PROBLEMAS CLÁSICOS DE PROGRAMACIÓN ENTERA

1. Problema de la mochila (Knapsack Problem)

   Un excursionista dispone de una cantidad ilimitada de objetos de n tipos para introducir en su mochila. Cada objeto de tipo $j \in \{1, \ldots, n\}$ tiene una utilidad c_j y un peso a_j. Se desea maximizar la suma de las utilidades de los objetos elegidos sabiendo que el peso máximo que soporta la mochila es b.

2. Problema del transporte (Transportation Problem)

   Dados m orígenes y n destinos, para cada par de índices $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$ se denota por c_{ij} al coste de transportar una unidad de un determinado producto indivisible desde el origen i hasta el destino j.

   Sabiendo que en cada origen $i \in \{1, \ldots, m\}$ se dispone de a_i unidades del producto y que la demanda del mismo en cada destino $j \in \{1, \ldots, n\}$ es de b_j unidades, se trata de satisfacer dichas demandas minimizando el coste total del transporte.

3. Problema de asignación lineal pura (Pure Linear Assignment Problem)

   Se dispone de n máquinas y de n tareas a realizar, de forma que cada máquina ha de efectuar exactamente una tarea. Sabiendo que, para cada par de índices $i, j \in \{1, \ldots, n\}$, c_{ij} es el coste de asignar a la máquina i la tarea j, se desea obtener una asignación de las máquinas a las tareas (o viceversa) que minimice el coste total.

4. Problema de asignación lineal generalizada (Generalized Linear Assignment Problem)

   Dadas m máquinas y n tareas a realizar, para cada par de índices $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$ se denota por c_{ij} al coste de asignar a la máquina i la tarea j, y por a_{ij} al tiempo que la máquina i requiere para efectuar la tarea j.

   Sabiendo que, para cada $i \in \{1, \ldots, m\}$, b_i es el tiempo máximo que la máquina i puede estar en funcionamiento, se trata de asignar cada tarea a una única máquina de forma que se minimice el coste total.

5. Problema de asignación cuadrática (Quadratic Assignment Problem)

   El fabricante de un determinado producto ha seleccionado n lugares para construir una planta industrial en cada uno de ellos. Para cada par de índices $i, i' \in \{1, \ldots, n\}$, se denota por $a_{i,i'}$ al número de unidades del producto que deben transportarse desde la planta i hasta la planta i' y, para cada par de índices $j, j' \in \{1, \ldots, n\}$, se denota por $c_{j,j'}$ al coste de transportar una unidad del producto desde el lugar j hasta el lugar j'. Se desea determinar en qué lugar ha de construirse cada planta industrial para minimizar el coste total del transporte.

6. Problema de cubrimiento (Set-Covering Problem) y problema de particionamiento (Set-Partitioning Problem)

   Sean $I = \{1, \ldots, m\}$ un conjunto de elementos, $P = \{P_1, \ldots, P_n\}$ una familia de subconjuntos de I, y c_j el coste de seleccionar el subconjunto P_j, para cada $j \in \{1, \ldots, n\}$.

   Dado $F \subseteq \{1, \ldots, n\}$, se dice que $\{P_j\}_{j \in F}$ es un **cubrimiento** de I si $\bigcup_{j \in F} P_j = I$. Si, además, $P_j \cap P_{j'} = \emptyset \quad \forall j, j' \in F$ con $j \neq j'$, se dice que $\{P_j\}_{j \in F}$ es una **partición** de I.

   Los problemas de cubrimiento y particionamiento consisten, respectivamente, en obtener un cubrimiento y una partición de mínimo coste.

7. Problema del viajante (Traveling Salesman Problem)

   Dadas n ciudades, para cada par de índices distintos $i, j \in \{1, \ldots, n\}$ se denota por c_{ij} al tiempo requerido para ir directamente desde la ciudad i hasta la ciudad j.

   Partiendo de la ciudad 1, un agente comercial debe visitar exactamente una vez las ciudades $2, \ldots, n$ y regresar a la ciudad 1. Se desea determinar una ruta que minimice la duración total del viaje.

   Si $c_{ij} = c_{ji} \quad \forall i, j \in \{1, \ldots, n\}$ tales que $i \neq j$, se dice que el problema es **simétrico**; en caso contrario, el problema es **asimétrico**.