Capítulo 2

Problemas Clásicos de Optimización

2.1. Introducción

En este capítulo se presenta una colección de problemas clásicos de optimización. Debido a la gran cantidad de problemas diferentes descritos en la literatura, el conjunto de problemas recogidos aquí no es exhaustivo. Existen libros [10] e incluso páginas web [74] que recogen compendios muy amplios de descripciones de problemas de optimización. Además existen libros monográficos dedicados a problemas de optimización populares, como el problema del viajante de comercio [144] o del enrutamiento de vehículos [310]. En las sucesivas secciones se presenta una breve descripción de cada problema considerado, acompañada de aplicaciones.

2.2. El problema del viajante de comercio

El Problema del Viajante de Comercio (*Travelling Salesman Problem* - TSP) es, probablemente, el problema de optimización combinatoria más estudiado y que mejor se conoce [144]. De manera informal, se puede describir como la:

"*búsqueda de la ruta más corta de un viajero que comienza su viaje en una ciudad de origen, visita un conjunto prescrito de ciudades y vuelve a la ciudad de partida*"

La distancia recorrida en el viaje depende del orden en que se visitan las ciudades. Por lo tanto, este problema también se puede enunciar como [144][97]:

"*búsqueda del orden óptimo de visita de las ciudades, de forma que la distancia recorrida sea mínima y cada ciudad se visite exactamente una vez*"

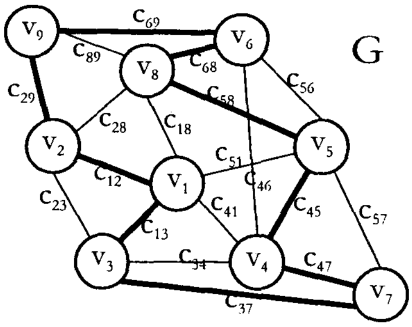

Figura 2.1: Problema del viajante en un grafo G. Las líneas gruesas representan un circuito hamiltoniano, es decir, una posible solución al problema dada por la permutación $[v_1, v_2, v_9,, v_6, v_8, v_5, v_4, v_7, v_3]$.

Como se puede inferir de esta descripción, no se necesitan herramientas matemáticas demasiado sofisticadas para formular matemáticamente el TSP. A pesar de ello, el TSP es un problema $\mathcal{NP}-$*duro* (como se demuestra en el trabajo de Karp [173]), y es muy difícil, e incluso imposible a veces [144], resolver instancias de gran tamaño de forma óptima. Para un grafo no dirigido completo de n vértices el tamaño del espacio de búsqueda es $(n-1)!/2$ [211].

Formalmente, el TSP puede representarse mediante un grafo no dirigido como el siguiente:

$$G = \{V, E, C\} \tag{2.1}$$

donde V es un conjunto de vértices que representa el conjunto de vértices, E es un conjunto de aristas con pesos que modela rutas entre ciudades y C es una matriz de pesos que representa el coste del viaje entre pares de ciudades adyacentes. Se supone que existe una arista conectando cada par de ciudades y los pesos $c_{ij} \in C$, asociados a las aristas $(i, j) \in E$ representan el coste entre las ciudades $i, j \in V$. En la figura 2.1 se representa gráficamente una solución posible al problema del TSP sobre un grafo G. Desde este punto de vista, el TSP se puede enunciar como el problema que consiste en encontrar el circuito hamiltoniano con menor longitud en el grafo G [333]. Si consideramos el problema de las n ciudades, es posible expresar una solución como el vector $[\pi(1), \ldots, \pi(n)]$ donde $\pi(i)$ representa el orden en el cual se visita la ciudad i. Considerando esta descripción, el problema del viajante se puede formular del siguiente modo:

$$TSP = \left\{ \; min: \quad c_{\pi(1)\pi(n)} + \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)} \right. \tag{2.2}$$

Las aplicaciones del TSP van más allá del problema de la planificación de rutas de un viajante de comercio y abarcan diferentes áreas de conocimiento, incluyendo las Matemáticas, Ciencias de la Computación, Investigación Operativa, Genética, Ingeniería y Electrónica [253]. Entre ellas destacan la planificación de máquinas, los procesos de fabricación, los problemas de asignación de frecuencias, etc. Existe una gran cantidad de versiones diferentes del TSP, como el TSP Simétrico (*Symmetric TSP*), TSP Máximo (*MAX TSP*), TSP con Cuello de Botella (*Bottleneck TSP*), TSP con Múltiples Visitas (*TSP with Multiple Visits*), Problema del Mensajero (*Messenger Problem*), TSP Agrupado (*Clustered TSP*), TSP Generalizado (*Generalized TSP*), TSP Dinámico (*Dynamic TSP*), etc [253].

2.3. El problema del enrutamiento de vehículos

El problema de enrutamiento de vehículos o *Vehicle Routing Problem* (VRP) consiste en:

"*obtener el conjunto de rutas más cortas posibles utilizando un conjunto de vehículos lo más pequeño posible tal que, partiendo de un almacén y regresando sucesivamente a él, abastecen a una serie de clientes distribuidos geográficamente, teniendo en cuenta que cada vehículo tiene una capacidad máxima propia y la cantidad de producto demandada por los clientes es diferente.*"

En términos de teoría de grafos, el VRP clásico se puede definir como sigue. sea G un grafo donde $V = \{v_1, \ldots, v_n\}$ es el conjunto de vértices los vértices y $A = \{(v_i, v_j) | v_i, v_j \in V \text{ and } i \neq j\}$ es el conjunto de arcos. El vértice v_0 representa el depósito, donde tiene su base una flota de m vehículos idénticos de capacidad Q, y los vértices restantes $V' = V \setminus \{v_0\}$ representan la localización de los clientes. Se define una matriz de costes $C = \{c_{ij}\}$ no negativa que satisface que $c_{ij} \leq c_{ik} + c_{kj}$. Se dice que el problema es simétrico cuando $c_{ij} = c_{ji}\ \forall (v_i, v_j) \in A$. En este caso, se suele sustituir A por el vector de aristas $E = \{(v_i, v_j)\}\ |v_i, v_j \in V,\ i \neq j$. Se asume que $m \in [\overline{m}, \underline{m}]$ con $\underline{m} = 1$ y $\overline{m} = n - 1$. El valor de m puede ser una variable o una constante, dependiendo de la aplicación. A cada vértice se le asocia una cantidad q_i que representa la cantidad de producto que demanda. El VRP consiste en determinar un conjunto m de rutas con coste total mínimo, comenzando y terminando en el depósito v_0, tal que cada vértice $v_{ij} \in V'$ se visite exactamente una vez por un vehículo y la cantidad total de producto repartido en cada ruta no exceda la capacidad Q del vehículo que sirve la ruta.

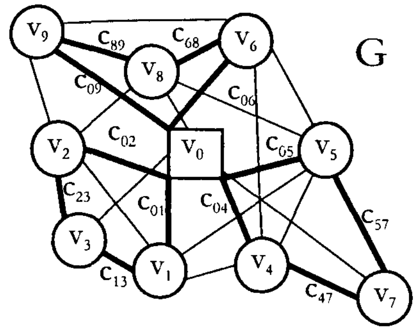

Figura 2.2: Problema del enrutamiento de vehículos en un grafo G. Las líneas gruesas representan una posible solución al problema dada por las rutas $m_1 = [v_0, v_2, v_3, v_1, v_0]$, $m_2 = [v_0, v_4, v_7, v_5, v_0]$ y $m_3 = [v_0, v_6, v_8, v_9, v_0]$.

Se define una solución al VRP como un conjunto de m rutas, $s = \{R_1, \ldots, R_m\}$, donde cada $R_k = (v_0, v_{k_1}, v_{k_2}, \ldots, v_0)$ es un conjunto ordenado que representa vertices consecutivos en la ruta k. Si seguimos una representación similar a la del TSP, podríamos representar una solución como $[\pi^1(1), \ldots, \pi^1(n_1), \ldots, \pi^m(1), \ldots, \pi^m(n_m)]$. De modo que, VRP puede formularse matemáticamente de la siguiente manera:

$$VRP = \begin{cases} min: \quad \sum_{k=1}^{m} \sum_{i=1}^{n_k} c_{\pi^k(i)\pi^k(i+1)} \\ s.a., \\ \sum_{i=1}^{n_k} q_{\pi^k(i)} \leq Q_k \ \forall k \in [1, m] \\ \sum_{k=1}^{m} n_k \leq n \end{cases} \tag{2.3}$$

Existe una gran cantidad de versiones derivadas del VRP, como el VRP con capacidades (*Capacitated VRP*), que es su versión homogénea. VRP con múltiples almacenes (*Multiple Depot VRP*), VRP cpn ventanas temporales (*VRP with Time Windows*) en el que se asocia una ventana temporal a cada cliente (con instante de inicio y de fin de servicio), VRP estocástico (*Stochastic VRP*) si cada componente tiene un comportamiento aleatorio, VRP periódico (*Periodic VRP*), VRP con atención dividida (*Split Delivery VRP*) cuando varios vehículos sirven a un mismo cliente, etc.

2.4. El problema de la mochila

El problema de la mochila o *Knapsack Problem* (KP) consiste en:

"dados n objetos, cada un con un peso w_j y un valor v_j, se debe seleccionar el conjunto de objetos cuyo valor total sea máximo, sin exceder un peso máximo W."

En este problema, una solución se suele representar mediante un vector $[x_1, \ldots, x_n]$, donde cada componente x_i representa un objeto i y su valor indica si el objeto se ha seleccionado (en cuyo caso el valor es 1) o no (en cuyo caso el valor es 0). Considerando esta representación, el problema de la mochila se puede formular del siguiente modo:

$$KP = \begin{cases} max: & \sum_{i=1}^{n} v_i x_i \\ s.a., & \\ \sum_{i=1}^{n} w_i x_i \leq W & \end{cases} \tag{2.4}$$

Este problema tiene numerosas aplicaciones como la denominada *Cutting Stock*, que consiste en cortar una plancha de acero en diferentes piezas. También se puede utilizar para modelar el problema de la determinación de los artículos que puede almacenar un depósito para maximizar su valor total. Indicar que el problema es relativamente fácil de resolver (algoritmos exactos en tiempo polinómico) cuando se permite que los objetos que se introducen en la mochila se puedan partir. En cambio, cuando dichos objetos no se pueden partir, el problema es $\mathcal{NP}-$*duro*.

2.5. El problema del corte máximo sobre grafos

El problema del *corte máximo* o *max-cut* consiste en:

"dado un grafo ponderado y no dirigido $G = \{V, A, W\}$, donde V es un conjunto de vértices, A es un conjunto de arcos no orientados o aristas y W es un conjunto de pesos w_{ij} asociados a cada arista $(i, j) \in A$, con $i, j \in V$, encontrar una partición del conjunto de vértices V de G en dos subconjuntos disjuntos $V = \{V_1, V_2\}$: $V_1 \cap V_2 = \emptyset$ $V_1 \cup V_2 = V$, tal que maximice la suma de los pesos de los arcos cuyos extremos se encuentran en subconjuntos distintos".

En la figura 2.3 se representa gráficamente un corte sobre un grafo G. La bipartición de V en V_1 y V_2 se denomina *grupo de corte* y el valor numérico de ese corte viene dado por la siguiente expresión:

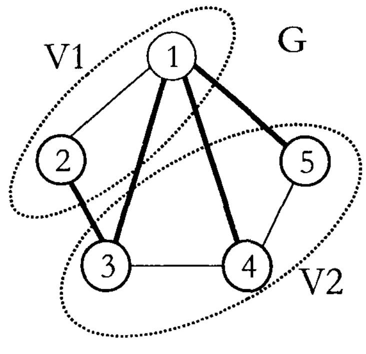

Figura 2.3: Grupo de corte para un grafo G.

$$cut(V_1,V_2) = \sum_{u\in V_1, v\in V_2} w_{uv} \tag{2.5}$$

Naturalmente, la solución a este problema consiste en la maximización de esta ecuación. En [173] se demuestra que la versión de decisión de *max-cut* es un problema $\mathcal{NP}-Completo$. Para grafos planos existen algoritmos con complejidad polinómica que resuelven dicho problema de forma exacta [146]. En [248] aparece una revisión muy extensa y detallada de este problema.

La resolución del problema del *max-cut* tiene importantes aplicaciones prácticas, entre las cuales se pueden destacar el diseño VLSI, el diseño de ASICS o la física estadística. En [14][55][61] se pueden encontrar descripciones de estas aplicaciones.

El problema del corte máximo se puede reformular como un problema de programación cuadrática entera P en los siguientes términos [81][102][134]:

$$MCP = \begin{cases} max: & \frac{1}{2} \sum_{1\leq i\leq j\leq n} w_{ij}(1-y_iy_j) \\ s.a., & \\ & y_i \in \{-1,1\}, \quad \forall i \in V \end{cases} \tag{2.6}$$

donde y_i se corresponde con la variable de decisión que determina si el vértice i pertenece a V_1 o a V_2.

Los conjuntos V_1 y V_2 en este formalismo vienen representados por:

$$V_1 = \{i \in V: \quad y_i = 1\} \qquad V_2 = \{i \in V: \quad y_i = -1\} \tag{2.7}$$

Luego, el grupo de corte viene definido en esta formulación como:

$$cut(V_1, V_2) = \frac{1}{2} \sum_{1 \leq i \leq j \leq n} w_{ij}(1 - y_i y_j) \tag{2.8}$$

En el caso general, este problema no se puede resolver de forma eficiente porque pertenece a la clase $\mathcal{NP} - Completa$. Debido al interés práctico que tiene su resolución, en la literatura se han propuesto varias relajaciones, entre las que destacan: la relajación a un problema continuo, la relajación a un problema de programación lineal y la relajación a un problema de programación semidefinida o SDP (*Semi-Definite Programming*). De estas alternativas, probablemente la más interesante es la SDP, ya que se resuelve en tiempo polinómico [139] y suele proporcionar soluciones más próximas al óptimo que los otros enfoques.

2.6. El problema de la máxima diversidad

El problema de la máxima diversidad o *maximum diversity problem* (MDP) consiste en:

"seleccionar un conjunto de elementos de una colección más grande de tal forma que los elementos seleccionados tengan las características más variadas entre sí"

En la figura 2.4 se representa el MDP sobre distancias euclídeas. Desde un punto de vista matemático, siendo d_{ij} la distancia entre los elementos i y j, el MDP se puede formular como el siguiente problema cuadrático binario:

$$MDP = \begin{cases} max: & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} x_i x_j \\ s.a., & \\ \sum_{i=1}^{n} x_i = m & \\ x_i = \{0,1\} \mid 1 \leq i \leq n & \end{cases} \tag{2.9}$$

Un ejemplo de aplicación de este problema aparece en la preservación de la biodiversidad. En este ámbito, se dispone de un número limitado de recursos para salvar únicamente a un número determinado de especies. En este escenario es más adecuado salvar a aquellas especies que muestren el conjunto más variado de características. En [184] se usa esta formulación para mostrar que el problema del clique (que es $\mathcal{NP} - completo$) es reducible al MDP.

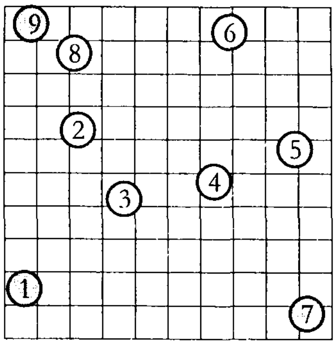

Figura 2.4: El problema de la selección de los tres elementos más diversos entre un conjunto de nueve localizados en el plano euclídeo. Los elementos seleccionados aparecen sombreados. En este caso, la solución viene determinada por la expresión: [1,0,0,0,0,0,1,0,1]

2.7. El problema de la ordenación lineal

El problema de la ordenación lineal o *lineal ordering problem* (LOP) se puede enunciar del siguiente modo:

"dada una matriz de pesos $E = \{e_{ij}\}$ de tamaño $n \times n$, encontrar una permutación p de las columnas (y filas) que maximice la suma de los pesos en el triángulo superior"

En términos matemáticos podemos expresarlo de este modo:

$$LOP = \left\{ \; max: \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} e_{p(i)p(j)} \right. \tag{2.10}$$

donde $p(i)$ es el índice de la columna (y fila) i en la permutación. Se debe señalar que en el LOP, la permutación p proporciona el ordenamiento tanto de la fila como de la columna. El problema equivalente en grafos consiste en encontrar, en un grafo completo ponderado, un torneo acíclico con una suma máxima de los pesos de los arcos [262].

En economía, LOP es equivalente al llamado problema de la triangulación (*triangulation problem*) para tablas de entrada-salida, que puede describirse como sigue: la economía de una región (generalmente un país) se divide en n sectores y se construye una tabla de entrada-salida E, donde la entrada e_{ij} representa el total de entregas (en valores monetarios) del sector i al sector j en un año dado. El problema de la triangulación consiste en permutar simultáneamente filas y columnas de E, para hacer la suma de las entradas sobre la diagonal principal lo más grande posible. Una solución óptima ordena los sectores de modo que los proveedores (es decir, los sectores que producen materiales para otras industrias) van primero, seguido de los consumidores (es decir, los sectores que fabrican el producto final que llega a los consumidores) [196].