Cátedra Nissan

-PROTHIUS-

Diseño de sistemas productivos y logísticos. Distribución física.

Joaquín Bautista Valhondo

D-06/2010
(Rec. OP-BCC)

Departamento de Organización de Empresas

Universidad Politécnica de Cataluña

Publica:

Universitat Politècnica de Catalunya
www.upc.edu

Edita:

Cátedra Nissan
www.nissanchair.com
director@nissanchair.com

Distribución física

NISSAN
CATEDRA NISSAN

DOE
Departament d'Organització d'Empreses

Contenido

- *Cubrimiento*
  - Preliminares
  - Mínimo número de instalaciones
  - Máxima cobertura
- *Abastecimiento*
- *Itinerarios*
  - Preliminares
  - Clasificación de problemas
  - Procedimientos de resolución
  - Problema del viajante de comercio
  - Problema del diseño de rutas

Cubrimiento. Preliminares

Condiciones:

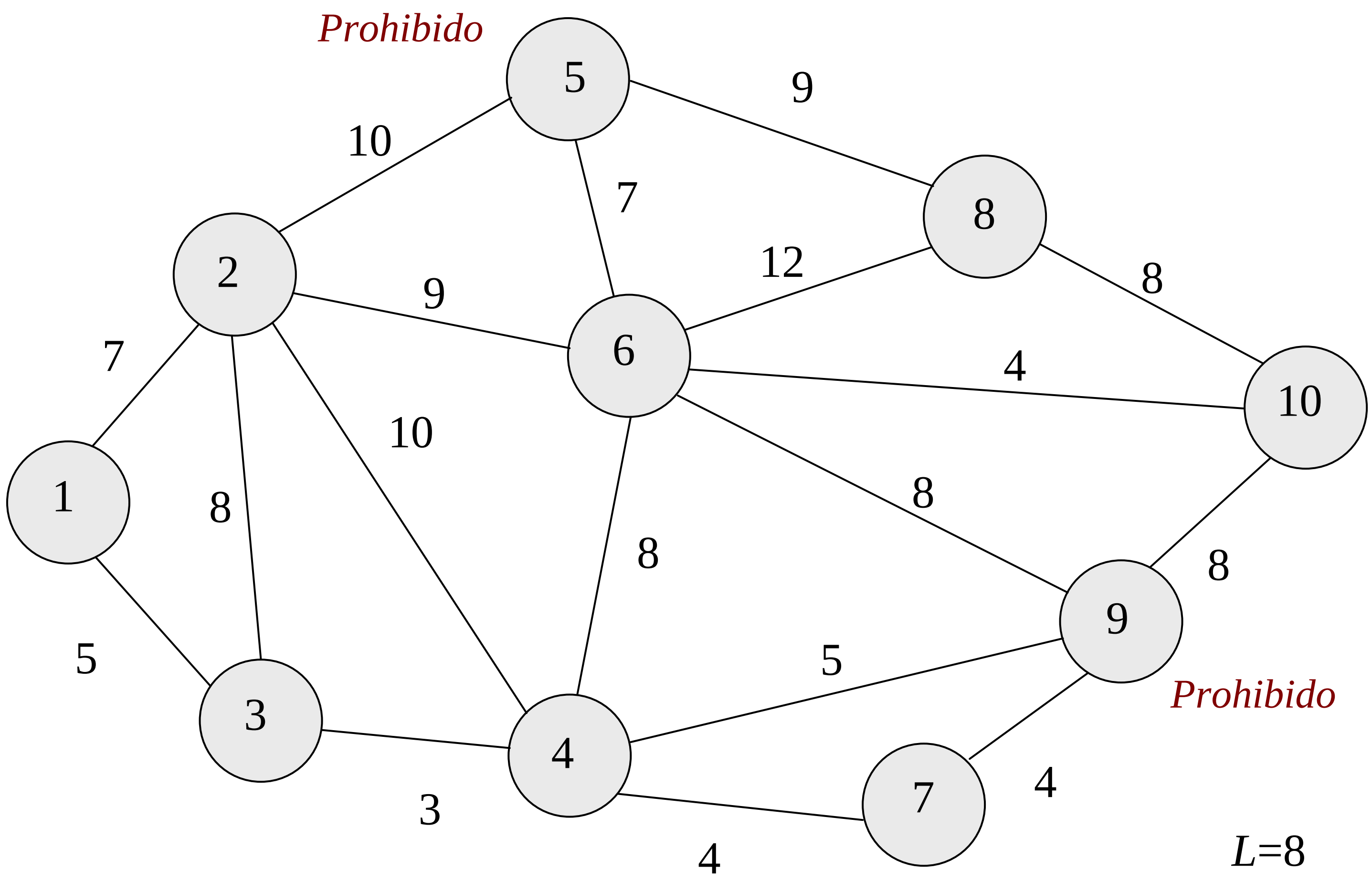
- Se dispone de un conjunto de emplazamientos a cubrir
- Sea L la distancia máxima permitida para cubrir una instalación
- Sea d_i el peso (demanda) asociado al emplazamiento i.
- Se dispone de un grafo G de estructura de comunicaciones
- Se dispone de un conjunto de emplazamientos que no admiten una instalación

Objetivos:

- *Minimizar el número de instalaciones* de forma que todos los emplazamientos queden cubiertos (todo emplazamiento está a una distancia menor o igual a L de la instalación más próxima)

- *Maximizar la suma de pesos* (Cobertura) de los emplazamientos cubiertos con un número de instalaciones prefijado.

Cubrimiento. Ejemplo prototipo



$d_1 = 2$
$d_2 = 6$
$d_3 = 3$
$d_4 = 8$
$d_5 = 5$
$d_6 = 4$
$d_7 = 7$
$d_8 = 10$
$d_9 = 9$
$d_{10} = 6$

L=8

Cubrimiento. Mínimo número de instalaciones (1/2)

- *Nomenclatura básica:*

J: conjunto de emplazamientos ($j = 1,..,|J|$)

$I(\subseteq J)$: conjunto de instalaciones potenciales ($i = 1,..,|I|$)

L: distancia máxima de cobertura entre una instalación y un emplazamiento

l_{ij}: distancia mínima entre la instalación $i \in I$ y el emplazamiento $j \in J$

$I_j(L) = \{i \in I : l_{ij} \leq L\}$: conjunto de instalaciones que cubren el emplazamiento $j \in J$

x_i: variable binaria que vale 1 si en $i \in I$ se fija una instalación y vale 0 en caso contrario.

- *Modelo:*

$$Min\ z_1 = \sum_{i \in I} x_i \qquad (0)$$

$$s.a.:$$

$$\sum_{i \in I_j(L)} x_i \geq 1 \quad \forall j \in J \qquad (1)$$

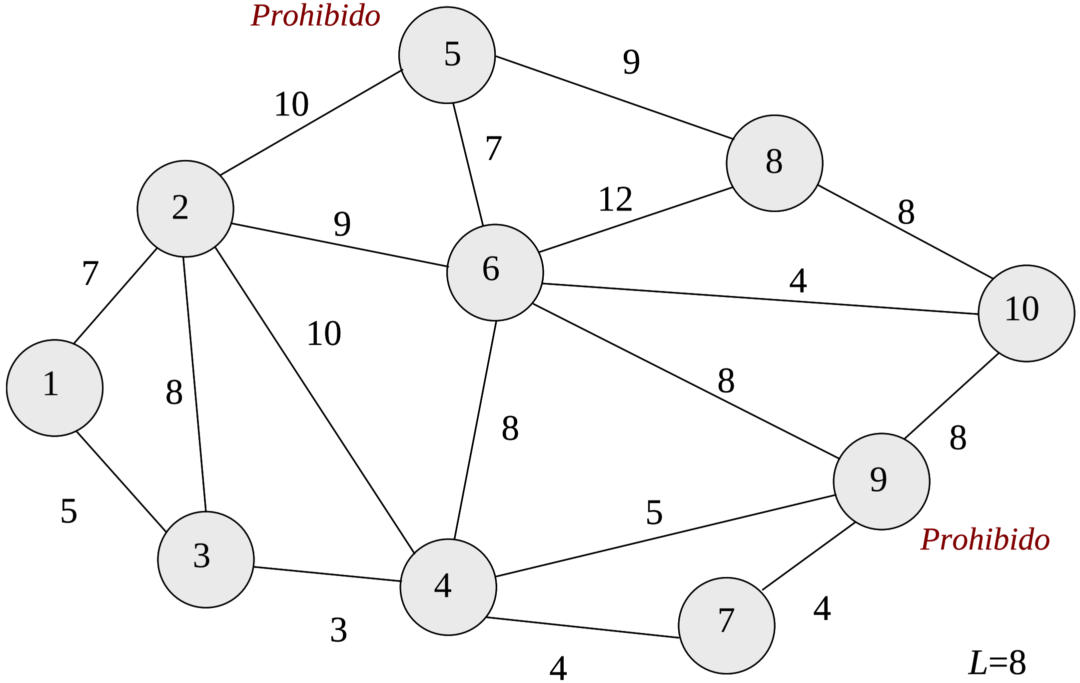
$$x_i \in \{0,1\} \quad \forall i \in I \qquad (2)$$

Cubrimiento. Mínimo número de instalaciones (2/2)

$$Min\ \ z_1 = \sum_{i=1}^{10} x_i \qquad (0)$$

$s.a.:$

$$x_1 + x_2 + x_3 + x_4 \geq 1 \qquad (1)$$
$$x_1 + x_2 + x_3 \geq 1 \qquad (2)$$
$$x_1 + x_2 + x_3 + x_4 + x_7 \geq 1 \qquad (3)$$
$$x_1 + x_3 + x_4 + x_6 + x_7 \geq 1 \qquad (4)$$
$$x_6 \geq 1 \qquad (5)$$
$$x_4 + x_6 + x_{10} \geq 1 \qquad (6)$$
$$x_3 + x_4 + x_7 \geq 1 \qquad (7)$$
$$x_8 + x_{10} \geq 1 \qquad (8)$$
$$x_3 + x_4 + x_6 + x_7 + x_{10} \geq 1 \qquad (9)$$
$$x_6 + x_8 + x_{10} \geq 1 \qquad (10)$$
$$x_i \in \{0,1\} \qquad \forall i = 1,..,10 \qquad (11)$$



$$x_1 + x_2 + x_3 \geq 1 \qquad (2)$$
$$x_6 = 1 \qquad (5')$$
$$x_3 + x_4 + x_7 \geq 1 \qquad (7)$$
$$x_8 + x_{10} \geq 1 \qquad (8)$$
$$x_i \in \{0,1\} \qquad \forall i = 1,..,10 \qquad (11)$$

$$x_3 = x_6 = x_8 = 1$$
$$x_3 = x_6 = x_{10} = 1$$

Cubrimiento. Máxima cobertura o satisfacción de la demanda (1/2)

- *Nomenclatura adicional:*

n : número máximo de instalaciones permitido ($n \le |I|$)

d_j : demanda o peso del emplazamiento $j \in J$

x_i : variable binaria que vale 1 si en $i \in I$ se fija una instalación y vale 0 en caso contrario

y_j : variable binaria que vale 1 si se cubre el emplazamiento $j \in J$ y vale 0 en caso contrario

- *Modelo:*

$$Max\ z_2 = \sum_{j \in J} d_j y_j \qquad (0)$$

$s.a.:$

$$y_j \le \sum_{i \in I_j(L)} x_i \qquad \forall j \in J \qquad (1)$$

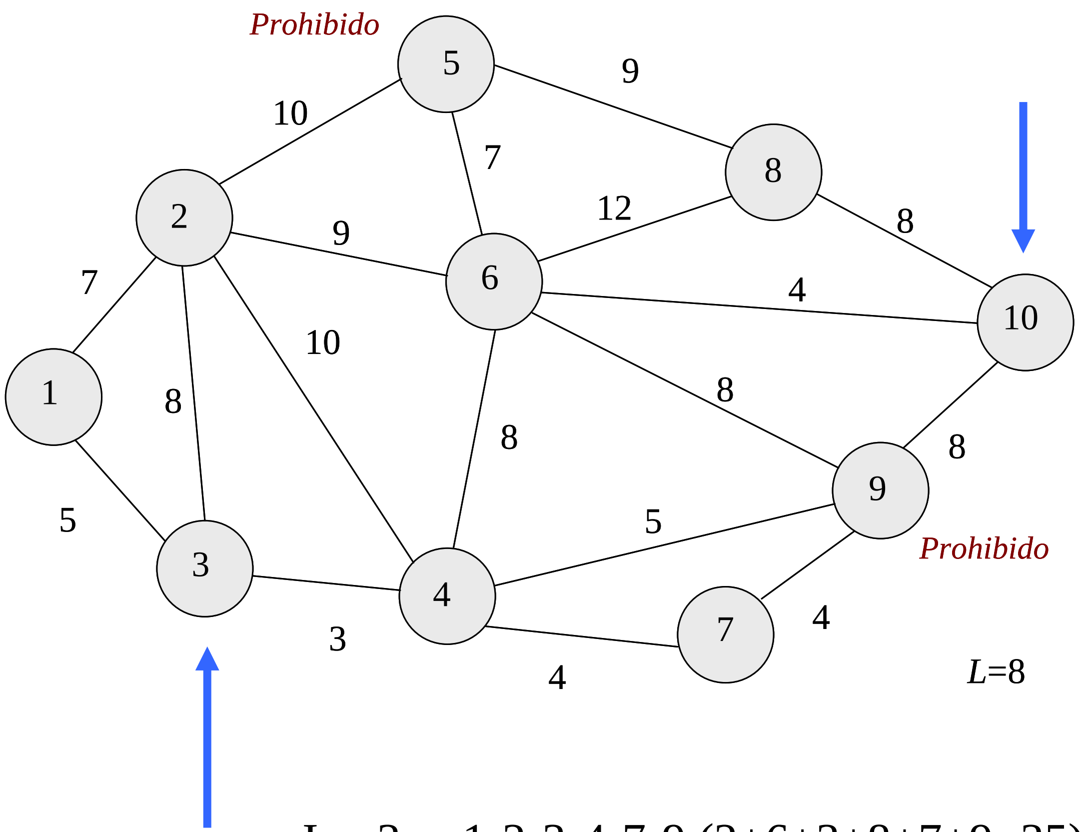
$$\sum_{i \in I} x_i \le n \qquad (2)$$

$$x_i \in \{0,1\} \qquad \forall i \in I \qquad (3)$$

$$y_j \in \{0,1\} \qquad \forall j \in J \qquad (4)$$

Cubrimiento. Máxima cobertura o satisfacción de la demanda (2/2)

$d_1 = 2$
$d_2 = 6$
$d_3 = 3$
$d_4 = 8$
$d_5 = 5$
$d_6 = 4$
$d_7 = 7$
$d_8 = 10$
$d_9 = 9$
$d_{10} = 6$



Ins.	*C.it.1*	*C.it.2*
1	19	---
2	11	---
3	**35**	---
4	33	4
5	*Prohibido*	
6	32	15
7	27	---
8	16	16
9	*Prohibido*	
10	29	**20**

Ins-3 : 1-2-3-4-7-9 (2+6+3+8+7+9=35)

Ins-10 : 6-8-10 (4+10+6=20)

Abastecimiento desde instalaciones hasta emplazamientos

- *Nomenclatura adicional:*

o_i : oferta o capacidad de la instalación $i \in I$

c_{ij} : coste de transportar una unidad desde $i \in I$ hasta $j \in J$; $\left(i \notin I_j(L)\right) \Rightarrow (c_{ij} = \infty)$

$J_i(L) = \left\{ j \in J : l_{ij} \leq L \right\}$: conjunto de emplazamientos cubiertos por la instalación $i \in I$

φ_{ij} : unidades transportadas desde $i \in I$ hasta $j \in J$

- *Modelo:*

$$Min \ z_3 = \sum_{i \in I} \sum_{j \in J} c_{ij} \varphi_{ij} \qquad (0)$$

$s.a.:$

$$\sum_{j \in J_i(L)} \varphi_{ij} \leq o_i \qquad \forall i \in I \qquad (1)$$

$$\sum_{i \in I_j(L)} \varphi_{ij} = d_j \qquad \forall j \in J \qquad (2)$$

$$\varphi_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J \qquad (3)$$

Cubrimiento y abastecimiento combinados

- *Nomenclatura adicional:*

C_i : coste de activar la instalación $i \in I$

x_i : variable binaria que vale 1 si se activa la instalación $i \in I$ y vale 0 en caso contrario

- *Modelo:*

$$Min\ z_4 = \sum_{i \in I} C_i x_i + \sum_{i \in I} \sum_{j \in J} c_{ij} \varphi_{ij} \qquad (0)$$

$s.a.:$

$$\sum_{j \in J_i(L)} \varphi_{ij} \le o_i x_i \qquad \forall i \in I \qquad (1)$$

$$\sum_{i \in I_j(L)} \varphi_{ij} = d_j \qquad \forall j \in J \qquad (2)$$

$$\varphi_{ij} \ge 0 \qquad \forall i \in I, \forall j \in J \qquad (3)$$

$$x_i \in \{0,1\} \qquad \forall i \in I \qquad (4)$$

Itinerarios. Preliminares

- *Concepto:* En el marco de la *distribución física*, el *diseño de itinerarios* consiste en establecer el camino que deberá seguir uno o más medios de transporte (reparto o recogida) sujetos a unas condiciones

- *Condiciones (ej.):*
  - Paso obligatorio por una serie de puntos de servicio.
  - Paso obligatorio por una serie de tramos.
  - Oferta del servicio en cierto intervalos horarios.
  - Limitación del tiempo de servicio.
  - Limitación de la capacidad de carga de las unidades de transporte.

- *Actividades productivas relacionadas (ej.):*
  - *Distribución de bienes* : alimentos, cosméticos, dinero, correo, paquetería, etc.
  - *Recogida en servicios* : residuos, limpieza de aceras, retirada de nieve, etc.
  - *Inspección o acción en servicios* : poda de árboles en entornos urbanos, registro de contadores, estado de luminarias, estado de una red de carreteras y su señalización, visitas médicas, etc.

Problemas clásicos (1/2)

- *Problema del viajante de comercio (TSP: Travelling Salesman o Salesperson Problem):*

  Dados n+1 puntos (el depósito más n clientes) se trata de determinar un circuito (o ciclo, si hay simetría) en que el vehículo se detenga una vez y una sola en cada punto y que tenga coste mínimo: hallar un circuito (ciclo) hamiltoniano de coste mínimo.

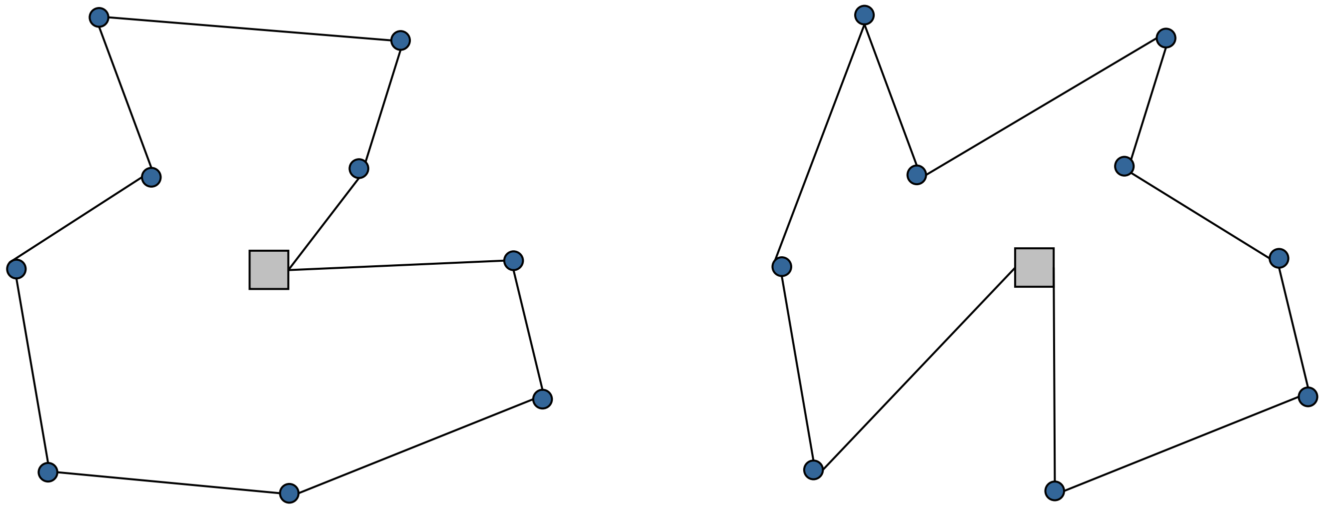


Figura: Representación esquemática de un ciclo (simetría) o circuito (asimetría) para 9 puntos y un almacén.

Problemas clásicos (2/2)

- *Problema de diseño de rutas (VRP: Vehicle Routing Problem):*

  Dados n+1 puntos (el depósito más n clientes) y unas limitaciones (tiempo, longitud o capacidad) se trata de determinar un sistema de recorridos en que se ofrezca servicio a todos los puntos con coste mínimo.

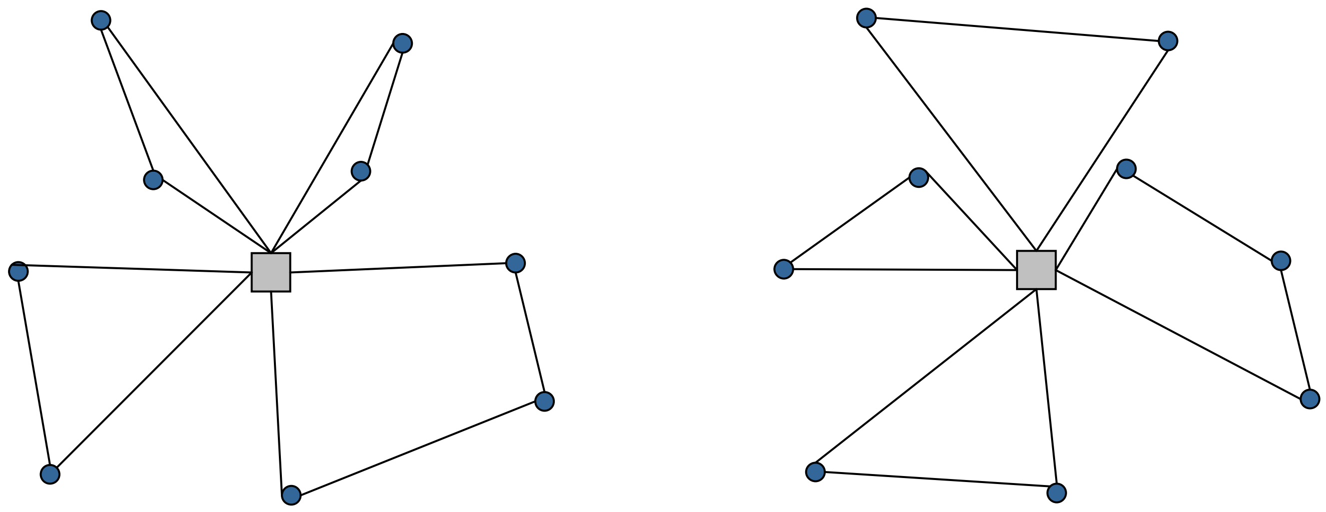


Figura: Representación esquemática de los recorridos
desde los vehículos con salida y llegada en el almacén (pétalos)

Clasificación de problemas. Aspectos generales (Rousseau)

- *Aspectos*:
  - Estructura del sistema de servicio
  - Demanda
  - Programación de actividades
  - Flota de vehículos
  - Tripulaciones
  - Datos requeridos o disponibles
  - Tráfico y seguridad

Clasificación de problemas. Demanda (1/2)

- *Naturaleza de la demanda*:
  - Reparto puro o recogida pura
  - Primero reparto y después recogida en un mismo pétalo
  - Reparto y recogida mezclados en un mismo pétalo
  - Un solo tipo de producto o elemento o varios
  - Restricción de servir toda la demanda
  - Admisibilidad de servicio partido (se satisface la demanda de un cliente por medio de más de un vehículo o pétalo)
  - Prioridades de los clientes o de los pedidos

Clasificación de problemas. Demanda (2/2)

- *Localización de la demanda*:
  - Localización en los vértices
  - Localización en los arcos
  - Localización Mixta

- *Información disponible sobre la demanda*:
  - Se conoce toda la demanda antes de que el vehículo salga del almacén
  - Demandas iguales repetidas a lo largo del tiempo
  - Frecuencia prefijada de las visitas a los clientes
  - Demandas no completamente conocidas antes de salir el vehículo
  - Demandas en tiempo real

Clasificación de problemas. Programación de actividades

- *Condiciones a respetar en la programación de actividades*:
  - Sólo algunos días de la semana (o del mes) admisibles para ciertos clientes
  - Ventanas temporales (intervalos temporales) duras o blandas a respetar para la entrega o recogida en ciertos clientes (imposición del cliente o por normativa: ordenanzas municipales)
  - Tiempos de carga o descarga
  - Ventanas temporales de la tripulación
  - Ventanas temporales en ciertos tramos para ciertos productos (por ejemplo, productos inflamables o tóxicos)

Clasificación de problemas. Flota de vehículos

- *Casos habituales*:
  - Homogénea o varios tipos de vehículos
  - Limitaciones de peso y volumen
  - Compartimentos
  - Restricciones relativas a los equipos de carga y descarga
  - Pares vehículo-emplazamiento no permitidos
  - Incompatibilidades vehículos-tipo de carga
  - Costes
  - Tamaño de la flota fijo o variable
  - Posibilidad de subcontratación
  - Uno o más emplazamientos de los vehículos de la flota (con asignación fija o variable)
  - Uno o más de un recorrido por vehículo en un intervalo de tiempo dado (jornada, etc.).

Clasificación de problemas. Tripulaciones

- *Condiciones sobre la tripulación*:
  - Tipo de jornada de trabajo: duración fija o entre un máximo y un mínimo, opción de horas extras
  - Hora de comienzo y localización en ese momento
  - Estructura de la retribución: por tiempo, por distancia recorrida, etc.
  - Pausas para comidas o descansos
  - Número fijo o variable de conductores
  - Posibilidad de que un recorrido comprenda más de un día y reglas asociadas
  - Asignación a zona

Clasificación de problemas. Datos requeridos o disponibles

- *Casos en cuanto a necesidad y disponibilidad*:
  - Base de datos de información geográfica, red viaria
  - Direcciones y localización de los clientes
  - Tiempos de desplazamiento
  - Tiempos de carga y descarga
  - Información sobre la localización de los vehículos
  - Información sobre las condiciones de facturación a los clientes

Clasificación de problemas. Tráfico y seguridad

- *Restricciones derivadas de la regulación*:
  - Direcciones únicas
  - Giros prohibidos (a la izquierda, en U, etc.)
  - Obligación de servir a ciertos clientes desde un lado especificado de la calle
  - Reglas para servir a los clientes situados en las esquinas
  - Reglas especiales para ciertos tramos de calle (por ejemplo, prohibición de circular con mercancías de cierto tipo en ciertas calles a ciertas horas o en cualquier momento)

Itinerarios. Procedimientos de resolución

Procedimientos exactos:

- Programación lineal entera
- Programación dinámica
- Procedimientos de separación y acotación (*branch and bound*)
- Propagación de restricciones

Procedimientos heurísticos. Esquema general:

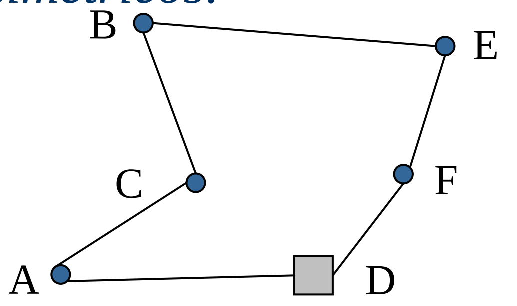
- *Fase-I. Iniciación* : necesaria en algunas ocasiones para fijar los valores de algunos parámetros o fijar clientes “semilla” en la construcción de pétalos.
- *Fase-II. Determinación de una solución*: obligatoria en todos los casos.
- *Fase-III. Mejora* : procedimientos de búsqueda local de intercambio e inserción.

Problema del viajante de comercio. Un ejemplo

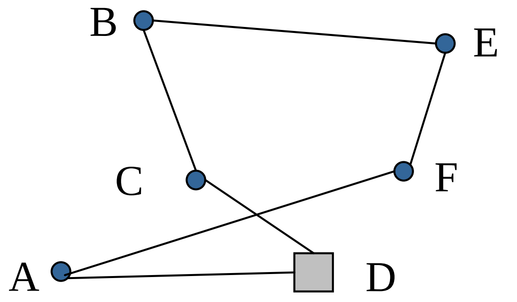
- *TSP costes (tiempos, distancias, fiabilidad) simétricos:*

	A	B	C	D	E	F
A	-	100	150	110	70	90
B		-	90	130	60	80
C			-	100	90	80
D				-	80	70
E					-	20
F						-

Tabla: costes simétricos entre emplazamientos. Depósito en “D”.



Solución-1: heurística *vecino más próximo*.
Secuencia: D-F-E-B-C-A. Valor 500.



Solución-2: Mejora por heurística 2-*opt*.
Secuencia: D-C-B-E-F-A. Valor 470.

Problema del diseño de rutas. Fase para determinar una solución

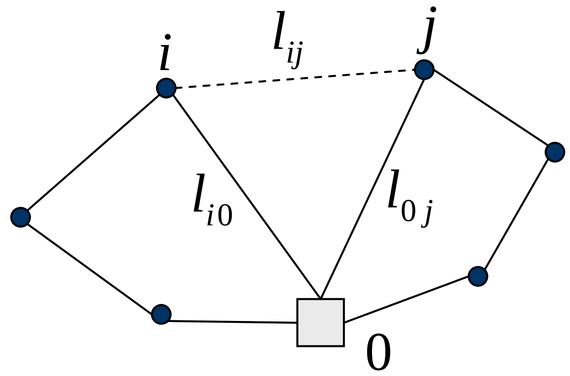
- *Una clasificación de algoritmos:*

$$\text{Algoritmos}: \left\{ \begin{array}{l} \text{1.Constructivos}: \left\{ \begin{array}{l} \text{1.1. Inserción} \\ \text{1.2. Fusión de pétalos}: \left\{ \begin{array}{l} \text{1.2.1. Paralelo} \\ \text{1.2.2. Secuencial} \end{array} \right\} \end{array} \right\} \\ \text{2. De 2 Fases}: \left\{ \begin{array}{l} \text{Fase - 1: Partición del conjunto de clientes} \\ \text{Fase - 2 : Viajante de comercio} \end{array} \right\} \end{array} \right\}$$

- *Fusión de pétalos. Concepto de ahorro:*

$\text{Dados}: \; l_{i0}, l_{0j}, l_{ij}$

$\text{Se define}: a_{ij} = l_{i0} + l_{0j} - l_{ij}$



Problema del diseño de rutas. Ejemplo prototipo (1/3)

- *VRP. Coordenadas:*

Vértice	*Coordenada* x	*Coordenada* y	*Pedido* (u)
1	20	10	3
2	30	40	3
3	-40	40	2
4	-20	15	3
5	-50	20	6
6	-15	5	4
7	-50	-5	1
8	-5	-45	4
9	4	-20	4
10	15	-40	4
11	60	-5	3
12	25	5	5
13	65	20	2

Tabla: Datos para el ejemplo prototipo de diseño de rutas. Coordenadas en Km. Coordenadas del depósito (código 0): (0,0). Capacidad de los vehículos: 10 unidades de producto

Problema del diseño de rutas. Ejemplo prototipo (2/3)

- *VRP distancias simétricas:*

	1	2	3	4	5	6	7	8	9	10	11	12	13
0	22	50	57	25	54	16	50	45	20	43	60	25	68
1	-	32	67	40	71	35	72	60	34	50	43	7	46
2	-	-	70	56	82	57	92	92	65	81	54	35	40
3	-	-	-	32	22	43	46	92	74	97	110	74	107
4	-	-	-	-	30	11	36	62	42	65	82	46	85
5	-	-	-	-	-	38	25	79	67	88	113	76	115
6	-	-	-	-	-	-	36	51	31	54	76	40	81
7	-	-	-	-	-	-	-	60	56	74	110	76	118
8	-	-	-	-	-	-	-	-	27	21	76	58	96
9	-	-	-	-	-	-	-	-	-	23	58	33	73
10	-	-	-	-	-	-	-	-	-	-	57	46	78
11	-	-	-	-	-	-	-	-	-	-	-	36	25
12	-	-	-	-	-	-	-	-	-	-	-	-	43

Tabla: Distancias euclídeas (sin decimales) en Km para el ejemplo prototipo de diseño de rutas.

Problema del diseño de rutas. Ejemplo prototipo (3/3)

- *VRP ahorros simétricos:*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1	-	41	12	7	6	3	1	7	9	15	40	41	44
2	-	-	37	19	21	9	8	3	5	11	56	40	78
3	-	-	-	50	88	29	61	10	3	2	7	8	18
4	-	-	-	-	48	30	39	8	3	3	3	4	8
5	-	-	-	-	-	32	79	20	7	8	1	3	7
6	-	-	-	-	-	-	30	10	5	4	0	1	2
7	-	-	-	-	-	-	-	35	15	19	0	0	1
8	-	-	-	-	-	-	-	-	39	67	29	12	18
9	-	-	-	-	-	-	-	-	-	40	23	13	15
10	-	-	-	-	-	-	-	-	-	-	46	22	33
11	-	-	-	-	-	-	-	-	-	-	-	49	103
12	-	-	-	-	-	-	-	-	-	-	-	-	51

Tabla: Ahorros (sin decimales) en Km para el ejemplo prototipo de diseño de rutas.

Ideas básicas para Algoritmos de Inserción:

- Se busca el cliente más alejado del depósito entre aquellos clientes cuya demanda no ha sido satisfecha. Sea *C*, dicha cliente.
- El cliente *C* será la semilla de una de las rutas.
- Se determina el camino mínimo entre *C* y el depósito 0 y se van insertando nuevos clientes en la ruta, siempre que haya capacidad suficiente, de forma que el recorrido de la ruta aumente lo menos posible.
- Para cada cliente candidato (demanda no superior a la capacidad remanente) se ha de evaluar el incremento del recorrido al insertarlo en la ruta en cada una de las posiciones posibles) hasta que se satura la capacidad del vehículo o no queda carga por asignar.

Problema del diseño de rutas. Algoritmos de fusión de pétalos (1/6)

Algoritmo de Clarke y Wright. Versión paralelo (simétrica-euclídea):

Paso -1: Calcular para todo par clientes (i, j) el ahorro:

$$a_{ij} = l_{0i} + l_{j0} - l_{ij} \quad \forall i \forall j;\ a_{i0} = a_{0i} = 0 \,\forall i$$

Paso - 2: Ordenar los pares de clientes de mayor a menor valor de a_{ij}.

Paso - 3: Mientras hayan clientes por asignar, *Hacer*:

- Seleccionar el par (i, j) no asignado (total o parcialmente) con demanda menor o igual a la capacidad máxima de una ruta y con mayor ahorro.

Casos:

1 - Si ni i ni j están asignados $\Rightarrow$ abrir nueva ruta con *extremos* i y j.

2 - Si i está asignado como extremo de una ruta abierta R con capacidad remanente mayor o igual que la demanda de $j \Rightarrow$ asignar j a R.

3 - Si j está asignado como extremo de una ruta abierta R con capacidad remanente mayor o igual que la demanda de $i \Rightarrow$ asignar i a R.

4 - Si i es extremo de una ruta abierta R y j es extremo de una ruta abierta R', y la suma de cargas de R y R' no supera la capacidad máxima de una ruta, entonces, *fundir* R y R'.

Problema del diseño de rutas. Algoritmos de fusión de pétalos (2/6)

Algoritmo de Clarke y Wright – paralelo-. Ejemplo:

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
d_i	3	3	2	3	6	4	1	4	4	4	3	5	2

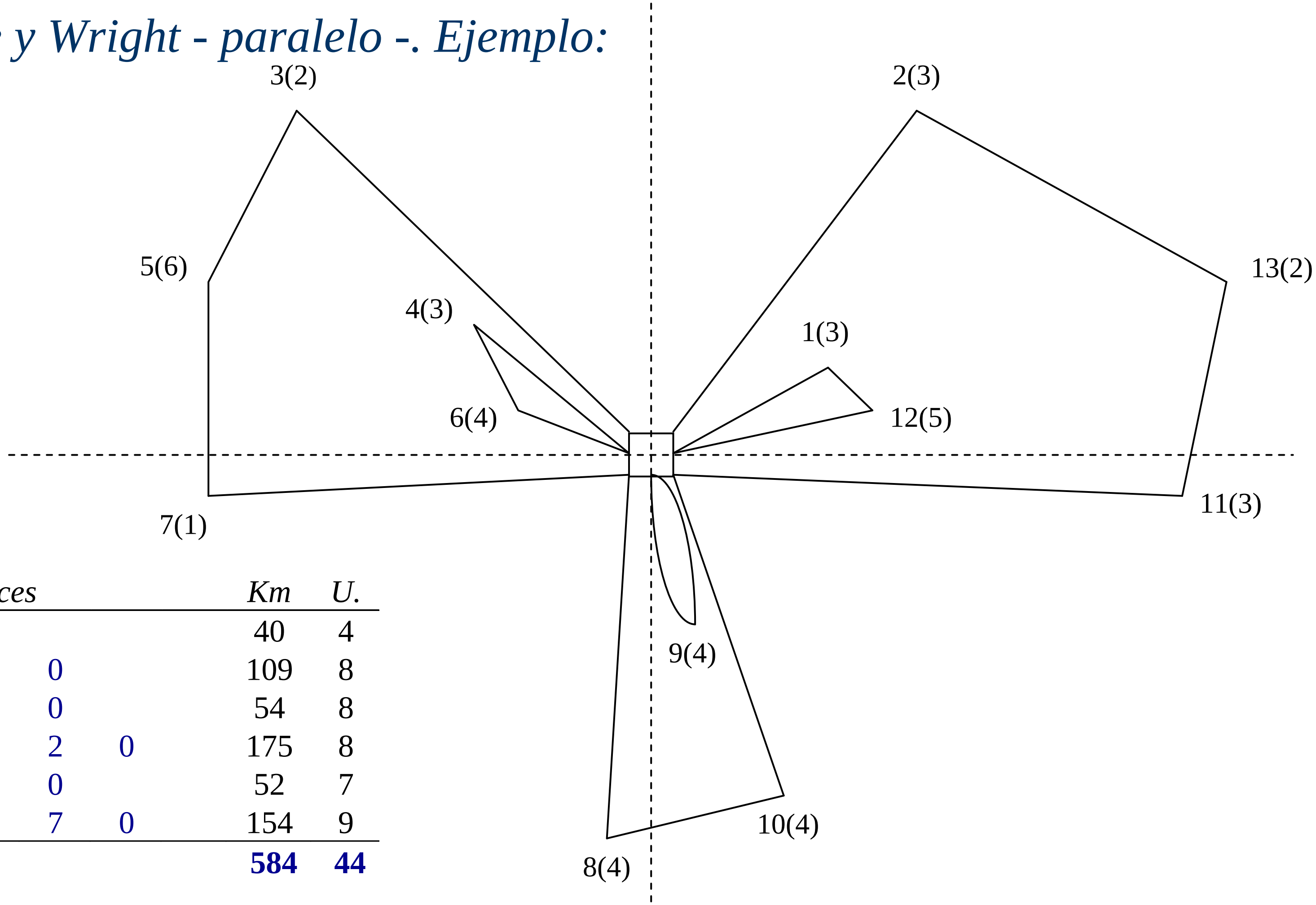
Tabla: Demandas de los clientes para el ejemplo prototipo. La capacidad máxima de una ruta es 10 *u*.

Etapa	Par	Ahorro	Fusión	Carga
1	11-13	103	#4	5
2	3-5	88	#3	8
3	5-7	79	3-5-7	9
4	2-13	78	11-13-2	8
5	8-10	67		8
6	1-12	41		8
7	4-6	30		7
8	0-9	0		4

Tabla: algoritmo de Clarke y Wright (paralelo). Solución: 6 pétalos. Los pares que dan lugar a la fusión de pétalos son: 11-13, 3-5, 5-7, 2-13, 8-10, 1-12 y 4-6.

Problema del diseño de rutas. Algoritmos de fusión de pétalos (3/6)

Algoritmo de Clarke y Wright - paralelo -. Ejemplo:



	Vértices					Km	U.
Ruta-1	0	9	0			40	4
Ruta-2	0	8	10	0		109	8
Ruta-3	0	12	1	0		54	8
Ruta-4	0	11	13	2	0	175	8
Ruta-5	0	4	6	0		52	7
Ruta-6	0	3	5	7	0	154	9
						584	**44**

Figura: algoritmo de Clarke y Wright (paralelo). Solución: 6 pétalos / 584 Km. Los pares que dan lugar a la fusión de pétalos son (según algoritmo): 11-13, 3-5, 5-7, 2-13, 8-10, 1-12 y 4-6.

Problema del diseño de rutas. Algoritmos de fusión de pétalos (4/6)

Algoritmo de Clarke y Wright. Versión serie o secuencial (simétrica-euclídea):

Paso - 1: Calcular para todo par clientes (i, j) el ahorro: $a_{ij} = l_{0i} + l_{j0} - l_{ij} \quad \forall i \forall j;\ a_{i0} = a_{0i} = 0\ \forall i$

Paso - 2: Ordenar los pares de clientes de mayor a menor valor de a_{ij}.

Paso - 3: *Mientras* hayan clientes por asignar, *Hacer*:

3.1 - Abrir ruta: Seleccionar el par (i, j) con vértices no asignados, con demanda menor o igual a la capacidad máxima de la ruta y con mayor ahorro (se contempla el depósito).

3.2 - *Mientras* la ruta abierta tenga capacidad remanente suficiente, *Hacer*:

- 3.2.1 - Si existe un par (i, j) con uno y solo uno de sus vértices coincidente con uno de los extremos de la ruta, con demanda del vértice no coincidente menor o igual que la capacidad remanente de la ruta, y con mayor ahorro; entonces: (1) asignar el vértice no coincidente a la ruta (nuevo extremo) y (2) actualizar la capacidad de la ruta.
- 3.2.2 - Si no existe un par (i, j) cumpliendo 3.2.1, entonces: (1) cerrar ruta abierta y (2) *Ir* a 3.1 (nueva ruta), si hay clientes por asignar, o (3) *Finalizar*, en caso contrario.

Problema del diseño de rutas. Algoritmos de fusión de pétalos (5/6)

Algoritmo de Clarke y Wright – serie -. Ejemplo:

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
d_i	3	3	2	3	6	4	1	4	4	4	3	5	2

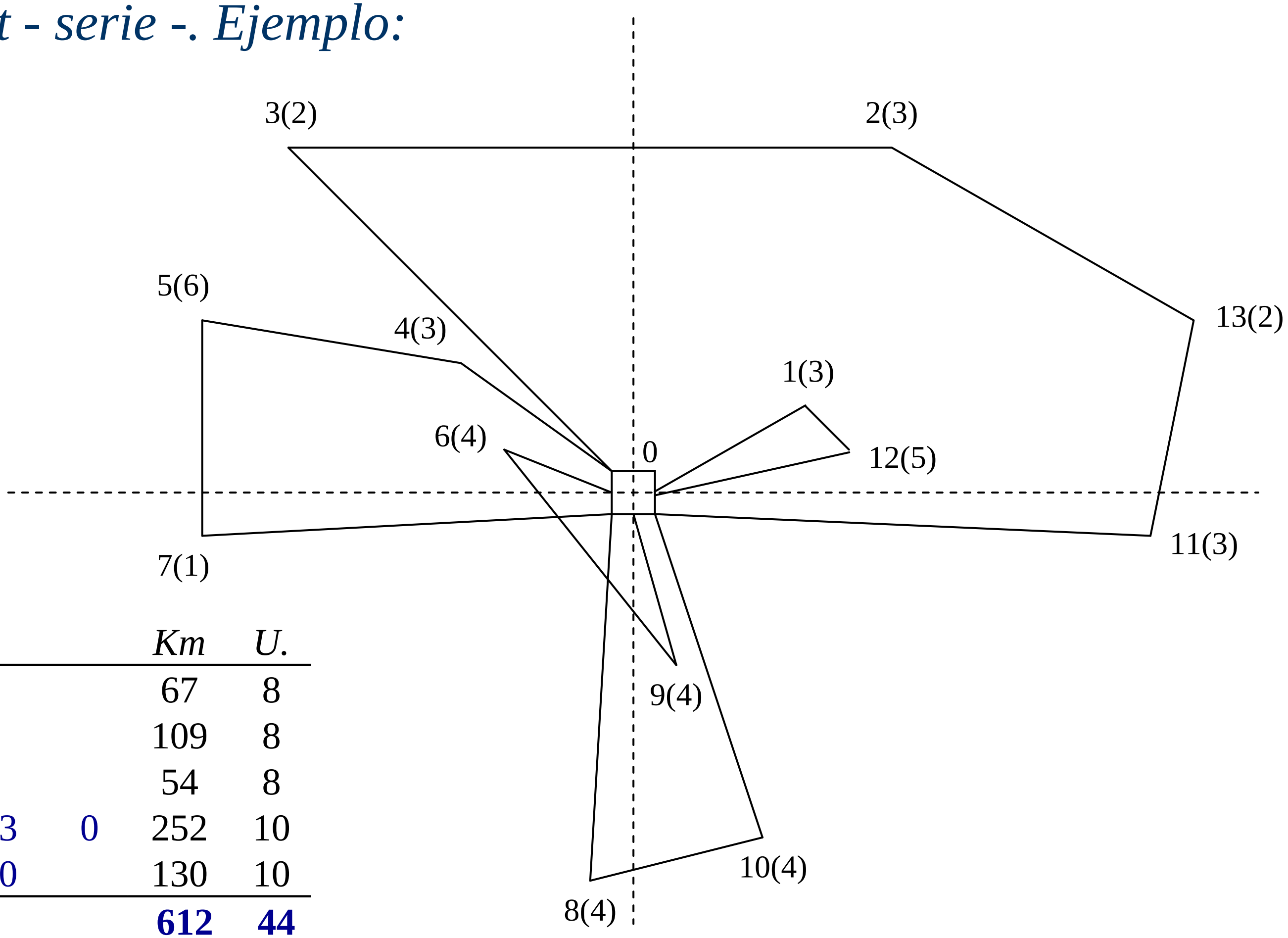
Tabla: Demandas de los clientes para el ejemplo prototipo. La capacidad máxima de una ruta es 10 *u*.

Etapa	Par	Ahorro	Fusión	Carga
1	11-13	103	(o)	5
2	2-13	78	11-13-2	8
3	2-3	37	11-13-2-3	10 (*)
4	5-7	79	(o)	7
5	4-5	48	4-5-7	10 (*)
6	8-10	67	(o)	8 (*)
7	1-12	41	(o)	8 (*)
8	6-9	5	(o)	8 (*)

Tabla: algoritmo de Clarke y Wright (serie). Solución: 5 pétalos / 612 Km. Los pares que dan lugar a la fusión de pétalos son: 11-13, 2-13, 2-3, 5-7, 4-5, 8-10, 1-12 y 6-9.

Problema del diseño de rutas. Algoritmos de fusión de pétalos (6/6)

Algoritmo de Clarke y Wright - serie -. Ejemplo:



	Vértices						*Km*	*U.*
Ruta-1	0	6	9	0			67	8
Ruta-2	0	8	10	0			109	8
Ruta-3	0	12	1	0			54	8
Ruta-4	0	11	13	2	3	0	252	10
Ruta-5	0	4	5	7	0		130	10
							612	**44**

Figura: algoritmo de Clarke y Wright (serie). Solución: 5 pétalos / 612 Km. Los pares que dan lugar a la fusión de pétalos son (según algoritmo): 11-13, 2-13, 2-3, 5-7, 4-5, 8-10, 1-12 y 6-9.

Problema del diseño de rutas. Algoritmos de 2 fases (1/2)

Algoritmo de Gillet y Miller (de barrido):

Fase - 1: Partición del conjunto de clientes:

Mientras hayan clientes por asignar, *Hacer* :

- Abrir ruta R : Seleccionar el vértice i primer extremo de la ruta.
- *Mientras* haya capacidad remanente en R, *Hacer* :
  - Seleccionar el vértice no asignado más cercano, en coordenada angular polar, al último seleccionado.
    Alternativas: sentido horario o antihorario.

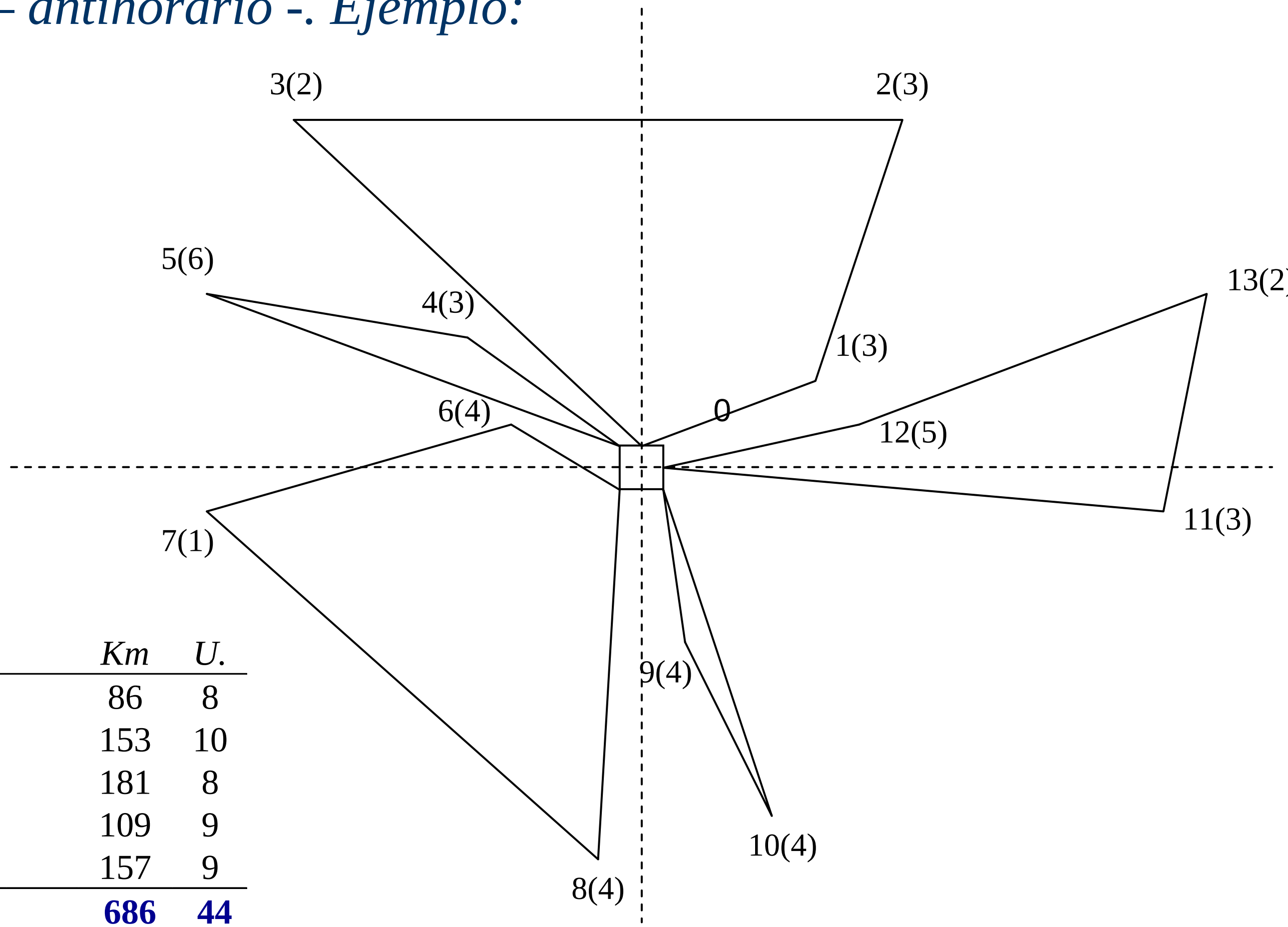
Fase - 2 : Determinar la secuencia de clientes en cada pétalo:

Para todo conjunto de clientes resultado de la partición, *Hacer* :

- Aplicar algoritmo para el TSP (p.e. algoritmos de fusión de pétalos).

Problema del diseño de rutas. Algoritmos de 2 fases (2/2)

Algoritmo de Gillet y Miller – antihorario -. Ejemplo:

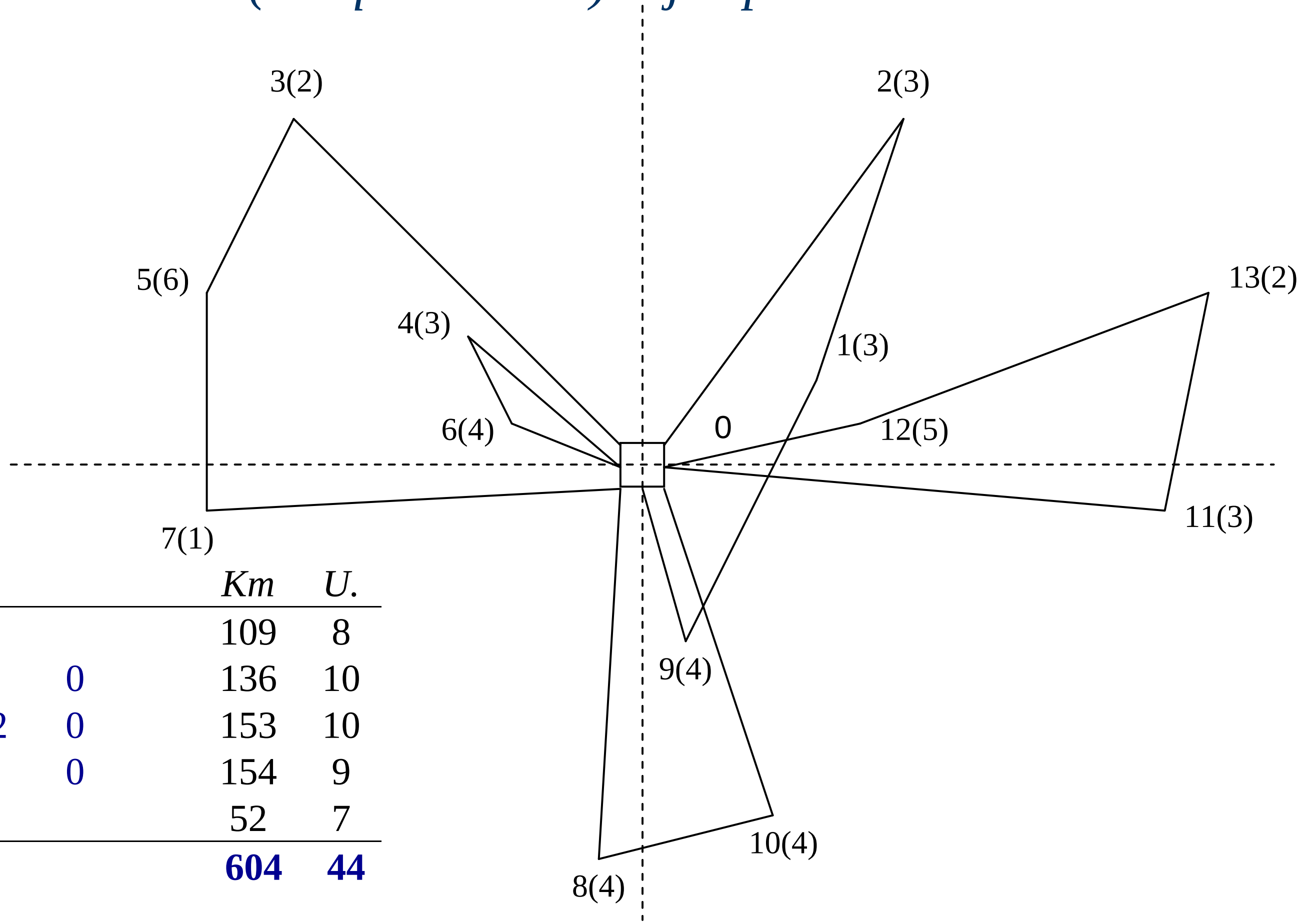


	Vértices					*Km*	*U.*
Ruta-1	0	9	10	0		86	8
Ruta-2	0	11	13	12	0	153	10
Ruta-3	0	1	2	3	0	181	8
Ruta-4	0	4	5	0		109	9
Ruta-5	0	6	7	8	0	157	9
						686	**44**

Figura: Solución obtenida (barriendo desde 1 en sentido contrario a las agujas del reloj) con el algoritmo de Gillet y Miller (5 pétalos / 686 Km)

Problema del diseño de rutas. Búsqueda local

Algoritmo de Exploración de entornos (búsqueda local). Ejemplo:



	Vértices					*Km*	*U.*
Ruta-1	0	8	10	0		109	8
Ruta-2	0	9	1	2	0	136	10
Ruta-3	0	11	13	12	0	153	10
Ruta-4	0	3	5	7	0	154	9
Ruta-5	0	4	6	0		52	7
						604	**44**

Figura: solución hallada para la jerarquía de criterios número de pétalos / distancia total (5 pétalos / 604 km) mediante búsqueda local.