alternativas se tiene la matriz de concordancia.

----	0.76	0.67	0.43	0.52
0.33	----	0.67	0.24	0.52
0.33	0.33	----	0.10	0.33
0.76	0.76	0.90	----	0.52
0.48	0.48	0.67	0.48	----

Paso 2. Se calculará la matriz de discordancia tomando en cuenta que no se desea una discordancia dinámica sino por el contrario una estática. Los valores de s(k) son: s(1)=5000, s(2)=4.5, s(3)=30, s(4)=3400, s(5)=4 y por último se tiene que s(6)=4.

D(1,4)criterio 1 =(27000-24000)/5000= 0.6

D(1,4)criterio 4 =(5500-4800)/3400= 0.21

D(1,4)criterio 5 =(4-1)/4= 0.75

por lo tanto

D(1,4)= máximo(0.6,0.21,0.75)= 0.75

D(5,2)criterio 1 =(23000-22000)/5000= 0.2

D(5,2)criterio 3 =(40-30)/30= 0.33

D(5,2)criterio 5 =(5-4)/4= 0.25

por lo que

D(5,2)= máximo(0.2,0.33,0.25)= 0.33

La matriz de discordancia se dá a continuación:

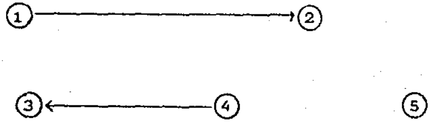
----	0.33	0.50	0.75	0.67
0.33	----	0.70	0.80	0.41
1.00	0.67	----	1.00	1.00
0.50	0.75	0.25	----	1.00
0.67	0.33	0.90	1.00	----

Paso 3. Los indices de concordancia son: 0.4,0.6,0.65 y los de discordancia: 0.2,0.4

Paso 4. Las parejas de alternativas que cumplen con la relación fuerte son: (1,2) y (4,3).

Paso 5. Las parejas de alternativas que cumplen con la relación débil son: (1,2) y (4,3).

Paso 6. La gráfica Gf y Gd son iguales:

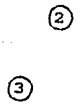


Paso 7. Seguiremos con el ordenamiento fuerte:

Y(0)= Gf

Primera iteración:

1. k=0
2. D={1,4,5}
3. U={vacio}
4. B={vacio}
5. A(0)={1,4,5}
6. v1(1)=1,v1(4)=1,v1(5)=1
7. La gráfica Y(1) es:



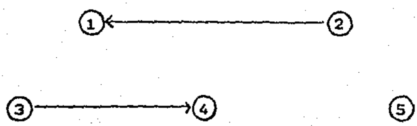
8. k=1

Segunda iteración:

2. D={2,3}
3. U={vacío}
4. B={vacío}
5. A(1)={2,3}
6. v1(2)=2, v1(3)=2
7. Y(2) no tiene ningún nodo
8. alto

Paso 8. Se calculará el ordenamiento débil:

La inversa de la gráfica Gf y Gd son iguales:



Primera iteración:

1. k=0
2. D={2,3,5}
3. U={vacio}
4. B={vacio}
5. A(0)={2,3,5}
6. a(2)=1, a(3)=1, a(5)=1
7. Y(1) es:

8. k=1

Segunda iteración:

2. D={1,4}
3. U={vacio}
4. B={vacio}
5. A(1)={1,4}
6. a(1)=2, a(4)=2
7. La gráfica Y(2) no tiene nodos
8. alto

por lo tanto:

v2(1)=1+2-1= 1

v2(2)=1+2-1= 2

v2(3)=1+2-1= 2

v2(4)=1+2-1= 1

v2(5)=1+2-1= 2

Paso 9. Obtendremos el ordenamiento medio:

m(1)=(1+1)/2= 1

m(2)=(2+2)/2= 2

m(3)=(2+2)/2= 2

m(4)=(1+1)/2= 1

m(5)=(1+2)/2= 1.5

Paso 10. El ordenamiento final se dá a continuación:

alternativa	rango
1	1
4	2
5	3
2	4
3	5

Esto implica que la Universidad I es la de mayor conveniencia para el Profesor Ramirez de acuerdo a sus intereses. La siguiente Universidad en orden de conveniencia es la IV, después la V, a continuación la II y la de menor conveniencia es la Universidad III.

Anteriormente, al aplicar el método Electra I a este problema, se llegó a una reducción del mismo: elegir la mejor opción entre dos Universidades y no entre las cinco que se tenian inicialmente. Con un análisis de sensitividad se puede llegar a reducir el problema encontrando la mejor opción.

Con las diferentes soluciones obtenidas anteriormente se puede observar la ventaja que posee el método Electra II sobre el Electra I. Del Electra II se obtiene un ordenamiento completo de las alternativas no-dominadas mientras que en el Electra I se llega unicamente a reducir el problema ordenando las alternativas en alternativas en el kernel y fuera del kernel.

3.5 METODO ELECTRA III

Este método, al igual que los anteriores, permite asesorar a un tomador de decisiones en la elección de una alternativa dentro de un conjunto finito de ellas tomando en cuenta criterios múltiples.

Para lograr este objetivo, el Electra III clasifica al conjunto de alternativas en clases o rangos. Estas clases se encuentran en orden de preferencia o satisfacción de mayor a menor, es decir, la primera clase contiene a aquella(s) alternativa(s) de mayor preferencia para el tomador de decisiones de acuerdo a sus intereses, la segunda clase contiene a aquella(s) de menor grado de satisfacción que la(s) anterior(es) y así sucesivamente, por lo que la última clase contiene a la(s) alternativa(s) de menor preferencia.

La diferencia entre el método anterior y el Electra III es que en este último se involucran de una manera más significativa los intereses del tomador de decisiones.

Esta diferencia es una gran ventaja ya que mientras mayor sea la participación de los intereses del tomador de decisiones, la elección de una alternativa sobre un conjunto finito de ellas es más adecuada y precisa.

Además, esta característica crea cierta flexibilidad ya que deja al interesado en libertad de definir:

- Una función de indiferencia y una función de preferencia de una alternativa sobre otra.

- El rechazo a la afirmación de que la alternativa x sea preferible a la alternativa y a través de una función.

- Un límite de discriminación para comparar los diferentes grados de satisfacción de la alternativa x sobre la alternativa y.

Debe notarse que las funciones citadas anteriormente, intervienen en el cálculo de los índices de concordancia y discordancia a diferencia de los métodos anteriores donde sólo se consideran los pesos relativos de los criterios.

DESCRIPCION DEL METODO ELECTRA III.

Para lograr una clasificación de las alternativas, el método Electra III realiza inicialmente dos clasificaciones en rangos o clases. Una de las clasificaciones es llamada destilación descendente y la otra destilación ascendente.

A continuación se explicará el procedimiento de cada una de ellas.

DESTILACION DESCENDENTE

El objetivo de este proceso es particionar al conjunto de alternativas en clases mutuamente excluyentes. Como se analiza en los párrafos posteriores, la primera clase que se obtiene de este proceso contiene a las alternativas de mayor preferencia para el interesado, la segunda contiene las alternativas con un grado menor de satisfacción que las anteriores y asi sucesivamente; es ésta la razón por la que este proceso recibe el nombre de DESTILACION DESCENDENTE.

Esta destilación consiste en tomar al conjunto X de alternativas y asignarlo a un nuevo conjunto B1; de este conjunto se obtiene otro al que llamaremos O1 (O1 C B1). O1 es el subconjunto de aquellas alternativas que tienen mayor preferencia para el tomador de decisiones de acuerdo a sus intereses. Después de obtener el subconjunto O1, se vuelven a analizar las preferencias del interesado y se extraen las de mayor preferencia formando otro conjunto O2 el cual está contenido en O1 (O2 C O1).

Se continúa sucesivamente hasta llegar a un conjunto Ok con el cual ya no se puede aplicar el procedimiento, es decir, cuando las alternativas que formen el nuevo conjunto tengan la misma preferencia. Al conjunto Ok lo llamaremos C1 (primera clase). En seguida se eliminan las alternativas que formen el conjunto C1 del B1 y al conjunto resultante lo llamaremos B2, es decir:

$$B2 = B1 - C1.$$

Con el conjunto B2 repetimos el procedimiento o la destilación que se realizó con B1 y al conjunto que se obtiene finalmente lo llamamos C2 (segunda clase).

Con el conjunto $B3 = B2 - C2$, repetimos la operación y así sucesivamente hasta obtener un conjunto Bn que no contenga elementos.

De esta manera se ha dividido al conjunto X de alternativas, en clases C1,C2,,,Cn-1, donde Cn-1 contiene las alternativas de menor preferencia para el tomador de decisiones.

DESTILACION ASCENDENTE

La diferencia de este proceso con el anterior, es que inicialmente se obtienen las clases de menor a mayor grado de satisfacción. Sin embargo, el objetivo es el mismo en ambas destilaciones: lograr una clasificación en las alternativas de acuerdo al orden de preferencias de mayor a menor.

Este proceso consiste en tomar al conjunto X de alternativas y asignarlo al conjunto B1 del cual obtenemos aquellas alternativas que son de menor preferencia o satisfacción para el tomador de decisiones de acuerdo a sus intereses; al conjunto resultante lo llamamos O1 ($O1 \subset B1$). O1 es el subconjunto de alternativas que tienen menor preferencia para el tomador de decisiones de acuerdo a sus intereses. Después de obtener este subconjunto O1, se vuelven a analizar las preferencias del interesado y se extraen las de menor preferencia formando otro conjunto O2 el cual está contenido en O1 ($O2 \subset O1$) y así sucesivamente hasta llegar al conjunto Ok el cual ya no puede ser destilado, es decir, no se le puede aplicar el procedimiento pues las alternativas que lo forman tienen la misma preferencia. Al conjunto Ok lo llamamos C1 (última clase).

Se toma en seguida al conjunto B1 eliminando las alternativas que están en C1 denotando al nuevo conjunto B2, así:

$$B2 = B1 - C1.$$

Al conjunto B2 le aplicamos el procedimiento que realizamos con B1 y al conjunto que se obtiene lo llamamos C2 (penúltima clase). Se repite la operación con $B3 = B2 - C2$ y así sucesivamente hasta llegar a un conjunto Bn que no contenga elementos.

De esta manera el conjunto X de alternativas queda dividido por las clases C1,C2,,,Cn-1 donde Cn-1 contiene las alternativas de mayor preferencia para el tomador de decisiones.

Finalmente se invierte el orden de las clases de la siguiente forma:

C1 = Cn-1
C2 = Cn-2
.
.
.
Cn-1 = C1

De esta manera se obtiene la primera clase C1, la segunda C2,,,etc.

En consecuencia, para obtener una clasificación final de las alternativas se tiene que realizar una clasificación por suma de rangos que consiste en:

1. Para cada alternativa, sumar la clase o rango que se obtuvo en la destilación descendente con la que se obtuvo en la ascendente.

2. Asignar a la primera clase a aquellas alternativas con el valor menor obtenido en el paso 1, la segunda clase a las alternativas con el valor siguiente y así sucesivamente hasta clasificar todas las alternativas.

Para poder llevar a cabo tanto la destilación descendente como la ascendente es necesario definir varios conceptos:

Sabemos que se prefiere la alternativa x a la alternativa y si los valores de la alternativa x son mayores que los de la alternativa y bajo casi todos los puntos de vista o criterios por lo que es necesario que:

$$a(x,j) > a(y,j) \quad \text{para todo criterio } j$$

donde

a(k,j) es el valor de la alternativa k bajo el criterio j.

LLamaremos UU_j a la diferencia entre el valor de la alternativa x y la alternativa y con respecto al criterio j.

Así, si

$$a(x,j) - a(y,j) = UU_j > 0 \quad \text{para toda } j$$

se dice que la alternativa x es preferible a la alternativa y.

Sin embargo, esta diferencia puede ser tan pequeña (cercana a cero) que resulte indiferente la alternativa x a la alternativa y con respecto al criterio j.

Para diferenciar estos dos casos, el tomador de decisiones fija dos limites llamados umbrales de acuerdo a sus intereses. Uno de ellos marca el limite de la indiferencia y el segundo la preferencia. Estos dos umbrales se denotan a lo largo del capitulo como Q_j y S_j respectivamente teniendo las siguientes afirmaciones:

1) Si $0 < UU_j < Q_j$ entonces se dice que la alternativa x es indiferente a la alternativa y con respecto al criterio j.

2) Si $Q_j < UU_j < S_j$ entonces la alternativa x es debilmente preferida a la alternativa y bajo el criterio j.

3) Si $S_j < UU_j$ se tiene que la alternativa x es estrictamente preferida a la alternativa y bajo el criterio j.

Para expresar en qué medida se puede afirmar que x es preferible a y bajo el criterio j se calcula el cociente $L_j(x,y)$ el cual lo llamaremos grado de credibilidad y se define como:

$$L_j(x,y) = (S_j(a(x,j))-\min\{\ a(y,j)-a(x,j)),\ S_j(a(x,j)\ \})\ /\ (S_j(a(x,j))-\min\{\ a(y,j)-a(x,j),\ Q_j(a(x,j))\ \}).$$

Si $a(y,j) <= a(x,j)$ implica que:

$$\min\{\ a(y,j)-a(x,j),\ S_j(a(x,j)\ \} = a(y,j)-a(x,j) \text{ pues } S_j(a(x,j))>0$$

además:

$$\min\{\ a(y,j)-a(x,j),\ Q_j(a(x,j))\ \} = a(y,j)-a(x,j) \text{ pues } Q_j(a(x,j))>0$$

entonces:

$$L_j(a(x,y)) = 1.$$

Si $a(y,j)-a(x,j) >= S_j(a(x,j))$ implica que :

$$\min\{\ a(y,j)-a(x,j),\ S_j(a(x,j))\ \} = S_j(a(x,j))$$

entonces:

$$L_j(a(x,j)) = 0.$$

En el cálculo del índice de concordancia del método que se describe, se toman en cuenta no únicamente el peso relativo de los criterios sino también el grado de credibilidad con el fin de reafirmar la preferencia de una alternativa sobre otra.

Es ésta una de las diferencias que se tienen respecto a los dos métodos anteriores y se define como:

$$C(x,y) = \sum_j \left(\left(w_j \,/\, \sum_j w_j\right) L_j(x,y) \right)$$

donde

w_j = peso del criterio j asignado por el tomador de decisiones.

También hay que tomar en cuenta la discordancia con la afirmación de que la alternativa x es al menos tan buena como la alternativa y bajo el criterio j.

Para lograr lo anterior, se calcula el índice de discordancia para todo par de alternativas x,y de la siguiente manera:

$$D_j(x,y) = \min(1, \max(0, (a(y,j)-a(x,j)-S_j(a(x,j)) \,/\, (V_j(a(x,j))-S_j(a(x,j)))))$$

para cada criterio j,

donde

$V_j(a(x,j))$ = función de veto definida por el tomador de decisiones para tomar en cuenta la diferencia $a(y,j)-a(x,j)$ y rechazar toda afirmación de que la alternativa x es preferibe a la alternativa y.

por lo que:

Si $a(y,j) - a(x,j) >= V_j(a(x,j))$ entonces $D_j(x,y) = 1$

Si $a(y,j) - a(x,j) < S_j(a(x,j))$ entonces $D_j(x,y) = 0$

Nótese que se guarda la siguiente relación:

$$0 < Q_j(a(x,j)) < S_j(a(x,j)) < V_j(a(x,j))$$

donde:

$a(x,j)$ es el valor de la alternativa x con respecto al criterio j.

Ya que se cuenta con los índices de concordancia y discordancia, el primero se debilita o disminuye con aquellos índices de discordancia que sean mayores que él si es el caso, de lo contrario, conserva su valor. Al índice debilitado lo llamaremos grado de credibilidad global y se define como:

$$L(x,y) = C(x,y) \quad \text{si} \quad F(x,y) = \text{ vacío}$$

donde

$$F(x,y) = \{ j / D_j(x,y) > C(x,y) \}$$

o

$$L(x,y) = \pi_{j \in F} \; (1 - D_j(x,y)) \; / \; (1 - C(x,y)) \quad \text{si} \quad F(x,y) \text{ es distinto del vacío.}$$

Se considera, en este método, que el grado de credibilidad global no es suficiente por lo que se define un umbral que se denota como $s(\sigma)$ y lo llamaremos umbral de discriminación.

Este umbral debe reflejar los intereses del tomador de decisiones, razón por la cual es él mismo quien lo define.

Una posible definición de este umbral sería:

$$s(\sigma) = \alpha + \beta(\sigma) \qquad 0 < \sigma < 1$$

donde α y β son constantes.

Dicho umbral, se aplica para comparar las alternativas de tal forma que:

$$\text{si } L(x,y) >= L(y,x) + s(L(x,y))$$

podemos afirmar con un grado de credibilidad significativo que la alternativa x es preferible a la alternativa y, y formar subconjuntos del conjunto total de alternativas en orden de preferencia de mayor a menor en el caso de la destilación descendente y de menor a mayor para el de la ascendente para finalmente obtener las clases.

A continuación se enumeran los pasos a seguir en el método Electra III:

3.6 ALGORITMO ELECTRA III.

El procedimiento utilizado se describe a continuación:

Paso 1. Calcular la matriz de concordancia entre dos alternativas (x,y) de la siguiente manera:

$$C(x,y) = \sum_j \left(\left(w_j / \sum_j w_j \right) \cdot L_j(x,y) \right)$$

con

$$L_j(x,y) = \left(S_j(a(x,j)) - \min\{ a(y,j)-a(x,j),\ S_j(a(x,j)) \}\right) / \left(S_j(a(x,j)) - \min\{ a(y,j)-a(x,j),\ Q_j(a(x,j)) \}\right).$$

donde

w_j = peso del criterio j asignado por el tomador de decisiones.

Paso 2. Calcular la matriz de discordancia para todo par de alternativas (x,y) tomando en cuenta el criterio j.

$$D_j(x,y) = \min(1, \max\{ 0,\ (a(y,j)-a(x,j)-S_j(a(x,j))) / (V_j(a(x,j)-S_j(a(x,j)))\}).$$

Paso 3. Calcular la matriz de grados de credibilidad para todo par de alternativas (x,y) como :

$$L(x,y) = C(x,y) \quad \text{si } F(x,y) = \{ j \mid D_j(x,\acute{y}) > C(x,y) \} = \text{vacio}$$

o en otro caso como

$$L(x,y) = C(x,y) \cdot \prod_{j \in F} (1-D_j(x,y)) / (1-C(x,y))$$

Paso 4. Se obtiene una destilación descendente.

Paso 5. Se obtiene una destilación ascendente.

Paso 6. Se obtiene una clasificación por suma de rangos.

A continuación se describen detalladamente los pasos 4 y 5 correspondientes a la destilación descendente y ascendente respectivamente aplicando los conceptos de umbral de discriminación y grado de credibilidad global.

ALGORITMO DE LA DESTILACION DESCENDENTE

1. Fijar: i=1
   Bi=X

2. Fijar: k=1
   Ok=Bi

3. Fijar: lambdak= $\max_{(x,y)}$ L(x,y) con (x,y)∈ Ok x Ok

4. Fijar: lambdak+1= $\max_{(x,y)}$ L(x,y) con (x,y)∈ Ok x Ok
   y L(x,y) < lambdak - s(lambdak).

5. Calcular para toda x∈ Ok:

i) pOk(x)=||{y|L(x,y)>lambdak+1, L(x,y)>L(y,x)+s(L(x,y)),y∈Ok}||
donde || {*} || es la cardinalidad del conjunto entre llaves.

ii) fOk(x)=||{y|L(x,y)>lambdak+1, L(y,x)>L(x,y)+s(l(y,x)),x∈Ok}||

iii) qOk(x) = pOk(x) - fOk(x).

6. Seleccionar aquellas alternativas $x \in Ok$, tales que:

$$qOk(x) = \max_{y} qOk(y) \text{ con } x \in Ok$$

a las alternativas seleccionadas las llamaremos conjunto Ok+1.

7. Si la cardinalidad de Ok+1 > 1 o lambdak+1 > 0:

i) Fijar: lambdak = lambdak+1

ii) Fijar: k = k+1

iii) ir al punto 4

en caso contrario: ir al punto 8

8. Fijar: Ci = Ok+1

9. Fijar: Bi+1 = Bi - Ci

10. Si la cardinalidad de Bi+1 > 1

i) Fijar i = i + 1

ii) ir al punto 2

en caso contrario: Ci+1 = Bi+1

11. Parar el algoritmo pues el conjunto X se ha dividido en las clases C1,C2,C3,,,Cn.

ALGORITMO DE LA DESTILACION ASCENDENTE

El algoritmo es igual al anterior excepto en el paso 6 en que:

$$qOk(x) = \min_{y} qOk(y) \text{ con } y \in Ok$$

y en el paso 11 que se remplaza por :

$$
\begin{aligned}
C1 &= Cn \\
C2 &= Cn-1 \\
C3 &= Cn-2 \\
&\vdots \\
Cn &= C1.
\end{aligned}
$$

Ejemplo 1.

Considérense ocho alternativas de un Aprovechamiento Hidráulico las cuales, se necesitan ordenar de acuerdo al grado de satisfacción que produce. Para ello, se ha considerado lo siguiente:

El conjunto de criterios de evaluación está constituido por:

Criterio 1	Máxima prevensión de inundaciones
Criterio 2	Uso múltiple del canal
Criterio 3	Incremento de plusvalía del área
Criterio 4	Máximo valor estético
Criterio 5	Incremento de solidaridad vecinal
Criterio 6	Mínimo número de relocalizaciones
Criterio 7	Mínimo costo del proyecto
Criterio 8	Mínimo costo de operación
Criterio 9	Mínima molestia en construcción
Criterio 10	Mínimos obstáculos legales

Los pesos de los criterios del 1 al 10 respectivamente son:

10,7,3,5,6,8,9,4,2 y 1.

La matriz de pesos de las alternativas se dá a continuación:

CRITERIOS:	1	2	3	4	5	6	7	8	9	10
ALTERNATIVA 1	1	3	1	1	1	1	8	6	8	2
ALTERNATIVA 2	6	7	5	4	2	8	2.8	8	7	2
ALTERNATIVA 3	5	5	7	7	7	5.9	4	1	4	7
ALTERNATIVA 4	6	6	8	8	8	4.3	3.3	1	4	6
ALTERNATIVA 5	4	4	4	5	5	6.6	4.7	1	3	5
ALTERNATIVA 6	8	2	3	2	4	7.4	5.6	5.3	2	4
ALTERNATIVA 7	4	3	3	3	4	7.7	2.7	8	1	8
ALTERNATIVA 8	7	8	6	6	6	2.7	1	1	6	1

Además se han definido las siguientes funciones de indiferencia, de preferencia estricta y de veto:

$Q_j(a(x,j)) = 0$, para $j=1,2,3,,,10$

$S_j(a(x,j)) = 2\ a(x,j)$, para $j=1,2,3,,,10$

$V_j(a(x,j)) = 10 + 0.8\ a(x,j)$ para $j=1,2,3,,,10$.

El umbral de discriminación está dado por:

s(lambda) = 0.4 + 0.1(lambda), con 0<lambda<1

A continuación se aplica el método Electra III para resolver el problema considerado.

Para obtener la matriz de concordancia es necesario calcular los indices de corcordancia para todo par de alternativas x,y.

Veamos como se calcula para las alternativas (8,2) y (1,3). Para ello, es necesario obtener el grado de credibilidad asociado a cada criterio ($L_j(8,2)$ y $L_j(1,3)$).

$L_1(8,2) = (2-\min\{6-7,2\}) / (2-\min\{6-7,0\}) = 1$
$L_2(8,2) = (2-\min\{7-8,2\}) / (2-\min\{7-8,0\}) = 1$
$L_3(8,2) = (2-\min\{5-6,2\}) / (2-\min\{5-6,0\}) = 1$
$L_4(8,2) = (2-\min\{4-6,2\}) / (2-\min\{4-6,0\}) = 1$
$L_5(8,2) = (2-\min\{2-6,2\}) / (2-\min\{2-6,0\}) = 1$
$L_6(8,2) = (2-\min\{8-2.7,2\}) / (2-\min\{8-2.7,0\}) = 0$
$L_7(8,2) = (2-\min\{2.8-1,2\}) / (2-\min\{2.8-1,0\}) = 0.1$
$L_8(8,2) = (2-\min\{8-1,2\}) / (2-\min\{8-1,0\}) = 0$
$L_9(8,2) = (2-\min\{7-6,2\}) / (2-\min\{7-6,0\}) = 0.5$
$L_{10}(8,2) = (2-\min\{2-1,2\}) / (2-\min\{2-1,0\}) = 0.5$

por lo que

$$C(8,2) = (10/55)(1)+(7/55)(1)+(3/55)(1)+(5/55)(1)+(6/55)(1)+ (8/55)(0)+(9/55)(0.1)+(4/55)(0)+(2/55)(1/2)+(1/55)(1/2).$$

por lo tanto $C(8,2) = 0.607 \sim 0.61$

$$L_1(1,3) = (2-\min\{5-1,2\}) / (2-\min\{5-1,0\}) = 0$$
$$L_2(1,3) = (2-\min\{5-3,2\}) / (2-\min\{5-3,0\}) = 0$$
$$L_3(1,3) = (2-\min\{7-1,2\}) / (2-\min\{7-1,0\}) = 0$$
$$L_4(1,3) = L_5(1,3) = L_3(1,3)$$
$$L_6(1,3) = (2-\min\{5.9-1,2\}) / (2-\min\{5.9-1,0\}) = 0$$
$$L_7(1,3) = (2-\min\{4-8,2\}) / (2-\min\{4-8,0\}) = 1$$
$$L_8(1,3) = (2-\min\{1-6,2\}) / (2-\min\{1-6,0\}) = 1$$
$$L_9(1,3) = L_7(1,3)$$
$$L_{10}(1,3) = (2-\min\{7-2,2\}) / (2-\min\{7-2,0\}) = 0$$

por lo tanto

$$C(1,3) = (10/55)(0)+(7/55)(0)+(3/55)(0)+(5/55)(0)+(6/55)(0)+ (8/55)(0)+(9/55)(1)+(4/55)(1)+(2/55)(1)+(1/55)(0).$$

$C(1,3) = 0.27.$

Siguiendo el mismo procedimiento se llega a la siguiente matriz de concordancia:

1.00	0.27	0.27	0.27	0.34	0.45	0.33	0.31
0.82	1.00	0.63	0.69	0.67	0.53	0.87	0.62
0.73	0.53	1.00	0.72	0.89	0.51	0.79	0.65
0.73	0.68	0.82	1.00	0.74	0.44	0.76	0.75
0.73	0.45	0.55	0.41	1.00	0.61	0.83	0.50
0.71	0.57	0.56	0.56	0.67	1.00	0.78	0.58
0.80	0.52	0.38	0.35	0.56	0.64	1.00	0.40
0.72	0.61	0.55	0.45	0.67	0.51	0.62	1.00

Para obtener la matriz de grados de credibilidad, se debe calcular primero la matriz de discordancia.

Como ejemplo se obtendrán los indices de discordancia para las alternativas (8,2) y (3,1).

$D_1(8,2) = \min\{1,\max\{0,(6-7-2)/(10+0.8(7)-2)\}\} = 0$

$D_2(8,2) = \min\{1,\max\{0,(7-8-2)/(10+0.8(8)-2)\}\} = 0$

$D_3(8,2) = \min\{1,\max\{0,(5-6-2)/(10+0.8(6)-2)\}\} = 0$

$D_4(8,2) = \min\{1,\max\{0,(4-6-2)/(10+0.8(6)-2)\}\} = 0$

$D_5(8,2) = \min\{1,\max\{0,(2-6-2)/(10+0.8(6)-2)\}\} = 0$

$D_6(8,2) = \min\{1,\max\{0,(8-2.7-2)/(10+0.8(2.7)-2)\}\} =0.3248$

$D_7(8,2) = \min\{1,\max\{0,(2.8-1-2)/(10+0.8(1)-2)\}\} = 0$

$D_8(8,2) = \min\{1,\max\{0,(8-1-2)/(10+0.8(1)-2)\}\} = 0.5681$

$D_9(8,2) = \min\{1,\max\{0,(7-6-2)/(10+0.8(6)-2)\}\} = 0$

$D_{10}(8,2) = \min\{1,\max\{0,(2-1-2)/(10+0.8(1)-2)\}\} = 0$

Como $C(8,2) = 0.61 > 0.5681 > 0.3248 > 0$

implica que $L(8,2)= 0.61$

$D_1(1,3) = \min\{1,\max(0,(5-1-2)/(10=0.8(1)-2))\} = 0.27$

$D_2(1,3) = \min\{1,\max(0,(5-3-2)/(10+0.8(3)-2))\} = 0$

$D_3(1,3) = \min\{1,\max(0,(7-1-2)/(10+0.8(1)-2))\} = 0.4545$

$D_5(1,3) = D_4(1,3) = D_3(1,3)$

$D_6(1,3) = \min\{1,\max(0,(5.9-1-2)/(10+0.8(1)-2))\} = 0.3295$

$D_7(1,3) = \min\{1,\max(0,(4-8-2)/(10+0.8(8)-2))\} = 0$

$D_8(1,3) = \min\{1,\max(0,(1-6-2)/(10+0.8(6)-2))\} = 0$

$D_9(1,3) = \min\{1,\max(0,(4-8-2)/(10+0.8(8)-2))\} = 0$

$D_{10}(1,3) = \min\{1,\max(0,(7-2-2)/(10+0.8(2)-2))\} = 0$

entonces:

$$\begin{aligned} L(1,3) &= 0.27[(1-0.4545)/(1-0.27)]\ \ [(1-0.3295)/(1-0.27)] \\ &\quad [(1-0.3125)(1-0.27)] \\ &= 0.097 \sim 0.10 \end{aligned}$$

Siguiendo el mismo procedimiento se llega a la matriz de grados de credibilidad:

1.00	0.15	0.10	0.05	0.30	0.31	0.20	0.22
0.82	1.00	0.63	0.69	0.67	0.53	0.87	0.62
0.73	0.48	1.00	0.72	0.89	0.51	0.79	0.65
0.73	0.68	0.82	1.00	0.74	0.44	0.76	0.75
0.73	0.36	0.55	0.41	1.00	0.61	0.83	0.50
0.71	0.57	0.56	0.56	0.67	1.00	0.78	0.58
0.80	0.52	0.38	0.35	0.56	0.64	1.00	0.40
0.72	0.61	0.55	0.45	0.67	0.51	0.62	1.00

Con la matriz anterior se puede obtener la destilación descendente y ascendente.

Daremos un ejemplo para obtener la lambda-potencia de la alternativa x y la lambda-debilidad de x (pok(x) y fok(x) respectivamente.

Para lambda1 = 0.89 y lambda2 = 0 se tiene:

po1(1)=0 y fo1(1)=5   ya que

1.00 < 1.00+0.4+0.1(1.00),
0.15 < 0.82+0.4+0.1(0.15),
0.10 < 0.73+0.4+0.1(0.10),
0.05 < 0.73+0.4+0.1(0.05),
0.30 < 0.73+0.4+0.1(0.30),
0.31 < 0.71+0.4+0.1(0.31),
0.20 < 0.80+0.4+0.1(0.20),
0.22 < 0.72+0.4+0.1(0.22)

lo cual implica que po1(1)=0.

Además:

1.00 < 1.00+0.4+0.1(1.00),
0.82 > 0.15+0.4+0.1(0.82),
0.73 > 0.10+0.4+0.1(0.73),
0.73 > 0.05+0.4+0.1(0.73),
0.73 < 0.30+0.4+0.1(0.73),
0.71 < 0.31+0.4+0.1(0.71),
0.80 > 0.20+0.4+0.1(0.80),
0.72 > 0.22+0.4+0.1(0.72).

lo cual implica que fo1(1)=5.

Siguiendo los pasos descritos en el algoritmo anterior se llega a:

DESTILACION DESCENDENTE

ALTERNATIVA	CLASE
1	2
2	1
3	1
4	1
5	2
6	2
7	1
8	1

DESTILACION ASCENDENTE

ALTERNATIVA	CLASE
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	1

SOLUCION

ALTERNATIVA	CLASE
1	8
2	2
3	3
4	4
5	7
6	1
7	5
8	6

Esta solución implica que dentro de las ocho alternativas que se tienen, la de mayor preferencia es la alternativa 6, le sigue en menor preferencia la alternativa 2, a continuación la alternativa 3, después la alternativa 4, la alternativa 7, en seguida la alternativa 8, después la alternativa 5 y finalmente, la alternativa 1 es la que produce un menor grado de satisfacción dentro del conjunto de alternativas.

CAPITULO IV

En el capitulo anterior se explicó ampliamente los métodos Electra I, Electra II y Electra III. Además, se ejemplificó su aplicación a través de problemas que llevan tiempo en resolverlos debido a la cantidad de operaciones que es necesario realizar.

Mientras mayor sea el número de alternativas y de criterios mayor es el número de operaciones y por lo tanto se vuelve engorroso el aplicar manualmente alguno de estos métodos. Es por ello que en este trabajo se crean programas de cómputo para que su manejo sea rápido y sencillo.

Estos programas fueron creados en una microcomputadora PC IBM-compatible utilizando el lenguaje C. Este lenguaje es de alto nivel y el manejo de cadenas es sencillo lo que facilita la impresión de datos en un archivo.

En este capitulo se desarrolla un manual que describe la organización de los programas de cómputo realizados incluyendo la explicación de las subrutinas utilizadas. También se explica el manejo de dichos programas y ejemplos de los mismos.

Algunos de estos ejemplos son resueltos en el capitulo III con el fin de comparar las soluciones y observar que efectivamente son idénticas.

4.1 PROGRAMA ELECTRA I.

Para llamar a este programa se teclea: electra1.c. Dicho programa ofrece 5 opciones:

1) Listado de un archivo
2) Generar un archivo
3) Modificación de un archivo
4) Proceso de un archivo
5) Terminar

A continuación se pide el número de la opción deseada y el nombre del archivo a utilizar.

Si se desea generar un archivo se teclea la opción 2. En caso de que ya no se requiera un archivo creado previamente y se necesite generar otro, se puede escoger dicha opción y dar el nombre del archivo existente.

Posteriormente se piden los siguientes datos:

- Número de alternativas
- Número de criterios
- Nivel de concordancia P
- Nivel de discordancia Q
- Valores de criterios para cada alternativa
- Tipo de criterio
- Pesos de los criterios

Para aquella información en donde se requiere teclear más de un número es necesario separar los datos por comas y al terminar oprimir la tecla ‹enter›. En caso contrario basta teclear el número y oprimir ‹enter›.

Los valores de criterios para cada alternativa deben ser positivos y el tipo de criterio será 1 si se prefiere la alternativa i a la alternativa j cuando el valor de la primera es mayor que el de la segunda. Sin embargo, si se prefiere la alternativa i a la alternativa j cuando el valor de i es menor que el de j, el tipo de criterio será -1. Con ésto no es necesario dar valores de criterios negativos para las alternativas por lo que se ha descartado dicho caso.

Si se desea listar un archivo hay que escoger la opción 1. En dicha opción se despliega la información o datos que contiene el archivo que se está utilizando.

Para modificar un archivo se teclea la opción 3. Esta subrutina nos permite modificar cualquiera de los siguientes datos:

1) Nivel de concordancia P
2) Nivel de discordancia Q
3) Valores de alternativas
4) Tipo de criterio
5) Pesos de los criterios

Después de escoger el tipo de dato a modificar se teclean los nuevos valores (separados por comas en caso de que sea más de un dato) y terminamos oprimiendo <enter>.

Para el cambio en valores de alternativas se pregunta al usuario por el número de alternativa en la cual se desea hacer la modificación por lo que no es necesario actualizar los valores de todas las alternativas si unicamente se quiere cambiar los de una en particular.

Se procesa un archivo si tecleamos el número 4. Con dicho procedimiento obtenemos la información del archivo que se está utilizando y posteriormente se obtienen los siguientes resultados:

- matriz de concordancia y matriz de discordancia: el elemento (i,j) de la primera matriz equivale al valor $C(i,j)$ mientras que en la segunda corresponde al valor $D(i,j)$.

- gráfica-matriz: en donde si la coordenada $(i,j)=1$ implica que existe un arco de la alternativa i a la alternativa j pues $C(i,j)>=P$ y $D(i,j)<=Q$.

- matriz acíclica: representa la gráfica anterior pero sin ciclos.

- solución: se presenta un conjunto de números que equivale a las alternativas que forman el kernel.

Finalmente, se escoge la opción 5 en caso de que se desee salir del programa.

A continuación, se ejemplificará el modo de operación del programa Electra I tomando como referencia el ejemplo II de la página 75 cuyo problema es el de escoger la universidad de mayor conveniencia para el profesor Ramírez de cinco universidades en donde le ofrecen trabajo.

Para llamar al programa se teclea Electrai y se presiona ‹enter›; inmediatamente se observa en la pantalla cinco opciones las cuales fueron descritas anteriormente. Debido a que no existe un archivo con los datos del problema a tratar, es necesario crearlo a través de la opción 2 que nos ofrece generarlo.

Después de presionar el número 2, se teclea el nombre del archivo para identificarlo y se comienza a proporcionar los datos de acuerdo al orden pedido, recordando, separarlos por comas como se muestra posteriormente en la ilustración.

Para proporcionar los datos del tipo de criterio, estos deben ser 1 ó -1: en el ejemplo , el tipo de criterio 5 es -1 ya que se prefiere la alternativa i a la alternativa j mientras **menor** sea el costo del equipo y de la planta, a diferencia del tipo de criterio 7 que es 1 ya que se prefiere la alternativa i a la j mientras **mayor** sea el aprovechamiento económico que proporcione.

Al finalizar es conveniente escoger la primera opción del menú principal para listar el archivo generado y en caso de haber tecleado incorrectamente un dato, continuar con la opción número tres para la modificación del mismo.

En caso de que los datos sean los correctos, el siguiente paso es procesar un archivo. Al teclear el número 4 dentro del menú principal, se muestran los datos que se encuentran en el archivo y posteriormente se genera la matriz de concordancia y la de discordancia; así mismo, se genera la gráfica matriz en donde la coordenada (i,j) = 1 implica que iRj, es decir, C(i,j) >=p y D(i,j) <=q ; en caso contrario, no se cumple esta relación.

Para el ejemplo que nos ocupa, en la gráfica matriz existe un uno en la coordenada (2,1), (4,2) y (4,5) lo que implica que existe un arco del nodo 2 al nodo 1, del 4 al 2 y del nodo 4 al 5. Esta gráfica es acíclica por lo cual es idéntica a la matriz acíclica.

Finalmente, se obtiene la solución {3,4} lo que implica que el problema se ha reducido a únicamente dos alternativas en lugar de las cinco alternativas que se tenían inicialmente.

Para un análisis de sensibilidad es necesario modificar los índices de concordancia y discordancia por lo que la opción a escoger es la número 3. Después de teclear el nombre del archivo a modificar se elige la opción 1 para cambiar el indice de concordancia de 0.6 a 0.5 y se responde que si se desea seguir modificando para alterar el indice de discordancia de 0.8 a 0.7 a través de la opción número 2.

Para terminar de modificar se niega a la pregunta del deseo de seguir modificando.

Al procesar el nuevo archivo, se obtiene como solución la alternativa 4 lo que significa que es este el sistema de mayor conveniencia.

A continuación se ilustrará el modo de operación del programa Electra I.

```
B>electral
Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   2

NOMBRE DEL ARCHIVO DE DATOS A GENERAR: ramirez.dat
NUMERO DE ALTERNATIVAS (2-30)  ?   5

NUMERO DE CRITERIOS (2-20)  ?   6

NIVEL DE CONCORDANCIA P (0<P<1)  ?   0.3

NIVEL DE DISCORDANCIA Q (0<P<1)  ?   0.25

VALORES DE CRITERIOS PARA LA ALTERNATIVA  1
80,76,20,80.3,65.2,30

VALORES DE CRITERIOS PARA LA ALTERNATIVA  2
60,40,65,60,62.5,10

VALORES DE CRITERIOS PARA LA ALTERNATIVA  3
120,130,140,30,25,70

VALORES DE CRITERIOS PARA LA ALTERNATIVA  4
150,100,27.5,120,-12.5,100

VALORES DE CRITERIOS PARA LA ALTERNATIVA  5
30,10,102.5,150,100,-20

TIPO DE CRITERIO (VALORES SEPARADOS POR COMAS) 1,-1,-1,1,-1,-1

PESOS DE LOS CRITERIOS: 4,2,3.5,4.3,2,1



Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
```

```
  Opcion deseada (1-5) ?   1

NOMBRE DEL ARCHIVO DE DATOS A LISTAR: ramirez.dat
NUMERO DE ALTERNATIVAS=  5
NUMERO DE CRITERIOS=  6

NIVEL DE CONCORDANCIA P=  0.300000
NIVEL DE DISCORDANCIA Q=  0.250000

V A L O R E S  D E  L A S  A L T E R N A T I V A S  VS.  C R I T E R I O S
 CRIT.     1       2       3       4       5       6
ALT. 1    80.00   76.00   20.00   80.30   65.20   30.00
ALT. 2    60.00   40.00   65.00   60.00   62.50   10.00
ALT. 3    120.00  130.00  140.00  30.00   25.00   70.00
ALT. 4    150.00  100.00  27.50  120.00  -12.50  100.00
ALT. 5    30.00   10.00  102.50  150.00  100.00  -20.00


CRITERIO          TIPO
    1             1
    2             -1
    3             -1
    4             1
    5             -1
    6             -1


CRITERIO          PESO
    1          4.000000
    2          2.000000
    3          3.500000
    4          4.300000
    5          2.000000
    6          1.000000

Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
```

Opcion deseada (1-5) ?   4

NOMBRE DEL ARCHIVO DE DATOS A PROCESAR: ramirez.dat
NUMERO DE ALTERNATIVAS =5
NUMERO DE CRITERIOS =6

COEFICIENTE DE CONCORDANCIA = 0.300000
COEFICIENTE DE DISCORDANCIA = 0.250000
TABLA DE UTILIDADES : ALTERNATIVA VS. CRITERIO

CRIT.	1	2	3	4	5	6
ALT. 1	80.00	76.00	20.00	80.30	65.20	30.00
ALT. 2	60.00	40.00	65.00	60.00	62.50	10.00
ALT. 3	120.00	130.00	140.00	30.00	25.00	70.00
ALT. 4	150.00	100.00	27.50	120.00	-12.50	100.00
ALT. 5	30.00	10.00	102.50	150.00	100.00	-20.00

TIPO ASIGNADO AL CRITERIO 1 = 1
TIPO ASIGNADO AL CRITERIO 2 = -1
TIPO ASIGNADO AL CRITERIO 3 = -1
TIPO ASIGNADO AL CRITERIO 4 = 1
TIPO ASIGNADO AL CRITERIO 5 = -1
TIPO ASIGNADO AL CRITERIO 6 = -1

PESO ASIGNADO AL CRITERIO 1 = 4.000000
PESO ASIGNADO AL CRITERIO 2 = 2.000000
PESO ASIGNADO AL CRITERIO 3 = 3.500000
PESO ASIGNADO AL CRITERIO 4 = 4.300000
PESO ASIGNADO AL CRITERIO 5 = 2.000000
PESO ASIGNADO AL CRITERIO 6 = 1.000000

MATRIZ DE CONCORDANCIA

0.00	0.70	0.64	0.39	0.57
0.30	0.00	0.64	0.18	0.57
0.36	0.36	0.00	0.06	0.36
0.61	0.82	0.94	0.00	0.57
0.43	0.43	0.64	0.43	0.00

MATRIZ DE DISCORDANCIA

0.00	0.24	0.27	0.52	0.46
0.30	0.00	0.40	0.60	0.60
0.80	0.60	0.00	0.75	0.80
0.47	0.60	0.20	0.00	0.80
0.55	0.25	0.60	0.80	0.00

GRAFICA-MATRIZ

0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0

MATRIZ ACICLICA:

```
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 1 0 0 0
```

NOTA: si la coordenada (i,j)=1 significa que existe un arco del nodo i al nodo j.

```
S O L U C I O N
alternativas:  1, 4, 5,

Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   3

NOMBRE DEL ARCHIVO DE DATOS A MODIFICAR: ramirez.dat
EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVEL DE CONCORDANCIA P
2) NIVEL DE DISCORDANCIA Q
3) VALORES DE ALTERNATIVAS
4) TIPO DE CRITERIO
5) PESOS DE LOS CRITERIOS
Opcion deseada (1-5) ?  1

DAR NIVEL DE CONCORDANCIA P (0<P<1) ?   0.5
DESEA SEGUIR MODIFICANDO (SI,NO) ?  si

EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVEL DE CONCORDANCIA P
2) NIVEL DE DISCORDANCIA Q
3) VALORES DE ALTERNATIVAS
4) TIPO DE CRITERIO
5) PESOS DE LOS CRITERIOS
Opcion deseada (1-5) ?  2

DAR NIVELES DE DISCORDANCIA Q (0<Q<1) ?   0.8
DESEA SEGUIR MODIFICANDO (SI,NO) ?  si

EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVEL DE CONCORDANCIA P
2) NIVEL DE DISCORDANCIA Q
3) VALORES DE ALTERNATIVAS
4) TIPO DE CRITERIO
5) PESOS DE LOS CRITERIOS
Opcion deseada (1-5) ?  3
```

```
EN QUE ALTERNATIVA ?  1
DAR VALORES DE LOS CRITERIOS  90,70,27.5,90,62.5,40

CAMBIOS EN OTRA ALTERNATIVA (SI,NO) ?  no
DESEA SEGUIR MODIFICANDO (SI,NO) ?  si

EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVEL DE CONCORDANCIA P
2) NIVEL DE DISCORDANCIA Q
3) VALORES DE ALTERNATIVAS
4) TIPO DE CRITERIO
5) PESOS DE LOS CRITERIOS
Opcion deseada (1-5) ?  5

DAR PESOS A LOS CRITERIOS 5,3,4,5,2,2
DESEA SEGUIR MODIFICANDO (SI,NO) ?  no


Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   1

NOMBRE DEL ARCHIVO DE DATOS A LISTAR: ramirez.dat
NUMERO DE ALTERNATIVAS=  5
NUMERO DE CRITERIOS=  6

NIVEL DE CONCORDANCIA P=  0.500000
NIVEL DE DISCORDANCIA Q=  0.800000

V A L O R E S  D E  L A S  A L T E R N A T I V A S  VS.  C R I T E R I O S
 CRIT.     1        2        3        4        5        6
ALT. 1    90.00   70.00   27.50   90.00   62.50   40.00
ALT. 2    60.00   40.00   65.00   60.00   62.50   10.00
ALT. 3    120.00  130.00  140.00  30.00   25.00   70.00
ALT. 4    150.00  100.00  27.50   120.00  -12.50  100.00
ALT. 5    30.00   10.00   102.50  150.00  100.00  -20.00


CRITERIO          TIPO
    1             1
    2             -1
    3             -1
    4             1
    5             -1
    6             -1


CRITERIO          PESO
    1           5.000000
    2           3.000000
    3           4.000000
    4           5.000000
    5           2.000000
```

```
    6          2.000000

Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   4

NOMBRE DEL ARCHIVO DE DATOS A PROCESAR: ramirez.dat
NUMERO DE ALTERNATIVAS =5
NUMERO DE CRITERIOS =6

COEFICIENTE DE CONCORDANCIA = 0.500000
COEFICIENTE DE DISCORDANCIA = 0.800000
TABLA DE UTILIDADES : ALTERNATIVA VS. CRITERIO

 CRIT.    1       2       3       4       5       6
ALT. 1   90.00   70.00   27.50   90.00   62.50   40.00
ALT. 2   60.00   40.00   65.00   60.00   62.50   10.00
ALT. 3   120.00  130.00  140.00  30.00   25.00   70.00
ALT. 4   150.00  100.00  27.50   120.00  -12.50  100.00
ALT. 5   30.00   10.00   102.50  150.00  100.00  -20.00

TIPO ASIGNADO AL CRITERIO  1 =  1
TIPO ASIGNADO AL CRITERIO  2 =  -1
TIPO ASIGNADO AL CRITERIO  3 =  -1
TIPO ASIGNADO AL CRITERIO  4 =  1
TIPO ASIGNADO AL CRITERIO  5 =  -1
TIPO ASIGNADO AL CRITERIO  6 =  -1

PESO ASIGNADO AL CRITERIO  1 =   5.000000
PESO ASIGNADO AL CRITERIO  2 =   3.000000
PESO ASIGNADO AL CRITERIO  3 =   4.000000
PESO ASIGNADO AL CRITERIO  4 =   5.000000
PESO ASIGNADO AL CRITERIO  5 =   2.000000
PESO ASIGNADO AL CRITERIO  6 =   2.000000


MATRIZ DE CONCORDANCIA

0.00  0.71  0.67  0.33  0.52
0.29  0.00  0.67  0.24  0.52
0.33  0.33  0.00  0.10  0.33
0.67  0.76  0.90  0.00  0.52
0.48  0.48  0.67  0.48  0.00


MATRIZ DE DISCORDANCIA

0.00  0.20  0.25  0.50  0.40
0.25  0.00  0.40  0.60  0.60
0.75  0.60  0.00  0.75  0.80
0.40  0.60  0.20  0.00  0.80
0.50  0.25  0.60  0.80  0.00
```

```
GRAFICA-MATRIZ
0 1 1 0 1
0 0 1 0 1
0 0 0 0 0
1 1 1 0 1
0 0 1 0 0


MATRIZ ACICLICA:
0 1 1 0 1
0 0 1 0 1
0 0 0 0 0
1 1 1 0 1
0 0 1 0 0


NOTA: si la coordenada (i,j)=1 significa que existe un arco del
      nodo i al nodo j.


S O L U C I O N
alternativas:  4,

Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   5

termina el proceso

B>
```

4.2 ESTRUCTURA DEL PROGRAMA.

El programa consta de las siguientes subrutinas:

main: subrutina que despliega el menú principal.

abrir: subrutina para abrir un archivo (todos los archivos tienen que abrirse antes de introducir información).

cerrar: subrutina que cierra un archivo (todos los archivos tienen que cerrarse después de terminar de introducir la información para no perderla).

generar: subrutina que genera un archivo.

modificar: subrutina que modifica un archivo.

listado: subrutina que lista la información del archivo especificado.

proceso: procesa un archivo. Además, contiene a la subrutina ciclos que obtiene una gráfica aciclica y una función mx que calcula el máximo entre dos valores.

Para generar un archivo basta con preguntar los datos, leerlos y grabarlos en el archivo.

La subrutina proceso es la más compleja pues como su nombre lo indica realiza todas las operaciones para obtener la solución final.

Lo primero que realiza es la lectura de los datos que se tienen en el archivo y desplegar la información; al mismo tiempo, calcula el rango máximo de las escalas a través de la función mx.

Para calcular la matriz de concordancia y la de discordancia suma todos los pesos de los criterios guardándo el resultado en una variable y en un registro el tipo de criterio. En caso de que el tipo de criterio sea positivo pregunta si el valor de la alternativa i es mayor que el de la j o si son iguales y de acuerdo a la respuesta acumula el peso del criterio en las variables correspondientes, pero si el valor de la alternativa i es menor que el de j, calcula el índice de discordancia utilizando la función mx. Si el tipo de criterio es negativo y el valor de la alternativa i es menor que el de la j o si son iguales acumula el peso del criterio en las variables correspondientes, en caso contrario, calcula el indice de discordancia.

A continuación se obtiene el indice de concordancia y si éste es mayor o igual al nivel de concordancia y el indice de discordancia menor al nivel de discordancia se guarda un 1 en la gráfica matriz Gc, en caso contrario se guarda un cero. Se despliega la matriz de concordancia, discordancia y la gráfica matriz.

Para obtener una gráfica aciclica, existe el algoritmo de Floyd para rutas más cortas en donde se detectan circuitos negativos por lo que con una modificación se logra encontrar los dos que forman un ciclo en una gráfica. Dicho procedimiento se explica a continuación:

Se construye una matriz de nxn donde n es el número de nodos.

$$M_{ij} = \begin{cases} 0 & \text{si } i = j. \\ 99 & \text{si la coordenada } (i,j) \text{ de la gráfica matriz es cero.} \\ -1 & \text{en otro caso.} \end{cases}$$

Paso 1. Hacer k = 0

Paso 2. Hacer k = k + 1. Para i distinta de k tal que M_{ik} es diferente de 99 y para j distinta de k tal que M_{kj} es diferente de 99 hacer:

$$M_{ij} = \min\{ M_{ij},\ M_{ik} + M_{kj} \}$$

Paso 3. Si $M_{ii} < 0$ para alguna i, entonces existe un ciclo.

Si $M_{ii} >= 0$ para toda i y k = n, entonces no existen ciclos.

Si $M_{ii} >= 0$ para toda i y k < n, ir al paso 2.

Si existe un ciclo, es necesario saber que nodos lo forman por lo que se utiliza una matriz P de nxn en donde la coordenada (i,j) es el predecesor del nodo j.

Esta gráfica se inicializa P_{ij} = i para todo nodo i,j. La gráfica se modifica en el paso 2 del algoritmo anterior de la siguiente manera:

$$P_{ij} = \begin{cases} P_{kj} & \text{si } M_{ik} + M_{kj} < M_{ij} \\ \text{no cambia en otro caso} & \end{cases}$$

Al detectar un ciclo en el nodo i basta con fijarnos en la coordenada (i,i) de la matríz P para obtener cual es el predecesor (supongamos que es j), después fijarnos en el predecesor de la coordenada (i,j) (supongamos que es k) y así sucesivamente hasta llegar a que el predecesor de (i,s) es i por lo que el ciclo está formado por los nodos i,j,k,,,s.

A continuación, de la gráfica matriz , se cambian las entradas (i,j), (j,k),,,(k,s) y (s,i) por cero y se despliega la nueva matríz llamada matriz acíclica.

Finalmente, para obtener el kernel, se recorre esta última por columnas y en caso de que las entradas sean cero, el número de columna pertenece al conjunto buscado.

Para modificar un archivo se leen los datos que contiene éste, se pregunta por los nuevos datos y se graban en el mismo archivo.

4.3 PROGRAMA ELECTRA II.

La llamada del programa se realiza tecleando electra2. Dicho programa se inicia con cinco opciones:

1) Listado de un archivo
2) Generar un archivo
3) Modificación de un archivo
4) Proceso de un archivo
5) Terminar

A continuación se pide una opción y el nombre del archivo a utilizar.

Se iniciará con la opción 2) para generar un archivo.

NOTA:

En caso de que ya no se requiera un archivo creado previamente y se necesite generar otro, se puede escoger dicha opción y dar el nombre del archivo existente.

En esta etapa del programa se proporciona la siguiente información:

- Número de alternativas
- Número de criterios
- Niveles de concordancia: P-,P0,P^*
- Niveles de discordancia: O0, O^*
- Valores de las alternativas desde los diferentes puntos de vista o criterios
- Valores de S(k)
- Respuesta al requerimiento de una discordancia dinámica
- Pesos de los criterios

Para el caso de que se requiera un solo valor numérico para dar la información, basta con teclear dicho valor y oprimir la tecla ‹enter›. Para el caso contrario (varios valores numéricos) los valores deben de ir separados por comas y al finalizar oprimir la tecla ‹enter›.

La información se teclea en el lugar donde se encuentra el cursor ya que si queremos dejar un espacio o un renglón en blanco antes de dar la información se tomarán dichos caracteres como el valor pedido.

Una vez generado un archivo (o varios) éste puede ser listado, procesado o modificado.

En caso de que se desee la opción de listado del programa se proporcionará la información que se tecleó al crear el archivo.

Para la modificación de un archivo se elige la opción 3. Dentro de dicha opción se puede modificar la siguiente información:

- Niveles de concordancia P-,P0,P*
- Niveles de discordancia Q0,Q*
- Valores de alternativas
- Valores de teta
- Pesos de los criterios

Para la primera y segunda opción el usuario proporcionará los nuevos niveles.

En caso de elegir la tercera opción es necesario indicar cual alternativa se desea modificar y a continuación proporcionar los nuevos valores.

Para la cuarta opción se tienen que teclear los nuevos valores de teta separados por comas así como para la quinta opción al indicar los nuevos pesos de los criterios.

Para procesar un archivo se indica la cuarta opción. Finalmente para salir del programa se elige la opción 5.

A continuación se ejemplificará el modo de operación del programa Electra II a través del problema descrito en la página 90 el cual considera un Banco con cinco áreas y el propósito de conocer cuáles se encuentran en mejores condiciones y cuáles tienen más dificultades.

Para utilizar este programa, se teclea Electra2 e inmediatamente se despliegan cinco opciones y en este caso se escogerá la opción número dos para generar un archivo ya que no se cuenta con uno que contenga los datos del problema a tratar.

En seguida, se teclea el nombre del archivo que se va a generar e inmediatamente se preguntan los datos que se proporcionan separados por comas como se muestra en la ilustración presentada posteriormente.

Al finalizar, se despliega de nuevo el menú principal y para estar seguros de que los datos del archivo son los correctos es conveniente elegir la opción uno para listarlos. En caso de que exista algún dato erróneo, se escoge la opción tres para modificarlos. En caso contrario, se puede procesar el archivo a través de la opción cuatro.

El proceso de un archivo consiste en listar los datos proporcionados, desplegar la matriz de concordancia y la de discordancia. A continuación, se despliega la gráfica matriz (acíclica) de relaciones fuertes que en el problema utilizado existe un uno en las coordenadas (1,5),(2,1),(2,3),(2,5),(3,5) y (4,5) lo que implica que se cumple cualquiera de las siguientes restricciones:

$$C(i,j) >= p^{*} \qquad\qquad C(i,j) >= p0$$
$$D(i,j) <= q^{*} \qquad \text{o} \qquad D(i,j) <= q0$$
$$\sum_{k\in I^{+}} W_{k} >= \sum_{k\in I^{-}} W_{k} \qquad\qquad \sum_{k\in I^{+}} W_{k} >= \sum_{k\in I^{-}} W_{k}$$

Así mismo, se despliega la gráfica matriz (acíclica) de relaciones débiles que en el ejemplo utilizado es idéntica a la de relaciones fuertes ya que se cumple que:

$$C(i,j) >= p-$$
$$D(i,j) <= q+$$
$$\sum_{k\in I^{+}} W_{k} >= \sum_{k\in I^{-}} W_{k}$$

para las mismas parejas de alternativas.

Estas dos matrices representan gráficas acíclicas, es decir, sin ciclos. Finalmente, se obtiene la siguiente solución:

ALTERNATIVA 2	RANGO 1
ALTERNATIVA 4	RANGO 2
ALTERNATIVA 1	RANGO 3
ALTERNATIVA 3	RANGO 4
ALTERNATIVA 5	RANGO 5

lo que implica que el área con mejores condiciones es la número dos, le sigue la número cuatro, después la uno, la tres y la de peores condiciones es la número cinco.

En caso de corregir algún dato en el archivo, se escoge la opción tres y si se requiere de cambios en todos los datos se consigue afirmando el deseo de continuar modificando como se demuestra en la siguiente ilustración:

```
electra2
Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   2

NOMBRE DEL ARCHIVO DE DATOS A GENERAR: banco.dat
NUMERO DE ALTERNATIVAS (2-30)  ?   5

NUMERO DE CRITERIOS (2-20)  ?   4

NIVELES DE CONCORDANCIA P-,PO,P*  (0<P-<PO<P*<1) ?   0.6,0.65,0.75

NIVELES DE DISCORDANCIA QO,Q*  (0<QO<Q*<1) ?   0.2,0.65

VALORES DE CRITERIOS PARA LA ALTERNATIVA  1
0.81,0.5,-0.27,-0.6

VALORES DE CRITERIOS PARA LA ALTERNATIVA  2
0.8,0.52,-0.3,-0.58

VALORES DE CRITERIOS PARA LA ALTERNATIVA  3
0.9,0.47,-0.22,-0.59

VALORES DE CRITERIOS PARA LA ALTERNATIVA  4
0.75,0.6,-0.4,-0.62

VALORES DE CRITERIOS PARA LA ALTERNATIVA  5
0.78,0.41,-0.25,-0.62

CRITERIO                 VALOR DE S(K) CORRESPONDIENTE
   1                            0
   2                            0.3
   3                            0
   4                            0.2


SE QUIERE UNA DISCORDANCIA DINAMICA (SI,NO)  ? si

PESOS DE LOS CRITERIOS: 10 ,4,1,3



Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   1
```

NOMBRE DEL ARCHIVO DE DATOS A LISTAR: banco.dat

NUMERO DE ALTERNATIVAS= 5

NUMERO DE CRITERIOS= 4

NIVEL DE CONCORDANCIA P-= 0.600000
NIVEL DE CONCORDANCIA PO= 0.650000
NIVEL DE CONCORDANCIA P*= 0.750000

NIVEL DE DISCORDANCIA QO= 0.200000
NIVEL DE DISCORDANCIA Q*= 0.650000

V A L O R E S D E L A S A L T E R N A T I V A S VS. C R I T E R I O S

CRIT.	1	2	3	4
ALT.1	0.81	0.50	-0.27	-0.60
ALT.2	0.80	0.52	-0.30	-0.58
ALT.3	0.90	0.47	-0.22	-0.59
ALT.4	0.73	0.60	-0.40	-0.62
ALT.5	0.78	0.41	-0.25	-0.62

CRITERIO	VALOR DE S(K)
1	0.000000
2	0.300000
3	0.000000
4	0.200000

VALORES DE TETA = VARIAN

CRITERIO	PESO
1	1.000000
2	4.000000
3	1.000000
4	3.000000

Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
Opcion deseada (1-5) ? 4

NOMBRE DEL ARCHIVO DE DATOS A PROCESAR: banco.dat
NUMERO DE ALTERNATIVAS =5
NUMERO DE CRITERIOS =4

COEFICIENTE DE CONCORDANCIA DEBIL = 0.600000
COEFICIENTE DE CONCORDANCIA MEDIO = 0.650000
COEFICIENTE DE CONCORDANCIA FUERTE = 0.750000

COEFICIENTE DE DISCORDANCIA DEBIL = 0.200000
COEFICIENTE DE DISCORDANCIA FUERTE = 0.650000

TABLA DE UTILIDADES : ALTERNATIVA VS. CRITERIO

CRIT.	1	2	3	4
ALT.1	0.81	0.50	-0.27	-0.60
ALT.2	0.80	0.52	-0.30	-0.58

ALT.3	0.90	0.47	-0.22	-0.59
ALT.4	0.75	0.60	-0.40	-0.62
ALT.5	0.78	0.41	-0.25	-0.62

MEDIDA DE DISCORDANCIA PARA EL CRITERIO 1 = 0.000000
MEDIDA DE DISCORDANCIA PARA EL CRITERIO 2 = 0.300000
MEDIDA DE DISCORDANCIA PARA EL CRITERIO 3 = 0.000000
MEDIDA DE DISCORDANCIA PARA EL CRITERIO 4 = 0.200000

PESO ASIGNADO AL CRITERIO 1 = 1.000000
PESO ASIGNADO AL CRITERIO 2 = 4.000000
PESO ASIGNADO AL CRITERIO 3 = 1.000000
PESO ASIGNADO AL CRITERIO 4 = 3.000000

matriz de concordancia

0.00	0.22	0.44	0.56	0.89
0.78	0.00	0.78	0.56	0.89
0.56	0.22	0.00	0.56	1.00
0.44	0.44	0.44	0.00	0.78
0.11	0.11	0.00	0.56	0.00

matriz de discordancia

0.00	0.04	0.19	0.20	0.07
0.10	0.00	0.27	0.15	0.17
0.06	0.11	0.00	0.28	0.00
0.32	0.25	0.45	0.00	0.38
0.22	0.27	0.15	0.46	0.00

GRAFICA-MATRIZ (ACICLICA) DE RELACIONES FUERTES

0	0	0	0	1
1	0	1	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	0

NOTA: si a(i,j)=1, entonces existe un arco de la alternativa i a la j

GRAFICA-MATRIZ (ACICLICA) DE RELACIONES DEBILES

0	0	0	0	1
1	0	1	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	0

S O L U C I O N

ALTERNATIVA	2	RANGO	1
ALTERNATIVA	4	RANGO	2
ALTERNATIVA	1	RANGO	3
ALTERNATIVA	3	RANGO	4
ALTERNATIVA	5	RANGO	5

Lista de opciones :

```
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   3

NOMBRE DEL ARCHIVO DE DATOS A MODIFICAR: banco.dat
EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVELES DE CONCORDANCIA P-,PO,P*
2) NIVELES DE DISCORDANCIA QO,Q*
3) VALORES DE ALTERNATIVAS
4) VALORES DE TETA
5) PESOS DE LOS COpcion deseada (1-5) ?  1

DAR NIVELES DE CONCORDANCIA P-,PO,P* (0<P-<PO<P*<1) ?   0.3,0.4,0.6
DESEA SEGUIR MODIFICANDO (SI,NO) ?  si

EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVELES DE CONCORDANCIA P-,PO,P*
2) NIVELES DE DISCORDANCIA QO,Q*
3) VALORES DE ALTERNATIVAS
4) VALORES DE TETA
5) PESOS DE LOS COpcion deseada (1-5) ?  2

DAR NIVELES DE DISCORDANCIA QO,Q* (0<QO<Q*<1) ?   0.3,0.6
DESEA SEGUIR MODIFICANDO (SI,NO) ?  si

EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVELES DE CONCORDANCIA P-,PO,P*
2) NIVELES DE DISCORDANCIA QO,Q*
3) VALORES DE ALTERNATIVAS
4) VALORES DE TETA
5) PESOS DE LOS COpcion deseada (1-5) ?  3

EN QUE ALTERNATIVA ?  1
DAR VALORES DE LOS CRITERIOS  0.9,0.8,-0.4,-0.9

CAMBIOS EN OTRA ALTERNATIVA (SI,NO) ?  no
DESEA SEGUIR MODIFICANDO (SI,NO) ?  si

EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVELES DE CONCORDANCIA P-,PO,P*
2) NIVELES DE DISCORDANCIA QO,Q*
3) VALORES DE ALTERNATIVAS
4) VALORES DE TETA
5) PESOS DE LOS COpcion deseada (1-5)'?  4

DAR VALORES DE S[K]  0,0,0,0.2
DESEA SEGUIR MODIFICANDO (SI,NO) ?  si

EN QUE PARTE DESEA HACER CAMBIOS ?
1) NIVELES DE CONCORDANCIA P-,PO,P*
2) NIVELES DE DISCORDANCIA QO,Q*
3) VALORES DE ALTERNATIVAS
4) VALORES DE TETA
```

```
5) PESOS DE LOS COpcion deseada (1-5) ?  5

DAR PESOS A LOS CRITERIOS 5,4,3,2
DESEA SEGUIR MODIFICANDO (SI,NO) ?  no


Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   1

NOMBRE DEL ARCHIVO DE DATOS A LISTAR: banco.dat

NUMERO DE ALTERNATIVAS=  5

NUMERO DE CRITERIOS=  4

NIVEL DE CONCORDANCIA P-=  0.300000
NIVEL DE CONCORDANCIA PO=  0.400000
NIVEL DE CONCORDANCIA P*=  0.600000

NIVEL DE DISCORDANCIA QO=  0.300000
NIVEL DE DISCORDANCIA Q*=  0.600000

V A L O R E S  D E  L A S  A L T E R N A T I V A S  VS.  C R I T E R I O S

 CRIT.     1     2      3      4
ALT.1     0.90  0.80  -0.40  -0.90
ALT.2     0.80  0.52  -0.30  -0.58
ALT.3     0.90  0.47  -0.22  -0.59
ALT.4     0.75  0.60  -0.40  -0.62
ALT.5     0.78  0.41  -0.25  -0.62


CRITERIO        VALOR DE S(K)
    1             0.000000
    2             0.000000
    3             0.000000
    4             0.200000

VALORES DE TETA = VARIAN

CRITERIO          PESO
    1           5.000000
    2           4.000000
    3           3.000000
    4           2.000000


Lista de opciones :
```

```
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   4

NOMBRE DEL ARCHIVO DE DATOS A PROCESAR: banco.dat
NUMERO DE ALTERNATIVAS =5
NUMERO DE CRITERIOS =4

COEFICIENTE DE CONCORDANCIA DEBIL = 0.300000
COEFICIENTE DE CONCORDANCIA MEDIO = 0.400000
COEFICIENTE DE CONCORDANCIA FUERTE = 0.600000

COEFICIENTE DE DISCORDANCIA DEBIL = 0.300000
COEFICIENTE DE DISCORDANCIA FUERTE = 0.600000

TABLA DE UTILIDADES : ALTERNATIVA VS. CRITERIO

 CRIT.    1     2      3      4
ALT.1    0.90  0.80  -0.40  -0.90
ALT.2    0.80  0.52  -0.30  -0.58
ALT.3    0.90  0.47  -0.22  -0.59
ALT.4    0.75  0.60  -0.40  -0.62
ALT.5    0.78  0.41  -0.25  -0.62

MEDIDA DE DISCORDANCIA PARA EL CRITERIO 1  = 0.000000
MEDIDA DE DISCORDANCIA PARA EL CRITERIO 2  = 0.000000
MEDIDA DE DISCORDANCIA PARA EL CRITERIO 3  = 0.000000
MEDIDA DE DISCORDANCIA PARA EL CRITERIO 4  = 0.200000


PESO ASIGNADO AL CRITERIO  1 =   5.000000
PESO ASIGNADO AL CRITERIO  2 =   4.000000
PESO ASIGNADO AL CRITERIO  3 =   3.000000
PESO ASIGNADO AL CRITERIO  4 =   2.000000


matriz de concordancia
0.00 0.64 0.64 0.86 0.64
0.36 0.00 0.43 0.71 0.79
0.71 0.57 0.00 0.71 1.00
0.36 0.29 0.29 0.00 0.43
0.36 0.21 0.00 0.71 0.00


matriz de discordancia
0.00 0.36 0.45 0.31 0.38
0.54 0.00 0.27 0.15 0.17
0.70 0.11 0.00 0.28 0.00
0.33 0.25 0.45 0.00 0.38
0.95 0.27 0.15 0.46 0.00
```

```
GRAFICA-MATRIZ (ACICLICA) DE RELACIONES FUERTES
0 1 0 1 1
0 0 0 1 1
0 1 0 1 1
0 0 0 0 0
0 0 0 1 0

NOTA: si a(i,j)=1, entonces existe un arco de la alternativa i a la j

GRAFICA-MATRIZ (ACICLICA) DE RELACIONES DEBILES
0 1 0 1 1
0 0 0 1 1
0 1 0 1 1
0 0 0 0 0
0 0 0 1 0

S O L U C I O N

ALTERNATIVA    1    RANGO    1
ALTERNATIVA    3    RANGO    2
ALTERNATIVA    2    RANGO    3
ALTERNATIVA    5    RANGO    4
ALTERNATIVA    4    RANGO    5

Lista de opciones :
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) Terminar
  Opcion deseada (1-5) ?   5

termina el proceso
```

4.4 ESTRUCTURA DEL PROGRAMA.

El programa consta de las siguientes subrutinas:

main: subrutina principal que indica el menú.

abrir: subrutina para abrir un archivo (todos los archivos tienen que abrirse antes de introducir información).

cerrar: subrutina que cierra un archivo (todos los archivos tienen que cerrarse después de terminar de introducir la información para no perderla).

generar: subrutina que genera un archivo.

modificar: subrutina que modifica un archivo.

listado: subrutina que lista la información del archivo especificado.

proceso: subrutina que procesa un archivo.

Esta última subrutina consta de 4 subrutinas más:

a) ordenf: subrutina para obtener el ordenamiento fuerte y están involucradas 3 subrutinas:

kernel: subrutina que obtiene el kernel de la gráfica indicada.

conju: subrutina que obtiene el conjunto U.

resta: subrutina para obtener la diferencia entre 2 conjuntos.

b) ordend: subrutina para obtener el ordenamiento débil.

c) ordenm: subrutina para obtener el ordenamiento medio.

d) final: subrutina que obtiene la solución final.

La subrutina proceso utiliza dos funciones:

- mx: función que obtiene el máximo entre dos valores.

- mn: función que obtiene el mínimo entre dos valores.

La subrutina que genera un archivo, se inicia preguntando los datos, después, lee la información recibida y la graba en el archivo correspondiente.

La subrutina proceso es la más compleja pues es donde se realizan los cálculos para obtener la solución final. El procedimiento principia con la lectura de los datos que se encuentran en el archivo a procesar y los imprime. A continuación, se suman los pesos de los criterios y el resultado se guarda en una variable.

Para calcular el índice de concordancia y el de discordancia, se pregunta para todo par de alternativas i,j si el valor de la alternativa i es mayor, igual o menor al valor de la alternativa j con respecto al criterio k; se acumulan los pesos de los criterios en las respectivas variables según sea el caso y para el tercer caso se agrega el cálculo del índice de discordancia de la siguiente manera:

- Si se desea una discordancia estática, únicamente se obtiene la diferencia entre los valores de las alternativas y se divide por el máximo entre el valor de la alternativa i y el de teta.

- Si se desea una discordancia dinámica, se obtiene la diferencia entre los valores de las alternativas y se divide por el mínimo entre el valor de la alternativa i y el de teta.

El índice de discordancia es entonces el máximo entre este valor y el calculado anteriormente.

Las funciones mx y mn son utilizadas para las operaciones anteriores.

Posteriormente, se calcula el índice de concordancia y se pregunta si las siguientes restricciones son satisfechas:

- Si $C(i,j) >= p^*$, $D(i,j) <= q^*$

o

si $C(i,j) >= p0$, $D(i,j) <= q0$

y

$$\sum_{k \in I^+} W_k >= \sum_{k \in I^-} W_k$$

entonces, la entrada de la matriz de relaciones fuertes es uno, en otro caso es cero.

- Si $C(i,j) >= p^-$, $D(i,j) <= q^*$ y $\sum_{k \in I^+} w_k >= \sum_{k \in I^-} w_k$

entonces, la entrada (i,j) es uno en la matriz de relaciones débiles.

Se despliega la matriz de concordancia y la de discordancia. A continuación, por medio de una modificación del algoritmo de Floyd explicado anteriormente (programa Electra I), se obtienen dos matrices: la gráfica matríz aciclica de relaciones fuerte (gf) y débil (gd) a partir de las cuales, se crean dos más (gf1 y gd1) para obtener el ordenamiento débil.

Si la coordenada (i,j) de la matriz gf es uno entonces lo es también la coordenada (j,i) de la matriz gf1, así mismo, se construye la matriz gd1 de la matriz gd.

Se despliega la gráfica matriz acicilica de relaciones fuerte y débil. A continuación, se llama a la subrutina ordenf la cual obtiene el orden fuerte y para ello utiliza tres subrutinas:

kernel (obtiene los nodos que no tienen predecesores en una gráfica), conju (obtiene el conjunto u) y resta (resta dos conjuntos).

La subrutina kernel recorre las entradas de la matríz por columnas y en caso de que todas sean cero indica que no existen predecesores para el nodo cuyo número es el número de la columna que se está utilizando.

Para obtener el conjunto u, se toman parejas de alternativas i,j que se encuentran en el kernel y se pregunta si existe un uno en la coordenada (i,j) o (j,i) de la gráfica matriz de relaciones débiles; en caso afirmativo se registran ambas alternativas en un vector.

Para restar dos conjuntos, se utilizan dos vectores cuyos elementos son los elementos de los conjuntos a restar. Se toma el primer elemento de un vector y se busca en el otro recorriendo todos sus elementos y en caso de no encontrarlo se almacena en el vector resultante, se continúa con el segundo elemento y asi sucesivamente hasta cubrir todos sus elementos.

Ya explicadas las subrutinas, el procedimiento a seguir es:

Paso 1. El vector yk se inicializa como

yk[i] = i   1<=i<=n, n es el número de alternativas.

Paso 2. Inicializar contador = 0

Se utiliza la subrutina kernel para obtener el kernel del vector yk y los nodos resultantes se almacenan el el vector d.

Paso 3. A través de la subrutina conju se obtiene otro vector llamado u.

Paso 4. Se utiliza de nuevo la subrutina kernel para obtener un vector llamado b cuyos elementos sno tienen predecesores en la gráfica matriz gd.

Paso 5. Con ayuda de la subrutina resta se obtiene el vector ak cuyos elementos son el resultado de restar los vectores d y u. Además, se le agregan los elementos del vector b.

Paso 6. Se crea el vector v1 (ordenamiento fuerte) como:

v1[i] = contador + 1

Paso 7. Utilizando la subrutina resta con los vectores yk y ak se actualiza el primero.

Hacer contador = contador + 1 e ir al paso 2.

Posteriormente, se llama a la subrutina ordend que obtiene el ordenamiento débil construyendo el vector v2 de la misma forma como se originó el vector v1 pero utilizando la matriz gráfica gd1 y gf1.

Se obtiene el máximo valor de los elementos del nuevo vector y se almacena en una variable llamada maxi, y finalmente:

v2[i] = 1 + maxi - v2[i]

En seguida, se llama a la subrutina ordenm que obtiene el ordenamiento medio como:

v1[i] = (v1[i] + v2[i])/2

y a la subrutina final que calcula el último ordenamiento organizando los elementos del vector v1 en forma decreciente.

La solución se obtiene desplegando la alternativa v2[i] con su respectivo rango i.

Para modificar un archivo se leen los datos que contiene éste, se pregunta por los nuevos datos y se graban en el mismo archivo.

Interpretación de resultados:

Si se escoge la opción 4 en el menú principal se obtiene un listado de la información que se tiene en el archivo a utilizar. Así mismo, se obtiene la matriz de concordancia donde la coordenada (i,j) corresponde al índice de concordancia entre la alternativa i y la alternativa j. Posteriormente se obtiene la matriz de discordancia. A continuación, se obtiene la matriz de relaciones fuertes y la de relaciones débiles. Dichas matrices representan la gráfica obtenida de la relación fuerte Rf y de la relación débil Rd respectivamente.

Ambas matrices constan de los elementos 0 o 1. Si la coordenada (i,j) es 1 implica que existe un archo de la alternativa i a la alternativa j, en caso de que sea cero implica que no existe dicho arco.

Por último se obtiene el rango de las alternativas, así, si la alternativa j tiene rango 1 implica que es la alternativa de mayor conveniencia.

4.5 PROGRAMA ELECTRA III.

La llamada del programa se realiza tecleando electra3. Dicho programa tiene 5 opciones en el meú principal:

1) Listado de un archivo.
2) Generar un archivo.
3) Modificación de un archivo.
4) Proceso de un archivo.
5) Terminar.

El programa pide una opción y el nombre del archivo que se va a utilizar.

Para crear o generar un archivo se pide la opción 2.

NOTA:

En caso de que ya no se requiera un archivo creado y se necesite generar otro, se podrá escoger la opción anterior y dar el nombre del archivo existente pues automáticamente deja de existir este último para generar el nuevo.

En esta etapa del programa se piden los siguientes datos:

- Número de alternativas
- Número de criterios
- Valores de las alternativas de acuerdo a los diferentes criterios
- Pesos de los criterios
- Umbrales de indiferencia
- Umbrales de preferencia estricta
- Umbrales de veto
- Umbrales de discriminación

Para el caso de que se requiera un solo valor numérico al dar la información, basta con teclear dicho valor y oprimir la tecla <enter>. Para el caso contrario (varios valores numéricos) los valores deben de ir separados por comas y al finalizar oprimir la tecla <enter>.

Una vez generado un archivo (o varios), éste puede ser listado, procesado o modificado.

En caso de que se desee la opción de listado, el programa proporcionará la información que se tecleó al crear el archivo.

Para modificar un archivo se pide la opción 3 del menú principal.

Dentro de dicha opción se pueden modificar los siguientes datos:

- Valores de las alternativas
- Pesos de los criterios
- Umbrales de indiferencia
- Umbrales de preferencia estricta
- Umbrales de veto
- Umbrales de discriminación

Para realizar un cambio en algún valor de cierta alternativa, el programa pregunta en que alternativa se desea hacer el cambio y después pide los valores desde todos los puntos de vista o criterios los cuales tienen que ir separados por comas.

Si se requiere modificar el peso de algún criterio, es necesario actualizar los pesos de todos los criterios.

Para la modificación de los umbrales de discriminación se teclean los 2 umbrales separados por comas.

Para los demás umbrales, el programa pregunta en que criterio se quiere hacer el cambio.

Para procesar un archivo se teclea la opción 4. Con dicho procedimiento se obtiene la información del archivo utilizado y los siguientes resultados:

- la matriz de concordancia y la matriz de grados de credibilidad: el elemento (i,j) de la primera matriz, equivale al valor C(i,j) mientras que en la segunda matriz, corresponde al valor L(i,j).

- resultados de la destilación descendente y ascendente: se obtiene la alternativa y su clase correspondiente.

- solución: se presentan las alternativas con su clase correspondiente.

Finalmente, se escoge la opción 5 para salir del programa.

Para mostrar el modo de operación de este programa se tomará como referencia el ejemplo 1 de la página 116 que considera ocho alternativas de un Aprovechamiento Hidraúlico y el objetivo es ordenarlas de acuerdo al grado de satisfacción que producen.

Al teclear Electra3 se presenta el menú prncipal; se iniciará por generar un archivo a través de la opción 2 y después de teclear el nombre de éste para poder identificarlo.

Los datos se proporcionan separados por comas y conforme se van solicitando.

En el caso de los umbrales de veto, se muestran tres tipos de funciones:

1) C1+C2*G(A)

2) SJ(GJ(A))+C1*G/P

3) SJ(GJ(A))+C1*P

en donde

G(A) es el valor del criterio para la alternativa A

S es el umbral de preferencia estricta

P es el peso del criterio.

Para cada criterio es necesario proporcionar que tipo de función se desea y en el caso de elegir la primera función se teclean los valores para C1 Y C2, en otro caso, se teclea únicamente el valor de C1.

En el ejemplo que nos ocupa se escogió el tipo de función número uno y los umbrales de veto son 10 y 0.8 para todos los criterios.

Posteriormente se presenta una ilustración de la forma como se generó este archivo y la manera de listarlo.

En caso de que algún dato sea incorrecto, es posible corregirlo a través de la opción 3 que modifica un archivo.

La opción 4 procesa un archivo desplegando los datos proporcionados, la matriz de concordancia, la matriz de grados de credibilidad y la destilación descendente y ascendente.

En el ejemplo, tomando en cuenta la primera destilación, se llega a que las alternativas que producen mayor satisfacción son la número 2,3,4,7 y 8 (primera clase) mientras que la número 1,5 y 6 producen menor satisfacción; si se toma en cuenta la destilación ascendente, la de menor preferencia es la alternativa 8.

Para el ordenamiento final, se suman las clases que se obtuvieron en las dos destilaciones anteriores; en el ejemplo a tratar corresponde a:

ALTERNATIVA	SUMA CLASES
1	2+2 = 4
2	1+3 = 3
3	1+2 = 3
4	1+2 = 3
5	2+2 = 4
6	2+2 = 4
7	1+2 = 3
8	1+1 = 2

y se ordenan de acuerdo al valor de esta suma. Por lo tanto, la alternativa 8 es la de mayor preferencia por corresponderle el mínimo de la suma de las clases y las de menor preferencia son las alternativas 1,5 y 6.

En la ilustración, después de obtener esta solución, se modifica el archivo al elegir la opción 3 del menú principal. En dicha opción se despliegan los tipos de datos a corregir y al teclear uno de ellos se pregunta por el nuevo dato. Si se desea continuar, se afirma el deseo de seguir modificando.

```
electra3

Lista de opciones:
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) terminar
Opcion deseada (1-5) ?  2


NOMBRE DEL ARCHIVO DE DATOS A GENERAR:  hidra.dat
NUMERO DE ALTERNATIVAS (2-15) ?  8

NUMERO DE CRITERIOS (2-15) ?  10

VALORES DE CRITERIOS PARA LA ALTERNATIVA  1 (SEPARADOS POR COMAS) ?  1,3,1,1,1,1
VALORES DE CRITERIOS PARA LA ALTERNATIVA  2 (SEPARADOS POR COMAS) ?  6,7,5,4,2,8
VALORES DE CRITERIOS PARA LA ALTERNATIVA  3 (SEPARADOS POR COMAS) ?  5,5,7,7,7,5
VALORES DE CRITERIOS PARA LA ALTERNATIVA  4 (SEPARADOS POR COMAS) ?  6,6,8,8,8,4
VALORES DE CRITERIOS PARA LA ALTERNATIVA  5 (SEPARADOS POR COMAS) ?  4,4,4,5,5,6
VALORES DE CRITERIOS PARA LA ALTERNATIVA  6 (SEPARADOS POR COMAS) ?  8,2,3,2,4,7
VALORES DE CRITERIOS PARA LA ALTERNATIVA  7 (SEPARADOS POR COMAS) ?  4,3,3,3,4,7
VALORES DE CRITERIOS PARA LA ALTERNATIVA  8 (SEPARADOS POR COMAS) ?  7,8,6,6,6,2


PESOS DE LOS CRITERIOS (SEPARADOS POR COMAS) ?  10,7,3,5,6,8,9,4,2,1


UMBRALES DE INDIFERENCIA
UMBRALES DEL CRITERIO 1  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 2  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 3  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 4  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 5  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 6  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 7  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 8  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 9  (SEPARADOS POR COMAS)  ?  0,0
UMBRALES DEL CRITERIO 10  (SEPARADOS POR COMAS)  ?  0,0


UMBRALES DE PREFERENCIA ESTRICTA
UMBRALES DEL CRITERIO 1 (SEPARADOS POR COMAS)  ?  2,0
UMBRALES DEL CRITERIO 2 (SEPARADOS POR COMAS)  ?  2,0
UMBRALES DEL CRITERIO 3 (SEPARADOS POR COMAS)  ?  20,
0
UMBRALES DEL CRITERIO 4 (SEPARADOS POR COMAS)  ?  2,0
UMBRALES DEL CRITERIO 5 (SEPARADOS POR COMAS)  ?  2,0
UMBRALES DEL CRITERIO 6 (SEPARADOS POR COMAS)  ?  2,0
UMBRALES DEL CRITERIO 7 (SEPARADOS POR COMAS)  ?  2,0
UMBRALES DEL CRITERIO 8 (SEPARADOS POR COMAS)  ?  2,0
UMBRALES DEL CRITERIO 9 (SEPARADOS POR COMAS)  ?  2,0
UMBRALES DEL CRITERIO 10 (SEPARADOS POR COMAS)  ?  2,0


UMBRALES DE VETO
tipos de funciones
1) C1+C2*G(A)
2) SJ(GJ(A))+C1*G/P
3) SJ(GJ(A))+C1/P(J)
```

```
TIPO DE FUNCION PARA EL CRITERIO 1 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8
TIPO DE FUNCION PARA EL CRITERIO 2 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8
TIPO DE FUNCION PARA EL CRITERIO 3 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8
TIPO DE FUNCION PARA EL CRITERIO 4 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8
TIPO DE FUNCION PARA EL CRITERIO 5 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8
TIPO DE FUNCION PARA EL CRITERIO 6 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10. ,.8
TIPO DE FUNCION PARA EL CRITERIO 7 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8
TIPO DE FUNCION PARA EL CRITERIO 8 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8
TIPO DE FUNCION PARA EL CRITERIO 9 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8
TIPO DE FUNCION PARA EL CRITERIO 10 (1-3)  ?  1
UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  10,.8


UMBRALES DE DISCRIMINACION
PARAMETROS ALFA Y BETA (SEPARADOS POR COMAS)  ?  .4,.1



Lista de opciones:
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) terminar
Opcion deseada (1-5) ?  3


NOMBRE DEL ARCHIVO DE DATOS A MODIFICAR   :  hidra.dat
EN QUE PARTE DESEA HACER CAMBIOS  ?
1) VALORES DE ALTERNATIVAS
2) PESOS DE LOS CRITERIOS
3) UMBRALES DE INDIFERENCIA
4) UMBRALES DE PREFERENCIA ESTRICTA
5) UMBRALES DE VETO
6) UMBRALES DE DISCRIMINACION
Opcion deseada (1-6)  ?  4
EN QUE CRITERIO  ?  3
DAR UMBRALES DE PREFERENCIA ESTRICTA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE PREFERENCIA ESTRICTA (SI,NO)  ?  no
DESEA SEGUIR MODIFICANDO (SI,NO)  ?  no


Lista de opciones:
1) Listado de un archivo
```

```
2) Generar ur archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) terminar
Opcion deseada (1-5) ?  1


NOMBRE DEL ARCHIVO DE DATOS A LISTAR: hir dra.dat
NUMERO DE ALTERNATIVAS=  8
NUMERO DE CRITERIOS=  10
```

V A L O R E S   D E   L A S   A L T E R N A T I V A S   VS.   C R I T E R I O S

CRIT.	1	2	3	4	5	6	7	8	9	10
ALT 1	1.00	3.00	1.00	1.00	1.00	1.00	8.00	6.00	8.00	2.00
ALT 2	6.00	7.00	5.00	4.00	2.00	8.00	2.80	8.00	7.00	2.00
ALT 3	5.00	5.00	7.00	7.00	7.00	5.90	4.00	1.00	4.00	7.00
ALT 4	6.00	6.00	8.00	8.00	8.00	4.30	3.30	1.00	4.00	6.00
ALT 5	4.00	4.00	4.00	5.00	5.00	6.60	4.70	1.00	3.00	5.00
ALT 6	8.00	2.00	3.00	2.00	4.00	7.40	5.60	5.30	2.00	4.00
ALT 7	4.00	3.00	3.00	3.00	4.00	7.70	2.70	8.00	1.00	8.00
ALT 8	7.00	8.00	6.00	6.00	6.00	2.70	1.00	1.00	6.00	1.00

CRITERIO	PESO
1	10.00
2	7.00
3	3.00
4	5.00
5	6.00
6	8.00
7	9.00
8	4.00
9	2.00
10	1.00

CRITERIO	FUN. DE INDIFERENCIA
1	0.00+0.00*G(A)
2	0.00+0.00*G(A)
3	0.00+0.00*G(A)
4	0.00+0.00*G(A)
5	0.00+0.00*G(A)
6	0.00+0.00*G(A)
7	0.00+0.00*G(A)
8	0.00+0.00*G(A)
9	0.00+0.00*G(A)
10	0.00+0.00*G(A)

CRITERIO	FUN. DE PREFERENCIA ESTRICTA
1	2.00+0.00*G(A)
2	2.00+0.00*G(A)
3	2.00+0.00*G(A)
4	2.00+0.00*G(A)
5	2.00+0.00*G(A)
6	2.00+0.00*G(A)
7	2.00+0.00*G(A)
8	2.00+0.00*G(A)
9	2.00+0.00*G(A)
10	2.00+0.00*G(A)

CRITERIO	FUN. DE VETO
1	10.00+0.00*G(A)
2	10.00+0.00*G(A)
3	10.00+0.00*G(A)
4	10.00+0.00*G(A)
5	10.00+0.00*G(A)
6	10.00+0.00*G(A)
7	10.00+0.00*G(A)
8	10.00+0.00*G(A)
9	10.00+0.00*G(A)
10	10.00+0.00*G(A)

UMBRAL DE DISCRIMINACION:
S(LAMBDA) = 0.40+LAMBDA*0.10   0<LAMBDA<1

Lista de opciones:
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) terminar
Opcion deseada (1-5) ? 4

NOMBRE DEL ARCHIVO DE DATOS A PROCESAR: hidra.dat
NUMERO DE ALTERNATIVAS= 8
NUMERO DE CRITERIOS= 10

TABLA DE UTILIDADES : ALTERNATIVA VS. CRITERIO

CRIT.:	1	2	3	4	5	6	7	8	9	10
ALT. 1	1.00	3.00	1.00	1.00	1.00	1.00	8.00	6.00	8.00	2.00
ALT. 2	6.00	7.00	5.00	4.00	2.00	8.00	2.80	8.00	7.00	2.00
ALT. 3	5.00	5.00	7.00	7.00	7.00	5.90	4.00	1.00	4.00	7.00
ALT. 4	6.00	6.00	8.00	8.00	8.00	4.30	3.30	1.00	4.00	6.00
ALT. 5	4.00	4.00	4.00	5.00	5.00	6.60	4.70	1.00	3.00	5.00
ALT. 6	8.00	2.00	3.00	2.00	4.00	7.40	5.60	5.30	2.00	4.00
ALT. 7	4.00	3.00	3.00	3.00	4.00	7.70	2.70	8.00	1.00	8.00
ALT. 8	7.00	8.00	6.00	6.00	6.00	2.70	1.00	1.00	6.00	1.00

PESO ASIGNADO AL CRITERIO 1 = 10.000000
PESO ASIGNADO AL CRITERIO 2 = 7.000000
PESO ASIGNADO AL CRITERIO 3 = 3.000000
PESO ASIGNADO AL CRITERIO 4 = 5.000000
PESO ASIGNADO AL CRITERIO 5 = 6.000000
PESO ASIGNADO AL CRITERIO 6 = 8.000000
PESO ASIGNADO AL CRITERIO 7 = 9.000000
PESO ASIGNADO AL CRITERIO 8 = 4.000000
PESO ASIGNADO AL CRITERIO 9 = 2.000000
PESO ASIGNADO AL CRITERIO 10 = 1.000000

MATRIZ DE CONCORDANCIA

1.00 0.27 0.27 0.27 0.34 0.45 0.33 0.31

0.82	1.00	0.63	0.69	0.67	0.53	0.87	0.62
0.73	0.53	1.00	0.72	0.89	0.51	0.79	0.65
0.73	0.68	0.82	1.00	0.74	0.44	0.76	0.75
0.73	0.45	0.55	0.41	1.00	0.61	0.83	0.50
0.71	0.57	0.56	0.56	0.67	1.00	0.78	0.58
0.80	0.52	0.38	0.35	0.56	0.64	1.00	0.40
0.72	0.61	0.55	0.45	0.67	0.51	0.62	1.00

MATRIZ DE GRADOS DE CREDIBILIDAD

1.00	0.12	0.07	0.03	0.28	0.24	0.15	0.16
0.82	1.00	0.63	0.69	0.67	0.53	0.87	0.62
0.73	0.42	1.00	0.72	0.89	0.51	0.79	0.65
0.73	0.68	0.82	1.00	0.74	0.44	0.76	0.75
0.73	0.31	0.55	0.41	1.00	0.61	0.83	0.50
0.71	0.57	0.56	0.56	0.67	1.00	0.78	0.58
0.80	0.52	0.38	0.33	0.56	0.64	1.00	0.40
0.72	0.58	0.55	0.45	0.67	0.51	0.62	1.00

DESTILACION DESCENDENTE

ALTERNATIVA	CLASE
1	2
2	1
3	1
4	1
5	2
6	2
7	1
8	1

DESTILACION ASCENDENTE

ALTERNATIVA	CLASE
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	1

S O L U C I O N:

ALTERNATIVA	CLASE
1	8
2	2
3	3
4	4
5	7
6	1
7	5
8	6

Lista de opciones:
1) Listado de un archivo
2) Generar un archivo

```
3) Modificacion de un archivo
4) Proceso de un archivo
5) terminar
Opcion deseada (1-5) ?  3


NOMBRE DEL ARCHIVO DE DATOS A MODIFICAR    :  hidra.dat
EN QUE PARTE DESEA HACER CAMBIOS  ?
1) VALORES DE ALTERNATIVAS
2) PESOS DE LOS CRITERIOS
3) UMBRALES DE INDIFERENCIA
4) UMBRALES DE PREFERENCIA ESTRICTA
5) UMBRALES DE VETO
6) UMBRALES DE DISCRIMINACION
Opcion deseada (1-6)  ?  1
EN QUE ALTERNATIVA  ?  1
DAR VALORES DE LOS CRITERIOS (SEPARADOS POR COMAS) :  6,3,5,4,7,9,8,3,8,1

CAMBIOS EN OTRA ALTERNATIVA (SI,NO)  ?  si
EN QUE ALTERNATIVA  ?  6
DAR VALORES DE LOS CRITERIOS (SEPARADOS POR COMAS) :  8,2,5,1,8,9,1,8,3,1

CAMBIOS EN OTRA ALTERNATIVA (SI,NO)  ?  no
DESEA SEGUIR MODIFICANDO (SI,NO)  ?  si

EN QUE PARTE DESEA HACER CAMBIOS  ?
1) VALORES DE ALTERNATIVAS
2) PESOS DE LOS CRITERIOS
3) UMBRALES DE INDIFERENCIA
4) UMBRALES DE PREFERENCIA ESTRICTA
5) UMBRALES DE VETO
6) UMBRALES DE DISCRIMINACION
Opcion deseada (1-6)  ?  2
DAR PESOS A LOS CRITERIOS (SEPARADOS POR COMAS) :  5,6,7,3,6,2,10,4,4,5
DESEA SEGUIR MODIFICANDO (SI,NO)  ?  si

EN QUE PARTE DESEA HACER CAMBIOS  ?
1) VALORES DE ALTERNATIVAS
2) PESOS DE LOS CRITERIOS
3) UMBRALES DE INDIFERENCIA
4) UMBRALES DE PREFERENCIA ESTRICTA
5) UMBRALES DE VETO
6) UMBRALES DE DISCRIMINACION
Opcion deseada (1-6)  ?  3
EN QUE CRITERIO  ?  1
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  2
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  3
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0
```

```
CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  4
DESEA SEGUIR MODIFICANDO (SI,NO)  ?  si

EN QUE PARTE DESEA HACER CAMBIOS  ?
1) VALORES DE ALTERNATIVAS
2) PESOS DE LOS CRITERIOS
3) UMBRALES DE INDIFERENCIA
4) UMBRALES DE PREFERENCIA ESTRICTA
5) UMBRALES DE VETO
6) UMBRALES DE DISCRIMINACION
Opcion deseada (1-6)  ?  3
EN QUE CRITERIO  ?  4
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  5
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  6
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  7
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  8
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  9
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  10
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  si
EN QUE CRITERIO  ?  1
DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  2,0

CAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  no
DESEA SEGUIR MODIFICANDO (SI,NO)  ?  si

EN QUE PARTE DESEA HACER CAMBIOS  ?
1) VALORES DE ALTERNATIVAS
2) PESOS DE LOS CRITERIOS
3) UMBRALES DE INDIFERENCIA
4) UMBRALES DE PREFERENCIA ESTRICTA
5) UMBRALES DE VETO
6) UMBRALES DE DISCRIMINACION
Opcion deseada (1-6)  ?  6
DAR UMBRALES DE DISCRIMINACION (SEPARADOS POR COMAS):  0,0.4
DESEA SEGUIR MODIFICANDO (SI,NO)  ?  no
```

```
Lista de opciones:
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) terminar
Opcion deseada (1-5) ?  4


NOMBRE DEL ARCHIVO DE DATOS A PROCESAR:  hidra.dat
NUMERO DE ALTERNATIVAS= 8
NUMERO DE CRITERIOS= 10

TABLA DE UTILIDADES : ALTERNATIVA VS. CRITERIO

 CRIT.:   1      2      3      4      5      6      7      8      9      10
ALT. 1   6.00   3.00   5.00   4.00   7.00   9.00   8.00   3.00   8.00   1.00
ALT. 2   6.00   7.00   5.00   4.00   2.00   8.00   2.80   8.00   7.00   2.00
ALT. 3   5.00   5.00   7.00   7.00   7.00   5.90   4.00   1.00   4.00   7.00
ALT. 4   6.00   6.00   8.00   8.00   8.00   4.30   3.30   1.00   4.00   6.00
ALT. 5   4.00   4.00   4.00   5.00   5.00   6.60   4.70   1.00   3.00   5.00
ALT. 6   8.00   2.00   5.00   1.00   8.00   9.00   1.00   8.00   3.00   1.00
ALT. 7   4.00   3.00   3.00   3.00   4.00   7.70   2.70   8.00   1.00   8.00
ALT. 8   7.00   8.00   6.00   6.00   6.00   2.70   1.00   1.00   6.00   1.00

PESO ASIGNADO AL CRITERIO  1 =  5.000000
PESO ASIGNADO AL CRITERIO  2 =  6.000000
PESO ASIGNADO AL CRITERIO  3 =  7.000000
PESO ASIGNADO AL CRITERIO  4 =  3.000000
PESO ASIGNADO AL CRITERIO  5 =  6.000000
PESO ASIGNADO AL CRITERIO  6 =  2.000000
PESO ASIGNADO AL CRITERIO  7 =  10.000000
PESO ASIGNADO AL CRITERIO  8 =  4.000000
PESO ASIGNADO AL CRITERIO  9 =  4.000000
PESO ASIGNADO AL CRITERIO  10 =  5.000000


MATRIZ DE CONCORDANCIA

1.00 0.81 0.60 0.60 0.90 0.83 0.83 0.83
0.69 1.00 0.60 0.60 0.79 0.79 0.79 0.83
0.62 0.69 1.00 1.00 1.00 0.79 0.92 0.71
0.62 0.81 1.00 1.00 0.96 0.79 0.79 0.81
0.40 0.63 0.60 0.48 1.00 0.67 0.83 0.58
0.67 0.75 0.40 0.40 0.54 1.00 0.85 0.75
0.38 0.58 0.50 0.40 0.87 0.58 1.00 0.40
0.62 0.88 0.67 0.40 0.67 0.77 0.79 1.00

MATRIZ DE GRADOS DE CREDIBILIDAD

1.00 0.81 0.60 0.60 0.90 0.83 0.83 0.83
0.69 1.00 0.60 0.60 0.79 0.79 0.79 0.83
0.62 0.69 1.00 1.00 1.00 0.79 0.92 0.71
0.62 0.81 1.00 1.00 0.96 0.79 0.79 0.81
0.40 0.63 0.60 0.48 1.00 0.67 0.83 0.58
0.67 0.75 0.28 0.25 0.54 1.00 0.85 0.75
0.22 0.58 0.50 0.40 0.87 0.58 1.00 0.40
0.60 0.88 0.67 0.40 0.67 0.77 0.79 1.00
```

```
  DESTILACION DESCENDENTE
ALTERNATIVA           CLASE
    1                  3
    2                  5
    3                  2
    4                  1
    5                  5
    6                  5
    7                  5
    8                  4

  DESTILACION ASCENDENTE
ALTERNATIVA           CLASE
    1                  4
    2                  1
    3                  3
    4                  2
    5                  5
    6                  5
    7                  5
    8                  5

      S O L U C I O N:
ALTERNATIVA       CLASE
     1              4
     2              3
     3              2
     4              1
     5              8
     6              5
     7              6
     8              7

Lista de opciones:
1) Listado de un archivo
2) Generar un archivo
3) Modificacion de un archivo
4) Proceso de un archivo
5) terminar
Opcion deseada (1-5) ?  5

TERMINA EL PROCESO

B>
```

4.6 ESTRUCTURA DEL PROGRAMA.

El programa consta de las siguientes subrutinas:

main: subrutina principal que indica el menú.

abrir: subrutina para abrir un archivo (todos los archivos tienen que abrirse antes de introducir información).

cerrar: subrutina que cierra un archivo (todos los archivos tienen que cerrarse después de terminar de introducir la información para no perderla).

generar: subrutina que genera un archivo.

modificar: subrutina que modifica un archivo.

listado: subrutina que lista la información del archivo especificado.

proceso: subrutina que procesa un archivo.

Esta última subrutina consta de una subrutina más:

destil: obtiene la destilación descendente y ascendente por lo que está formada por 2 subrutinas:

a) desc: subrutina para obtener la destilación descendente.

b) asc: subrutina para obtener la destilación ascendente.

La subrutina proceso utiliza 2 funciones:

- mx: función que obtiene el máximo entre dos valores.
- mn: función que obtiene el mínimo entre dos valores.

Al igual que en los programas anteriores, la subrutina modificar, pregunta por los datos, los lee y los graba en el archivo correspondiente.

La subrutina proceso, inicia con la lectura de los datos del archivo a procesar y los despliega. A continuación, calcula la función de indiferencia y la de preferencia estricta almacenando dichos valores en las matrices q y s respectivamente:

$$q(i,j) = q1 + q2 * a(i,j)$$

donde

q1 y q2 son los umbrales de indiferencia

a(i,j) es el valor de la alternativa i con respecto al criterio j.

$$s(i,j) = q3 + q4 * a(i,j)$$

donde

q3 y q4 son los umbrales de preferencia estricta.

Utilizando estos datos y la función mn que obtiene el mínimo entre dos números, se calcula el grado de credibilidad y se almacena en una matriz de triple dimensión L(i,j,k) donde i y j son alternativas y k es un criterio.

Se obtiene la función de veto según el tipo de función seleccionada y a los valores proporcionados.

Para adquirir el índice de concordancia, se acumula en una variable los pesos de los criterios y realizando las operaciones adecuadas este índice se almacena en una matriz denotada con(i,j) y se despliega con el nombre de matriz de concordancia.

En seguida, se realiza el cálculo para el índice de discordancia y la matriz no es desplegada por tener tres entradas. Se pregunta si el índice de discordancia es mayor que el de concordancia y según sea el caso, se calcula la matriz de grados de credibilidad. A continuación, se despliega dicha matriz.

La subrutina destil es llamada para llevar a cabo la destilación ascendente o descendente según sea el caso. El procedimiento que sigue es:

Paso 1. Inicialización del vector b que guarda las alternativas del conjunto Bi:

$$b[i] = i \quad \text{para} \quad 1 <= i <= n$$

donde n es el número de alternativas

y hacer contc = 0 que lleva el número de clase.

Paso 2. Inicialización del vector ok que guarda las alternativas del conjunto Ok:

$$ok[i] = b[i] \quad \text{para } 1 <= i <= n$$

Paso 3. Calcular el valor de lambdak como el máximo valor que contiene la matriz de grados de credibilidad y que cumple que 0 <= lambdak < 1.

Paso 4. Calcular el valor de lambdak1 que es el máximo valor de la matriz de grados de credibilidad y que cumple con la siguiente restricción:

$$L(x,y) < lambdak - s(lambdak)$$

para x,y elementos de ok. Para ello, es necesario variables temporales que guardan los elementos del vector ok.

Paso 5. Para todo elemento x del vector ok:

Si $L(x,y) > lambdak1$ y $L(y,x) > L(x,y) + s(L(y,x))$ $y \in ok$

hacer: $p[x] = p[x] + 1$

Si $L(x,y) > lambdak1$ y $L(y,x) > L(x,y) + s(L(y,x))$ $y \in ok$

hacer: $f[x] = f[x] + 1$

De acuerdo al valor de una bandera que es parámetro de la subrutina destil, se llama a la subrutina desc que obtiene la destilación descendente o a asc que obtiene la ascendente.

En ambas subrutinas para todo elemento x de ok se calcula:

$$p[x] = p[x] - f[x]$$

La diferencia entre ambas subrutinas es que en la descendente se calcula el máximo valor de la diferencia anterior mientras que en la ascendente se calcula el mínimo.

Paso 6. Almacenar en el vector ok1 aquellos elementos que pertenecen al vector ok y cuyo valor p sea igual al obtenido en el paso anterior.

Paso 7. Si lambdak1 > 0 y el número de elementos del vector ok1 es mayor a 1 entonces:

lambdak = lambdak1

ok[i] = ok1[i] para 1 <= i <= n

ir al paso 4

en caso contrario, ir al paso 8.

Paso 8. Hacer contc = contc + 1

c[x] = contc para $x \in ok1$

Paso 9. El vector b se actualiza eliminando aquellos elementos que se encuentren en el vector c.

Paso 10. Si existe más de un elemento en el vector b entonces ir al paso 2, en caso contrario, hacer:

c[x] = contc + 1 $x \in b$

Después de llamar a la subrutina destil y obtener la destilación descendente utilizando del vector ca, se despliega la alternativa x con su clase ca[x]. Posteriormente, se llama a la subrutina destil para obtener la destilación ascendente utilizando el vector cd y se cambia el orden de los elementos del vector: el primer elemento pasa al último, el segundo al penúltimo y asi sucesivamente hasta que el último sea el primer elemento.

Se despliega la destilación ascendente con la alternativa x y la clase cd[x]. Finalmente, se suma el valor del primer elemento del vector ca con el de cd, el segundo elemento de ca con el de cd y así sucesivamente hasta recorrer todos los elementos de los vectores. La solución se despliega a través del orden ascendente de los elementos del vector cd.

Para modificar un archivo se leen los datos que contiene éste, se pregunta por los nuevos y se graban en el mismo archivo.

C O N C L U S I O N E S.

Es de gran importancia en la toma de decisiones, el estudio de la información que se tiene de cada alternativa y de los atributos bajo los cuales se tomará la decisión ya que a pesar de que existen varios modelos para elegir la mejor opción cada uno tiene su utilidad de acuerdo a lo dicho anteriormente.

A continuación se describirán algunas ventajas y desventajas al aplicar uno u otro de los modelos presentados en los capítulos anteriores y algunas diferencias que existen entre ellos.

Como se mencionó en el primer capítulo, el modelo de dominación se utiliza cuando se cuenta con varias alternativas, no es necesario tener conocimiento del valor numérico de los atributos y que el tomador de decisiones dé importancia relativa a cada uno de ellos.

Así mismo, el objetivo del interesado debe ser unicamente el reducir su problema pues este modelo no llega a una decisión final ya que puede suceder el caso en que un gran número de alternativas permanezca después de ser aplicado.

Una ventaja del modelo de satisfacción sobre el de dominación es que reduce el problema al tamaño que el interesado requiera, ya que por ser un método iterativo se pueden elevar o bajar los niveles de aspiración, llegando si así lo desea a una única alternativa.

Para aplicar estte modelo es necesario tener un conocimiento adicional sobre el de dominación que es el de los niveles de aspiración.

Una ventaja que existe en ambos modelos es que no se requiere de comparaciones entre atributos, de la información de la importancia relativa de los mismos y tampoco de una evaluación numérica para las alternativas.

Si los atributos tienen diferente importancia y se desea una evaluación de atributo por atributo, el método lexicográfico nos ayudará a resolver el problema. Debe considerarse que este modelo no brinda la posibilidad de obtener una utilidad global de cada alternativa.

En estos tres métodos a las alternativas no se les puede otorgar una calificación global ya que no existen comparaciones entre atributos lo que lleva a que su evaluación sea en base a atributo por atributo.

Estas tres características son una ventaja sobre otros modelos ya que los valores de las alternativas pueden expresarse en diferentes escalas.

Otra ventaja es que son sencillos y es posible aplicar primero el modelo de satisfacción, después el de dominación y por último el lexicográfico para obtener una mejor decisión.

Si no existe diferencia entre atributos, es conveniente aplicar el método maximáx y maximin ya que la decisión final se realiza considerando solamente un atributo por alternativa.

El modelo aditivo que es muy sencillo y fácil de aplicar, se puede utilizar si se tienen atributos con diferente prioridad e independientes entre sí, mientras que si son dependientes, el modelo de utilidad configural puede ser aplicado.

Una diferencia entre los modelos anteriores y los de representación espacial estriba en el resultado obtenido después de aplicarlos. Los primeros reducen el problema o se llega a la alternativa de mayor grado de satisfacción, mientras que los segundos, ordenan al conjunto de alternativas de acuerdo a las preferencias del interesado.

Existe una desventaja en los modelos que requieren de la construcción de una representación espacial cuando no se cuenta con el conocimiento indispensable para crearla, tal es el caso de las Curvas de Indiferencia donde son necesarios los conocimientos en modelos de Economía para poder construirlas.

Los modelos de programación matemática se aplican a otro tipo de problemas en los que se tiene un objetivo sujeto a varias restricciones; estos modelos proporcionan una solución óptima que satisface dichas restricciones.

La diferencia entre este último modelo y los anteriores es que estos últimos se basan en un conjunto de alternativas definido al igual que los criterios bajo los cuales se evalúan las alternativas y establecen un ordenamiento de las alternativas o la alternativa de mayor preferencia. En los modelos de programación matemática se tienen varias soluciones que satisfacen varias restricciones y se obtiene entre este conjunto la solución óptima.

El método Electra I al igual que el modelo de dominación, divide al conjunto total de alternativas en alternativas dominadas y alternativas no-dominadas. La diferencia es que en el primero, además de realizar esta división, se trata de reducir el problema buscando el conjunto más pequeño de alternativas no-dominadas.

Una de las ventajas de este método sobre los mencionados anteriormente, es que la reducción del problema a la que se llega es diferente para cada tomador de decisiones de acuerdo a sus preferencias y además permite un análisis de sensibilidad ya que es el interesado quien fija la prioridad de los criterios y el peso de los mismos así como los límites de los índices que permiten descartar o aceptar una alternativa.

El método Electra II tiene una gran ventaja sobre los modelos de representación espacial que es la obtención de un ordenamiento completo de las alternativas no-dominadas permitiendo la elección de varias alternativas de acuerdo a un orden de preferencias.

El método Electra III tiene la ventaja de lograr una clasificación en orden de preferencia de alternativas donde se involucran los intereses del tomador de decisiones de una manera más significativa que en todos los demás modelos.

El tomador de decisiones debe analizar cuidadosamente los modelos existentes y dependiendo de sus objetivos e intereses aplicar el más conveniente para llegar a una solución satisfactoria.

B I B I O G R A F I A

Decision with Conflicting Objetives

Keeney R. and Raiffa H.

Wiley New York, 1976

Electre

R. Benayoun, B. Roy, B. Sussmann

Societe D'Economie et de Mathematique Appliquees, Juin 1966

Investigación de Operaciones

Herbert Moskowitz, Gordon P. Wright

Prentice-Hall Hispanoamérica S.A.

Investigación de Operaciones en la Ciencia Administrativa

G. D. Eppen, F. J. Gould

Prentice-Hall Hispanoamérica S.A.

Método Electra como ayuda a los procesos de Toma de Decisiones

Alejandro Mendóza

D .E. P. F. I.

Métodos y Modelos de Investigación de Operaciones

Juan Prawda

Volúmen II, Limusa.

Notas para el Curso Seminario de Investigación de Operaciones

Ma. del Cármen Hernández

Facultad de Ciencias

L I S T A D O   E L E C T R A   I

```
#include <stdio.h>
FILE    *fp;           /* apuntador del archivo de datos */
int     archivo,       /* archivo a utilizar */
        numalt,        /* numero de alternativas */
        numcrit;       /* numero de criterios */
float   p,             /* nivel de concordancia */
        q,             /* nivel de discordancia */
        mx(),          /* funcion para obtener el maximo */
                       /* entre dos valores */
        a[30][20],     /* matriz de pesos */
        juego[20];     /* arreglo para los pesos de los criterios */
int     tipo[20],      /* tipo de criterios */
        gc[30][30];    /* matriz para representar la grafica Gc */
char    coma;          /* caracter auxiliar */

main()
{
 int opcion;          /* entero auxiliar para escoger cualquiera de */
                      /* las cinco opciones */
 char nomar[15];      /* nombre del archivo de datos a utilizar */

 do
 {
  printf("Lista de opciones :\n");
  printf("1) Listado de un archivo\n");
  printf("2) Generar un archivo\n");
  printf("3) Modificacion de un archivo\n");
  printf("4) Proceso de un archivo\n");
  printf("5) Terminar\n");
  printf("  Opcion deseada (1-5) ?   ");
  scanf("%d",&opcion);
  printf("\n");
  if (opcion==1)
  {
   printf("NOMBRE DEL ARCHIVO DE DATOS A LISTAR: ");
   scanf("%s",nomar);
   abrir(opcion,nomar);
   listado();
   fclose(fp);
   printf("\n\n");
  }
  if (opcion==2)
  {
   printf("NOMBRE DEL ARCHIVO DE DATOS A GENERAR: ");
   scanf("%s",nomar);
   abrir(opcion,nomar);
   generar();
   fclose(fp);
   printf("\n\n");
  }
  if (opcion==3)
  {
```

```
   printf("NOMBRE DEL ARCHIVO DE DATOS A MODIFICAR: ");
   scanf("%s",nomar);
   abrir(opcion,nomar);
   modificar(nomar);
   fclose(fp);
   printf("\n\n");
  }
  if (opcion==4)
  {
   printf("NOMBRE DEL ARCHIVO DE DATOS A PROCESAR: ");
   scanf("%s",nomar);
   abrir(opcion,nomar);
   proceso();
   fclose(fp);
   printf("\n\n");
  }
  if (opcion==5)
  {
   printf("termina el proceso\n");
   exit();
  }
  if (opcion>=6)
  {
   printf("ERROR\n");
   exit();
  }
 } /* del do */
 while (opcion!=5);
} /* de main */

/* subrutina que abre el archivo a utilizar  */

abrir(num,nom)
int num;                    /* opcion deseada */
char nom[];                /* nombre del archivo de datos */
{
 if (num==1 || num==3 || num==4)
  fp=fopen(nom,"r");
 if (num==2)
  fp=fopen(nom,"w");
}  /* de abrir   */


/* subrutina para generar los datos de un archivo */
generar()
{
  int i,j;

  printf("NUMERO DE ALTERNATIVAS (2-30)  ?   ");
  scanf("%d",&numalt);
  fprintf(fp,"%d\n",numalt);
  printf("\n");
  printf("NUMERO DE CRITERIOS (2-20)  ?   ");
```

```
  scanf("%d",&numcrit);
  fprintf(fp,"%d\n",numcrit);
  printf("\n");
  printf("NIVEL DE CONCORDANCIA P (0<P<1)  ?   ");
  scanf("%f",&p);
  fprintf(fp,"%f\n",p);
  printf("\n");
  printf("NIVEL DE DISCORDANCIA Q (0<P<1)  ?   ");
  scanf("%f",&q);
  fprintf(fp,"%f\n",q);
  printf("\n");
  for (i=1;i<=numalt;i++)
   {
    printf("VALORES DE CRITERIOS PARA LA ALTERNATIVA  %d\n",i);
    for (j=1;j<=numcrit;j++)
     {
      scanf("%f %c",&(a[i][j]),&coma);
      fprintf(fp,"%f ",a[i][j]);
     }
    printf("\n");
    fprintf(fp,"\n");
   } /* del for */
  printf("TIPO DE CRITERIO (VALORES SEPARADOS POR COMAS) ");
  for (i=1;i<=numcrit;i++)
   {
    scanf("%d %c",&(tipo[i]),&coma);
    fprintf(fp,"%d ",tipo[i]);
   }
  printf("\n");
  fprintf(fp,"\n");
  printf("PESOS DE LOS CRITERIOS: ");
  for (i=1;i<=numcrit;i++)
   {
    scanf("%f %c",&(juego[i]),&coma);
    fprintf(fp,"%f ",juego[i]);
   }
  printf("\n\n");
}  /* de generar */

/* subrutina que modifica un archivo */
modificar (nom)
char nom[];
{
 int  res,                   /* respuesta al tipo de dato a */
                             /* modificar */
      alt,                   /* numero de alternativa a modificar */
      i,j;                   /* enteros auxiliares */
 char dime[3];               /* respuesta al deseo de modificar */
                             /* otro dato */

 fscanf(fp,"%d\n",&numalt);
 fscanf(fp,"%d\n",&numcrit);
 fscanf(fp,"%f\n",&p);
 fscanf(fp,"%f\n",&q);
```

```
for (i=1;i<=numalt;i++)
 for (j=1;j<=numcrit;j++)
  fscanf(fp,"%f ",&(a[i][j]));
for (i=1;i<=numcrit;i++)
 fscanf(fp,"%d ",&(tipo[i]));
for (i=1;i<=numcrit;i++)
 fscanf(fp,"%f ",&(juego[i]));
fclose(fp);
do
{
 printf("EN QUE PARTE DESEA HACER CAMBIOS ?\n");
 printf("1) NIVEL DE CONCORDANCIA P\n");
 printf("2) NIVEL DE DISCORDANCIA Q\n");
 printf("3) VALORES DE ALTERNATIVAS\n");
 printf("4) TIPO DE CRITERIO\n");
 printf("5) PESOS DE LOS CRITERIOS\n");
 printf("Opcion deseada (1-5) ?  ");
 scanf("%d",&res);
 printf("\n");
 if (res==1)
 {
  printf("DAR NIVEL DE CONCORDANCIA P (00<P<1) ?    ");
  scanf("%f",&p);
  printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
  scanf("%s",dime);
  printf("\n");
 }
 if (res==2)
 {
  printf("DAR NIVELES DE DISCORDANCIA Q (0<Q<1) ?    ");
  scanf("%f",&q);
  printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
  scanf("%s",dime);
  printf("\n");
 }
 if (res==3)
 {
  do
  {
   printf("EN QUE ALTERNATIVA ?  ");
   scanf("%d",&alt);
   printf("DAR VALORES DE LOS CRITERIOS  ");
   for (i=1;i<=numcrit;i++)
    scanf("%f %c",&(a[alt][i]));
   printf("\n");
   printf("CAMBIOS EN OTRA ALTERNATIVA (SI,NO) ?  ");
   scanf("%s",dime);
  }
  while (dime[0]=='s');
  printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
  scanf("%s",dime);
  printf("\n");
 }
 if (res==4)
 {
  printf("DAR TIPO DE CRITERIO ");
```

```
   for (i=1;i<=numcrit;i++)
    scanf("%d %c",&(tipo[i]));
   printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
   scanf("%s",dime);
   printf("\n");
  }
  if (res==5)
  {
   printf("DAR PESOS A LOS CRITERIOS ");
   for (i=1;i<=numcrit;i++)
    scanf("%f %c",&(juego[i]));
   printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
   scanf("%s",dime);
   printf("\n");
  }
 } /* del do */
 while (dime[0]!='n');
 abrir(2,nom);
 fprintf(fp,"%d\n",numalt);
 fprintf(fp,"%d\n",numcrit);
 fprintf(fp,"%f\n",p);
 fprintf(fp,"%f\n",q);
 for (i=1;i<=numalt;i++)
 {
  for (j=1;j<=numcrit;j++)
   fprintf(fp,"%f ",a[i][j]);
  fprintf(fp,"\n");
 }
 for (i=1;i<=numcrit;i++)
  fprintf(fp,"%d ",tipo[i]);
 fprintf(fp,"\n");
 for (i=1;i<=numcrit;i++)
  fprintf(fp,"%f ",juego[i]);
 printf("\n\n");
} /* de modificar */

/* subrutina que lista un archivo */

 listado()
 {
  int i,j;   /* enteros auxiliares */

  fscanf(fp,"%d\n",&numalt);
  printf("NUMERO DE ALTERNATIVAS=  %d\n",numalt);
  fscanf(fp,"%d\n",&numcrit);
  printf("NUMERO DE CRITERIOS=  %d\n\n",numcrit);
  fscanf(fp,"%f\n",&p);
  printf("NIVEL DE CONCORDANCIA P=  %f\n",p);
  fscanf(fp,"%f\n",&q);
  printf("NIVEL DE DISCORDANCIA Q=  %f\n\n",q);
  printf("V A L O R E S  D E  L A S  A L T E R N A T I V A S  VS.  C R I T E R I
  printf(" CRIT.   ");
  for (i=1;i<=numcrit;i++)
   printf("%2d      ",i);
  printf("\n");
  for (i=1;i<=numalt;i++)
   {
    printf("ALT.%2d    ",i);
    for(j=1;j<=numcrit;j++)
```

```
      {
       fscanf(fp,"%f ",&(a[i][j]));
       printf("%3.2f  ",a[i][j]);
      }
     printf("\n");
    }
   printf("\n\n");
   printf("CRITERIO            TIPO\n");
   for (i=1;i<=numcrit;i++)
   {
    fscanf(fp,"%d ",&(tipo[i]));
    printf("    %d              %d\n",i,tipo[i]);
   }
   printf("\n\n");
   printf("CRITERIO            PESO\n");
   for (i=1;i<=numcrit;i++)
    {
     fscanf(fp,"%f ",&(juego[i]));
     printf("    %d           %f\n",i,juego[i]);
    }
  }  /* de listado */

/* subrutina que procesa el archivo */
proceso()
{
 int    i,j,k,                  /* enteros auxiliares */
        m,                      /* longitud del arreglo cuyos elementos */
                                /* son la solucion */
        v[30];                  /* arreglo cuyos elementos son la */
                                /* solucion */
 float sumap,                   /* suma total de pesos */
        pmas,                   /* w+ */
        pigual,                 /* w= */
        temp,
        maximo,                 /* reales auxiliares */
        dis[30][30],            /* matriz de discordancia */
        con[30][30],            /* matriz de concordancia */
        aux[30][20];            /* matriz para revalorizar los */
                                /* valores de las alternativas */

 fscanf(fp,"%d\n",&numalt);
 printf("NUMERO DE ALTERNATIVAS =%d\n",numalt);
 fscanf(fp,"%d\n",&numcrit);
 printf("NUMERO DE CRITERIOS =%d\n\n",numcrit);
 fscanf(fp,"%f\n",&p);
 printf("COEFICIENTE DE CONCORDANCIA = %f\n",p);
 fscanf(fp,"%f\n",&q);
 printf("COEFICIENTE DE DISCORDANCIA = %f\n",q);
 printf("TABLA DE UTILIDADES : ALTERNATIVA VS. CRITERIO\n\n");
 printf(" CRIT.    ");
 for (i=1;i<=numcrit;i++)
  printf("%2d      ",i);
 printf("\n");
 for (i=1;i<=numalt;i++)
 {
  printf("ALT.%2d   ",i);
  for (j=1;j<=numcrit;j++)
   {
    fscanf(fp,"%f ",&(a[i][j]));
    printf("%3.2f  ",a[i][j]);
```

```
 printf("\n");
}
printf("\n");
for (i=1;i<=numcrit;i++)
{
 fscanf(fp,"%d ",&(tipo[i]));
 printf("TIPO ASIGNADO AL CRITERIO  %d =  %d\n",i,tipo[i]);
}
printf("\n");
maximo=0.0;
for (j=1;j<=numcrit;j++)
{
 for (i=1;i<=numalt;i++)
   maximo=mx(maximo,a[i][j]);
}
for (i=1;i<=numcrit;i++)
{
 fscanf(fp,"%f ",&(juego[i]));
 printf("PESO ASIGNADO AL CRITERIO  %d =   %f\n",i,juego[i]);
}
/* sumamos los pesos de los criterios */
sumap=0;
for (i=1;i<=numcrit;i++)
 sumap+=juego[i];
for (i=1;i<=numalt;i++)
 for (j=1;j<=numalt;j++)
  {
   gc[i][j]=0;
   dis[i][j]=0;
   con[i][j]=0;
  }

/* obtenemos la matriz de concordancia y la de discordancia */
for (i=1;i<=numalt;i++)
 for (j=1;j<=numalt;j++)
  {
   if (i!=j)
   {
    pmas=0;
    pigual=0;
    for (k=1;k<=numcrit;k++)
    {
     if (tipo[k]>0)
     {
      if (a[i][k]>a[j][k])
       pmas+=juego[k];
      if (a[i][k]==a[j][k])
       pigual+=juego[k];
      if (a[i][k]<a[j][k])
      {
       temp=(a[j][k]-a[i][k])/maximo;
       dis[i][j]=mx(dis[i][j],temp);
      } /* del if */
     }  /* del if */
     if (tipo[k]<0)
     {
      if (a[i][k]<a[j][k])
```

```
          pmas+=juego[k];
         if (a[i][k]==a[j][k])
          pigual+=juego[k];
         if (a[i][k]>a[j][k])
         {
          temp=(a[i][k]-a[j][k])/maximo;
          dis[i][j]=mx(dis[i][j],temp);
         } /* del if */
        }  /* del if */
      }  /* del for */
     con[i][j]=(pmas+pigual/2)/sumap;
     if (con[i][j]-p>=-0.000001 && dis[i][j]-q<=0.000001)
      gc[i][j]=1;
     } /* del if */
     if (i==j)
      dis[i][j]=0;
    } /* del for */
   printf("\n\nMATRIZ DE CONCORDANCIA\n\n");
   for (i=1;i<=numalt;i++)
   {
    for (j=1;j<=numalt;j++)
     printf("%3.2f  ",con[i][j]);
    printf("\n");
   }
   printf("\n\n");
   printf("MATRIZ DE DISCORDANCIA\n\n");
   for (i=1;i<=numalt;i++)
   {
    for (j=1;j<=numalt;j++)
     printf("%3.2f  ",dis[i][j]);
    printf("\n");
   }
   printf("\n\n");
   printf("GRAFICA-MATRIZ\n");
   for (i=1;i<=numalt;i++)
   {
    for (j=1;j<=numalt;j++)
     printf("%d ",gc[i][j]);
    printf("\n");
   }
   /* obtenemos una grafica aciclica */
   ciclos();
   /* obtenemos el kernel */
   m=0;
   for (j=1;j<=numalt;j++)
   {
    k=0;
    for (i=1;i<=numalt;i++)
     if (gc[i][j]==1)
      {
       i=numalt;
       k=1;
      }
    if (k==0)
     {
      m+=1;
```

```
   v[m]=j;
  }
 } /* del for */
 /* obtenemos la solucion */
 printf("\n\nS O L U C I O N\n");
 printf("alternativas:  ");
 if (m==0)
  for (i=1;i<=numalt;i++)
   printf("%d, ",i);
 else
 for (i=1;i<=m;i++)
  printf("%d, ",v[i]);
} /* del proceso */

/* subrutina para obtiener el maximo entre dos reales */
float mx(p,q)
float p,q;
{
 if (p>=q)
  return(p);
 else
  return(q);
}
/* subrutina para obtener una grafica aciclica */
ciclos()
{
 int ban,           /* entero para guardar un nodo del ciclo */
     numciclo,      /* numero de ciclo */
     i,j,k,         /* enteros auxiliares */
     ciclo[30],     /* arreglo para guardar el numero de ciclo */
                    /* de cada nodo */
     aux1,aux2,     /* enteros auxiliares */
     pred[30][30],  /* matriz para guardar los predecesores de */
                    /* todo arco */
     mat[30][30];   /* matriz auxiliar */

 for (i=0;i<=numalt;i++)
  ciclo[i]=0;
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numalt;j++)
   pred[i][j]=i;
 numciclo=0;
 do
 {
  for (i=1;i<=numalt;i++)
   for (j=1;j<=numalt;j++)
    if (i==j)
     mat[i][j]=0;
    else
     if (gc[i][j]==1)
      mat[i][j]=-1;
     else
      mat[i][j]=99;
  ban=0;
  for (k=1;k<=numalt;k++)
  {
   for (i=1;i<=numalt;i++)
```

```
 {
  while ((i==k || mat[i][k]==99) && i<=numalt)
   i+=1;
  if (i<=numalt)
  {
   for (j=1;j<=numalt;j++)
   {
    while ((j==k || mat[k][j]==99) && j<=numalt)
     j+=1;
    if (j<=numalt)
    {
     if (mat[i][j]>(mat[i][k]+mat[k][j]))
     {
      mat[i][j]=mat[i][k]+mat[k][j];
      pred[i][j]=pred[k][j];
     }
     if ((i==j) && mat[i][j]<0)
     {
      ban=i;
      k=i=j=numalt;
     }
    } /* del if */
   }  /* del for */
  }   /* del if */
 }    /* del for */
}     /* del for */
if (ban!=0)
{
 numciclo+=1;
 aux1=ban;
 do
 {
  if (ciclo[aux1]==0)
   ciclo[aux1]=numciclo;
  else
  {
   aux2=ciclo[aux1];
   for (i=1;i<=numalt;i++)
   if (ciclo[i]==aux2)
    ciclo[i]=numciclo;
  }
  aux1=pred[ban][aux1];
 }
 while (aux1!=ban);
 for (i=1;i<=numalt;i++)
  if (ciclo[i]==numciclo && gc[ban][i]==1)
   gc[ban][i]=0;
 for (i=1;i<=numalt;i++)
   if (ciclo[i]==numciclo && i!=ban)
    for (j=1;j<=numalt;j++)
     if (gc[i][j]==1)
     {
      gc[i][j]=0;
      if (ciclo[j]!=numciclo)
```

```
        gc[ban][j]=1;
       } /* del if */
   ban=0;
   for (i=1;i<=numalt;i++)
    if (ciclo[i]!=numciclo)
     ban=1;
   } /* del if */
  } /* del do */
  while (ban!=0 );
  printf("\n\nMATRIZ ACICLICA: \n");
  for (i=1;i<=numalt;i++)
  {
   for (j=1;j<=numalt;j++)
    printf("%d ",gc[i][j]);
   printf("\n");
  }
  printf("\n\n");
  printf("NOTA: si la coordenada (i,j)=1 significa que existe un arco del\n");
  printf("      nodo i al nodo j.\n");
 } /* de ciclos */
```

L I S T A D O E L E C T R A II

```
#include <stdio.h>
FILE     *fp;            /* apuntador del archivo de datos */
int      archivo,        /* archivo a utilizar */
         numalt,         /* numero de alternativas */
         numcrit;        /* numero de criterios */
float    pd,             /* nivel de concordancia debil */
         po,             /* nivel de concordancia medio */
         pf,             /* nivel de concordancia fuerte */
         qo,             /* nivel de discordancia debil */
         qf,             /* nivel de discordancia fuerte */
         mx(),mn(),      /* funciones para obtener el maximo y el minimo */
         a[30][20],      /* matriz de pesos */
         s[20],          /* arreglo para los parametros de teta */
         juego[20];      /* arreglo para los valores del juego */
int      gf[30][30],     /* matriz de relaciones fuertes */
         gf1[30][30],    /* copia de la matriz de relaciones fuertes */
         gd[30][30],     /* matriz de relaciones debiles */
         gdi[30][30],    /* copia de la matriz de relaciones debiles */
         yk[30],         /* conjunto yk */
         d[30],          /* conjunto d */
         b[30],          /* conjunto b */
         u[30],          /* conjunto u */
         ak[30],         /* conjunto ak */
         v1[30],         /* arreglo para el ordenamiento fuerte */
         v2[30];         /* arreglo para el ordenamiento debil */
char     coma,           /* caracter auxiliar */
         teta[3];        /* arreglo para la respuesta a una */
                         /* discordancia dinamica */
main()
{
 int opcion;            /* entero auxiliar para escoger cualquiera */
                        /* de las cinco opciones */
 char nomar[15];        /* nombre del archivo de datos a utilizar */

 do
 {
  printf("Lista de opciones :\n");
  printf("1) Listado de un archivo\n");
  printf("2) Generar un archivo\n");
  printf("3) Modificacion de un archivo\n");
  printf("4) Proceso de un archivo\n");
  printf("5) Terminar\n");
  printf("  Opcion deseada (1-5) ?   ");
  scanf("%1d\n",&opcion);
  printf("\n");
  if (opcion==1)
  {
   printf("NOMBRE DEL ARCHIVO DE DATOS A LISTAR: ");
   scanf("%s",nomar);
   abrir(opcion,nomar);
```

```
  listado();
  fclose(fp);
  printf("\n\n");
 }
 if (opcion==2)
 {
  printf("NOMBRE DEL ARCHIVO DE DATOS A GENERAR: ");
  scanf("%s",nomar);
  abrir(opcion,nomar);
  generar();
  fclose(fp);
  printf("\n\n");
 }
 if (opcion==3)
 {
  printf("NOMBRE DEL ARCHIVO DE DATOS A MODIFICAR: ");
  scanf("%s",nomar);
  abrir(opcion,nomar);
  modificar(nomar);
  opcion=3;
  fclose(fp);
  printf("\n\n");
 }
 if (opcion==4)
 {
  printf("NOMBRE DEL ARCHIVO DE DATOS A PROCESAR: ");
  scanf("%s",nomar);
  abrir(opcion,nomar);
  proceso();
  fclose(fp);
  printf("\n\n");
 }
 if (opcion==5)
 {
  printf("termina el proceso\n");
  exit();
 }
 if (opcion>=6)
 {
  printf("ERROR\n");
  exit();
 }
} /* del do */
while (opcion!=5);
} /* de main */

/* subrutina que abre el archivo a utilizar  */

abrir(num,nom)
int num;                /* opcion deseada */
char nom[];             /* nombre del archivo de datos */
{
 if (num==1 || num==3 || num==4)
  fp=fopen(nom,"r");
 if (num==2)
  fp=fopen(nom,"w");
```

```
} /* de abrir */

/* subrutina para generar los datos de un archivo */
generar()
{
  int i,j;

  printf("NUMERO DE ALTERNATIVAS (2-30)  ?   ");
  scanf("%d",&numalt);
  fprintf(fp,"%d\n",numalt);
  printf("\n");
  printf("NUMERO DE CRITERIOS (2-20)  ?   ");
  scanf("%d",&numcrit);
  fprintf(fp,"%d\n",numcrit);
  printf("\n");
  printf("NIVELES DE CONCORDANCIA P-,PO,P*  (0<P-<PO<P*<1) ?   ");
  scanf("%f %c %f %c %f",&pd,&coma,&po,&coma,&pf);
  fprintf(fp,"%f %f %f\n",pd,po,pf);
  printf("\n");
  printf("NIVELES DE DISCORDANCIA QO,Q*  (0<QO<Q*<1) ?   ");
  scanf("%f %c %f",&qo,&coma,&qf);
  fprintf(fp,"%f %f\n\n",qo,qf);
  printf("\n");
  for (i=1;i<=numalt;i++)
   {
    printf("VALORES DE CRITERIOS PARA LA ALTERNATIVA  %d\n",i);
    for (j=1;j<=numcrit;j++)
     {
      scanf("%f %c",&(a[i][j]),&coma);
      fprintf(fp,"%f ",a[i][j]);
     }
    printf("\n");
    fprintf(fp,"\n");
   }
  printf("CRITERIO                 VALOR DE S(K) CORRESPONDIENTE\n");
  for (i=1;i<=numcrit;i++)
   {
    printf("   %d                                   ",i);
    scanf("%f",&(s[i]));
    fprintf(fp,"%f ",s[i]);
   }
  printf("\n\n");
  fprintf(fp,"\n");
  printf("SE QUIERE UNA DISCORDANCIA DINAMICA (SI,NO)  ?  ");
  scanf("%s",teta);
  fprintf(fp,"%s\n",teta);
  printf("\n");
  printf("PESOS DE LOS CRITERIOS: ");
  for (i=1;i<=numcrit;i++)
   {
    scanf("%f %c",&(juego[i]),&coma);
```

```
    fprintf(fp,"%f ",juego[i]);
   }
  printf("\n\n");
 } /* de generar */

/* subrutina que modifica un archivo */
modificar (nom)
char nom[];
{
 int  res,                /* respuesta al tipo de dato a modificar */
      alt,                /* numero de alternativa a modificar */
      i,j;                /* enteros auxiliares */
 char dime[3];            /* respuesta al deseo de modificar otro dato */

 fscanf(fp,"%d\n",&numalt);
 fscanf(fp,"%d\n",&numcrit);
 fscanf(fp,"%f %f %f\n",&pd,&po,&pf);
 fscanf(fp,"%f %f\n",&qo,&qf);
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numcrit;j++)
   fscanf(fp,"%f ",&(a[i][j]));
 for (i=1;i<=numcrit;i++)
  fscanf(fp,"%f ",&(s[i]));
 fscanf(fp,"%s",teta);
 for (i=1;i<=numcrit;i++)
  fscanf(fp,"%f ",&(juego[i]));
 fclose(fp);
 do
 {
  printf("EN QUE PARTE DESEA HACER CAMBIOS ?\n");
  printf("1) NIVELES DE CONCORDANCIA P-,PO,P*\n");
  printf("2) NIVELES DE DISCORDANCIA QO,Q*\n");
  printf("3) VALORES DE ALTERNATIVAS\n");
  printf("4) VALORES DE TETA\n");
  printf("5) PESOS DE LOS CRITERIOS\n");
  printf("Opcion deseada (1-5) ?  ");
  scanf("%d",&res);
  printf("\n");
  if (res==1)
  {
   printf("DAR NIVELES DE CONCORDANCIA P-,PO,P* (0<P-<PO<P*<1) ?   ");
   scanf("%f %c %f %c %f",&pd,&coma,&po,&coma,&pf);
   printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
   scanf("%s",dime);
   printf("\n");
  }
  if (res==2)
  {
   printf("DAR NIVELES DE DISCORDANCIA QO,Q* (0<QO<Q*<1) ?   ");
   scanf("%f %c %f",&qo,&coma,&qf);
   printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
   scanf("%s",dime);
   printf("\n");
  }
  if (res==3)
  {
   do
```

```
    {
     printf("EN QUE ALTERNATIVA ?  ");
     scanf("%d",&alt);
     printf("DAR VALORES DE LOS CRITERIOS  ");
     for (i=1;i<=numcrit;i++)
      scanf("%f %c",&(a[alt][i]));
     printf("\n");
     printf("CAMBIOS EN OTRA ALTERNATIVA (SI,NO) ?  ");
     scanf("%s",dime);
    }
    while (dime[0]=='s');
    printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
    scanf("%s",dime);
    printf("\n");
   }
   if (res==4)
   {
    printf("DAR VALORES DE S[K]  ");
    for (i=1;i<=numcrit;i++)
     scanf("%f %c",&(s[i]));
    printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
    scanf("%s",dime);
    printf("\n");
   }
   if (res==5)
   {
    printf("DAR PESOS A LOS CRITERIOS ");
    for (i=1;i<=numcrit;i++)
     scanf("%f %c",&(juego[i]));
    printf("DESEA SEGUIR MODIFICANDO (SI,NO) ?  ");
    scanf("%s",dime);
    printf("\n");
   }
  } /* del do */
  while (dime[0]!='n');
  abrir(2,nom);
  fprintf(fp,"%d\n",numalt);
  fprintf(fp,"%d\n",numcrit);
  fprintf(fp,"%f %f %f\n",pd,po,pf);
  fprintf(fp,"%f %f\n",qo,qf);
  for (i=1;i<=numalt;i++)
  {
   for (j=1;j<=numcrit;j++)
    fprintf(fp,"%f ",a[i][j]);
   fprintf(fp,"\n");
  }
  for (i=1;i<=numcrit;i++)
   fprintf(fp,"%f ",s[i]);
  fprintf(fp,"\n");
  fprintf(fp,"%2s\n",teta);
  for (i=1;i<=numcrit;i++)
   fprintf(fp,"%f ",juego[i]);
 } /* de modificar */

 /* subrutina que lista un archivo */

  listado()
```

```
{
 int i,j;   /* enteros auxiliares */

 printf("\n");
 fscanf(fp,"%d\n",&numalt);
 printf("NUMERO DE ALTERNATIVAS=  %d\n\n",numalt);
 fscanf(fp,"%d\n",&numcrit);
 printf("NUMERO DE CRITERIOS=  %d\n\n",numcrit);
 fscanf(fp,"%f %f %f\n",&pd,&po,&pf);
 printf("NIVEL DE CONCORDANCIA P-=  %f\n",pd);
 printf("NIVEL DE CONCORDANCIA PO=  %f\n",po);
 printf("NIVEL DE CONCORDANCIA P*=  %f\n\n",pf);
 fscanf(fp,"%f %f\n",&qo,&qf);
 printf("NIVEL DE DISCORDANCIA QO=  %f\n",qo);
 printf("NIVEL DE DISCORDANCIA Q*=  %f\n\n",qf);
 printf("V A L O R E S  D E  L A S  A L T E R N A T I V A S  VS.  C R I T E R I
 printf(" CRIT.    ");
 for (i=1;i<=numcrit;i++)
  printf("%d       ",i);
 printf("\n");
 for (i=1;i<=numalt;i++)
  {
   printf("ALT.%d     ",i);
   for(j=1;j<=numcrit;j++)
    {
     fscanf(fp,"%f ",&(a[i][j]));
     printf("%3.2f  ",a[i][j]);
    }
   printf("\n");
  }
 printf("\n\n");
 printf("CRITERIO       VALOR DE S(K)\n");
 for (i=1;i<=numcrit;i++)
  {
   fscanf(fp,"%f ",&(s[i]));
   printf("    %d              %f\n",i,s[i]);
  }
 printf("\n");
 fscanf(fp,"%s",teta);
 if (teta[0]=='s')
    printf("VALORES DE TETA = VARIAN\n\n");
 else;
 printf("CRITERIO          PESO\n");
 for (i=1;i<=numcrit;i++)
  {
   fscanf(fp,"%f ",&(juego[i]));
   printf("    %d              %f\n",i,juego[i]);
  }
}  /* de listado */

/* subrutina que procesa el archivo */

proceso()
{
 int   i,j,k;                 /* enteros auxiliares */
 float sumap,                 /* suma total de pesos */
       pmas,                  /* w+ */
```

```
     pigual,                   /* w= */
     pmenos,                   /* w- */
     temp,                    /* real auxiliar */
     dis[30][30],             /* matriz de discordancia */
     con[30][30];             /* matriz de concordancia */

fscanf(fp,"%d\n",&numalt);
printf("NUMERO DE ALTERNATIVAS =%d\n",numalt);
fscanf(fp,"%d\n",&numcrit);
printf("NUMERO DE CRITERIOS =%d\n\n",numcrit);
fscanf(fp,"%f %f %f\n",&pd,&po,&pf);
printf("COEFICIENTE DE CONCORDANCIA DEBIL = %f\n",pd);
printf("COEFICIENTE DE CONCORDANCIA MEDIO = %f\n",po);
printf("COEFICIENTE DE CONCORDANCIA FUERTE = %f\n\n",pf);
fscanf(fp,"%f %f\n",&qo,&qf);
printf("COEFICIENTE DE DISCORDANCIA DEBIL = %f\n",qo);
printf("COEFICIENTE DE DISCORDANCIA FUERTE = %f\n\n",qf);
printf("TABLA DE UTILIDADES : ALTERNATIVA VS. CRITERIO\n\n");
printf(" CRIT.   ");
for (i=1;i<=numcrit;i++)
 printf("%d     ",i);
printf("\n");
for (i=1;i<=numalt;i++)
 {
  printf("ALT.%d   ",i);
  for (j=1;j<=numcrit;j++)
   {
    fscanf(fp,"%f ",&(a[i][j]));
    printf("%3.2f  ",a[i][j]);
   }
  printf("\n");
  fscanf(fp,"\n");
 }
printf("\n");
for (i=1;i<=numcrit;i++)
 {
  fscanf(fp,"%f ",&(s[i]));
  printf("MEDIDA DE DISCORDANCIA PARA EL CRITERIO %d  = %f\n",i,s[i]);
 }
printf("\n\n");
fscanf(fp,"\n");
fscanf(fp,"%s\n",teta);
for (i=1;i<=numcrit;i++)
 {
  fscanf(fp,"%f ",&(juego[i]));
  printf("PESO ASIGNADO AL CRITERIO  %d =    %f\n",i,juego[i]);
 }
sumap=0;
for (i=1;i<=numcrit;i++)
 sumap+=juego[i];
for (i=1;i<=numalt;i++)
 for (j=1;j<=numalt;j++)
  {
   gf[i][j]=0;
   gd[i][j]=0;
```

```
    gf1[i][j]=0;
    gd1[i][j]=0;
    dis[i][j]=0;
    con[i][j]=0;
   }
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numalt;j++)
   {
    if (i!=j)
    {
    pmas=0;
    pmenos=0;
    pigual=0;
    dis[i][j]=0;
    for (k=1;k<=numcrit;k++)
     {
      if (a[i][k]>a[j][k])
       pmas+=juego[k];
      if (a[i][k]==a[j][k])
       pigual+=juego[k];
      if (a[i][k]<a[j][k])
       {
        pmenos+=juego[k];
        if (teta[0]=='s' && a[j][k]>0)
          temp=(a[j][k]-a[i][k])/mx(a[i][k],s[k]);
        if (teta[0]=='n' && a[j][k]>0)
          temp=(a[j][k]-a[i][k])/s[k];
        if (teta[0]=='s' && a[j][k]<0)
          temp=(a[i][k]-a[j][k])/mn(a[i][k],s[k]);
        if (teta[0]=='n' && a[j][k]<0)
          temp=(a[j][k]-a[i][k])/s[k];
        dis[i][j]=mx(dis[i][j],temp);
       } /* del if */
     } /* del for */
    con[i][j]=(pmas+pigual)/sumap;
    if (con[i][j]-pf>=-0.000001 && dis[i][j]-qf<=0.000001 && pmas>=pmenos ||
      con[i][j]-po>=-0.000001 && dis[i][j]-qo<=0.000001 && pmas>=pmenos)
     gf[i][j]=1;
    if (con[i][j]-pd>=-0.000001 && dis[i][j]-qf<=0.000001 && pmas>=pmenos)
      gd[i][j]=1;
    } /* del if */
    if (i==j)
     dis[i][j]=0;
   } /* del for */
 printf("\n\n");
 printf("matriz de concordancia\n");
 for (i=1;i<=numalt;i++)
 {
  for (j=1;j<=numalt;j++)
   printf("%2.2f ",con[i][j]);
  printf("\n");
 }
 printf("\n\n");
 printf("matriz de discordancia\n");
 for (i=1;i<=numalt;i++)
 {
  for (j=1;j<=numalt;j++)
```

```
    printf("%2.2f ",dis[i][j]);
   printf("\n");
  }
  printf("\n\n");
  ciclos(gf);
  ciclos(gd);
  for (i=1;i<=numalt;i++)
   for (j=1;j<=numalt;j++)
   {
    if (gf[i][j]==1)
      gf1[j][i]=1;
    if (gd[i][j]==1)
     gd1[j][i]=1;
   }
  printf("GRAFICA-MATRIZ (ACICLICA) DE RELACIONES FUERTES\n");
  for (i=1;i<=numalt;i++)
   {
    for (j=1;j<=numalt;j++)
     printf("%d ",gf[i][j]);
    printf("\n");
   }
  printf("\n\n");
  printf("NOTA: si a(i,j)=1, entonces existe un arco de la alternativa i a la j\
  printf("GRAFICA-MATRIZ (ACICLICA) DE RELACIONES DEBILES\n");
  for (i=1;i<=numalt;i++)
   {
    for (j=1;j<=numalt;j++)
     printf("%1d ",gd[i][j]);
    printf("\n");
   }
  printf("\n\n");
  ordenf(v1,gf,gd);           /* orden fuerte */
  ordend();                   /* orden debil  */
  ordenm();                   /* orden medio  */
  final();                    /* solucion     */
} /* de proceso */

/* declaracion de funciones y subrutinas ha utilizar */
 float mx(p,q)
 float p,q;
 {
  if (p>=q)
   return(p);
  else
   return(q);
 }

 float mn(p,q)
 float p,q;
 {
  if (p<=q)
   return(p);
  else
   return(q);
 }
```

```
/* subrutina para obtener una grafica aciclica */
ciclos(g)
int g[30][30];
{
 int ban,            /* nodo que forma parte del ciclo */
     numciclo,       /* numero de ciclo */
     i,j,k,          /* enteros auxiliares */
     ciclo[30],      /* arreglo para guardar el ciclo al que */
                     /* pertenece cada */
                     /* nodo */
     aux1,aux2,      /* enteros auxiliares */
     pred[30][30],   /* predecesores de los arcos */
     mat[30][30];    /* matriz auxiliar */

 for (i=0;i<=numalt;i++)
  ciclo[i]=0;
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numalt;j++)
   pred[i][j]=i;
 numciclo=0;
do
{
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numalt;j++)
   if (i==j)
    mat[i][j]=0;
   else
    if (g[i][j]==1)
     mat[i][j]=-1;
    else
     mat[i][j]=99;
 ban=0;
 for (k=1;k<=numalt;k++)
 {
  for (i=1;i<=numalt;i++)
  {
   while ((i==k || mat[i][k]==99) && i<=numalt)
    i+=1;
   if (i<=numalt)
   {
    for (j=1;j<=numalt;j++)
    {
     while ((j==k || mat[k][j]==99) && j<=numalt)
      j+=1;
     if (j<=numalt)
     {
      if (mat[i][j]>(mat[i][k]+mat[k][j]))
      {
       mat[i][j]=mat[i][k]+mat[k][j];
       pred[i][j]=pred[k][j];
      }
      if ((i==j) && mat[i][j]<0)
      {
       ban=i;
       k=i=j=numalt;
      }
     } /* del if */
```

```
    }  /* del for */
   }   /* del if */
  }    /* del for */
 }     /* del for */
 if (ban!=0)
 {
 numciclo+=1;
 aux1=ban;
 do
 {
  if (ciclo[aux1]==0)
   ciclo[aux1]=numciclo;
  else
  {
   aux2=ciclo[aux1];
   for (i=1;i<=numalt;i++)
   if (ciclo[i]==aux2)
    ciclo[i]=numciclo;
  }
  aux1=pred[ban][aux1];
 }
 while (aux1!=ban);
 for (i=1;i<=numalt;i++)
  if (ciclo[i]==numciclo && g[ban][i]==1)
   g[ban][i]=0;
 for (i=1;i<=numalt;i++)
   if (ciclo[i]==numciclo && i!=ban)
    for (j=1;j<=numalt;j++)
     if (g[i][j]==1)
     {
      g[i][j]=0;
      if (ciclo[j]!=numciclo)
       g[ban][j]=1;
     } /* del if */
 ban=0;
 for (i=1;i<=numalt;i++)
  if (ciclo[i]!=numciclo)
   ban=1;
 } /* del if */
} /* del do */
while (ban!=0 );
} /* de ciclo */

/* subrutina para obtener el orden fuerte */
ordenf(vv,gff,gdd)
int  vv[],gff[30][30],gdd[30][30];
{
 int contador,
     temp,       /* enteros auxiliares */
     i,j,
     *longyk,
     *longd,
     *longu,
     *longb,
```

```
      *longak,
      *longymas;

 contador=0;
 /* inicializamos el vector YK */
 for (i=1;i<=numalt;i++)
   yk[i]=i;
 *longyk=numalt;
 /* inicializamos el vector vv */
 for (i=1;i<=numalt;i++)
  vv[i]=0;
 do
  {
   *longd=0;
   /* obtenemos el conjunto D */
   kernel(yk,gff,d,longyk,longd);
   /* obtenemos el conjunto  U */
   *longu=0;
   conju(gdd,longu,longd);
   /* obtenemos el conjunto B */
   *longb=0;
   kernel(u,gdd,b,longu,longb);
   /* obtenemos AK */
   *longak=0;
   resta(d,u,ak,longd,longu,longak);
   for (i=1;i<=*longb;i++)
   {
    *longak+=1;
    ak[*longak]=b[i];
   }
   /* obtenemos el vector v */
   contador+=1;
   for (i=1;i<=*longak;i++)
   {
    temp=ak[i];
    vv[temp]=contador;
    for (j=1;j<=numalt;j++)
    {
     gff[temp][j]=0;
     gdd[temp][j]==0;
    }
   } /* del for */
  /* actualizamos YK */
  *longymas=0;
  resta(yk,ak,yk,longyk,longak,longymas);
  *longyk=*longymas;
  } /* del do */
   while (*longyk !=0);
}   /* de ordenf */

/* subrutina para obtener el kernel */
kernel(aa,bb,cc,lon,m)
int  aa[],bb[30][30],cc[],
     *lon,*m;
{
 int   i,j,k,temp;

 for (j=1;j<=*lon;j++)
```

```
 {
  k=0;
  temp=aa[j];
  for (i=1;i<=numalt;i++)
  if (bb[i][temp]==1)
  {
   i=numalt;
   k=1;
  }
  if (k==0)
  {
   *m+=1;
   cc[*m]=temp;
  }
 }  /* del for   */
}   /* de kernel */

/* subrutina para obtener el conjunto U */
  conju(bb,plongu,plongd)
  int  bb[30][30],*plongu,*plongd;
  {
   int   i,j,temp1,temp2;

   if (*plongd!=1)
   for(i=1;i<=*plongd;i++)
   {
    temp1=d[i];
    for (j=2;j<=*plongd;j++)
    {
     temp2=d[j];
     if (bb[temp1][temp2]==1 || bb[temp2][temp1]==1)
     {
      u[*plongu+1]=temp1;
      u[*plongu+2]=temp2;
      *plongu+=2;
     } /* del if   */
    }  /* del for  */
   }   /* del for  */
 }     /* de conju */

 resta(aa,bb,cc,m,n,q)     /* restamos el conjunto aa con bb, el */
                           /* resultado se guarda en c */
 int aa[],bb[],cc[],*m,*n,*q;
 {
  int i,j,s;

  if (*n==0)
  for (i=1;i<=*m;i++)
  {
   *q+=1;
   cc[*q]=aa[i];
  }
  else
  for (i=1;i<=*m;i++)
  {
   s=0;
   for (j=1;j<=*n;j++)
```

```
    if (aa[i]==bb[j])
    {
     j=*n;
     s=1;
    }
   if (j==(*n+1) && s==0)
   {
    *q+=1;
    cc[*q]=aa[i];
   }
  } /* del for */
 } /* de resta */

/* subrutina para obtener el ordenamiento debil */
ordend()
{
 int i,j,maxi;

 ordenf(v2,gf1,gd1);
 maxi=0;
 for (i=1;i<=numalt;i++)
 if (v2[i]>maxi)
  maxi=v2[i];
 for (i=1;i<=numalt;i++)
  v2[i]=1+maxi-v2[i];
}  /* de ordend */

ordenm()      /* orden medio */
{
 int i;

 for (i=1;i<=numalt;i++)
  v1[i]=(v1[i]+v2[i])/2;
}

final()       /* orden final */
{
 int i,j,temp1;

 for (i=1;i<=numalt;i++)
  v2[i]=i;
 for (i=1;i<=numalt;i++)
  for (j=1;j<=(numalt-i);j++)
   if (v1[j]>v1[j+1])
   {
   temp1=v1[j];
   v1[j]=v1[j+1];
   v1[j+1]=temp1;
   temp1=v2[j];
   v2[j]=v2[j+1];
   v2[j+1]=temp1;
   }
 printf("S O L U C I O N\n\n");
 for (i=1;i<=numalt;i++)
  printf("ALTERNATIVA    %d    RANGO    %d\n",v2[i],i);
} /* de final */
```

L I S T A D O   E L E C T R A   III

```
#include <stdio.h>
FILE  *fp;                   /* apuntador al archivo a utilizar */
int   numalt,                /* numero de alternativas */
      numcrit,               /* numero de criterios */
      fun;                   /* entero auxiliar */
float q1,q2,                 /* reales auxiliares */
      a[15][15],             /* matriz de pesos para las alternativas */
      juego[15],             /* arreglo para el peso de cada criterio */
      mn(),                  /* funcion para obtener el minimo entre */
                             /* dos reales */
      mx();                  /* funcion para obtener el maximo entre */
                             /* dos reales */
char  coma;                  /* caracter auxiliar */
main()
{
 int opcion;          /* entero auxiliar para escoger cualquiera de las */
                      /* cinco alternativas */
 char nomar[15];      /* nombre del archivo de datos a utilizar */

 do
 {
  printf("Lista de opciones:\n");
  printf("1) Listado de un archivo\n");
  printf("2) Generar un archivo\n");
  printf("3) Modificacion de un archivo\n");
  printf("4) Proceso de un archivo\n");
  printf("5) terminar\n");
  printf("Opcion deseada (1-5) ?  ");
  scanf("%d",&opcion);
  printf("\n\n");
  if (opcion==1)
  {
   printf("NOMBRE DEL ARCHIVO DE DATOS A LISTAR: ");
   scanf("%s",nomar);
   abrir(opcion,nomar);
   listar();
   fclose(fp);
   printf("\n\n");
  }
  if (opcion==2)
  {
   printf("NOMBRE DEL ARCHIVO DE DATOS A GENERAR:  ");
   scanf("%s",nomar);
   abrir(opcion,nomar);
   generar();
   fclose(fp);
   printf("\n\n");
  }
  if (opcion==3)
  {
   printf("NOMBRE DEL ARCHIVO DE DATOS A MODIFICAR    :  ");
   scanf("%s",nomar);
```

```
   abrir(opcion,nomar);
   modificar(nomar);
   fclose(fp);
   printf("\n\n");
  }
  if (opcion==4)
  {
   printf("NOMBRE DEL ARCHIVO DE DATOS A PROCESAR:  ");
   scanf("%s",nomar);
   abrir(opcion,nomar);
   proceso();
   fclose(fp);
   printf("\n\n");
  }
  if (opcion==5)
  {
   printf("TERMINA EL PROCESO\n");
   exit();
  }
  if (opcion>=6)
  {
   printf("ERROR");
   exit();
  }
 } /* del do */
 while (opcion!=5);
} /* de main */

/* *********** DECLARACION DE SUBRUTINAS *************  */

/* subrutina que abre el archivo a utilizar */
abrir(num,nom)
int  num;          /* opcion deseada */
char nom[];        /* nombre del archivo de datos */
{
 if (num==1 || num==3 || num==4)
  fp=fopen(nom,"r");
 if (num==2)
  fp=fopen(nom,"w");
}

/* subrutina para generar un archivo */
generar()
{
 int i,j;   /* enteros auxiliares */

 printf("NUMERO DE ALTERNATIVAS (2-15) ?  ");
 scanf("%d",&numalt);
 fprintf(fp,"%d\n",numalt);
 printf("\nNUMERO DE CRITERIOS (2-15) ?  ");
 scanf("%d",&numcrit);
 fprintf(fp,"%d\n",numcrit);
```

```
printf("\n");
for (i=1;i<=numalt;i++)
{
 printf("VALORES DE CRITERIOS PARA LA ALTERNATIVA  %d (SEPARADOS POR ",i);
 printf("COMAS) ?  ");
 for (j=1;j<=numcrit;j++)
 {
  scanf("%f %c",&(a[i][j]),&coma);
  fprintf(fp,"%f ",a[i][j]);
 }
 fprintf(fp,"\n");
}
printf("\n\nPESOS DE LOS CRITERIOS (SEPARADOS POR COMAS) ?  ");
for (i=1;i<=numcrit;i++)
{
 scanf("%f %c",&(juego[i]),&coma);
 fprintf(fp,"%f ",juego[i]);
}
fprintf(fp,"\n");
printf("\n\nUMBRALES DE INDIFERENCIA\n");
for (i=1;i<=numcrit;i++)
{
 printf("UMBRALES DEL CRITERIO %d  (SEPARADOS POR COMAS)  ?  ",i);
 scanf("%f %c %f",&q1,&coma,&q2);
 fprintf(fp,"%f %f\n",q1,q2);
}
printf("\n\nUMBRALES DE PREFERENCIA ESTRICTA\n");
for (i=1;i<=numcrit;i++)
{
 printf("UMBRALES DEL CRITERIO %d (SEPARADOS POR COMAS)  ?  ",i);
 scanf("%f %c %f",&q1,&coma,&q2);
 fprintf(fp,"%f %f\n",q1,q2);
}
printf("\n\nUMBRALES DE VETO\n");
printf("tipos de funciones\n");
printf("1) C1+C2*G(A)\n");
printf("2) SJ(GJ(A))+C1*G/P\n");
printf("3) SJ(GJ(A))+C1/P(J)\n");
printf("\n");
for (i=1;i<=numcrit;i++)
{
 printf("TIPO DE FUNCION PARA EL CRITERIO %d (1-3)  ?  ",i);
 scanf("%d",&fun);
 fprintf(fp,"%1d\n",fun);
 if (fun==1)
 {
  printf("UMBRALES DE VETO (SEPARADOS POR COMAS)  ?  ");
  scanf("%f %c %f",&q1,&q2);
  fprintf(fp,"%f %f\n",q1,q2);
 }
 if (fun==2 || fun==3)
 {
  printf("UMBRAL DE VETO  ?  ");
  scanf("%f",q1);
  fprintf(fp,"%f\n",q1);
```

```
  }
 } /* del for */
 printf("\n\nUMBRALES DE DISCRIMINACION\n");
 printf("PARAMETROS ALFA Y BETA (SEPARADOS POR COMAS)  ?  ");
 scanf("%f %c %f",&q1,&coma,&q2);
 fprintf(fp,"%f %f\n",q1,q2);
 printf("\n\n");
} /* de generar */

/* subrutina para listar un archivo */
listar()
{
 int i,j;   /* enteros auxiliares */

 fscanf(fp,"%d\n",&numalt);
 printf("NUMERO DE ALTERNATIVAS=  %d\n",numalt);
 fscanf(fp,"%d\n",&numcrit);
 printf("NUMERO DE CRITERIOS=  %d\n\n",numcrit);
 printf("V A L O R E S  D E  L A S  A L T E R N A T I V A S  VS.");
 printf("  C R I T E R I O S\n");
 printf(" CRIT. ");
 for (i=1;i<=numcrit;i++)
  printf("   %d  ",i);
 printf("\n");
 for (i=1;i<=numalt;i++)
 {
  printf("ALT %2d    ",i);
  for (j=1;j<=numcrit;j++)
  {
   fscanf(fp,"%f ",&(a[i][j]));
   printf("%3.2f  ",a[i][j]);
  }
  printf("\n");
 }
 printf("\n\nCRITERIO              PESO\n");
 for (i=1;i<=numcrit;i++)
 {
  fscanf(fp,"%f ",&(juego[i]));
  printf("    %d              %3.2f\n",i,juego[i]);
 }
 fprintf(fp,"\n");
 printf("\n\nCRITERIO           FUN.  DE  INDIFERENCIA\n");
 for (i=1;i<=numcrit;i++)
 {
  fscanf(fp,"%f %f\n",&q1,&q2);
  printf("   %d              %3.2f+%3.2f*G(A)\n",i,q1,q2);
 }
 printf("\n\nCRITERIO           FUN. DE PREFERENCIA ESTRICTA\n");
 for (i=1;i<=numcrit;i++)
 {
  fscanf(fp,"%f %f\n",&q1,&q2);
  printf("   %d               %3.2f+%3.2f*G(A)\n",i,q1,q2);
 }
 printf("\n\nCRITERIO              FUN. DE VETO\n");
 for (i=1;i<=numcrit;i++)
```

```
 {
  fscanf(fp,"%d\n",&fun);
  if (fun==1)
  {
   fscanf(fp,"%f %f\n",&q1,&q2);
   printf("   %d                   %3.2f+%3.2f*G(A)\n",i,q1,q2);
  }
  if (fun==2)
  {
   fscanf(fp,"%f\n",&q1);
   printf("   %d                   S(G(A))+(%3.2f*G/P)\n",i,q1);
  }
  if (fun==3)
  {
   fscanf(fp,"%f\n",&q1);
   printf("   %d                   S(G(A))+(%3.2f*P)\n",i,q1);
  }
 } /* del for */
 printf("\n\nUMBRAL DE DISCRIMINACION: \n");
 fscanf(fp,"%f %f\n",&q1,&q2);
 printf("S(LAMBDA) = %3.2f+LAMBDA*%3.2f     0<LAMBDA<1\n",q1,q2);
 printf("\n\n");
} /* de listar */

/* subrutina que modifica un archivo */
modificar(nom)
char nom[];
{
 int  i,j,                /* enteros auxiliares */
      alt,                /* numero de alternati[illegible] */
      res,                /* respuesta al tipo de dato a modificar */
      tipo[15];           /* arreglo para guardar el tipo de funcion */
                          /* para umbrales de veto */
 char dime[3];            /* respuesta al desea de modificar otro dato */
 float  funind[15][2],    /* matriz para guardar umbrales */
                          /* de indiferencia */
        funpref[15][2],   /* matriz para guardar umbrales */
                          /* de preferencia */
        veto1[15][2],     /* arreglo para guardar umbrales */
                          /* de veto con respecto a la primera funcion */
        veto2[15],        /* arreglo para guardar umbrales de veto con */
                          /* respecto a la segunda funcion */
        veto3[15],        /* arreglo para guardar umbrales de veto con */
                          /* respecto a la tercera funcion */
        lambda1,lambda2;  /* reales para guardar umbrales */
                          /* de discriminacion */

 fscanf(fp,"%d\n",&numalt);
 fscanf(fp,"%d\n",&numcrit);
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numcrit;j++)
   fscanf(fp,"%f ",&(a[i][j]));
 for (i=1;i<=numcrit;i++)
  fscanf(fp,"%f ",&(juego[i]));
```

```
for (i=1;i<=numcrit;i++)
 fscanf(fp,"%f %f",&(funind[i][0]),&(funind[i][1]));
for (i=1;i<=numcrit;i++)
 fscanf(fp,"%f %f",&(funpref[i][0]),&(funpref[i][1]));
for (i=1;i<=numcrit;i++)
{
 fscanf(fp,"%d ",&(tipo[i]));
 if (tipo[i]==1)
  fscanf(fp,"%f %f",&(veto1[i][0]),&(veto1[i][1]));
 if (tipo[i]==2)
  fscanf(fp,"%f",&(veto2[i]));
 if (tipo[i]==3)
  fscanf(fp,"%f",&(veto3[i]));
}
fscanf(fp,"%f %f",&lambda1,&lambda2);
fclose(fp);
do
{
 printf("EN QUE PARTE DESEA HACER CAMBIOS  ?\n");
 printf("1) VALORES DE ALTERNATIVAS\n");
 printf("2) PESOS DE LOS CRITERIOS\n");
 printf("3) UMBRALES DE INDIFERENCIA\n");
 printf("4) UMBRALES DE PREFERENCIA ESTRICTA\n");
 printf("5) UMBRALES DE VETO\n");
 printf("6) UMBRALES DE DISCRIMINACION\n");
 printf("Opcion deseada (1-6)  ?  ");
 scanf("%d",&res);
 if (res==1)
 {
  do
  {
   printf("EN QUE ALTERNATIVA  ?  ");
   scanf("%d",&alt);
   printf("DAR VALORES DE LOS CRITERIOS (SEPARADOS POR COMAS) :  ");
   for (i=1;i<=numcrit;i++)
    scanf("%f %c",&(a[alt][i]));
   printf("\nCAMBIOS EN OTRA ALTERNATIVA (SI,NO)  ?  ");
   scanf("%s",dime);
  }
  while (dime[0]=='s');
  printf("DESEA SEGUIR MODIFICANDO (SI,NO)  ?  ");
  scanf("%s",dime);
  printf("\n");
 }
 if (res==2)
 {
  printf("DAR PESOS A LOS CRITERIOS (SEPARADOS POR COMAS) :  ");
  for (i=1;i<=numcrit;i++)
```

```
   scanf("%f %c",&(juego[i]));
  printf("DESEA SEGUIR MODIFICANDO (SI,NO)  ?  ");
  scanf("%s",dime);
  printf("\n");
 }
 if (res==3)
 {
  do
  {
   printf("EN QUE CRITERIO  ?  ");
   scanf("%d",&alt);
   printf("DAR UMBRALES DE INDIFERENCIA (SEPARADOS POR COMAS) :  ");
   scanf("%f %c %f",&(funind[alt][0]),&coma,&(funind[alt][1]));
   printf("\nCAMBIOS EN OTRO UMBRAL DE INDIFERENCIA (SI,NO)  ?  ");
   scanf("%s",dime);
  } /* del do */
  while (dime[0]=='s');
  printf("DESEA SEGUIR MODIFICANDO (SI,NO)  ?  ");
  scanf("%s",dime);
  printf("\n");
 } /* del if */
 if (res==4)
 {
  do
  {
   printf("EN QUE CRITERIO  ?  ");
   scanf("%d",&alt);
   printf("DAR UMBRALES DE PREFERENCIA ESTRICTA (SEPARADOS POR COMAS) :  ");
   scanf("%f %c %f",&(funpref[alt][0]),&coma,&(funpref[alt][1]));
   printf("\nCAMBIOS EN OTRO UMBRAL DE PREFERENCIA ESTRICTA (SI,NO)  ?  ");
   scanf("%s",dime);
  }
  while (dime[0]=='s');
  printf("DESEA SEGUIR MODIFICANDO (SI,NO)  ?  ");
  scanf("%s",dime);
  printf("\n");
 } /* del if */
 if (res==5)
 {
  do
  {
   printf("EN QUE CRITERIO  ?  ");
   scanf("%d",&alt);
   printf("\ntipos de funciones\n");
   printf("1) C1+C2*G(A)\n");
   printf("2) SJ(GJ(A))+C1*G/P\n");
   printf("3) SJ(GJ(A))+C1/P(J)\n");
   printf("TIPO DE FUNCION PARA EL CRITERIO %d (1-3)  ?  ",alt);
   scanf("%d",&(tipo[alt]));
   printf("DAR UMBRALES DE VETO:  ");
   if (tipo[alt]==1)
    scanf("%f %c %f",&(veto1[alt][0]),&coma,&(veto1[alt][1]));
   if (tipo[alt]==2)
    scanf("%f",&(veto2[alt]));
   if (tipo[alt]==3)
    scanf("%f",&(veto3[alt]));
   printf("\nCAMBIOS EN OTRO UMBRAL DE VETO (SI,NO)  ?  ");
   scanf("%s",dime);
  } /* del do */
  while (dime[0]=='s');
  printf("DESEA SEGUIR MODIFICANDO (SI,NO)  ?  ");
```

```
   scanf("%s",dime);
   printf("\n");
  } /* del if */
  if (res==6)
  {
   printf("DAR UMBRALES DE DISCRIMINACION (SEPARADOS POR COMAS):  ");
   scanf("%f %c %f",&lambda1,&coma,&lambda2);
   printf("DESEA SEGUIR MODIFICANDO (SI,NO)  ?  ");
   scanf("%s",dime);
   printf("\n");
  } /* del if */
 } /* del do */
 while (dime[0]=='s');
 abrir(2,nom);
 fprintf(fp,"%d\n",numalt);
 fprintf(fp,"%d\n",numcrit);
 for (i=1;i<=numalt;i++)
 {
  for (j=1;j<=numcrit;j++)
   fprintf(fp,"%f ",a[i][j]);
  fprintf(fp,"\n");
 }
 for (i=1;i<=numcrit;i++)
  fprintf(fp,"%f ",juego[i]);
 fprintf(fp,"\n");
 for (i=1;i<=numcrit;i++)
  fprintf(fp,"%f %f\n",funind[i][0],funind[i][1]);
 for (i=1;i<=numcrit;i++)
  fprintf(fp,"%f %f\n",funpref[i][0],funpref[i][1]);
 for (i=1;i<=numcrit;i++)
 {
  fprintf(fp,"%1d\n",tipo[i]);
  if (tipo[i]==1)
   fprintf(fp,"%f %f\n",veto1[i][0],veto1[i][1]);
  if (tipo[i]==2)
   fprintf(fp,"%f\n",veto2[i]);
  if (tipo[i]==3)
   fprintf(fp,"%f\n",veto3[i]);
 }
 fprintf(fp,"%f %f",lambda1,lambda2);
} /* de modificar */
/* subrutina que procesa un archivo */
proceso()
{
 int i,j,k,            /* enteros auxiliares */
     cd[15],ca[15]; /* arreglos para la destilacion descendente */
                    /* y ascendente */
 float *dis1,*dis2, /* umbrales de discriminacion */
       q[15][15],    /* matriz para la funcion de indiferencia y de veto */
       s[15][15],    /* matriz para la funcion de preferencia estricta */
       l[15][15][15], /* matriz para obtener L(x,y) y la matriz de */
                      /* discordancia */
       con[15][15],   /* matriz de concordancia */
       mult;          /* real auxiliar */

 fscanf(fp,"%d\n",&numalt);
 printf("NUMERO DE ALTERNATIVAS= %d\n",numalt);
 fscanf(fp,"%d\n",&numcrit);
 printf("NUMERO DE CRITERIOS= %d\n\n",numcrit);
 printf("TABLA DE UTILIDADES - ALTERNATIVA VS. CRITERIO\n\n");
```

```
printf(" CRIT.:");

 for (i=1;i<=numcrit;i++)
  printf("    %d  ",i);
 printf("\n");
 for (i=1;i<=numalt;i++)
 {
  printf("ALT. %d    ",i);
  for (j=1;j<=numcrit;j++)
  {
   fscanf(fp,"%f ",&(a[i][j]));
   printf("%3.2f   ",a[i][j]);
  }
  printf("\n");
  fscanf(fp,"\n");
 }
 printf("\n");
 for (i=1;i<=numcrit;i++)
 {
  fscanf(fp,"%f ",&(juego[i]));
  printf("PESO ASIGNADO AL CRITERIO  %d =  %f\n",i,juego[i]);
 }
 printf("\n\n");
 /* obtenemos la funcion de indiferencia para toda alternativa */
 for (j=1;j<=numcrit;j++)
 {
  fscanf(fp,"%f %f\n",&q1,&q2);
  for (i=1;i<=numalt;i++)
   q[i][j]=q1+q2*a[i][j];
 }
/* obtenemos la funcion de preferencia estricta para toda alternativa */
 for (j=1;j<=numcrit;j++)
 {
  fscanf(fp,"%f %f\n",&q1,&q2);
  for (i=1;i<=numalt;i++)
   s[i][j]=q1+q2*a[i][j];
 }
 /* obtenemos L(x,y) */
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numalt;j++)
   for (k=1;k<=numcrit;k++)
    l[i][j][k]=(s[i][k]-mn(a[j][k]-a[i][k],s[i][k]))/(s[i][k]-mn(a[j][k]-a[i][k]
 /* obtenemos la funcion de veto para toda alternativa */
 for (j=1;j<=numcrit;j++)
 {
  fscanf(fp,"%d\n",&fun);
  if (fun==1)
  {
   fscanf(fp,"%f %f\n",&q1,&q2);
   for (i=1;i<=numalt;i++)
    q[i][j]=q1+q2*a[i][j];
  }
  if (fun==2)
  {
   fscanf(fp,"%f\n",&q1);
   for (i=1;i<=numalt;i++)
```

```
   q[i][j]=s[i][j]+q1*a[i][j]/juego[j];
  }
  if (fun==3)
  {
   fscanf(fp,"%f\n",&q1);
   for (i=1;i<=numalt;i++)
    q[i][j]=s[i][j]+q1*juego[j];
  }
 } /* del for */
 q1=0;
 /* suma de pesos */
 for (i=1;i<=numcrit;i++)
  q1+=juego[i];
 /* indice de concordancia */
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numalt;j++)
  {
   con[i][j]=0;
   for (k=1;k<=numcrit;k++)
    con[i][j]+=juego[k]/q1*l[i][j][k];
  }
 printf("MATRIZ DE CONCORDANCIA\n\n");
 for (i=1;i<=numalt;i++)
 {
  for (j=1;j<=numalt;j++)
   printf("%3.2f ",con[i][j]);
  printf("\n");
 }
 /* obtenemos el indice de discordancia */
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numalt;j++)
   for (k=1;k<=numcrit;k++)
   {
    mult=(a[j][k]-a[i][k]-s[i][k])/(q[i][k]-s[i][k]);
    l[i][j][k]=mn(1.0,mx(0.0,mult));
   }
 /* obtenemos la matriz de grados de credibilidad */
 for (i=1;i<=numalt;i++)
  for (j=1;j<=numalt;j++)
  {
   mult=1.0;
   for (k=1;k<=numcrit;k++)
    if (l[i][j][k]>con[i][j])
     mult*=(1-l[i][j][k])/(1-con[i][j]);
   q[i][j]=con[i][j]*mult;
  } /* del for */
 printf("\nMATRIZ DE GRADOS DE CREDIBILIDAD\n\n");
 for (i=1;i<=numalt;i++)
 {
  for (j=1;j<=numalt;j++)
   printf("%3.2f ",q[i][j]);
  printf("\n");
 }
 /* leemos los umbrales de discriminacion */
 fscanf(fp,"%f %f",&q1,&q2);
 /* destilacion descendente */
 destil(q,1,ca);
 printf("\n  DESTILACION DESCENDENTE\n");
```

```
 printf("ALTERNATIVA            CLASE\n");
 for (i=1;i<=numalt;i++)
  printf("    %d                  %d\n",i,ca[i]);
 /* destilacion ascendente */
 destil(q,0,cd);
 i=1;
 do
 {
  j=cd[i];
  k=numalt-i+1;
  cd[i]=cd[k];
  cd[k]=j;
  i+=1;
 }
 while (i<=(numalt/2));
 printf("\n\n  DESTILACION ASCENDENTE\n");
 printf("ALTERNATIVA            CLASE\n");
 for (i=1;i<=numalt;i++)
  printf("    %d                  %d\n",i,cd[i]);

 /* ordenamiento final */

 for (i=1;i<=numalt;i++)
 {
  ca[i]=ca[i]+cd[i];
  cd[i]=i;
 }
 for (i=1;i<=numalt;i++)
  for (j=1;j<=(numalt-i);j++)
   if (ca[j]>ca[j+1])
   {
    k=ca[j];
    ca[j]=ca[j+1];
    ca[j+1]=k;
    k=cd[j];
    cd[j]=cd[j+1];
    cd[j+1]=k;
   }
 printf("\n\n      S O L U C I O N:\n");
 printf("ALTERNATIVA        CLASE\n");
 for (i=1;i<=numalt;i++)
  printf("%     %d                 %d\n",i,cd[i]);
} /* de proceso */
/* funcion para obtener el minimo entre dos reales */
float mn(p,r)
float p,r;
{
 if (p<=r)
  return(p);
 else
  return(r);
}

float mx(p,r)
float p,r;
{
 if (p>=r)
```

```
  return(p);
 else
  return(r);
}
destil(l,ban,c)
int   ban,
      c[];
float l[][15];
{
 int  longb,     /* longitud del arreglo b */
      longc,     /* longitud de la matriz c */
     *longok,    /* longitud del arreglo ok */
      longok1,   /* longitud del arreglo ok1 */
      contc,     /* contador */
      i,j,k,
      temp,
      temp1,     /* enteros auxiliares */
     *maxq,      /* lambda-calificacion maxima y minima */
      b[15],     /* arreglo para guardar las alternativas */
                 /* del conjunto Bi */
      ok[15],    /* arreglo para guardar las alternativas */
                 /* del conjunto Ok */
      ok1[15],   /* arreglo para guardar las alternativas */
                 /* del conjunto Ok+1 */
      p[15],     /* arreglo para calcular las lambda-potencia */
      f[15];     /* arreglo para calcular las lambda-debilidad */
 float lambdak,
       lambdak1;

/* inicializacion */
 for (i=1;i<=numalt;i++)
 {
  b[i]=i;
  c[i]=0;
 }
 longb=numalt;
 contc=0;
 do
 {
  for (i=1;i<=longb;i++)
   ok[i]=b[i];                    /* paso 2 */
  *longok=longb;
  lambdak=0.0;                    /* calculando lambdak */
  for (i=1;i<=*longok;i++)
  {
   temp=ok[i];
   for (j=1;j<=*longok;j++)
   {
    temp1=ok[j];
    if (l[temp][temp1]<1.0 && l[temp][temp1]>lambdak)
     lambdak=l[temp][temp1];
   }
  }
  do
  {
   lambdak1=0.0;                  /* calculando lambdak+1 */
   for (i=1;i<=*longok;i++)
   {
    temp=ok[i];
    for (j=1;j<=*longok;j++)
    {
     temp1=ok[j];
```

```
     temp1=ok[j];
     if (l[temp][temp1]<1.0 && l[temp][temp1]>lambdak1 &&
         l[temp][temp1]<(lambdak-(q1+q2*lambdak)))
      lambdak1=l[temp][temp1];
    }
   }
   for (i=1;i<=*longok;i++)        /* paso 5 */
   {
    temp=ok[i];
    p[temp]=f[temp]=0;
    for (j=1;j<=*longok;j++)
    {
     temp1=ok[j];
     if (l[temp][temp1]>lambdak1 &&
         l[temp][temp1]>(l[temp1][temp]+q1+q2*l[temp][temp1]))
      p[temp]+=1;
     if (l[temp1][temp]>lambdak1 &&
         l[temp1][temp]>(l[temp][temp1]+q1+q2*l[temp1][temp]))
      f[temp]+=1;
    }
   }
   if (ban==1)
   {
    *maxq=0;
    desc(p,f,ok,longok,maxq);
   }
   else
   {
    *maxq=numalt;
    asc(p,f,ok,longok,maxq);
   }
   longok1=0;                      /* paso 6 */
   for (i=1;i<=*longok;i++)
   {
    temp=ok[i];
    if (p[temp]==*maxq)
    {
     longok1+=1;
     ok1[longok1]=temp;
    }
   }
   if (longok1>1 && lambdak1>0.0)     /* paso 7 */
   {
    lambdak=lambdak1;
    *longok=longok1;
    for (i=1;i<=longok1;i++)
     ok[i]=ok1[i];
    i=0;
   } /* del if */
  } /* del do */
  while (i==0);
  contc+=1;
  for (i=1;i<=longok1;i++)         /* paso 8 */
  {
   temp=ok1[i];
   c[temp]=contc;
   for (j=1;j<=longb;j++)
    if (b[j]==temp)               /* paso 9 */
    {
     b[j]=0;
```

```
      j=longb;
     }
   }
   i=0;
   for (j=1;j<=longb;j++)
    if (b[j]!=0)
    {
     i+=1;
     b[i]=b[j];
    }
   longb=i;
  } /* del do */
  while (i>1);
  if (i==1)
  {
   contc+=1;
   temp=b[1];
   c[temp]=contc;
  }
} /* de destil */
/* subrutina desc */
desc(p,f,ok,longok,m)
int p[],f[],ok[],
    *longok,
    *m;

{
 int i,temp;       /* enteros auxiliares */

 for (i=1;i<=*longok;i++)
 {
  temp=ook[i];
  p[temp]=p[temp]-f[temp];
  if (p[temp]>*m)
   *m=p[temp];
  }
}
/* subrutina asc */
asc(p,f,ok,longok,m)
int p[],f[],ok[],
    *longok,
    *m;

{
 int i,temp;

 for (i=1;i<=*longok;i++)
 {
  temp=ok[i];
  p[temp]=p[temp]-f[temp];
  if (p[temp]<*m)
   *m=p[temp];
 }
}
```