BIBLIOTECA DE PROGRAMAS

Introducción al problema de la programación lineal.

Los problemas en que deseamos maximizar o minimizar una función numérica, dependiente de un cierto número de variables (y con estas) sujetas a ciertas restriciones, constituyen una clase de problemas que llamaremos, problemas de optimización.

Una parte importante de esta clase de problemas son los llamados problemas de programación, que consisten en determinar la disposición óptima de recursos limitados para alcanzar objetivos conocidos.

Cuando la función que queremos optimizar es lineal e igualmente lo son las relaciones que ligan las variables entre sí, nos encontramos en el caso particular de la programación lineal. Su importancia radica en que este modelo tiene mucha aplicabilidad en problemas de tipo industrial, económico, militar, etc. Ejemplo de situaciones que se adaapten al modelo matemático de la programación lineal son los siguientes.

a) Una fábrica dispone de un equipo de máquinas que puede utilizarse durante un cierto tiempo, para producir una serie de productos, cada uno de los cuales debe de pasar por cada máquina, una determinada cantidad de tiempo. El beneficio que obtenemos es directamente proporcional a las unidades producidas de cada tipo. Se trata de determinar la cantidad a producir de cada producto para maximizar el beneficio.

b) Deseamos transportar una cierta cantidad de mineral de unos ciertos centros de producción a unos ciertos centros de consumo y conocemos cual es el coste unitario por unidad de carga transportada en cada una de las trayectorias. Se trata de conseguir minimizar el costo de transporte, quedando satisfechas las demandas de los consumidores.

c) Una empresa trata de distribuir una serie de personas en unos puntos de acuerdo con unas características que deben de cumplir , y que se miden con una puntuación. Cada puesto exige unas cualidades mínimas y el rendimiento es proporcional al número de puntos obtenidos en cada característica.

Hay muchas razones para utilizar las técnicas de programación lineal para resolver este tipo de problemas: por una parte que las técnicas clásicas se han revelado insuficientes y por otra que la programación lineal usa técnicas facilmente implementables, permitiendo además obtener gran información, sin más que utilizar con sentido común las salidas estandar de ciertos paquetes de programas como el MPSX.

2. Conceptos fundamentales en programación lineal

El problema de programación lineal puede ser formulado de la manera siguiente: hallar los valores de r variables x_j de tal manera que satisfagan sus restricciones lineales de igualdad o desigualdad, siendo no negativas y optimizando una forma lineal, es decir

$$\text{Optimizar } z = c_1 x_1 \ldots + c_r x_r \qquad (1)$$

$$\left.\begin{array}{ll} \sum_{v=1}^{r} a_{ij} x_j \leq b_i & i \epsilon I_1 \\ \sum_{V=1}^{r} a_{ij} x_j \geq b_i & i \epsilon I_2 \\ \sum_{V=1}^{r} a_{ij} x_j = b_i & i \; I_3 \end{array}\right\} \quad (2) \qquad I_1 \cup I_2 \cup I_3 = M$$

$$x_j \geq 0 \qquad j=1, 2, \ldots r \qquad (3)$$

donde alguno o varios o ninguno de los I_1, I_2, I_3 pueden ser el conjunto vacio.

Cada conjunto de x_j $j=1 \ldots r$ que satisface (2) se llama solución, y si además satisface (3) se llama solución factible.

Cada solución factible en que se alcanza el óptimo de z se llama solución óptima factible.

Para el mejor tratamiento del problema buscamos un problema equivalente en el que transformamos todas las restricciones de tipo (2) en igualdades mediante la introducción de las llamadas variables de holgura $x_{r+1}, \ldots, x_{r+1}$

$$x_{r+i} = bi \quad \sum_{j=1}^{r} a_{ij} x_j \qquad i \epsilon I_1$$

$$x_{r+i} = -bi + \sum_{j=1}^{r} a_{ij} x_j \qquad i \epsilon I_2$$

que son no negativas y que se incorporan a (1) con coeficientes cero.

La forma lineal (1) recibe el nombre de función objetivo.

Con esto es problema nos queda: opt $\quad z = \vec{c}\,\vec{x} \quad (1')$

$$A\vec{x} = \vec{b} \qquad (2')$$

$$\vec{x} \geq o \qquad (3')$$

siendo A una matriz de mxn dimensiones, m n.

La matriz A sirve para definir un conjunto conexo en el espacio vectorial R^n a que pertenece X llamado espacio de las soluciones. Al mismo tiempo las columnas de A definen un cono perteneciente a R^m que se llama espacio de los requerimientos.

En el espacio de las soluciones, consideramos cada submatriz no singular -que denominarémos B- de orde mxn, si asociamos a las n-m variables asociadas con las columnas de A restantes valor cero, y resolvemos el sistema obtenemos una solución básica, es decir:

B es la matriz no singular básica

$$A = [B, R]$$

R es la matriz con el resto de columnas

Como las variables asociadas con las columnas de R son cero Ax=b $BX_B=b \quad X_B = B^{-1}b$.

Si alguna de las componentes de X_B es cero obtenemos una solución básica degenerada.

El proceso que resuelve el problema de hallar una solución óptima es el método del Simplex y consta de varias fases:

1) Hallar una solución básica factible inicial
2) Mejora de dicha solución
3) Criterios de optimalidad

1) En el primer paso se trata de situarnos en un vértice del polígono covexo en el que estan las soluciones factibles. Así hemos reducido el tener que examinar los infinitos puntos del recinto a sólo un número finito que son los vértices.

El problema que se plantea es el de situarnos en un punto inicial que sea extremo, este problema se puede resolver con criterios diferentes: directamente o bien con el método de las dos fases.

La característica de estos puntos iniciales es que el número máximo de elementos distintos de cero que poseen es igual al del rango de la ecuación de coeficientes.

2) Un segundo paso consiste en la mejora de la solución inicial, para ello obtendrémos, con algún criterio, una nueva base y darémos el valor de función objetivo en ambas. Naturalmente tratamos de que la segunda sea mejor que la primera. Imponiendo primeramente la condición de no negatividad de los componentes de una nueva base obtenemos un criterio para saber que vector debe ser sustituido, y observando el incremento de la función objetivo vemos que vector es el que sustituye.

Problemas particulares que afectan al tipo de base -si es degenerada o no etc.- surgen si el criterio de salida de un vector de la base, no establece uno de manera unívoca. Así mismo se puede obtener una mayor rapidez de cálculo cuando al tratar de fijar el vector que entra en la base, consideramos aspectos parciales, matemáticamente hablando, pero que son inmediatos al necesitar un mínimo de cálculo.

Una de las peculiaridades del método de resolución de la programación lineal es que dos bases consecutivas solo difieren en un único vector, cuyo signifcado geométrico es el pasar de un vector de un polígono convexo al adyacente, buscando el vértice, si la solución es única, en que el hiperplano formado por la función objetivo en dicho vértice es un hiperplano soporte. Si la solución no es única y existe n vertices solución; las infinitas soluciones estan en la variedad lineal de dimensión n-1 formada por las combinaciones convexas de esos n-vértices.

3) Un último punto importante es saber, cuando hemos terminado el problema. Para ello establecemos un criterio sencillo de optimalidad, que sirva para conocer cuando hemos llegado a la solución o soluciones de tipo óptimo. Si la solución óptima no es única aparece el problema de la alternativa, que dice que cualquier combinación covexa de las soluciones óptimas básica del problema es también solución óptima. Esto geometricamente signifca que el hiperplano que define la función objetivo es paralelo al hiperplano que es el borde de una de las restricciones.

4) La biblioteca de programas dispone del paquete 'MPSX (*) para resolver los problemas de programación lineal y entera. Este paquete, en su procedimiento computacional utiliza una variante del simplex, llamado método revisado del simplex, ya que goza de la ventaja de necesitar un mínimo almacenamiento.

Para la utilización del MPSX es necesario conocer los siguientes apartados:

(1) Sentencias de control de sistema operativo propio del lenguaje.

(2) Comandos propios del lenguaje, cuyo conocimiento consta de 2 fases:
  2.1. Significado de cada comando
  2.2. Combinación de comandos para la realización de un programa

(3) Introducción de datos para resolver un problema.

Un ejemplo muy sencillo de la estructura de un programa procesable en el C.C.U.C.M. sería:

```
(1)  //SIMPLEX      JOB
(1)  //             EXEC        MPSX
(1)  //JOBLIB       DD     DSN=MPSX.SYSTM360,
(1)  //    DISP=SHAR
(1)  //MPSCOMP.SYSIN       DD        *
(2)            PROGRAM
.              INITIALZ
.              MOVE (XDATA, 'PRUEVA')
.              MOVE (XPBNAME, 'PBFILE')
.               CONVERT
.               SETUP
.               MOVE (XOBJ, 'ECON')
.               MOVE (XRHS, 'DEM')
.               PRIMAL
.               SOLUTION
(2)            PEND
(1)  /*
(1)  //MPSEXEC.SYSIN       DD      *
(3)  Datos númericos
(1)  /*
(1)  //
```

(*) Mathematical Programming System Extended.

En donde (1) se refieren a las fichas de sentencias de control necesarias para acceder al paquete y lograr la realización del problema.

Las notas de tipo (2) se refieren a las sentecias que controlan la ejecución del procedimiento, sirven para definir y dar valores a los datos, y para obtener la mayor información en la salida, así:

PROGRAM que es una sentencia de inicialización,

MOVE que sirve para transferir datos de una posición de memoria a otra y

PEND que finaliza físicamente el programa

son sentecias de procedimiento.

En sucesivas comunicaciones se hará referencia más extensa a cada uno de estos tipos de sentencias.