Modelización de problemas de Programación Lineal

INVESTIGACIÓN OPERATIVA I.

2° Grado en Estadística

1. Introducción

Dentro de la Investigación Operativa aparecen una serie de problemas *tipo* de Programación Lineal y Entera, que suelen usarse de referencia para otros problemas más complejos. A continuación se muestra la formulación de los ejemplos más característicos, en los cuales no aparecen las restricciones de no negatividad de las variables que vienen implícitas en la mayoría de los casos.

2. Problema de la dieta

Este tipo de problemas es uno de los más básicos de la Programación Lineal (PL). El objetivo es determinar la combinación óptima de alimentos que satisfaga unos requerimientos nutricionales demandados, minimizando o maximizando algún objetivo, como el coste total de la dieta, la cantidad total de algún nutriente... El problema de la dieta ha ido evolucionando para incluir consideraciones más complejas, como restricciones dietéticas específicas, preferencias personales y limitaciones adicionales.

Se supone que se desea encontrar la combinación óptima de alimentos para satisfacer los requerimientos nutricionales diarios, minimizando el coste total de la dieta. Para lo cual se consideran n alimentos: A_1, A_2, ..., A_n, y una serie de requerimientos nutricionales mínimos que se expresan en función de diferentes nutrientes, R_1, R_2, ..., R_m. De forma que, cada alimento A_j, $(j = 1, ..., n)$ contiene a_{ij} unidades del nutriente o requerimiento i, $(i = 1, ..., m)$, con un coste de c_j por unidad del alimento A_j utilizado. Teniendo en cuenta que los requerimientos mínimos de cada uno de los nutrientes es b_i, con $i = 1, ..., m$.

El objetivo es minimizar el coste total de la dieta, sujeto a las restricciones de los requerimientos nutricionales. Por lo tanto, se definen las variables:

- x_j= unidades del alimento A_j que se incorporan en la dieta.

La función objetivo, en este caso, será minimizar los costes totales:

- Minimizar $z = c_1x_1 + c_2x_2 + ...c_nx_n$

Sujeto a las siguientes restricciones:

$$\begin{gathered}
\text{Requerimiento 1: } a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \geq b_1 \\
\text{Requerimiento 2: } a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \geq b_2 \\
...... \\
\text{Requerimiento m: } a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \geq b_m
\end{gathered}$$

3. Problema del transporte y derivados

En este tipo de problemas se pretende optimizar el transporte de productos entre diferentes puntos. Suelen representarse mediante grafos y redes. Un grafo es un conjunto de objetos, que está formado por vértices o nodos, conectados entre sí por arcos o aristas, donde los nodos suelen representan ubicaciones o actividades, y las aristas representan las conexiones posibles entre los distintos vértices. La mayoría de estos problemas tienen algoritmos específicos que optimizan el tiempo de resolución de los mismos cuando hay muchas variables. Toda la teoría correspondiente a Grafos y Redes se abordará en la asignatura de Investigación Operativa II. En esta sección se mostrará cuál es la modelización de este tipo de problemas desde el punto de vista de la Programación Lineal y Entera.

3.1. Problema del Transporte

El problema de transporte trata de determinar cómo distribuir eficientemente un recurso desde diferentes orígenes a múltiples destinos, minimizando los costes derivados de ese transporte.

Se parte de un conjunto de orígenes O_i, con $i = 1, ..., n$ desde los cuales se pueden mandar a lo más o_i unidades de un determinado producto y un conjunto de destinos D_j, con $j = 1, ...m$ que demandan unas cantidades mínimas d_j, con un valor o coste de envío c_{ij} entre el origen i y el destino j. Para plantear el correspondiente problema de programación lineal, se definen las variables siguientes:

- x_{ij} = unidades del producto que se transportan del origen i al destino j.

El planteamiento para minimizar los costes (también se podría maximizar el valor o la ganancia) teniendo en cuenta la oferta de los orígenes y las demandas de los destinos es:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \\ \text{Sujeto a:} \quad & \\ \text{Oferta de cada origen } i \quad & \sum_{j=1}^{m} x_{ij} \leq o_i \quad \forall i = 1, ...n \\ \text{Demanda de cada destino } j \quad & \sum_{i=1}^{n} x_{ij} \geq d_j \quad \forall j = 1, ..., m \end{aligned}$$

Si la oferta total que hay entre todos los orígenes es igual a la demanda total que piden los destinos, las restricciones anteriores se cumplen con igualdad y se dice que el problema del transporte está *balanceado*. Si la demanda es superior a la oferta, el problema tiene solución no factible porque no se cumplen las restricciones del problema. Para encontrar una solución que permita mandar lo que hay en los almacenes, aunque no se satisfaga la demanda, se puede construir un origen ficticio que oferte todo lo que falta en la demanda total, conectado con todos los destinos, con un coste muy alto, cuyo valor representa la penalización por no enviar los productos a los destinos.

3.2. Problema de Transbordo

Este tipo de problemas es un caso particular del problema del transporte, donde el envío del producto desde los orígenes a los destinos pasa por puntos de transbordo intermedio. En estos puntos intermedios se puede descargar, cargar o mantener las unidades del producto que llegan a cada uno de ellos.

El planteamiento es el mismo que en el problema del transporte, pero hay que añadir las restricciones correspondientes de los puntos intermedios. Sea k un punto intermedio del transporte, al que llegan x_{ik} unidades desde el origen i, y del que salen x_{kj} unidades del producto al destino j, la restricción de cada nodo k intermedio es:

$$\sum_i x_{ik} = \sum_j x_{kj}$$

que indica que todas las unidades que llegan al nodo k desde los diferentes nodos, salen del mismo, hacia otros nodos.

3.3. Problema de Flujo Fijo a Coste Mínimo

El problema de transporte y transbordo enlaza con otro tipo de problemas que reciben el nombre de *problemas de flujo fijo a coste mínimo*, en los cuales se trabaja con redes de flujo. Para ello se dibuja una red donde cada origen, destino y punto intermedio se representa por un nodo, los arcos o flechas indican las posibles uniones entre los nodos, el valor del arco v_{ij} es el coste de transporte de cada unidad de flujo entre los nodos que conecta y la capacidad del arco C_{ij} es lo máximo que se puede transportar de un nodo a otro. Toda red de flujo necesita un nodo origen O del que parte el flujo F conocido (suma total de las demandas/ofertas) que pasa por la red y un nodo final o destino D donde llega ese flujo, en caso de que no existan se crean de forma ficticia asignado un valor nulo a los arcos de unión. Además, hay que tener en cuenta la ley de conservación del flujo, es decir, todo lo que llega a un nodo debe salir de él. Su planteamiento es similar al del problema del transporte, pero teniendo en cuenta que conocemos la cantidad exacta de producto, en este caso flujo, que va a pasar por la red, F. El objetivo es determinar por donde mover el flujo para obtener el coste mínimo.

El planteamiento sería:

$$\begin{array}{rl}
Min & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}x_{ij} \\
\text{Sujeto a} & \\
\text{Nodo origen} & \displaystyle\sum_j x_{Oj} = F \\
\text{Nodo final} & \displaystyle\sum_i x_{iD} = F \\
\text{Nodos intermedios} & \displaystyle\sum_i x_{ik} = \sum_j x_{kj} \quad \forall \text{ nodo intermedio } k \\
\text{Capacidad del arco} & x_{ij} \leq C_{ij} \quad \forall \quad \text{arco} = (i,j)
\end{array}$$

3.4. Problema de Flujo Compatible a Coste Mínimo

Una variante del problema anterior es el *problema de flujo compatible con coste mínimo*, en estos casos no tiene por qué verificarse la ley de conservación del flujo en todos los nodos, pudiéndose quedar flujo en algún nodo (llamados pozos), o generando más flujo del que le llega en otros (fuentes). Por lo que hay que modificar la restrición de estos nodos, del mismo modo que se hacía en el problema del transbordo de la siguiente forma:

$$\sum_j x_{kj} - \sum_i x_{ik} = a_k$$

Donde:

- si $a_k = 0$ se cumple la ley de conservación del flujo en el nodo k, coincidiendo con el problema del transbordo.
- si $a_k > 0$, el nodo k genera a_k unidades de flujo que no le han llegado desde otros nodos de la red.
- si $a_k < 0$, el nodo k se queda con a_k unidades de flujo que no manda por la red.

3.5. Problema del Flujo Máximo

La mayoría de los problemas de flujos se centran en la optimización del flujo de recursos que pasa a través de una red, donde se busca maximizar la cantidad de flujo que pasa por las diferentes aristas, respetando las restricciones de capacidad y conservación de flujo en los nodos. Estos problemas tienen aplicaciones en el diseño de redes de transporte, comunicaciones, suministro de energía,...

El problema del flujo máximo consiste en determinar la máxima cantidad de flujo que pasa por la red desde un nodo origen O de donde sale el flujo a un nodo final o destino D donde llega todo el flujo F desconocido, teniendo en cuenta las limitaciones de capacidad de cada arco, C_{ij} y siendo x_{ij}, la cantidad de flujo que pasa del nodo i al nodo j.

$$\begin{array}{rl}
Max & F \\
\text{Sujeto a:} & \\
\text{Nodo origen} & \displaystyle\sum_j x_{Oj} = F \\
\text{Nodo final} & \displaystyle\sum_i x_{iD} = F \\
\text{Nodos intermedios} & \displaystyle\sum_i x_{ik} = \sum_j x_{kj} \quad \forall \text{ nodo intermedio } k \\
\text{Capacidad del arco} & x_{ij} \leq C_{ij} \quad \forall \quad \text{arco} = (i,j)
\end{array}$$

Es el mismo planteamiento que el problema de flujo fijo a coste mínimo, cambiando la función objetivo y teniendo en cuenta que F es una variable más del problema. En el planteamiento anterior se verifica la ley de conservación del flujo, en caso no ser así bastaría modificar las restricciones correspondientes como se hizo en el problema del flujo compatible a coste mínimo.

4. Problema de Planificación de la Producción

En todas las empresas en las que se producen productos hay que establecer un equilibrio óptimo entre la cantidad de unidades que se producen y se almacenan, teniendo en cuenta que la demanda suele ser variable. Se puede suponer que se fabrica un producto, cuya demanda puede variar en el tiempo y debe ser satisfecha por la empresa. Lo más habitual es realizar una producción constante, superior a la demanda y almacenar el sobrante para periodos con mayor demanda sin necesidad de aumentar el ritmo de producción. Sin embargo, hay que tener en cuenta los costes derivados del almacenamiento, incluso la pronta caducidad de los productos.

Cada fábrica tiene sus características propias, por lo que las restricciones a tener en cuenta pueden variar según el caso, pero todas presentan un objetivo común, llevar a cabo una planificación de la producción que maximice los beneficios finales considerando los costes de producción y almacenaje.

Si se suponen los siguientes datos:

- n periodos de tiempo a considerar.
- s_0: cantidad en el almacén al principio del primer periodo de tiempo.
- d_t: demanda del producto en el periodo t.
- s_{max}: cantidad máxima del almacén.
- v_t: precio de venta del producto en el periodo t.
- p_t: coste de producción en el periodo t.
- c_t: coste de almacenamiento en el periodo t.

Las variables de decisión a tener en cuenta son:

- x_t = unidades del producto producidas en el periodo t de tiempo.
- s_t = unidades del producto almacenadas en el periodo t.

La función objetivo maximiza las ganancias obtenidas descontando los costes derivados.

$$\text{Maximizar } z = \sum_{i=1}^{n} v_t d_t - p_t x_t - c_t s_t$$

Lo que equivale a minimizar los costes de almacén y producción.

Junto con las restricciones que limitan la producción en función de los recursos disponibles, que dependerán de las características propias de la empresa, hay que tener en cuenta las restricciones correspondientes con el almacenaje:

$$\begin{array}{rl} \text{Cantidad que se almacena en cada periodo} & s_{t-1} + x_t - d_t = s_t; \quad \forall t = 1, 2, \ldots, n \\ \text{Límite de almacenaje} & x_t \leq s_{max} \quad \forall t = 1, 2, \ldots, n \end{array}$$

5. Problema de Asignación

Los problemas de asignación tratan de asignar de manera óptima un conjunto de tareas a un conjunto de recursos, teniendo en cuenta las posibles restricciones que puedan derivarse, minimizando el coste o maximizando el beneficio asociado. Puede aplicarse a situaciones como la asignación de personal a proyectos, de máquinas a tareas o la asignación de rutas de distribución a vehículos.

Dado un conjunto de n recursos y un conjunto de n tareas, se quiere asignar cada recurso a una tarea de manera óptima. Para lo cual se definen variables de decisión binarias:

$$x_{ij} = \begin{cases} 1, & \text{si el recurso } i \text{ se asigna a la tarea } j, \\ 0, & \text{en caso contrario.} \end{cases}$$

El objetivo es encontrar la asignación óptima:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot x_{ij} \\ \text{Sujeto a:} \quad & \\ \text{Cada recurso se asigna a una tarea} \quad & \sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = 1, 2, \ldots, n \\ \text{Cada tarea se asigna a un recurso} \quad & \sum_{i=1}^{n} x_{ij} = 1 \quad \forall j = 1, 2, \ldots, n \\ & \text{Variables de decisión binarias} \\ & x_{ij} \in 0, 1 \quad \forall i, j = 1, 2, \ldots, n \end{aligned}$$

Donde c_{ij} representan el coste de asignar el recurso i a la tarea j.

Si el número de recursos y tareas no es el mismo, pueden sobrar recursos que no realicen ninguna tarea, o tareas que deben tener asignado más de un recurso. Estas variaciones se añaden al modelo modificando las correspondientes restricciones, o se puede equilibrar el problema añadiendo tantos recursos o tareas ficticias como sean necesarios.

El problema de asignación, se puede plantear como un caso particular del problema del transporte usando variables binarias, y tiene su propio algoritmo de resolución (Método Húngaro). En concreto, se puede resolver como problema de transporte de flujo fijo, usando el planteamiento visto anteriormente.

6. Problema de Camino Mínimo

Los caminos mínimos son una categoría de problemas que buscan encontrar la ruta más corta o de menor coste entre dos nodos específicos en un grafo. Estos problemas son fundamentales en la optimización de rutas, planificación de trayectorias y sistemas de navegación. Algoritmos clásicos como Dijkstra y Bellman-Ford se utilizan para resolverlos de manera eficiente, pero también se pueden resolver como problemas de PL, como un caso particular de una red de flujo. Para ello el objetivo es hacer pasar una unidad de flujo por el camino óptimo que es la que va marcando el camino y un flujo nulo por el resto de los arcos. Teniendo esto en cuenta se puede plantear como un problema de flujo fijo a coste mínimo usando variables binarias. Sea v_{ij} la distancia o coste del arco entre el nodo i y el j, se definen las variables: $x_{ij} = \begin{cases} 1, & \text{si el arco } (i,j) \text{ pertenece al camino óptimo,} \\ 0, & \text{en caso contrario.} \end{cases}$

La función objetivo a optimizar es: Min $\sum_i \sum_j v_{ij} x_{ij}$

Sujeto a:

$$\sum_h x_{Oh} - \sum_k x_{kO} = 1 \qquad \text{nodo origen}$$
$$\sum_h x_{ih} = \sum_k x_{ki} \quad \forall i \ \text{nodo intermedio}$$
$$\sum_h x_{Dh} - \sum_k x_{kD} = 1 \qquad \text{nodo destino}$$
$$x_{ij} \ \text{variables binarias}$$

Donde el vértice O se considera el origen y el vértice D sería el punto final del camino.

7. Problema de la Mochila

El llamado problema de la mochila (knapsack) presenta el caso de tener un espacio limitado donde almacenar unidades de diferentes productos que compiten por la capacidad del almacén. El ejemplo base es una mochila con una capacidad máxima C y un conjunto de n objetos, cada uno con un valor v_i y una ocupación de espacio c_i asociados. El objetivo es determinar la combinación de objetos que maximice el valor total que puede llevarse en la mochila, respetando la restricción de capacidad máxima permitida. Sea x_i la variable de decisión binaria que vale 1 si el objeto i se incluye en la mochila, y 0 si no se incluye:

$$\text{Maximizar} \quad \sum_{i=1}^{n} v_i \cdot x_i$$

Sujeto a:

$$\sum_{i=1}^{n} c_i x_i \leq C$$
$$x_{ij} \in \{0,1\} \quad \forall i = 1, 2, \ldots, n$$

Si hay que tener en cuenta el peso de cada objeto w_i y el que soporta la mochila W, se añade la restricción: $\sum_{i=1}^{n} w_i x_i \leq W$

Un ejemplo de aplicación es la elección de proyectos sin sobrepasar un presupuesto máximo disponible C , siendo v_i y c_i el valor y coste de cada proyecto respectivamente.

8. Problemas de cubrimiento

Los problemas de cubrimiento surgen ante la necesidad de decidir dónde ubicar instalaciones, radares, cámaras de vigilancia,... que cubran una determinada zona o área. Es decir, el objetivo es seleccionar un conjunto de elementos de un conjunto dado (cámaras) de manera que se cubran todos los elementos de otro conjunto (zonas a vigilar), minimizando cierta función de costes asociada a la selección de estos elementos.

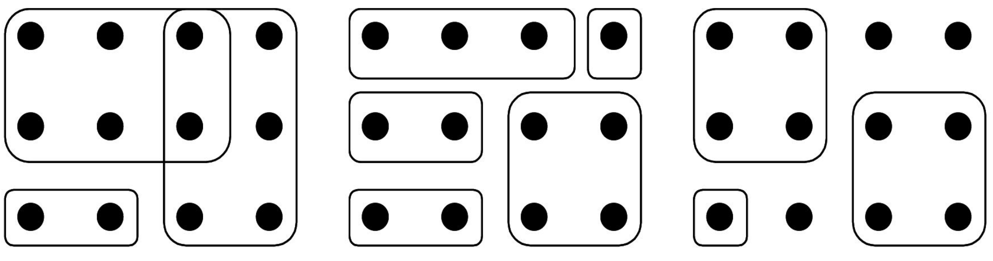
El problema parte de un conjunto de elementos que se deben cubrir (U) y un conjunto de subconjuntos (S) disponibles para seleccionar, los cuales llevan asociado un coste. El objetivo es seleccionar el subconjunto de S óptimo, de forma que los elementos contenidos en estos subconjuntos cubra todos los elementos en U. La restricción que hay que plantear es que cada elemento en U debe estar cubierto al menos una vez por los subconjuntos seleccionados.

El problema del cubrimiento se puede expresar en forma de programación lineal (PL) de la siguiente manera:

$$
\begin{aligned}
\text{Minimizar} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{Sujeto a} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad \forall\, i = 1, 2..., m \\
& x_j \in \{0, 1\} \qquad \forall\, j = 1, 2, ..., n
\end{aligned}
$$

Donde m es el número de elementos del conjunto U, y n es el número de subconjuntos de S. El coste del subconjunto j es c_j y las variables de decisión x_j son binarias que indican si el subconjunto j es seleccionado. La matriz a_{ij} es la matriz que indica si el elemento i está cubierto por el subconjunto j, (si lo cubre, $a_{ij} = 1$; de lo contrario, es 0).

El problema del cubrimiento es similar al problema de empaquetamiento y al problema de partición, ya que todos pertenecen a la categoría de problemas de optimización combinatoria y se relacionan con la selección de conjuntos de elementos para satisfacer ciertos criterios. En el problema de empaquetamiento $\sum_{j=1}^{n} a_{ij} x_j \leq 1$ el elemento no puede ser elegido más de una vez y en el problema de partición $\sum_{j=1}^{n} a_{ij} x_j = 1$ exactamente una.



Cubrimiento Partición Empaquetado

9. Problema de Planificación de Proyectos

Otro tipo de problemas que se pueden resolver con PL, aunque tienen sus propios algoritmos de resolución, son los de planificación de proyectos o método de ruta crítica (CPM), acrónimo de Critical Path Method.

Dado un proyecto, hay que conocer las actividades que lo forman, el tiempo que requieren para llevarlas a cabo y el orden de precedencia entre ellas. El proyecto se representa mediante una red según la relación de precedencia, donde los arcos representan las tareas, el valor del arco es el tiempo que tardan en realizarse y los nodos representan instantes temporales (comienzo o fin de tareas). Es necesario un nodo origen $i = 1$ y uno final $i = n$, y se numeran los nodos de menor a mayor, de forma que para todo arco (i, j), sea $i < j$. Para facilitar la interpretación del orden de precedencia entre tareas, hay que tener en cuenta que dos nodos a lo más se pueden conectar por un arco, añadiendo en caso necesario nodos y actividades ficticias. Una vez resuelto el camino óptimo, estas las tareas ficticias se eliminan para presentar el informe final.

Se denomina **actividad crítica** a aquella tarea que no puede modificar su comienzo o duración sin afectar a la duración total del proyecto. Se llama **camino crítico** al formado por las tareas críticas que determinan un camino entre el nodo origen y el final, dando la duración óptima del proyecto. Dada la red del proyecto, se definen las variables x_i como el instante de tiempo en el que ocurre lo correspondiente al nodo i (comienzo y fin de actividades), x_1 será el tiempo de inicio del proyecto y x_n el tiempo en el que acaba el proyecto. El objetivo de la planificación es terminar el proyecto en el menor tiempo posible, teniendo en cuenta el orden de precedencia de las tareas que lo forman:

$$\begin{aligned} \text{Min} \quad & x_n - x_1 \\ \text{Sujeto a:} \quad & \\ & x_j - x_i > t(i,j) \quad \forall (i,j) \end{aligned}$$

Donde $t(i, j)$ es la duración de la actividad que va del nodo i al j.

En ocasiones cuando se resuelve un problema de este tipo y se obtiene la solución final, no suele ser la deseable si lo que se pretende es realizar el proyecto lo antes posible, en estos casos, si se conoce el tiempo máximo que puede durar el proyecto y hay que terminarlo en dicho plazo, la única opción posible es reducir la duración de las actividades, con el correspondiente coste asociado. Para resolver este nuevo problema se puede replantear el problema anterior, definiendo, TM como el tiempo máximo de duración del proyecto, $C(i, j)$ coste por cada unidad de tiempo en que se reduzca la duración de la tarea (i, j), $r(i, j)$ es la máxima reducción de tiempo permitida para la tarea y X_{ij} unidades de tiempo en las que se reduce la duración de la actividad (i, j). Con todo esto, el problema a resolver sería:

$$\begin{aligned} \text{Min} \quad & \sum_{(i,j)} C(i,j) X_{ij} \\ \text{Sujeto a:} \quad & \\ & X_{ij} \leq r(i,j) \quad \forall (i,j) \\ & x_n - x_1 \leq TM \\ & x_j - x_i > t(i,j) - X_{ij} \quad \forall (i,j) \end{aligned}$$

10. Restricciones con variables binarias

La introducción de variables binarias en los problemas de programación lineal agrega una dimensión crucial para abordar decisiones discretas. Estas variables toman el valor de 0 o 1, lo que las convierte en una herramienta para expresar condiciones de si/no y restricciones condicionadas en el modelo matemático. En los siguientes apartados se muestra la modelización de las restricciones más usuales en la programación entera.

10.1. Restricciones: O BIEN

En muchas situaciones dadas dos restricciones en un problema, se quiere estar seguro de que se satisfaga por lo menos una de las dos restricciones.

$$h(x_1, ..., x_n) \leq 0 \quad \text{o bien} \quad g(x_1, ..., x_n) \leq 0$$

Para escribir esta condición se define una nueva variable $y = \{0, 1\}$ binaria, y las restricciones anteriores se sustituyen por:

$$h(x_1, ..., x_n) \leq My$$

$$g(x_1, ..., x_n) \leq M(1 - y)$$

donde M es un número suficientemente grande.

10.2. Restricciones: SI ENTONCES

Si lo que se desea es estar seguro de que se debe satisfacer una restricción $g(x_1, ..., x_n) \geq 0$ si se satisface la condición $h(x_1, ..., x_n) > 0$. Mientras que si no se cumple $h(x_1, ..., x_n) > 0$ entonces $g(x_1, ..., x_n) \geq 0$ puede o no verificarse. Es decir

$$Si \quad h(x_1, ..., x_n) > 0 \Rightarrow g(x_1, ..., x_n) \geq 0$$

Estas restricciones se sustituyen por

$$-g(x_1, ..., x_n) \leq My$$

$$h(x_1, ..., x_n) \leq M(1 - y)$$

donde M es un número suficientemente grande y la variable $y = \{0, 1\}$ es binaria.

10.3. Restricciones típicas con variables binarias

Dado que el uso de variables binarias representa la toma de decisiones en el planteamiento de muchos problemas, las condiciones a la hora de modelizar las correspondientes restricciones son muchas y variadas. A continuación, aparecen algunas de las más frecuentes y sencillas, que pueden servir de referencia para otras más complejas.

Suponiendo que $x_j = \{0, 1\}$ para todo j:

Si $x_i = 1 \Rightarrow x_j = 1$ se puede escribir $x_i \leq x_j$

Si $x_i = 0 \Rightarrow x_j = 0$ se puede escribir $x_j \leq x_i$

Si $x_i = 1 \Rightarrow x_j = 0$ se puede escribir $x_i \leq 1 - x_j$

Si $x_i = 0 \Rightarrow x_j = 1$ se puede escribir $1 - x_i \leq x_j$

El caso donde, o $x_i = 1$ o $x_j = 1$ se puede escribir $x_i + x_j \geq 1$

10.4. Deben cumplirse K de N restricciones

El modelo incluye un conjunto de N restricciones posibles de las que sólo K de ellas se deben cumplir. Las N-K restantes que no se eligen quedan eliminadas del problema, aun cuando por coincidencia la solución pueda satisfacer alguna de ellas. Sean las posibles restricciones:

$$g_1(x_1, ..., x_n) \leq b_1$$

$$g_2(x_1, ..., x_n) \leq b_2$$

$$...$$

$$g_N(x_1, ..., x_n) \leq b_N$$

Definiendo las correspondientes variables binarias, y_i, la formulación del problema es:

$$g_1(x_1, ..., x_n) \leq b_1 + My_1$$

$$g_2(x_1, ..., x_n) \leq b_2 + My_2$$

$$...$$

$$g_N(x_1, ..., x_n) \leq b_N + My_N$$

$$y_1 + y_2 + ... + y_N = N - K$$

10.5. Variables acotadas

Sea x_j la variable de decisión a calcular, la cual si es no nula tiene que tomar un valor entre dos cotas L_j y U_j. Para modelar esta condición se define la variable binaria y_j tal que:

$$y_j = \begin{cases} 1, & \text{si } x_j \neq 0 \\ 0, & \text{si } x_j = 0. \end{cases}$$

La restricción correspondiente sería:

$$L_j y_j \leq x_j \leq U_j y_j$$

10.6. Variables por niveles

Suponer que la variable x_j solo puede tomar unos valores o niveles definidos, v_1, v_2,...v_k. En este caso, se define la variable binaria y_{ij} tal que:

$$y_{ij} = \begin{cases} 1, & \text{si } x_j = v_i \\ 0, & \text{si } x_j \neq v_i. \end{cases}$$

Dado que x_j solo puede tomar uno de dichos valores, se escribe:

$$\sum_{i=1}^{k} y_{ij} = 1$$

Además, la variable x_j está expresada;

$$x_j = \sum_{i=1}^{k} v_i y_{ij}$$

10.7. Funciones lineales por partes

Una función lineal por partes no es realmente una función lineal, pero mediante el uso de variables binarias se puede expresar de forma lineal. Suponer que una función lineal por partes tiene los siguientes puntos de rotura $p_1, p_2, ..., p_n$, entonces para algún k se cumple que $p_k \leq x \leq p_{k+1}$, por lo que para algún número $0 \leq \alpha_k \leq 1$ se puede escribir $x = \alpha_k p_k + (1-\alpha_k)p_{k+1}$, y al ser $f(x)$ lineal podemos escribir $f(x) = \alpha_k f(p_k) + (1-\alpha_k)f(p_{k+1})$.

Por lo tanto, a la hora de modelizar se cambia $f(x)$ por $\alpha_1 f(p_1) + \alpha_2 f(p_2) + ... + \alpha_n f(p_n)$ y se añaden las siguientes condiciones de adyacencia:

$$\alpha_1 \leq y_1,$$
$$\alpha_2 \leq y_1 + y_2,$$
$$\alpha_3 \leq y_2 + y_3,$$
$$...$$
$$\alpha_{n-1} \leq y_{n-2} + y_{n-1}$$
$$\alpha_n \leq y_{n-1}$$

Teniendo en cuenta que:

$$y_1 + y_2 + ... + y_{n-1} = 1$$
$$\alpha_1 + \alpha_2 + ... + \alpha_n = 1$$
$$x = \alpha_1 p_1 + \alpha_2 p_2 + ... + \alpha_n p_n$$

donde $y_j = \{0, 1\}$ y solo dos α_j son no nulos y estos son consecutivos.