



Figura 1: Universidad Nacional Autónoma de México

FACULTAD DE INGENIERÍA  
 DIMEI  
 DEPTO. DE INGENIERÍA DE SISTEMAS  
 PROGRAMA DE POSGRADO EN IDEO  
 NOTAS DE CLASE  
 OPTIMIZACIÓN COMBINATORIA  
 SEMESTRE: 2024-1  
 DRA. PATRICIA E. BALDERAS CAÑAS

## 1. ÍNDICE TEMÁTICO

1. INTRODUCCIÓN
2. GRAFOS
3. PROBLEMAS Y ALGORITMOS
4. ÁRBOLES ÓPTIMOS Y RUTAS
5. PROBLEMAS DE FLUJO MÁXIMO
6. PROBLEMAS DE FLUJO A MÍNIMO COSTO
7. EMPAREJAMIENTOS ÓPTIMOS
8. PROBLEMA DEL CARTERO CHINO
9. PROBLEMA DE LA MOCHILA
10. PROBLEMA DEL AGENTE VIAJERO

## 2. INTRODUCCIÓN

La optimización combinatoria es una rama de la optimización en matemáticas aplicadas y en ciencias de la computación, relacionada a la investigación de operaciones, teoría de algoritmos y teoría de la complejidad computacional. También está relacionada con otros campos, como la inteligencia artificial e ingeniería de software. Los algoritmos de optimización combinatoria resuelven instancias de problemas que se creen ser difíciles en general, explorando el espacio de soluciones (usualmente grande) para estas instancias. Los algoritmos de optimización combinatoria logran esto reduciendo el tamaño efectivo del espacio, y explorando el espacio de búsqueda eficientemente.

Los algoritmos de optimización combinatoria a menudo son implementados en lenguajes como C y C++ entre otros softwares inteligentes, en lenguajes de programación lógicos tales como Prolog, o incluso en lenguajes multi-paradigma tales como Oz.

Mediante el estudio de la teoría de la complejidad computacional es posible comprender la importancia de la optimización combinatoria. Los algoritmos de optimización combinatoria se relacionan comúnmente con problemas NP-hard. Dichos problemas en general no son resueltos eficientemente, sin embargo, varias aproximaciones de la teoría de la complejidad sugieren que ciertas instancias (ej. "pequeñas instancias") de estos problemas pueden ser resueltas eficientemente. Dichas instancias a menudo tienen ramificaciones prácticas muy importantes.

El contenido de estas Notas de Clase está organizado de acuerdo con la figura:

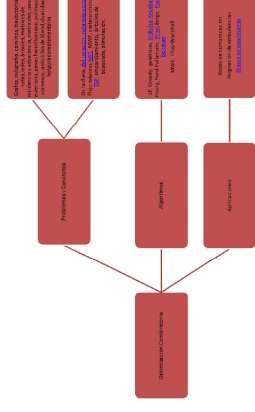


Figura 2: Síntesis

## 3. GRAFOS

### 3.1. Grafos y árboles

**Definition 1** Una gráfica  $G$  es una terna ordenada

$$(V(G), E(G), \Psi_G) \quad (1)$$

que consiste de un conjunto no vacío  $V(G)$  de vértices, un conjunto  $E(G)$ , ajeno a  $V(G)$ , de aristas o lados, y una función de incidencia  $\Psi_G$ , que asocia a cada lado de  $G$ , un par vértices, de  $G$ , no ordenado y no necesariamente distintos. Si  $e$  es una arista,  $u$  y  $v$  dos vértices tales que

$$\Psi_G(e) = uv$$

entonces se dice que  $e$  une a  $u$  y  $v$ , en este caso los vértices  $u$  y  $v$  se llaman extremos de  $e$ .

**Definition 2** El grado de un vértice  $\deg(v)$ , es el número de aristas de las cuales es un extremo, o que inciden en él.

**Example 3** Sea  $(V(G), E(G), \Psi_G)$ , donde

$$V(G) = \{v_1, v_2, v_3\} \quad (2)$$

$$E(G) = \{e_1, e_2\} \quad (3)$$

$$\Psi_G(e_1) = v_1v_2 \quad (4)$$

$$\Psi_G(e_2) = v_2v_3 \quad (5)$$

$$\Psi_G(e_3) = v_2v_3 \quad (6)$$

**Example 4** Sea  $(V(H), E(H), \Psi_H)$ , donde

$$V(H) = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E(H) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$$

$$\Psi_H(e_1) = v_1v_2$$

$$\Psi_H(e_2) = v_2v_3$$

$$\Psi_H(e_3) = v_3v_5$$

$$\Psi_H(e_4) = v_3v_4$$

$$\Psi_H(e_5) = v_2v_4$$

$$\Psi_H(e_6) = v_4v_5$$

$$\Psi_H(e_7) = v_2v_5$$

$$\Psi_H(e_8) = v_2v_5$$

la gráfica (el grafo) correspondiente es

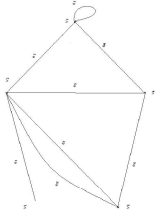


Figura 3: grafica2

**Example 5** Sea  $H = (V(HG), E(H), \psi_H)$ , donde

$$\begin{aligned}
 V(H) &= \{u, v, w, x, y\} \\
 E(H) &= \{a, b, c, d, e, f, g, h\} \\
 \psi_H(a) &= uv \\
 \psi_H(b) &= uu \\
 \psi_H(c) &= vw \\
 \psi_H(d) &= wx \\
 \psi_H(e) &= vx \\
 \psi_H(f) &= wx \\
 \psi_H(g) &= ux \\
 \psi_H(h) &= xy
 \end{aligned}$$

la gráfica (el grafo) correspondiente es

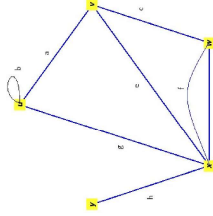


Figura 4: Grafo

La gráfica muestra la relación de incidencia. No es única la forma de mostrar el grafo. La posición relativa de los vértices no tiene significado. Las gráficas cuyas aristas sólo se intersectan en los vértices se llaman planas, por representarse en un plano de manera simple.

**Exercise 6** Dibujar las siguientes gráficas de manera que sean planas.

**Definition 7** Dos vértices que son incidentes a un lado común se llaman adyacentes. Así como los lados son adyacentes cuando tienen un vértice en común.

**Definition 8** Una arista con extremos idénticos, es decir, el mismo vértice, se llama lazo.

**Definition 9** Una arista con extremos distintos se llama vínculo.

**Definition 10** Una gráfica es finita si sus dos conjuntos que la definen,  $V$  y  $E$ , son finitos.

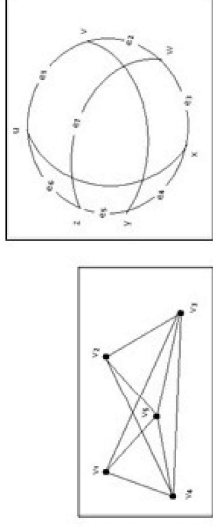


Figura 5: Gráficas no planas

**Definition 11** Una gráfica con un sólo vértice se denomina trivial y el resto de las gráficas no triviales.

**Definition 12** Una gráfica se denomina simple si no tiene lazos y dos vínculos no tienen el mismo par de vértices.

**Definition 13** Con la notación  $V(G)$  y  $E(G)$ , nos referimos a la cardinalidad de los conjuntos  $V(H)$  y  $E(H)$ , respectivamente.

**Definition 14** Dos gráficas son idénticas  $G = H$ , si

$$\begin{aligned}
 V(G) &= V(H) \\
 E(G) &= E(H) \\
 \psi_G &= \psi_H
 \end{aligned}$$

**Definition 15** Dos gráficas son isomórficas  $G \cong H$  si existen biyecciones, entre los vértices y las aristas,

$$\begin{aligned}
 \theta &: V(G) \rightarrow V(H) \\
 \phi &: E(G) \rightarrow E(H)
 \end{aligned}$$

tales que

$$\psi_G(\epsilon) = uv \Leftrightarrow \psi_H(\phi(\epsilon)) = \theta(u)\theta(v)$$

el par de mapeos  $(\theta, \phi)$  se llama isomorfismo entre  $G$  y  $H$ . El isomorfismo puede indicarse mediante los mapeos de incidencia.

**Example 16** Se presentan ambas gráficas de forma que se aprecie rápidamente el isomorfismo.

$$\begin{array}{l}
 G \\
 \theta(v_1) = y \\
 \theta(v_2) = x \\
 \theta(v_3) = u \\
 \theta(v_4) = v \\
 \theta(v_5) = w
 \end{array}
 \quad
 \begin{array}{l}
 H \\
 \phi(\epsilon_1) = h \\
 \phi(\epsilon_2) = g \\
 \phi(\epsilon_3) = b \\
 \phi(\epsilon_4) = a \\
 \phi(\epsilon_5) = e \\
 \phi(\epsilon_6) = c \\
 \phi(\epsilon_7) = d \\
 \phi(\epsilon_8) = f
 \end{array}
 \quad (7)$$

Una gráfica sin etiquetas es la representante de todas las gráficas isomórficas a ella.

**Definition 17** Una gráfica se llama completa cuando cada par de vértices distintos está unido por una arista.

**Notation 18** Salvo isomorfismo, sólo hay una gráfica completa con  $n$  vértices, la cual se representa por  $K_n$ .

**Example 19** Gráfica completa con cinco vértices:  $K_5$

**Definition 20** Una gráfica vacía es una sin aristas.

**Example 28** Gráfica y matriz de adyacencia

$$\begin{matrix} & v_1 & v_2 & v_3 & v_4 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix} \tag{8}$$

Propiedades. Sean  $M$  y  $A$  las matrices de incidencia y adyacencia de la gráfica  $G$ , respectivamente.

1. La suma de todas las celdas de una columna de  $M$  es 2.
2. La suma de todas las celdas de una columna de  $A$  es el grado del vértice, esto es, el número de aristas que lo tienen por extremo.

**Definition 29** Sobre subgráficas

1. Una gráfica  $H$  es una subgráfica de  $G$ ,  $H \subseteq G$ , si

$$V(H) \subseteq V(G) \tag{9}$$

$$E(H) \subseteq E(G) \tag{10}$$

$$\psi_H \text{ es la restricción de } \psi_G \text{ a } E(H) \tag{11}$$

En este caso, también se dice que  $G$  es una supergráfica de  $H$ .

2. Cuando  $H \subseteq G$ , pero  $H \neq G$  se dice que  $H$  es una subgráfica propia de  $G$ .
3. Una expansión  $H$  (subgráfica o supergráfica) de  $G$ , se da cuando  $V(H) = V(G)$ .
4. Quitando los lazos y dejando sólo una arista, de las que halla en cada par de vértices adyacentes, se tiene una subgráfica expansión simple de  $G$ , denominada la gráfica simple subyacente de  $G$ .

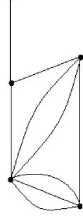


Figura 8: Gráfica simple subyacente

5. Un subgrafo de recubrimiento  $H$  de  $G$ , es un subgrafo tal que

$$V(H) = V(G) \tag{12}$$

**Definition 30** Supongamos que  $V' \subseteq V$ , no vacío. La gráfica de  $G$  cuyo conjunto de vértices es  $V'$  y el conjunto de aristas es el formado por aquellas de  $E(G)$ , que tienen sus dos extremos en  $V'$ , se llama subgráfica de  $G$  inducida por  $V'$  y se denota por  $G[V']$ .

**Definition 31** La subgráfica  $G[V - V'] = G - V'$ , de  $G$  es la subgráfica que se obtiene removiendo los vértices de  $V'$  junto con las aristas incidentes.

En particular, si  $V' = \{v\}$ ,  $G - v$  significa  $G - \{v\}$ .

**Definition 32** Supongamos que  $E'$  es un subconjunto no vacío de  $E$ . La subgráfica de  $G$  cuyo conjunto de vértices es el conjunto de extremos en  $E'$  y cuyo conjunto de aristas es  $E'$ , se llama la subgráfica de  $G$  inducida por  $E'$ , la cual denotamos por  $G[E']$ , esto es, la subgráfica de  $G$  inducida por el conjunto de aristas  $E'$ .

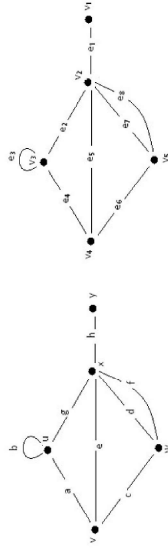


Figura 6: Gráficas isomorfas

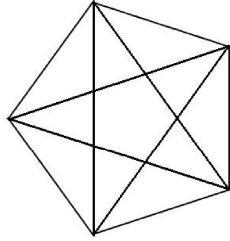


Figura 7: K5

**Definition 21** Una gráfica bipartita es una cuyo conjunto de vértices se puede partir en dos subconjuntos  $X$  y  $Y$ , de modo que, cada arista tiene un extremo en  $X$  y un extremo en  $Y$ .

**Notation 22** La partición se denota por  $(X, Y)$ .

**Definition 23** Una gráfica bipartita completa es una gráfica simple bipartita con partición  $(X, Y)$ , en la cual cada vértice de  $X$  está unido con un vértice de  $Y$ .

**Notation 24** Si  $v(X) = m$  y  $v(Y) = n$ , la gráfica bipartita completa se denota por  $K_{m,n}$ .

**3.2. Matrices de incidencia y adyacencia**

**Definition 25** Para toda gráfica  $G$ , con  $V(G) = \{v_1, \dots, v_m\}$  y  $E(G) = \{e_1, \dots, e_n\}$ , hay una matriz de dimensión  $m \times n$ , denominada matriz de incidencia, definida por

$$M(G) = [m_{ij}]$$

donde  $m_{ij}$  es el número de veces que el vértice  $v_i$  y la arista  $e_j$ , son incidentes. Nótese que  $m_{ij} = 0, 1, 2$ .

**Example 26** Matriz de incidencia

$$\begin{bmatrix} - & v_1 & v_2 & v_3 & v_4 & v_5 \\ v_1 & 1 & 1 & 0 & 0 & 1 \\ v_2 & 1 & 1 & 1 & 0 & 0 \\ v_3 & 0 & 0 & 1 & 1 & 0 \\ v_4 & 0 & 0 & 0 & 1 & 0 \\ v_5 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

**Definition 27** La matriz de adyacencia es una matriz de dimensión  $m \times m$ ,  $A(G) = [a_{ij}]$ , donde  $a_{ij}$  es el número de aristas que unen a los vértices  $v_i, v_j$ .

**Definition 33** La subgráfica expansión de  $G$  cuyo conjunto de aristas es  $E \setminus E'$ , se denota simplemente por  $G - E'$ , la cual se obtiene de  $G$  al remover las aristas en  $E'$ .

**Definition 34** Similarmente, la gráfica obtenida de  $G$ , en la que se agregan las aristas de  $E'$ , se denota por  $G + E'$ .

**Definition 35** Dos subgráficas  $G_1$  y  $G_2$  se dicen ajenas, si no tienen vértices en común.

**Definition 36** Se dicen ajenas en cuanto a las aristas, si no tienen aristas en común.

**Definition 37** La unión  $G_1 \cup G_2$ , de  $G_1$  y  $G_2$ , es la subgráfica donde el conjunto de vértices es  $V(G_1) \cup V(G_2)$  y el conjunto de aristas es  $E(G_1) \cup E(G_2)$ .

**Definition 38** En el caso de gráficas ajenas, la unión se suele representar por  $G_1 + G_2$ .

**Definition 39** La intersección  $G_1 \cap G_2$ , se define análogamente, solo se pide que, al menos, tengan un vértice en común, para que la intersección sea no vacía.

De la relación entre la suma de los grados y el número de aristas, en una gráfica, se tiene el siguiente:

**Corollary 40** En cualquier gráfica, el número de vértices de grado impar es par.

**Definition 41** Una gráfica  $G$  es  $k$ -regular si

$$d(v) = k \quad (13)$$

para todo  $v \in V(G)$ .

**Example 42** Las gráficas completas  $K_n$  y las gráficas bipartitas completas  $K_{m,n}$  son regulares. También los  $k$ -cubos. En gráfica se tiene el 3-cubo, con 8 vértices.

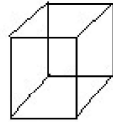


Figura 9: 3-cubo

### 3.3. Caminos, trayectorias y rutas

**Definition 43** Un camino (walk) en una gráfica  $G$  es una sucesión finita, no nula, de vértices y aristas, alternadas. El camino de  $v_0$  a  $v_k$ , denotado por  $(v_0, v_k)$ , es un camino con  $k+1$  vértices y  $k$  aristas:

$$W = v_0 e_1 v_1 e_2 v_2 \dots e_k v_k \quad (14)$$

cuyos términos son sucesivamente, vértices y aristas, tal que, los extremos de  $e_i$  son  $v_{i-1}$  y  $v_i$ , con  $1 \leq i \leq k$ . El origen es  $v_0$ , el final es  $v_k$ , y los vértices internos son  $v_1 \dots v_{k-1}$ . La longitud del camino es  $k$ .

**Example 44** En la gráfica se resalta el camino

$$W = v_1 e_1 v_2 e_3 v_3 e_4 v_5 e_5 v_6 e_7 v_6 \quad (15)$$

**Definition 45** Un camino cerrado tiene longitud positiva y su origen y fin son el mismo vértice.

**Definition 46** Un camino euleriano es un camino que pasa por cada arista una y solo una vez.

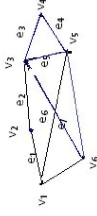


Figura 10: Camino

**Definition 47** Un ciclo o circuito euleriano es un camino cerrado que recorre cada arista exactamente una vez.

El problema de encontrar dichos caminos fue discutido por primera vez por Leonhard Euler, en el famoso problema de los puentes de Königsberg.

**Definition 48** Un paseo (trail) es un camino donde las aristas son diferentes.

**Definition 49** Un trayecto o ruta (path) es un paseo donde todos los vértices son distintos.

**Definition 50** Un trayecto cerrado con origen y vértices internos, distintos es un ciclo.

**Example 51** Se ilustra un ciclo y un trayecto cerrado:

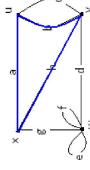


Figura 11: Ciclo

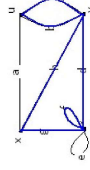


Figura 12: Camino cerrado

**Definition 52** Un grafo es conexo si, para cualquier par de sus vértices  $u$  y  $v$ , existe una  $(u, v)$ -trayectoria en el grafo.

**Definition 53** Una subgráfica  $H$  es maximal respecto a una propiedad, si  $H$  cumple la propiedad y no es subgráfica propia de otro grafo que cumpla la propiedad.

**Definition 54** Un subgrafo  $H$  es minimal con respecto a una propiedad, si  $H$  cumple la propiedad y no existe un subgrafo propio de  $H$  que cumpla la propiedad.

**Definition 55** Una componente de una gráfica es un subgrafo conexo maximal.

**Corollary 56** Una gráfica es conexa si tiene exactamente una componente.

**Definition 57** Una gráfica con más de una componente se denomina disconexo.





Figura 13: Gráfica bipartita

**Definition 58** Una gráfica es bipartita si el conjunto de sus vértices se puede partir en dos subconjuntos tales que cada arista tiene un extremo en cada uno de esos subconjuntos.

**Theorem 59** Una gráfica es bipartita si y sólo si contiene ciclos no impares.

**Example 60**

**Definition 61** Un árbol es un grafo acíclico conexo.

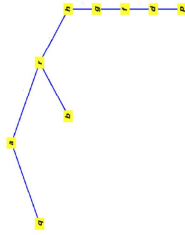


Figura 14: Árbol

**Example 62**

**Proposition 63** Si  $T$  es un árbol entonces para cualquier par,  $u$  y  $v$ , de sus vértices, existe una única trayectoria  $(u, v)$ .

**Proposition 64** Si  $T$  es un árbol no trivial, entonces existen al menos dos vértices distintos de grado uno.

**Proposition 65** Si  $T$  es un árbol con  $n$  vértices, entonces el número de aristas es  $n - 1$ .

**Definition 66** Un puente en un grafo es una arista  $e$  tal que el número de componentes de  $G - e$  es mayor que el número de componentes de  $G$ . En particular, si  $G$  es conexo entonces  $e$  es un puente de  $G$  si y sólo si  $G - e$  es disconexo.

**Example 67** La arista  $c$  es un puente de la gráfica siguiente.



Figura 15: Puente

## 4. PROBLEMAS Y ALGORITMOS

### 4.1. Problema del conector

Se tienen  $n$  ciudades y se quiere diseñar una red de comunicación conectando a todas ellas. Para ciertos pares de ciudades se puede construir una vía de comunicación directa entre ellas, y se conoce el costo de tal vía. El problema del conector

consiste en determinar qué vías de comunicación directa deben construirse, con el fin de que todas la  $n$  ciudades queden conectadas a un costo total mínimo.

Es decir, dado un grafo conexo  $G$  y una función de costo  $c : E(G) \rightarrow \mathfrak{R}$ , tal que  $c(e) > 0$  para toda  $e \in E(G)$ , el problema del conector consiste en hallar un subgrafo de recubrimiento conexo  $H$  tal que

$$c(H) = \sum_{e \in E(H)} c(e) \quad (16)$$

sea mínimo.

### 4.2. Problema del flujo máximo

**Example 68** Para la red que se muestra a continuación, use el algoritmo de la ruta de flujo máximo para encontrar el patrón de flujo que da el flujo máximo desde la fuente (source) hasta el sumidero (sink), dado que la capacidad del arco desde el nodo  $i$  al nodo  $j$  es el número del nodo  $i$  más cercano a lo largo del arco entre estos nodos.

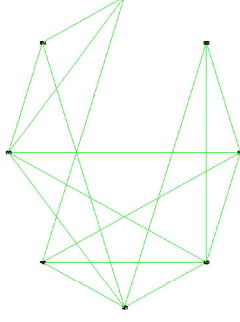


Figura 16: Encontrar el flujo máximo para este grafo.

**Exercise 69** Reproduzca en una hoja de cálculo el problema de la figura y dibuja la ruta máxima hallada.

### 4.3. Ruta más corta

Cuando cada arista  $e$  de  $G$  tiene asociado su peso  $w(e)$ , tenemos la siguiente:

**Definition 70** Una gráfica  $G$ , junto con sus pesos en las aristas se denomina gráfica ponderada (weighted graph).

**Definition 71** Si  $H$  es una subgráfica de una gráfica ponderada  $G$ , el peso

$$w(H) = \sum_{e \in H} w(e) \quad (17)$$

es decir, el peso de  $H$  es la suma de los pesos de sus aristas.

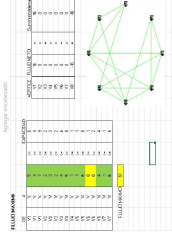


Figura 17: Solución hoja de cálculo

Muchos problemas de optimización de cantidades, tienen que ver con encontrar una subgráfica de una gráfica ponderada, de cierto tipo con mínimo (máximo) peso, como el problema de la ruta más corta, que dice que dada una red (ferroviaria) que conecta a varios pueblos o ciudades, se busca determinar la ruta más corta, entre dos pueblos en particular, en la red.

Lo que significa, encontrar en una gráfica ponderada, una ruta de mínimo peso, que conecte dos vértices  $u_0$  y  $v_0$ . Los pesos representan las distancias por la vía entre dos pueblos distintos, conectados directamente, es decir entre vértices distintos adyacentes, por tanto con distancia no negativa.

**Example 72** En la gráfica se resalta la subgráfica que es la  $(u_0, v_0)$  ruta de mínimo peso

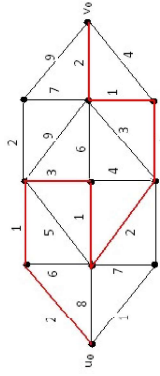


Figura 18: Ruta de mínimo peso

**Notation 73** La longitud de una subgráfica ponderada se representa por  $w(H)$ .

**Definition 74** El mínimo peso de una  $(u, v)$  ruta es la distancia entre  $u$  y  $v$ , la cual se denota por  $d(u, v)$ .

En las definiciones anteriores y en la formulación del problema de la ruta más corta, se asume que la gráfica  $G$  es simple.

**Example 75** Usted necesita hacer un viaje en automóvil a un pueblo que usted nunca antes ha visitado. Por consiguiente, está estudiando un mapa para determinar la ruta más corta a su destino. Dependiendo de la ruta que usted elija, hay otros cinco pueblos (llámelos A, B, C, D, E) a través de los cuales usted podría pasar en el camino. El mapa muestra la distancia en millas a lo largo de cada camino que conecta directamente dos pueblos sin ningún pueblo intermedio. Estos números se resumen en la tabla siguiente, donde un guión indica que no existe camino que conecte directamente estos dos pueblos, sin pasar por cualquier otro pueblo.

Pueblo	Millas entre los pueblos cercanos				
	A	B	C	D	E
Origen	40	60	50	-	-
A	-	10	-	70	-
B	-	-	20	55	-
C	-	-	-	-	50
D	-	-	-	-	.0
E	-	-	-	-	-

Figura 19: Ruta más corta.

1. Formule este problema como un problema de ruta más corta dibujando una red donde los nodos representen pueblos, los arcos representen caminos y los números indican la longitud de cada arco en millas.
2. Identifique la ruta más corta.
3. Si cada número en la tabla representara su costo (en U.M.) por manejar su automóvil de un pueblo al próximo, ¿daría ahora la respuesta en la parte b su ruta a costo mínimo?
4. Si cada número en la tabla representara su tiempo (en minutos) por manejar su automóvil de un pueblo al próximo, ¿daría la respuesta en la parte b a su ruta a tiempo mínimo?

**Definition 76** Un algoritmo es un conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas. También puede decirse que es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite solucionar un problema, realizar un cómputo, procesar datos o llevar a cabo otras tareas o actividades (Real Academia de la Lengua Española, <https://dle.rae.es/algorithm>, fecha de consulta 02/09/2021)

**Definition 77** Un árbol de expansión o recubrimiento es un subgrafo de expansión o recubrimiento acíclico, que contiene a todos los vértices del grafo.

**Algorithm 78** Dijkstra

1. Sea

$$l(u_0) = 0 \tag{18}$$

$$L(v) = \infty, v \neq u_0 \tag{19}$$

$$S = \{u_0\} \tag{20}$$

$$i = 0 \tag{21}$$

$$2. \text{ Para cada } v \in \bar{S}_i, \text{ reemplace } l(v) \text{ por } \min\{l(v), l(u_i) + w(u_i, v)\}. \text{ Calcule} \\ \min\{l(v)\} \\ v \in \bar{S}_i \tag{22}$$

y sea  $u_{i+1}$  un vértice para el cual este mínimo se alcanza. Sea  $S_{i+1} = S_i \cup \{u_{i+1}\}$ .

3. Si  $i = v - 1$ , pare. Si  $i < v - 1$ , reemplace  $i$  por  $i + 1$  y vaya al paso 2.

Cuando finaliza el algoritmo, la distancia de  $u_0$  a  $v$  está dada por el valor final de  $l(v)$ . Si el interés es determinar la distancia a un vértice específico  $v_0$ , nos detenemos en cuanto algún  $u_i$  es igual a  $v_0$ . En el siguiente diagrama se presenta el algoritmo de Dijkstra.

#### 4.4. Árboles de expansión a mínimo costo (MST)

**Definition 79** Arbol de expansión

**Example 80** Encuentre el árbol de expansión a costo mínimo MST, de la red con nodos A, B, C, D, E, F y G. Con la asignación de pesos para las aristas, como se ve en la figura.

**Example 81** Encontrar el MST por el algoritmo Kruskal, para el grafo<sup>1</sup>

**Solution 82** A continuación se muestra el MST, cuya longitud es 51.

**Exercise 83** Un avión carguero tiene tres compartimientos para guardar la carga: al frente, al centro y atrás. Estos compartimientos tienen límites de capacidad, tanto en el peso como en el espacio, como se resume en seguida.<sup>2</sup>

Compartimiento	Capacidad de Peso (Tons)	Capacidad de Espacio (Pies Cúbicos)
Frente	72	7,000
Centro	78	9,000
Atrás	10	5,000

Además, el peso de la carga en los compartimientos respectivos debe tener la misma proporción que la capacidad de peso de ese compartimiento para mantener el equilibrio del avión.

Se han propuesto las siguientes cuatro cargas para su embarque en un próximo vuelo considerando el espacio disponible. Puede aceptarse cualquier parte de estas cargas. El objetivo es determinar cuánto (si es posible) de cada carga debe aceptarse y cómo distribuir cada una entre los compartimientos para maximizar la utilidad total del vuelo.

Carga	Peso (Tons)	Volumen (Pies Cúbicos)	Utilidad (\$ / Tons)
1	20	500	320
2	16	700	400
3	25	600	360
4	13	400	290

<sup>1</sup> Tomado de Cook, W. et al. (1997) Combinatorial Optimization, p. 17

<sup>2</sup> Problema 3.20 tomado de (Hillier, et al. 2002)

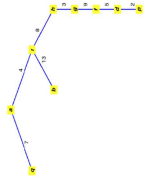


Figura 23: Solución para el MST

1. Exite es un problema mixto de distribución de recursos pero también incluye un número pequeño de restricciones de requerimientos fijos. Construya una tabla de parámetros que incluya todo excepto las restricciones de requerimientos fijos. Luego, construya por separado una tabla de parámetros con las restricciones de requerimientos fijos.
2. Formule y resuelva un modelo de programación lineal en una hoja de cálculo.
3. Exprese el modelo en forma algebraica.

Figura 24: Solución generada en Excel

**Solution 84**

**4.5. árboles de expansión mínima (MST) y programación lineal**

Hay una conexión interesante entre los árboles de expansión mínima y la programación lineal, esto es, hay un problema de programación lineal para el cual cada árbol de expansión mínima proporciona una solución óptima. Considere el siguiente problema de programación lineal:

Minimizar:  $c^T x$   
 Sujeta a:  
 $x(\gamma(S)) \leq |S| - 1$  para todo  $S \quad \emptyset \neq S \subset V$  (23)  
 $x(E) = |V| - 1$   
 $x_e \geq 0$  para todo  $e \in E$

**Theorem 85** Sea  $x^0$  el vector característico de un MST con respecto a los costos  $c_e$ . Entonces,  $x^0$  es una solución óptima del problema anterior.

**Proof 86** Se reformula el problema para tener un manejo más fácil, como sigue

Minimizar:  $c^T x$   
 Sujeta a:  
 $x(A) \leq |V| - \kappa(A)$  para todo  $A \subset E$  (24)  
 $x(E) = |V| - 1$   
 $x_e \geq 0$  para todo  $e \in E$

La afirmación es que ambos problemas tienen las mismas soluciones factibles, y de ahí que las mismas soluciones óptimas. Cada restricción de la forma

$$x(\gamma(S)) \leq |S| - 1 \quad (25)$$

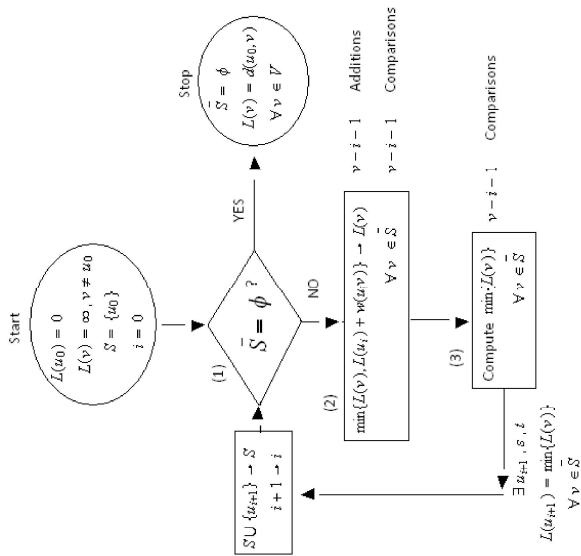


Figura 20: Algoritmo Dijkstra

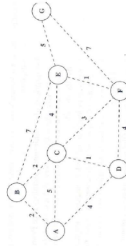


Figura 21: Red con 8 nodos

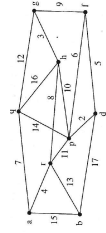


Figura 22: Red para ejemplo Kruskal

es consecuencia de

$$x(A) \leq |V| - \kappa(A) \quad (26)$$

tomado  $A = \gamma(S)$ . Nótese que:

$$\kappa(\gamma(S)) \geq |V| |S| + 1 \quad (27)$$

Recíprocamente, cada restricción

$$x(A) \leq |V| - \kappa(A) \quad (28)$$

es consecuencia de una combinación de las restricciones

$$\begin{aligned} x(\gamma(S)) \leq |S| - 1 & \text{ para todo } S \quad \phi \neq S \subset V \\ x(E) = |V| - 1 \end{aligned} \quad (29)$$

Sea  $A \subset E$  y  $S_1, \dots, S_k$  los conjuntos de nodos de la subgráfica  $(V, A)$ . Entonces

$$x(A) \leq \sum_{i=1}^k x(\gamma(S_i)) \leq \sum_{i=1}^k (|S_i| - 1) = |V| - k \quad (30)$$

Es suficiente probar que  $x^0$  es óptima para el problema reformulado, es más, que es suficiente probar que esto es cierto cuando  $x^0$  es el vector característico del árbol de expansión  $T$  generado por el algoritmo Kruskal. La optimalidad del vector característico se prueba mostrando que dicho algoritmo puede usarse para calcular una solución factible del problema dual que satisface la holgura complementaria con  $x^0$ . Para facilitar la construcción del problema dual, reemplazamos el objetivo de minimizar  $c^T x$ , por maximizar  $-c^T x$ . De ahí que el problema dual es

Minimizar:  $\sum_{A \in E} (|V| - \kappa(A)) y_A$

Sujeta a:

$$\sum_{y_A : e \in A} y_A \geq -c_e \quad \text{para todo } e \in E \quad (31)$$

$$y_A \geq 0 \quad \text{para todo } A \subset E$$

Nótese que no se requiere que  $y_E$  sea no-negativo. Sea  $e_1, \dots, e_m \in E$  el orden en el cual el algoritmo Kruskal considera los arcos.

Sea

$$R_i = \{e_1, \dots, e_i\} \quad (32)$$

para  $1 \leq i \leq m-1$  y

$$y_{R_i}^0 = c_{e_{i+1}} - c_{e_i} \quad (33)$$

para  $1 \leq i \leq m-1$  y

$$y_{R_m}^0 = -c_{e_m} \quad (34)$$

Se sigue del orden en los arcos que

$$y_A^0 \geq 0 \quad (35)$$

para  $A \neq E$ . De la primera restricción del problema dual, con  $e = e_i$  se tiene

$$\sum_{y_A : e \in A} y_A^0 = \sum_{j=1}^m y_{R_j}^0 = \sum_{j=1}^m (c_{e_{j+1}} - c_{e_j}) - c_{e_m} = -c_{e_i} = c_e \quad (36)$$

En otras palabras, todas las restricciones se cumplen con igualdad. Falta mostrar que  $y^0$  es una solución factible del problema dual, y que por holgura complementaria, la condición

$$x^0 > 0 \quad (37)$$

implica que la igualdad en la respectiva restricción dual, se satisface. Finalmente, falta verificar que

$$y_A^0 > 0 \quad (38)$$

implica que  $x^0$  satisface la primera restricción, con igualdad. Para ello, se sabe que

$$A = R_i \quad (39)$$

para alguna  $i$ . Si la restricción no se cumple con igualdad, entonces hay una arista en  $R_i$ , cuya adición a

$$T \cap R_i \quad (40)$$

15

reduce el número de componentes de

$$(V, T \cap R_i) \quad (41)$$

Pero una arista de ese tipo tendría sus extremos en dos componentes distintas de  $(V, T \cap R_i)$ , de ahí que se habría agregado a  $T$ , debido al algoritmo Kruskal. Por lo tanto,  $x^0$  y  $y^0$  satisfacen las condiciones de holgura complementaria. De donde se tiene que  $x^0$  es una solución óptima del problema reformulado y el problema original.

**Remark 87** Dado que cualquier árbol de expansión que proporciona una solución óptima  $n$  del problema de programación lineal debe ser un MST, y debido a que la demostración anterior solo utiliza el hecho de que  $T$  fue generado por el algoritmo Kruskal, esa prueba es una prueba de que dicho algoritmo calcula un MST.

**Problem 88** Dinero en movimiento<sup>3</sup>. Jake Nguyen pasa nerviosamente una mano a través de su pelo, alguna vez peinado finalmente, se afloja el mismo tiempo su corbata de seda perfectamente cuidada y se frota sus manos sudadas en sus, alguna vez pantalones impecablemente planchados. Hoy no ha sido ciertamente un buen día.

Durante los últimos meses, Jake había oído rumores que circulaban en Wall Street - rumores de los labios de banqueros inversionistas y corredores de la bolsa conocidos por su franqueza. Hacían murmuraciones acerca de un próximo colapso de la economía japonesa - las murmuraciones eran porque pensaban que haciendo públicos sus temores acelerarían el derrumbamiento.

Hoy, sus mismos temores se han hecho realidad. Jake y sus colegas se reúnen alrededor de una pequeña televisión dedicada exclusivamente al canal de Bloomberg (canal 315 de Sky). Jake mira fijamente con escepticismo y escucha los horrores que tienen lugar en el mercado japonés. El mercado japonés está arrastrando todos los mercados financieros de los demás países Asiáticos Orientales en su vertiginosa caída. Jake se queda paralizado. Como gerente de la inversión extranjera Asiática para Grant Hill Associates, una pequeña empresa de inversión de la Costa Oriental que se especializa en la comercialización del dinero. Jake lleva la responsabilidad personal por cualquier impacto negativo de este colapso. Y Grant Hill Associates experimentará impactos negativos.

Jake no había hecho caso de las murmuraciones que advertían un derrumbamiento japonés. En cambio, él había aumentado en gran manera el riesgo de Grant Hill Associates de quedar fuera del mercado japonés. Debido a que el mercado japonés se había comportado mejor de lo esperado durante el último año. Jake había aumentado las inversiones en Japón de 2.5 millones de a 15 millones de dólares, hace sólo un mes. En ese momento, la paridad del dólar era de 80 yens.

No más. Jake comprende que la devaluación de hoy del yen significa que un dólar equivale a 125 yens. El podrá liquidar las inversiones sin pérdida en yens. La pérdida de dólares cuando convirtiera los yens en dólares americanos sería enorme. Jake respira profundamente, cierra los ojos y mentalmente se prepara para reparar los severos daños.

La mediación de Jake es interrumpida por una potente voz que lo llama desde una esquina de su gran oficina. Grant Hill, el presidente de Grant Hill Associates, vocifera "¡Nguyen, esto es un infierno!"

Jake da un brinco y mira renuente hacia la esquina de la oficina donde aparece la figura de Grant Hill hecho una furia. Jake entonces se alisa el cabello, se aprieta la corbata, y camina aprisa por la oficina.

Grant Hill le sale al paso a Jake, lo mira fijamente a los ojos y le sigue gritando. No me diga ni una sola palabra. Nguyen! No hay pretextos: simplemente arregle este desastre! ¡¡¡Saque todo nuestro dinero de Japón! ¡ Mi instinto me dice que éste es sólo el principio! ¡ Invierta el dinero en bonos americanos confiables! ¡ ¡¡AHORA! Y no olvide sacar nuestro dinero en efectivo de Indonesia y Malasia!

Jake tiene bastante sentido común para no hacer ningún comentario. él inclina su cabeza, gira sobre los tacones de sus zapatos, y prácticamente sale volando de la oficina.

Seguramente atrás de su escritorio, Jake empieza a formular un plan para sacar sus inversiones de Japón, Indonesia y Malasia. Sus experiencias de invertir en mercados extranjeros le han enseñado que jugar con millones de dólares, así como sacar dinero de un mercado extranjero es casi tan importante como sacar dinero del mercado. Los socios de la banca de Grant Hill Associates cobran diferentes honorarios por la transacción de convertir un tipo de dinero en otro y remitir grandes sumas de dinero alrededor del mundo. Y ahora, las cosas han empeorado, los gobiernos en el Asia Oriental han impuesto límites muy estrictos en la cantidad de dinero que un individuo o compañía puede cambiar del dinero doméstico a una divisa particular y pueda sacarlo del país. El objetivo de esta dramática medida es reducir la salida de inversiones extranjeras de esos países para prevenir un colapso total de las economías en la región. Debido a las posiciones en efectivo de Grant Hill Associates de 10.5 billones de rúpias en Indonesia y 28 millones de ringgits en Malasia, junto con los yenes, no está claro cómo deben convertir esas divisas en dólares.

Jake quiere convertir esas divisas en dólares. Jake quiere encontrar siempre los tipos de cambio, en todo momento para la mayoría de las monedas en el mundo, vea la tabla:

Tipos de cambio de divisas

<sup>3</sup> Tomado y adaptado de Hillier, F., Hillier, M., Hillier, and Lieberman, G. (2001) Caso 6.2

DE/A	Yen	Rupia	Ringgit	US Dolar	C Dolar	Euro	Libra	Peso
Yen japonés	1	50	0.04	0.008	0.01	0.0064	0.0048	0.0768
Rupia hindia		1	0.0008	0.00016	0.0002	0.000128	0.00096	0.001536
Ringgit malasio			1	0.2	0.16	0.12	1.92	
US Dolar				1	1.25	0.8	0.6	9.6
C Dolar					1	0.64	0.48	7.68
Euro						1	0.75	12
Libra británica							1	16
Peso mexicano								1

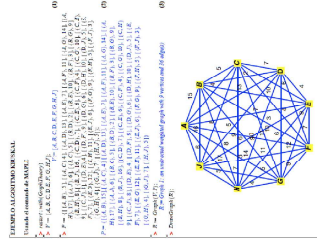


Figura 25: Algoritmo Kruskal con Maple 16

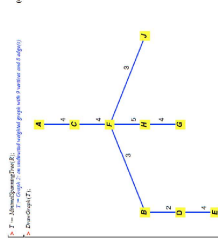


Figura 26: Solución MST

La tabla anterior indica que, por ejemplo, 1 yen japonés equivale a 0.008 dólares americanos. Haciendo unas llamadas telefónicas, conoce los costos de la transacción que su compañía debe pagar por las grandes transacciones de divisas durante estos tiempos críticos, vea la tabla 2.

Jake nota que al cambiar una moneda por cualquier otra se produce el mismo costo de la transacción que para una conversión inversa. Finalmente, Jake averigua las cantidades máximas de dineros domésticos que su compañía tiene permitido convertir a otros tipos de moneda en Japón, Indonesia y Malasia, vea la tabla 3.

	Yen	Rupia	Ringgit	US Dolar	C Dolar	Euro	Libra	Peso
Yen japonés	1	50	0.04	0.008	0.01	0.0064	0.0048	0.0768
Rupia hindia		1	0.0008	0.00016	0.0002	0.000128	0.00096	0.001536
Ringgit malasio			1	0.2	0.16	0.12	1.92	
US Dolar				1	1.25	0.8	0.6	9.6
C Dolar					1	0.64	0.48	7.68
Euro						1	0.75	12
Libra británica							1	16
Peso mexicano								1

Figura 27: Tabla 2. Costo por transacción (porcentaje)

	Yen	Rupia	Ringgit	US Dolar	C Dolar	Euro	Libra	Peso
Yen japonés	1000	50000	4000	800	1000	6400	4800	7680
Rupia hindia		1000000	160000	20000	20000	128000	96000	153600
Ringgit malasio			100000000	20000000	16000000	120000000	192000000	
US Dolar				1000000000	1250000000	800000000	600000000	9600000000
C Dolar					1000000000	640000000	480000000	7680000000
Euro						1000000000	750000000	12000000000
Libra británica							1000000000	16000000000
Peso mexicano								10000000000

Figura 28: Tabla 3 Límites de transacciones en miles de dólares equivalentes

El problema de Jake se formula como un problema de flujo a costo mínimo. La red se muestra a continuación. Hay tres nodos de suministro — Yen, Rupia y Ringgit. Hay un nodo de demanda — el nodo Dólares Americanos.

**Exercise 89** Complete la información faltante en la red anterior y encuentre la ruta de costo mínimo de los tres orígenes (yen, rupia y ringgit) al destino (USD).

**Solution 90** La gráfica queda como sigue

## 5. PROBLEMA DE EMPAREJAMIENTOS ÓPTIMOS

4

**Definition 91** Dada una gráfica  $G = (V, E)$  n emparejamiento  $M$  es un subconjunto de aristas o lados con la propiedad de que ningún par de lados comparten un nodo.

**Definition 92** El problema del emparejamiento consiste en encontrar el emparejamiento máximo.

La cardinalidad de un emparejamiento máximo denominado completo o perfecto es

$$\frac{|V|}{2} \quad (42)$$

Considérese una gráfica  $G$  junto con un emparejamiento fijo  $M$  de  $G$ ; los lados de  $M$  se denominan emparejados y el resto libres.

Sea  $\{v, w\}$  un lado emparejado entonces  $v$  es la pareja de  $w$ . Los nodos que no son incidentes en ningún lado emparejado se denominan expuestos. Los nodos restantes están emparejados.

<sup>4</sup>Tomado de Papadimitriou, C. and Steiglitz K., p. 218

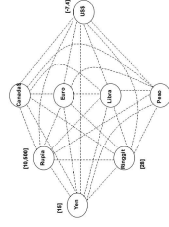


Figura 29: Red problema de divisas

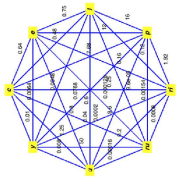


Figura 30: Red ponderada de divisas

**Definition 93** Una trayectoria  $p = [u_1, u_2, \dots, u_k]$  se llama *alternante* si dos lados  $[u_1, u_2]$  y  $[u_3, u_4]$  son libres, mientras que  $[u_2, u_3]$ ,  $[u_4, u_5], \dots, [u_{k-1}, u_k]$  son emparejados.

**Definition 94** Los vértices que caen en una trayectoria alternante comienzan con uno expuesto y tiene rango impar en esta trayectoria se llaman *exterior* y los de rango par se llaman *interiores*.

Se dice que el emparejamiento en este caso es perfecto cuando su cardinalidad es

$$|V|/2 \tag{43}$$

Otro problema es que dados los pesos de los lados, el reto es encontrar un emparejamiento de peso total máximo. En el emparejamiento siguiente se marcan en gris dos vértices exteriores. El resto son vértices interiores

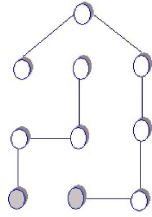


Figura 31: Vértices interiores y exteriores

**Example 95** Encuentre un emparejamiento aumentado para la siguiente gráfica:

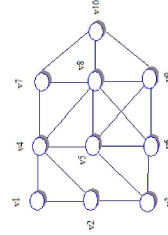


Figura.32: Gráfica

**Solution 96** Una solución quedaría como sigue:

La trayectoria  $p_1 = [v_1, v_2, v_3, v_5, v_4, v_8]$  es alternante y la trayectoria  $p_2 = [v_1, v_4, v_5, v_6, v_8, v_7, v_9, v_{10}]$  es aumentada por que los nodos  $v_1$  y  $v_9$  son expuestos.

**Lemma 97** Sea  $P$  el conjunto de vértices en una trayectoria aumentada  $M$  en la gráfica  $G$  bipartita con respecto al emparejamiento  $M$ . Entonces el emparejamiento  $M' = M \oplus P$  es un emparejamiento con cardinalidad 1

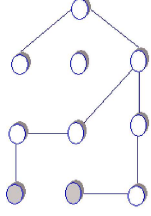


Figura 33: Emparejamiento aumentado.

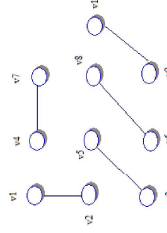


Figura 34: Trayectoria alternante

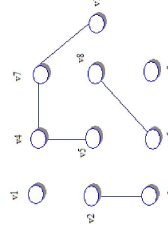


Figura 35: Trayectoria alternante aumentada

**Exercise 98** Encuentre una trayectoria aumentada para el emparejamiento mostrado en la definición anterior.

**Theorem 99** Un aumento  $M$  en una gráfica  $G$  es máxima si y solo si no hay una trayectoria aumentada en  $G$  con respecto a  $M$ .

**Remark 100** Debido a que el problema de emparejamiento para gráficas bipartitas es un caso especial del problema de emparejamiento, se aplica el teorema anterior y se resuelve descubriendo repetidamente una ruta de aumento  $p$  con respecto a la coincidencia actual  $M$ , aumentando el emparejamiento actual a  $M \oplus P$ .

## 6. PROBLEMA DEL CARTERO CHINO

5

El trabajo de un cartero es recoger el correo en la oficina postal, entregarlo a los destinatarios, y regresar a la oficina postal. Se espera que recorra cada calle de su área al menos una vez, de modo que camine lo menos posible.

**Definition 101** En una gráfica ponderada, se define el peso del paseo ponderado

$$\forall e \in \{e_1, \dots, e_n\} \tag{44}$$

como

$$\sum_{i=1}^n w_i(e_i) \tag{45}$$

El problema del cartero chino es encontrar un paseo con mínimo peso en una gráfica conexada ponderada, con pesos no-negativos, óptimo.

Si la gráfica  $G$  es euleriana, es decir que contiene un paseo euleriano, entonces cualquier paseo euleriano de  $G$  es un paseo óptimo. Recordamos que un paseo se denomina euleriano si recorre cada arco o arista exactamente una vez.

El problema en cuestión se resuelve con el algoritmo Fleury, el cual construye un paseo euleriano a partir de un trayecto, sujeto a la única condición de que, en cada etapa, un arco puente de la subgráfica no recorrida se toma si y sólo si no hay alternativa.

<sup>5</sup> Tomado de Bondy and Murty, p. 62.

### 6.1. Algoritmo Fleury

El algoritmo de Fleury se usa para mostrar la ruta de Euler o el circuito de Euler a partir de un gráfico dado. En este algoritmo, comenzando desde un borde, intenta mover otros vértices adyacentes eliminando los bordes anteriores. Usando este truco, el gráfico se vuelve más simple en cada paso para encontrar la ruta o circuito de Euler.

1. Escoger un vértice arbitrario  $v_0$  y establecer el trayecto

$$W_0 = v_0 \tag{46}$$

2. Supóngase que el trayecto

$$W_i = \forall e \in \{e_1, \dots, e_i\} \tag{47}$$

se ha escogido, entonces elegir

$$e_{i+1} \tag{48}$$

de

$$E \setminus \{e_1, \dots, e_i\} \tag{49}$$

de modo tal que

- i)  $e_{i+1}$  es incidente con  $v_i$
- ii) A menos que no haya una alternativa,  $e_{i+1}$  no es una arista de corte (puente) de

$$G_i = G - \{e_1, \dots, e_i\} \tag{50}$$

3. Detenerse cuando el paso 2 no puede implementarse más.

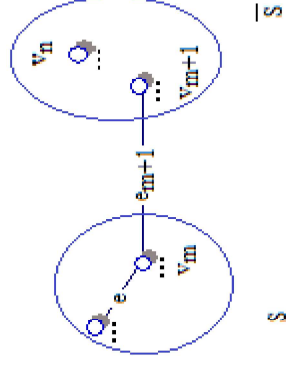


Figura 36: Algoritmo Fleury.

Por su definición el algoritmo de Fleury construye un trayecto en  $G$ .

**Theorem 102.** *Si  $G$  es euleriana, entonces un trayecto en  $G$  construido mediante el algoritmo de Fleury es un paseo euleriano de  $G$ .*

**Example 103.** *Se ilustra un ejemplo implementado con lenguaje C++, para el grafo:*

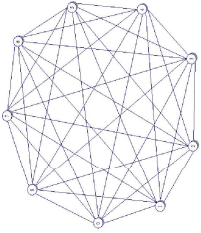


Figura 37: Ejemplo para el algoritmo Fleury

Las figuras 35 y 36 muestran la compilación en C++.



Figura 38: Implementación del Algoritmo Fleury p. 1



Figura 39: Implementación del Algoritmo Fleury p. 2

La ejecución del programa en C++ muestra cuatro circuitos eulerianos los cuales se ven en colores diferentes, véanse lasfigura siguiente:

Otra solución se ve en la figura 40.



Figura 40: Corrida generada en C++

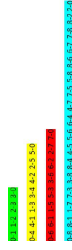


Figura 41: Circuitos Eulerianos

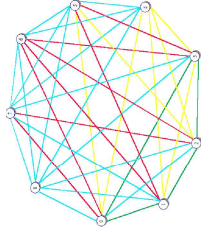


Figura 42: Solución al problema de divisas.

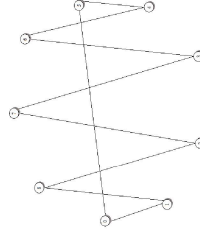


Figura 43: Circuito Euleriano



## 7. PROBLEMA DE LA MOCHILA

El problema de la mochila, comúnmente abreviado por KP (del inglés Knapsack problem) es un problema de optimización combinatoria, es decir, que busca la mejor solución entre un conjunto finito de posibles soluciones a un problema. Modela una situación análoga al llenar una mochila, incapaz de soportar más de un peso determinado, con todo o parte de un conjunto de objetos, cada uno con un peso y valor específicos. Los objetos colocados en la mochila deben maximizar el valor total sin exceder el peso máximo.

### 7.1. Formulación

Maximizar:  $\sum_{i=1}^n v_i x_i$

Sujeta a:

$$\sum_{i=1}^n w_i x_i \leq W$$

$$0 \leq x_i \leq q_i$$

Si  $q_i = 1$  para  $i = 1, 2, \dots, n$  se dice que se trata del problema de la mochila 0-1.

Si uno o más de los  $q_i$  es infinito entonces se dice que se trata del problema de la mochila no acotado también llamado a veces problema de la mochila entera.

En otro caso se dice que se trata del problema de la mochila acotado.

El siguiente ejemplo ilustra el caso de tres compartimientos.

**Example 104** (Problema 3.20. Hillier, et al. 2002) *Un avión carguero tiene tres compartimientos para guardar la carga: al frente, al centro y atrás. Estos compartimientos tienen límites de capacidad, tanto en el peso como en el espacio, como se resume en seguida.*

Compartimiento	Capacidad de Peso (Libras)	Capacidad de Espacio (Pie Cúbico)
Frente	12	7,000
Centro	18	9,000
Atrás	10	5,000

Figura 44: Capacidades por compartimiento

Además, el peso de la carga en los compartimientos respectivos debe tener la misma proporción que la capacidad de peso de ese compartimiento para mantener el equilibrio del avión.

Se han propuesto las siguientes cuatro cargas para su embarque en un próximo vuelo considerando el espacio disponible.

Carga	Peso (Libras)	Volumen (Pie Cúbico)	Utilidad (\$/ton)
1	20	300	250
2	15	200	180
3	10	400	300
4	25	400	375

Figura 45: Características de las cuatro cargas

Puede aceptarse cualquier parte de estas cargas. El objetivo es determinar cuánto (si es posible) de cada carga debe aceptarse y cómo distribuir cada una entre los compartimientos para maximizar la utilidad total del vuelo.

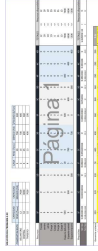
1. Este es un problema mixto de distribución de recursos pero también incluye un número pequeño de restricciones de requerimientos fijos. Construya una tabla de parámetros que incluya todo excepto las restricciones de requerimientos fijos. Luego, construya por separado una tabla de parámetros con las restricciones de requerimientos fijos.

2. Formule y resuelva un modelo de programación lineal en una hoja de cálculo.

3. Exprese el modelo en forma algebraica.

Solución generada en Excel

Figura 46: Solución en hoja de cálculo.



## 8. PROBLEMA DEL AGENTE VIAJERO

Consiste en visitar cierto número de ciudades, una sola vez y regresar al punto de partida. Conocidos los tiempos de viaje entre cada par de ciudades, se quiere hacer un itinerario para que se visite cada ciudad sólo una vez y emplear el menor tiempo posible en regresar al punto de partida. En términos de grafos, se desea encontrar un ciclo hamiltoniano con peso mínimo en una gráfica completa ponderada. Un ciclo de esa naturaleza se le denomina óptimo. A diferencia del problema de la ruta más corta y del problema del conector, no se conoce un algoritmo eficiente para resolver el problema del agente viajero. Por lo que, se desea encontrar una solución razonablemente buena (no necesariamente óptima).

Un enfoque posible es encontrar un primer ciclo hamiltoniano C, después buscar otro de menor peso que pueda sustituir al primero y así sucesivamente. Considere un ciclo hamiltoniano

$$C = v_1 v_2 \dots v_n v_1 \quad (51)$$

Para todo  $i$  y  $j$  tales que

$$1 < i+1 < j < n \quad (52)$$

se puede obtener un nuevo ciclo hamiltoniano

$$C_{ij} = v_1 v_2 \dots v_i v_j v_{j-1} \dots v_{i+1} v_{i+1} v_{j+1} v_{j+2} \dots v_n v_1 \quad (53)$$

quitando las aristas:

$$v_i v_{i+1} \quad (54)$$

$$v_j v_{j+1} \quad (55)$$

y

pero agregando

$$v_i v_j \quad (56)$$

y

$$v_{i+1} v_{j+1} \quad (57)$$

como se muestra en la siguiente figura<sup>6</sup>:

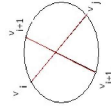


Figura 47: Estrategia para generar un ciclo hamiltoniano

Figura 48

<sup>6</sup> Tomado de Bondy and Murty (1976)

Si, para algunos  $i$  y  $j$

$$w(v_i v_j) + w(v_{i+1} v_{j+1}) < w(v_i v_{j+1}) + w(v_j v_{j+1}) \quad (58)$$

el ciclo  $C_{ij}$  mejora, reduce el peso de  $C$ .

Después de realizar una sucesión de modificaciones como la explicada, se llegará a un ciclo que ya no pueda mejorarse por este método. Este ciclo final, no será óptimo, pero se supone razonablemente que será mucho mejor. Para mayor precisión, el procedimiento anterior puede realizarse varias veces, iniciando en con ciclos diferentes, en cada ocasión.

**Example 105** Considere la siguiente tabla de distancias (en millas), por avión, entre seis ciudades principales, Londres, Mexico City, New York, Paris, Peking y Tokyo.<sup>7</sup>

	L	MC	NY	PA	PE	T
L	—	56	35	2	51	60
MC	56	—	21	57	78	70
NY	35	21	—	36	68	68
PA	2	57	36	—	51	61
PE	51	78	68	51	—	13
T	60	70	68	61	13	—

$$(59)$$

Use el enfoque explicado, iniciando con el ciclo:

$$(60)$$

$L, MC, NY, PA, PE, T$

para encontrar un ciclo hamiltoniano desde  $L$ , de menor peso.

**Solution 106** La gráfica correspondiente a la tabla de distancias se ve en la siguiente figura:

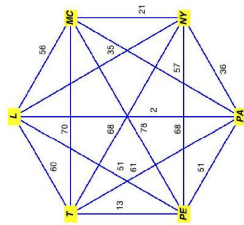


Figura 49: Red de distancias en millas

el paseo hamiltoniano se resalta en la figura 48.

el cual se obtiene de las iteraciones.

<sup>7</sup> Adaptado de Bondy and Murty (1976)

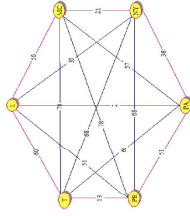


Figura 50: Solución con el paseo Hamiltoniano resaltado

### 8.1. Algoritmo inversión de sub-recorrido

**Exercise 107** Mejorar el camino euleriano  $NY - MC - L - T - PE - PA - NY$  con longitud total de 237 considerando la tabla de distancias siguiente<sup>8</sup>:

Salida	NY	MC	L	T	PE	PA	NY	Total
	21							
		56						
			60					
				13				
					51			
						36		
							237	

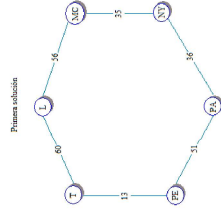


Figura 51: Recorrido inicial

### Solution 108

El recorrido inicial se mejora intercambiando los nodos  $PE$  y  $PA$  para generar el recorrido  $NY - MC - PE - T - L - PA - NY$  con longitud total de 224.

Salida	NY	MC	PE	T	L	PA	NY	Total
	35							
		78						
			13					
				60				
					2			
						36		
							224	

<sup>8</sup> Hillier and Lieberman, 2005, p. 623

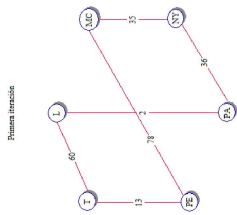


Figura 52: Primera iteración.

La segunda iteración se obtiene intercambiando PA con MC con la que se logra una longitud total de 206: NY - PA - L - T - PE - MC - NY

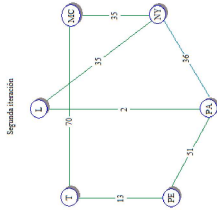


Figura 53: Segunda iteración

Salida	NY	35
L	2	
PA	51	
PE	35	
MC	70	
NY	35	
Total		206

Exercise 109 Aplicar el algoritmo por inversión de sub-recorrido, al paseo que se inicia y termina en L: L - MC - NY - PA - PE - T - L

Solution 110

i	Paseo hamiltoniano	Aristas que se quitan	Aristas que se agregan
0	L, MC, NY, PA, PE, T, L		
1	LPA, NY, MC, PE, T, L	LMC, PAPE	LPA, MCPE
2	L, PA, NY, MC, T, PE, L	TL, MCPE	LPE, MCT
3	L, PA, PE, T, MC, NY, L	LN, PAPE	NYL, PAPE

(61)

8.2. Búsqueda Tabú

Resumen adaptado de Hillier and Lieberman (2005) Inicio: con una solución inicial de ensayo.

**Iteración:** se usa un procedimiento de búsqueda local apropiado para definir los movimientos factibles hasta el vecino local de la solución de ensayo actual. Se elimina de la consideración cualquier movimiento de la actual lista tabú a menos que el movimiento produzca una mejor solución de ensayo encontrada hasta aquí.

Después se determina cuáles de los movimientos restantes proporcionan la mejor solución. Adoptar esta solución como la siguiente solución de ensayo sin importar si es mejor o peor que la actual solución de ensayo. Si la lista tabú se llenó, borrar el miembro más antiguo de la lista tabú para proveer mayor flexibilidad a los futuros movimientos.

**Regla de paro:** Use algún criterio de parada tal como un número fijo de iteraciones, o una cantidad fija de tiempo de CPU, o un número fijo de iteraciones consecutivas sin que se mejore el mejor valor de la función objetivo. También detener en cualquier iteración donde no haya movimientos factibles en la vecindad local de la actual solución de ensayo. Aceptar la mejor solución de ensayo en cualquier iteración como la solución final. El algoritmo de búsqueda tabú, genera preguntas para las cuales los autores resumen algunas respuestas.

Preguntas / Respuestas

1. ¿Qué procedimiento de búsqueda debería usarse? / En cada iteración escoger el mejor vecino inmediato de la actual solución de ensayo que se ha sacado de su estatus tabú.
2. ¿Cómo el procedimiento define la estructura del vecino que especifica que vecinos son inmediatos, es decir, elegibles en una sola iteración de cualquier solución de ensayo actual. / Estructura de la vecindad. Un vecino inmediato de la solución de ensayo actual es una que se alcanzó agregando solo un enlace y borrando uno de los otros ensayos en el ciclo que se forma por esta adición. El enlace borrado debe provenir de un ciclo a fin de que se conserve un árbol de expansión (ST).
3. ¿Cuál es la forma en las que los movimientos tabú deberían representarse en la lista tabú. / Forma de los movimientos tabú. Enliste los enlaces que no se borran.
4. ¿Cuál movimiento tabú debería agregarse a la lista tabú / Adición de un movimiento tabú en cada iteración después de escoger el enlace por agregarse a la red y también agregue este enlace en la lista tabú.
5. ¿Qué tanto un movimiento tabú debe conservarse en la lista tabú / Tamaño máximo de la lista tabú. dos. Siempre que un movimiento tabú sea agregado, borre el más antiguo.
6. ¿Qué regla de paro debería usarse? / Regla de paro: parar después de 3 iteraciones consecutivas en las que no hay un mejoramiento del mejor valor hallado. También parar en cualquier iteración donde la actual solución de ensayo no tiene vecinos inmediatos, que no fueron sacados por su estatus tabú.

Example 111. Se ilustra con el grafo de la figura siguiente<sup>9</sup>:

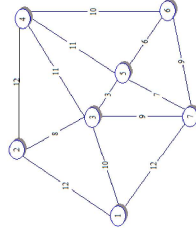


Figura 54: Ejemplo tomado de Hillier and Lieberman (2005)

Una solución de ensayo inicial es 1 - 2 - 3 - 4 - 5 - 6 - 7 - 1 con distancia 69.

Iteración 1

1. Seleccionar subconjunto para revertir: 3 - 4
2. Enlaces borrados 2 - 3 y 4 - 5 Enlaces agregados 2 - 4 y 3 - 5
3. Lista Tabú 2 - 4 y 3 - 5 Nueva solución de ensayo 1 - 2 - 4 - 3 - 5 - 6 - 7 - 1 con distancia 65

<sup>9</sup>Hillier and Lieberman, 2005, p. 622

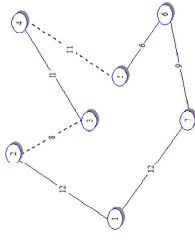


Figura 55: Enlaces que se quitan.

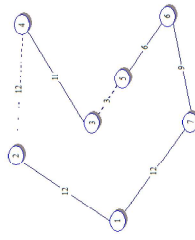


Figura 56: Enlaces que se agregan.

### Iteración 2

1. Selección subhour para revertir: 3 - 5 - 6 a 6 - 5 - 3
2. Enlaces borrados: 3 - 4 y 6 - 7 Enlaces agregados: 2 - 4 y 3 - 5
3. Lista Tabú: 2 - 4, 3 - 5, 4 - 6 y 3 - 7 Nueva solución de ensayo: 1 - 2 - 4 - 6 - 5 - 3 - 7 - 1 con distancia 64

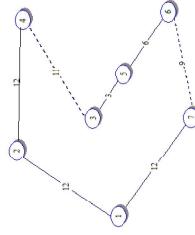


Figura 57: Enlaces que se borran.

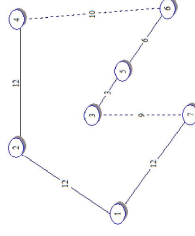


Figura 58: Enlaces que se agregan.

## 9. REFERENCIAS Y BIBLIOGRAFÍA

- Balderas, P. y Sánchez, G. (coords., 2011) Ingeniería de Sistemas. Investigación e Intervención. México: Plaza y Valdés-Facultad de Ingeniería, UNAM, 67 – 92
- Bazaraa, M., Jarvis, J., and Sherali, H. (1998) Programación lineal y flujo de redes. México LIMUSA, 2a.
- Bondy, J. A. and Murty, U.S.R. (1982) Graph Theory with Applications. N.Y. North-Holland, 5a.
- Chartrand, G. (1977) Introductory Graph Theory. New York: Dover Publications, Inc. pp. 294
- Cook, W., Cunningham, W., Pulleyblank, W., and Schrijver, A. (1998) Combinatorial optimization. New York : J. Wiley.
- Espinosa Armenta, R. (2010) Matemáticas Discretas. México: Alfaomega, pp. 467
- Harary, F. (1969) Graph Theory. Massachussets: Addison-Wesley, pp. 274
- Harary, F. Norman, F. y Cartwright, D. (1965, 1978) Structural Models: an Introduction to the Theory of Directed Graph. New York: John Wiley & Sons, pp. 415
- Hillier, F. and Lieberman, G. (2005) Introduction to Operations Research, 8th edition. New York: McGraw-Hill.
- Ortega, James M. (1987) Matrix Theory. A second course. The University Series in Mathematics. New York: Plenum Press, pp. 262
- Murthy, D., Page, N., & Rodin, E. (1990) Mathematical Modelling. A tool for Problem Solving Engineering, Physical, Biological and Social Sciences. N.Y.: Pergamon Press.
- Slomson, Alan (1991) An Introduction to Combinatorics. London: Capman & Hall/CRC

### 9.1. BIBLIOGRAFÍA ADICIONAL

- Pierluigi Crescenzi, Viggo Kann, Magnús Halldórsson, Marek Karpinski, Gerhard Woeginger, A Compendium of NP Optimization Problems.
- Christos H. Papadimitriou, and Kenneth Steiglitz: Combinatorial Optimization: Algorithms and Complexity; Dover Pubns; (paperback, Unabridged edition, July 1998) ISBN 0486402584.