Málaga, 10 de Julio de 2014

50 Años de Arc Routing

Ángel Corberán

Departament d'Estadística i Investigació Operativa

Universitat de València

Índice

- Introducción
- Problemas de rutas por arcos
  - Origen
  - Historia
  - El Problema del Cartero Chino
  - El Problema del Cartero Rural
- Aplicaciones
- Resolución exacta: Combinatoria Poliédrica

Introducción

El Problema del Cartero Chino

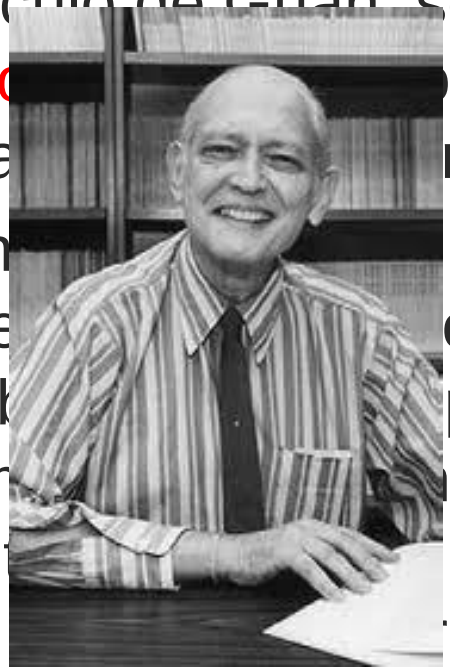
En los años seser un matemático trabajando en el S College, fue alentado (como muchos otr China) a resolver problemas reales ento del “Gran Salto Adelante” (1958-1 a transformar el país desde una econor economía moderna.

“When the author was plotting a diagram for a postman's route, he discovered the following problem: A postman has to cover his assigned segment before returning to the post office. The problem is to find the shortest walking distance for the postman”.

El Problema del Cartero Chino

El artículo de Guan, sobre la optimización de la ruta de un cartero
r un au
revista
mática
).

Basán
bien A
re
"Chine
oblem'
formal
po de
Goldm
al Bure
al
Institu
and Te
s
quien
e.

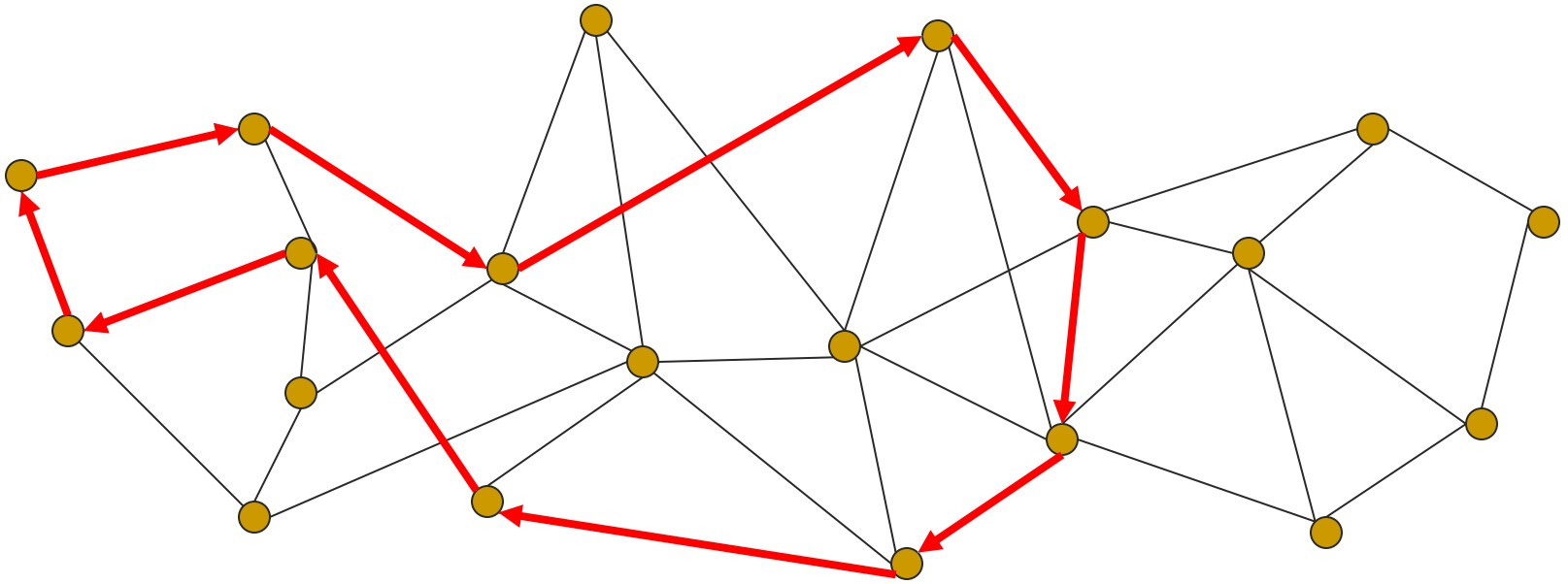


Lo que sí se sabe es que ese nombre apareció por 1ª vez en el título de un abstract de Edmonds para el 27 Congreso de la Operations Research Society of America (May 1965): "The Chinese Postman's Problem".

Problemas de rutas por arcos

Problemas de rutas por arcos

Son problemas relativos al recorrido de algunas (o todas) las aristas y/o arcos de un grafo



Problemas de rutas

Origen: problema de los puentes de Königsberg (Euler, 1736)

Rutas por vértices

- 1 Vehículo
  - TSP
  - GTSP
- k Vehículos
  - CVRP

Todos NP-difíciles

Rutas por arcos

- 1 Vehículo
  - CPP
  - RPP
- k Vehículos
  - CARP

Casi todos NP-difíciles

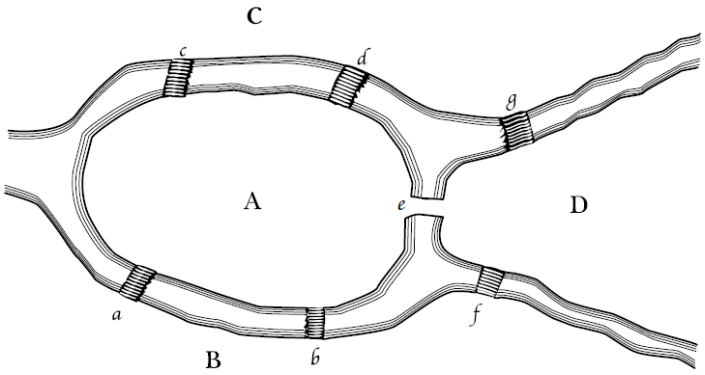
El problema de los puentes de Königsberg

Königsberg era la capital de Prusia oriental, pero ahora es conocida como Kaliningrado (Rusia). La ciudad está construida alrededor del río Pregel que la divide en 2 partes y deja una isla llamada Kneiphof en el centro. En el siglo XVII, 7 puentes unían las 4 partes de la ciudad.

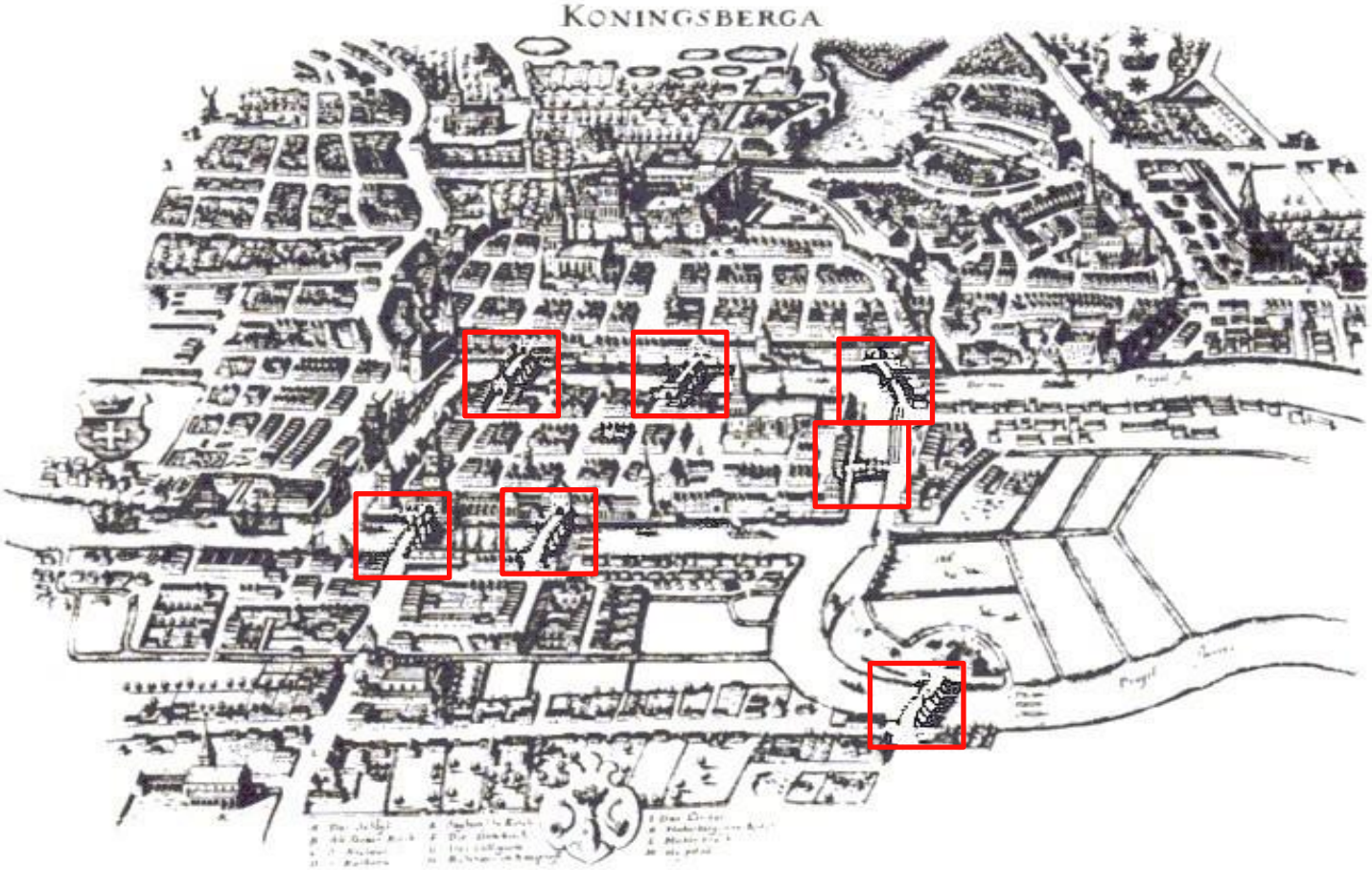
Se dice que sus habitantes intentaban encontrar un recorrido que atravesara exactamente una vez los 7 puentes y que volviera al punto de partida, pero nadie fue capaz de hallarlo.

El problema de los puentes de Königsberg

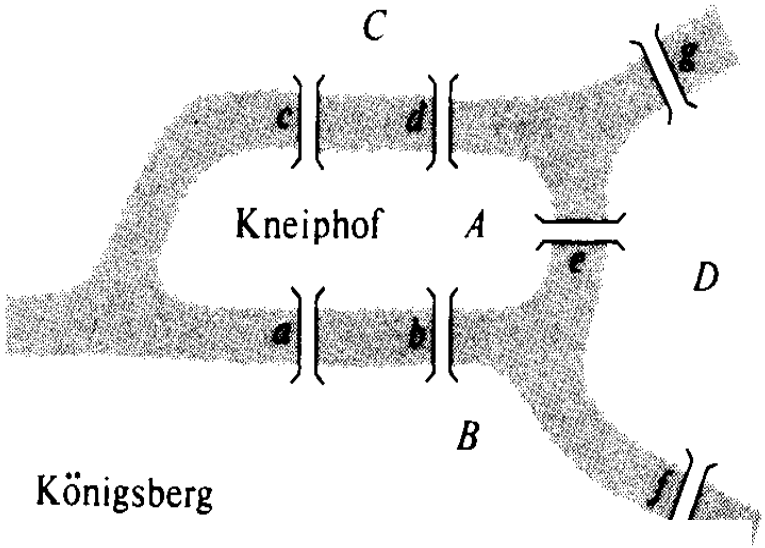
"The problem, which I am told is widely known, is as follows: in Königsberg in Prussia, there is an island A, called the Kneiphof; the river which surrounds it is divided into two branches, as can be seen [in the figure], and these branches are crossed by seven bridges a, b, c, d, e, f and g. Concerning these bridges, it was asked whether anyone could arrange a route in such a way that he would cross each bridge once and only once. I was told that some people asserted that this was impossible, while others were in doubt; but nobody would actually assert that it could be done. From this, I have formulated the general problem: whatever be the arrangement and division of the river into branches, and however many bridges there be, can one find out whether or not it is possible to cross each bridge exactly once?" (Euler, 1736)

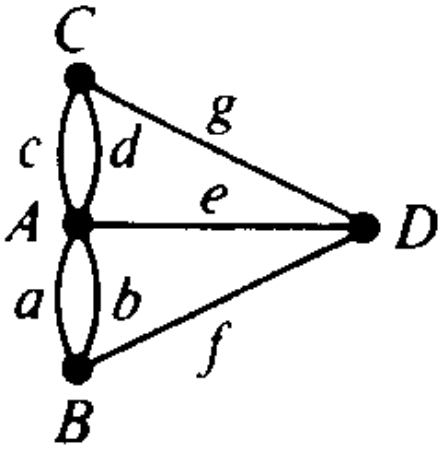


El problema de los puentes de Königsberg



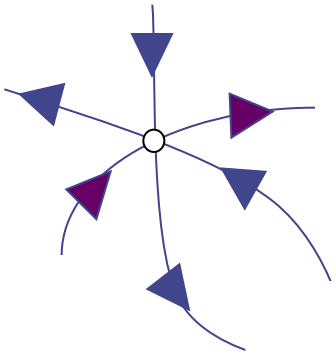
El problema de los puentes de Königsberg





El problema de los puentes de Königsberg

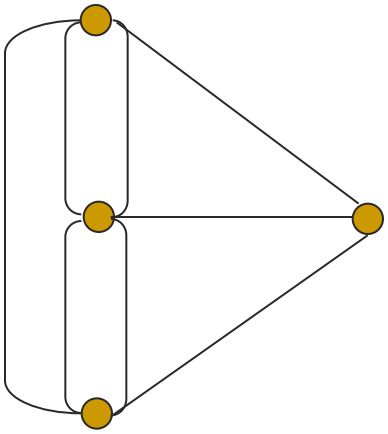
Euler señaló que encontrar una ruta que cruce cada puente una única vez es posible si se cumple que :
"cuando crucemos a una parte de la ciudad, hemos de poder salir por otro puente".



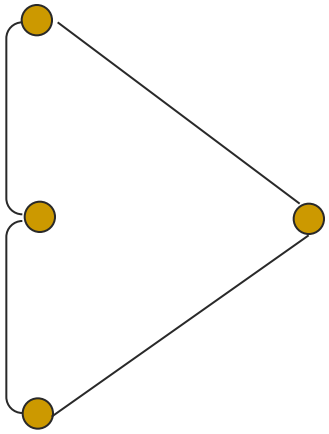
El grado de cada vértice debe ser una suma de 2's. Debe ser un número par.

Un grafo con todos sus vértices de grado par se llama grafo Euleriano.

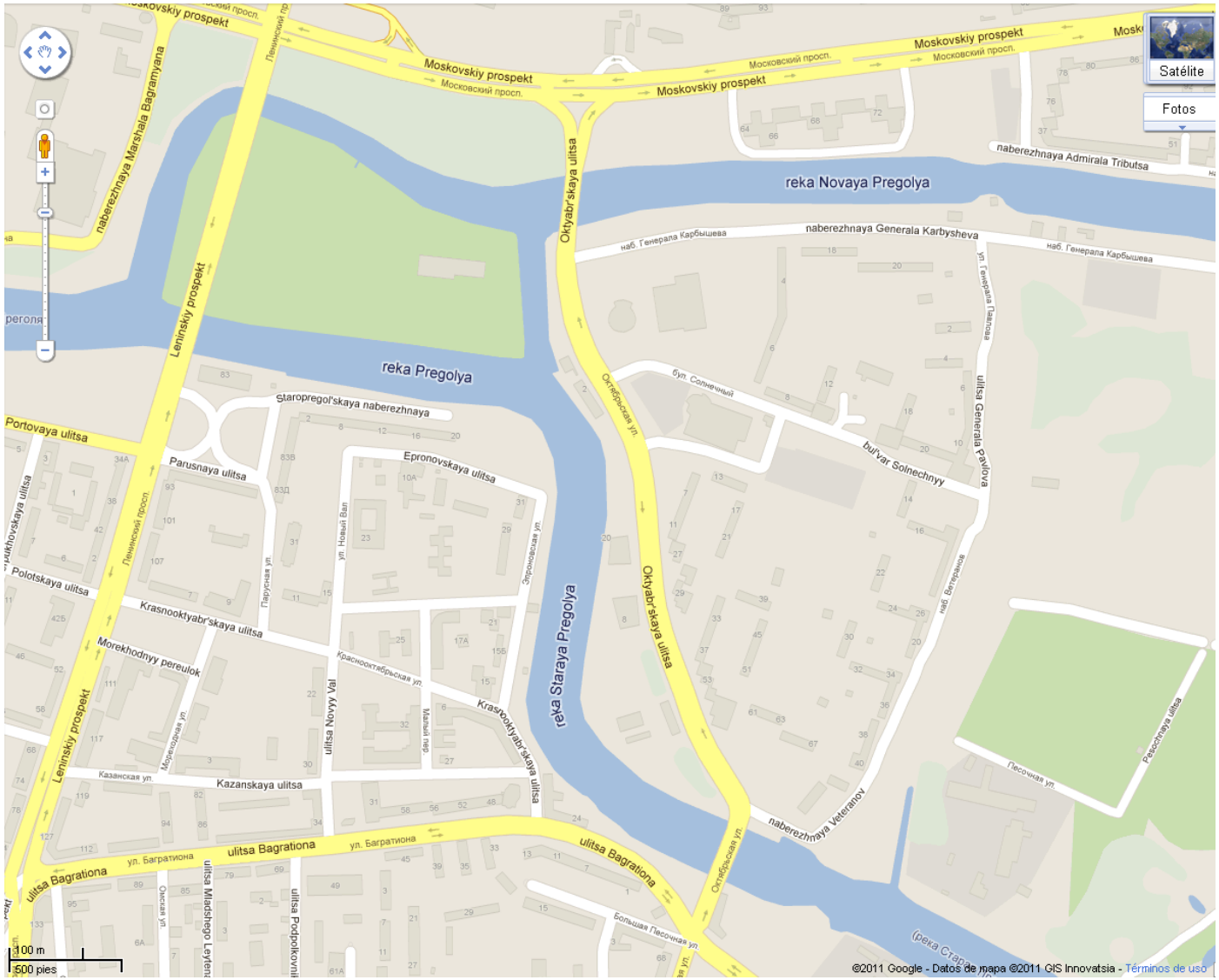
El problema de los puentes de Königsberg



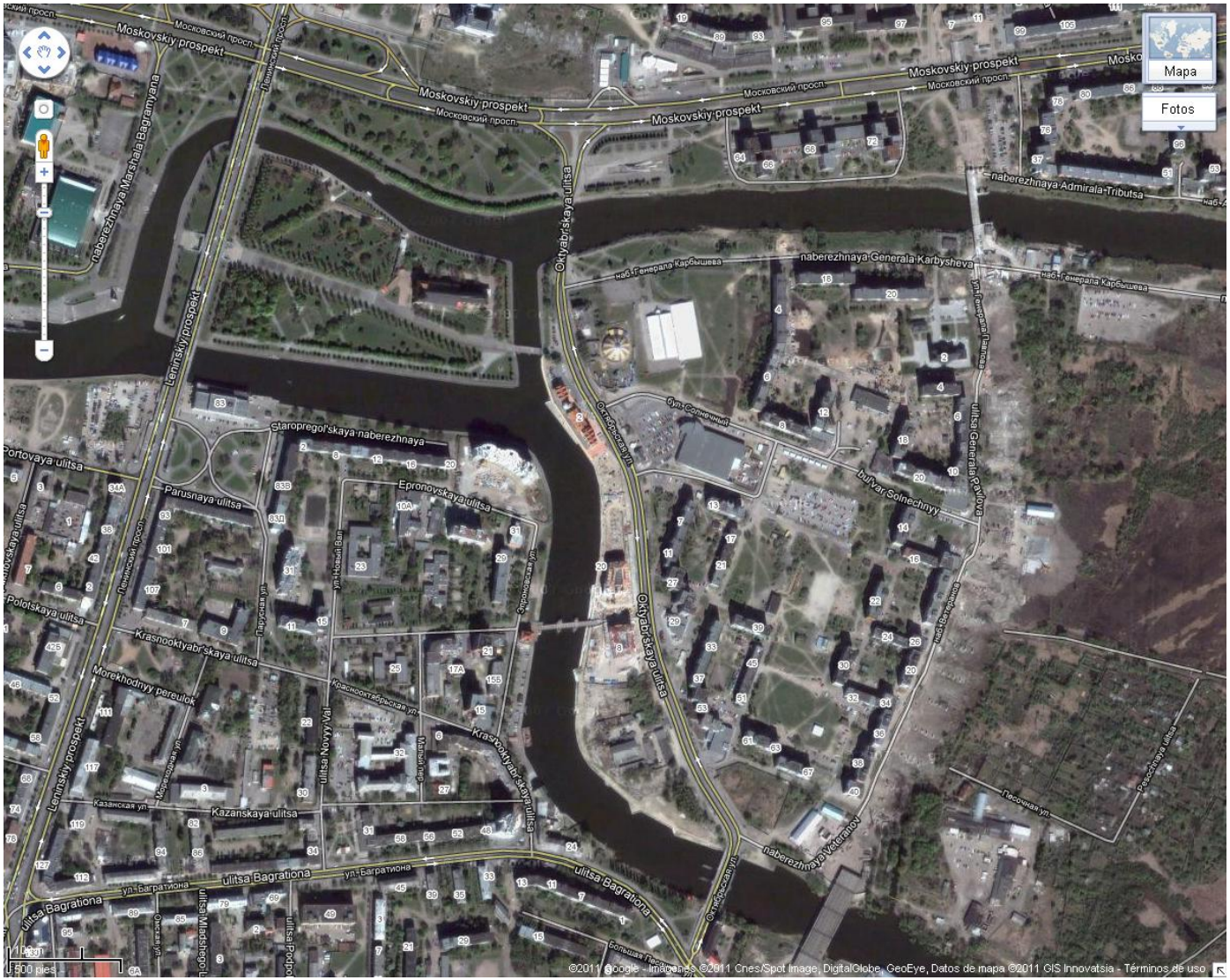
Königsberg a finales del siglo XIX



Kaliningrado



Kaliningrado, Google Maps



Kaliningrado, Google Earth

El problema de los puentes de Königsberg

Euler dió las condiciones necesarias para que un grafo G conexo y no dirigido sea Euleriano. Una demostración de que eran también suficientes llegó en 1873, en un artículo póstumo de Hierholzer.

G es Euleriano si y sólo si cada vértice de G tiene grado par.

Una segunda caracterización fue dada en 1912 por Veblen:

G es Euleriano si y sólo si G es la unión de ciclos disjuntos.

El problema de los puentes de Königsberg

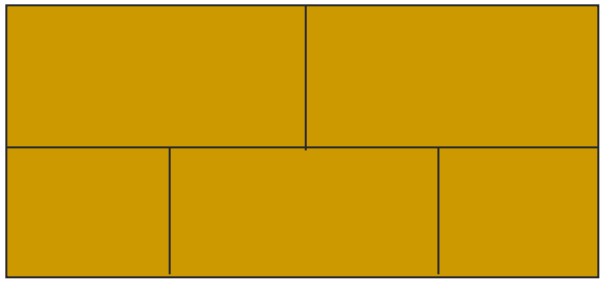
"When it has been determined that such a journey can be made, one still has to find how it should be arranged [. . .] It is [. . .] an easy task to construct the required route [. . .] I do not therefore think it worthwhile to give any further details concerning the finding of the routes" (Euler, 1736)

Los detalles fueron proporcionados por primera vez por Hierholzer (1873), quien demostró que hay un algoritmo que en tiempo lineal, dado un grafo G no dirigido, o bien encuentra un tour Euleriano de G o demuestra que G no es Euleriano.

Dibujo de figuras

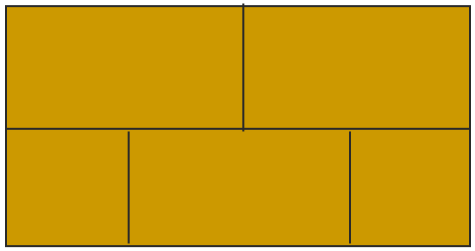
Otro problema relacionado es el de dibujar una figura con el menor número de trazos posible

Por ejemplo, ¿con cuántos trazos se puede dibujar la figura siguiente?



Dibujo de figuras

En 1847, J.B. Listing escribió un tratado que incluía una discusión sobre el dibujo de figuras. Observó que la figura

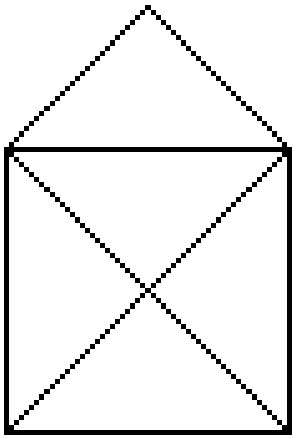


tiene 8 vértices impares y, por lo tanto, no puede dibujarse con menos de 4 trazos.

La casa de Santa Claus

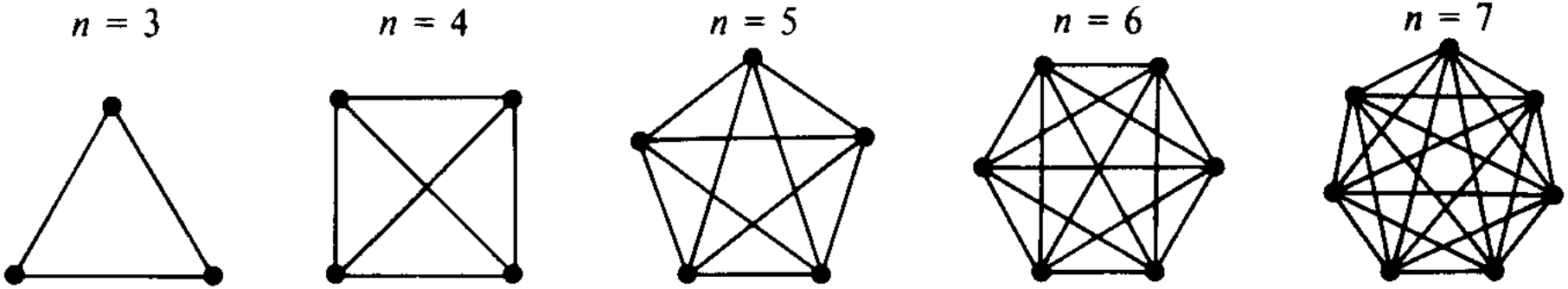
La casa de Santa Claus es un juego alemán para niños pequeños muy antiguo .

Hay que dibujar una casa con un sólo trazo. No hay que levantar el lápiz del papel mientras dibujas. No se puede repetir ninguna línea. Mientras se dibuja hay que recitar: "Das ist das Haus des Nikolaus" (“Esta es la casa de Santa Claus"). Hay que cantar una sílaba de esta frase cada línea recta dibujada.



Dibujo de figuras

En 1809, Poinsot demostró que figuras formadas por *n* puntos, todos unidos entre sí, pueden ser dibujadas con 1 único trazo si *n* es impar, pero no si *n* es par.



Laberintos

A finales del siglo XIX se prestó una gran atención al problema de escapar de un alberinto.

Este problema está estrechamente relacionado con los grafos Eulerianos.

Laberintos (Hever Castle, UK)

Laberinto de Hampton Court

El Problema del Cartero Chino (CPP)

El Problema del Cartero Chino

Guan, 1962

CPP: Encontrar un tour de longitud mínima que atraviese al menos una vez cada arista de un grafo dado.

Si un grafo es Euleriano, el mismo grafo es la solución al Problema del Cartero Chino.

Si no es Euleriano, al menos una de sus aristas tendrá que ser recorrida más de 1 vez. En el contexto de los problemas de rutas de vehículos, esto recibe el nombre de tiempo muerto, puesto que el cartero (vehículo) no está realizando ningún trabajo productivo.

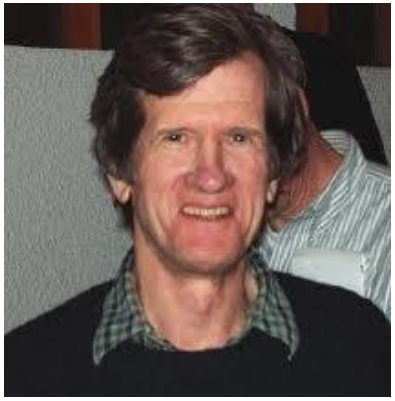
CPP: Resolución

"The Konigsberg Bridges Problem Generalized"
Bellman y Cooke, 1967 (J. Math. Analysis and App.)
"In connection with a given graph, we ask for the paths which traverse every edge at least once and for which the number of repetitions of edges is a minimum".

… "The (dynamic programming) procedure outlined can require a large number of comparisons. The number is difficult to estimate."

CPP: Resolución

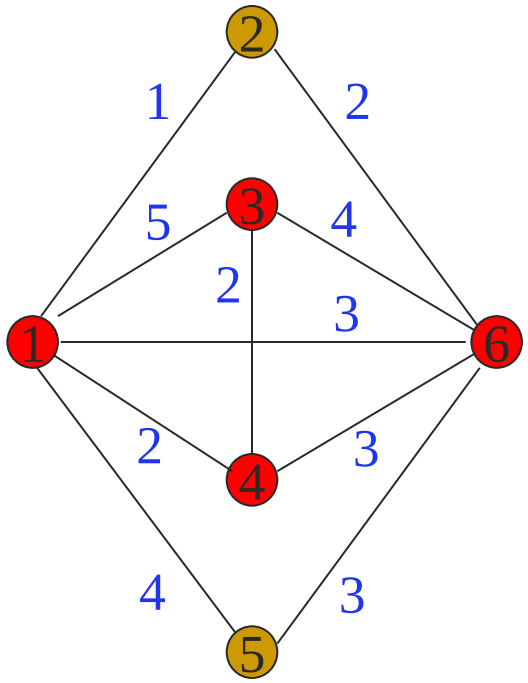
Edmonds, 1965
Christofides, 1973
Edmonds and Johnson, 1973



Debemos convertir los vértices im do algunas aristas, de forma que la lo mínima.

Para ello añadiremos caminos má es impares, cuya longitud total sea m

CPP: Resolución

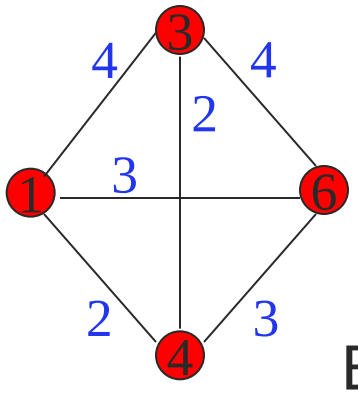


Los caminos más cortos son:

	1	2	3	4	5	6
1	0	1	4	2	4	3
2		0	5	3	5	2
3			0	2	7	4
4				0	6	3
5					0	3
6						0

CPP: Resolución

Los vértices impares son el 1, 3, 4 y 6. Usando las distancias más cortas como longitudes:



Acoplamientos: 1 – 3 y 4 – 6
1 – 4 y 3 – 6
1 – 6 y 3 – 4

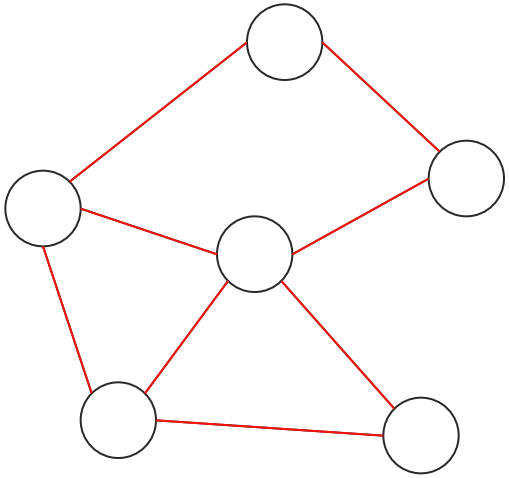
El acoplamiento 1 – 6 y 3 – 4 es el óptimo

Encontrar un acoplamiento de coste mínimo puede hacerse en tiempo polinómico (Edmonds)

CPP: Resolución

Tour óptimo : 1-2-6-5-1-3-6-4-3-4-1-6-1

El Problema del Cartero Chino



x_e=copias de e a añadir a G para obtener un grafo Euleriano.

CPP Formulación (Edmonds & Johnson, 1973) :

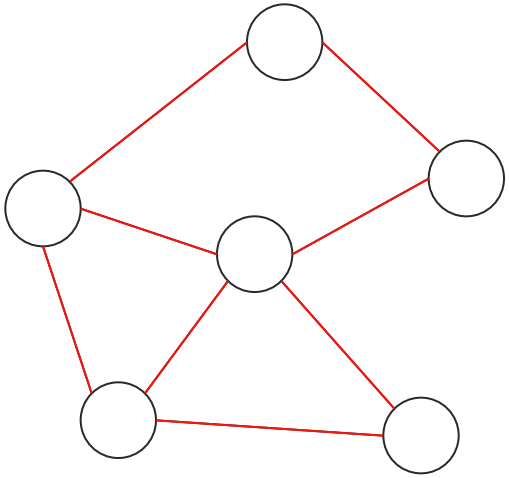
Minimizar $\sum c_e x_e$

$x(\delta(v)) \equiv d(v), \forall v \in V$

$x_e \geq 0$ y enteras, $\forall e \in E$

Restricciones no lineales!!

El Problema del Cartero Chino



x_e=copias de e a añadir a G para obtener un grafo Euleriano.

CPP Formulación (Edmonds & Johnson, 1973) :

Minimizar $\sum c_e x_e$

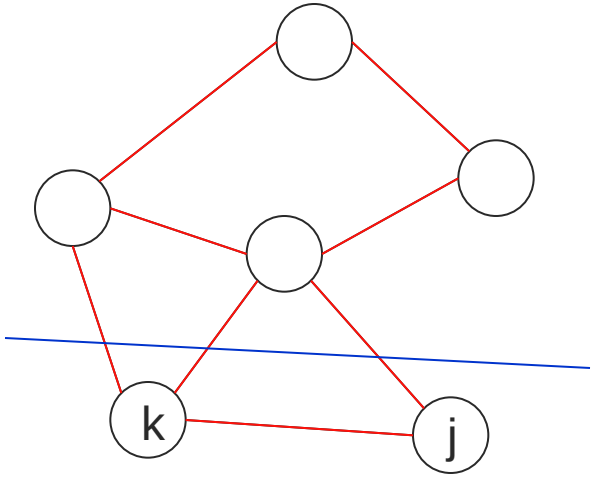
Número exponencial !! $\longrightarrow$ $x(\delta(S)) \geq 1, \quad \forall S \subset V, |\delta(S)|$ es impar

$x_e \geq 0, \qquad \forall e \in E$

Descripción completa (resoluble polinómicamente)

Desigualdades de cortadura impar

$x(\delta(S)) \geq 1, \forall S: |\delta(S)|$ es impar

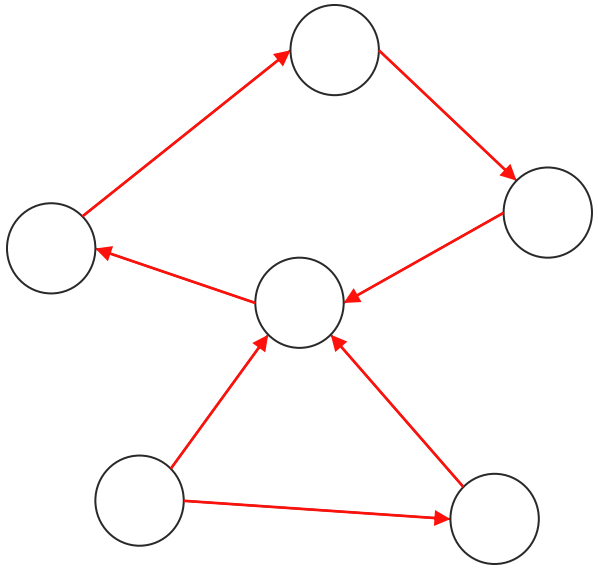


S={k,j}

$x(\delta(S)) \geq 1$

El Problema del Cartero Chino Dirigido (DCPP)

Grafos dirigidos Eulerianos



G=(V,A) fuertemente conexo

La paridad del gra tices es una condición ne o suficiente para qu igido sea Euleriano.

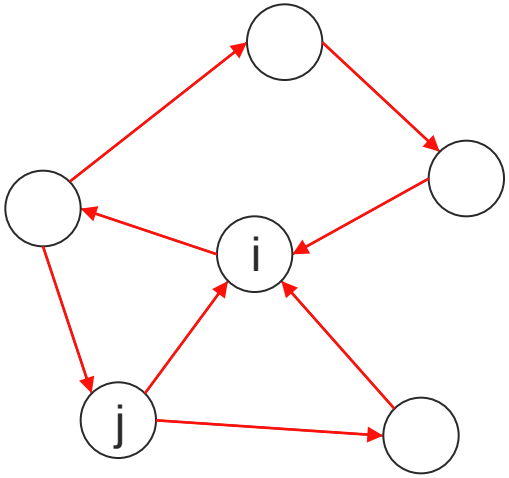
König (1936):

Un grafo dirigido f. conexo es Euleriano si y sólo si es simétrico

(G es simétrico si $\forall i \in V$, nº arcos entran = nº arcos salen)

DCPP: Resolución

Edmonds & Johnson, 1973



$d^+(i)=1$ $d^-(i)=3$ → oferta(i)= $s_i = d^-(i)-d^+(i)$

$d^+(j)=2$ $d^-(j)=1$ → demanda(j)= $t_j = d^+(j)-d^-(j)$

x_{ij}=copias de (i,j) a añadir a G para obtener un grafo Euleriano.

$$Min \sum_{i \in S,\, j \in T} c_{ij} x_{ij}$$

$$\sum_{i \in S} x_{ij} = t_j \quad \forall j \in T$$

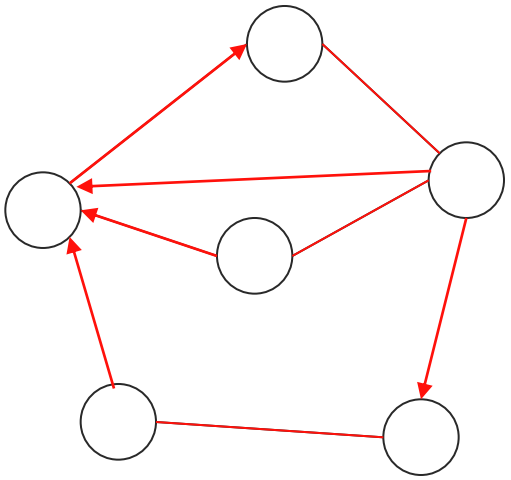
$$\sum_{j \in T} x_{ij} = s_i \quad \forall i \in S$$

$$x_{ij} \geq 0$$

Resoluble en tiempo polinómico

El Problema del Cartero Chino en un grafo mixto (MCPP)

Grafos mixtos Eulerianos



No Euleriano

G=(V,E,A) fuertemente conexo

La paridad del grado de los vértices es, de nuevo, una condición necesaria pero no suficiente para que un grafo mixto sea Euleriano.

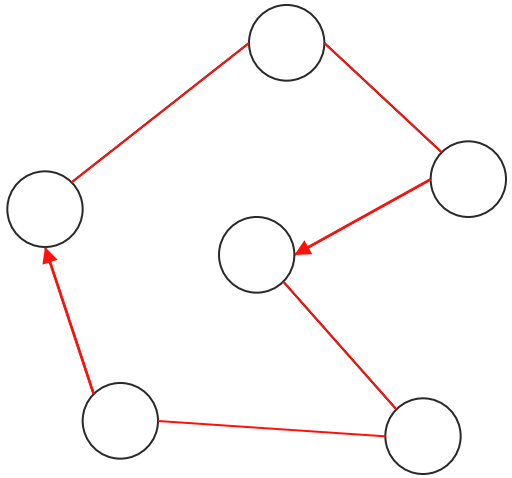
G=(V,E,A) es Euleriano si

G es par, y
G es simétrico

¿Son esas condiciones también necesarias para que G sea Euleriano?

Grafos mixtos Eulerianos

La respuesta es no, como demuestra la siguiente figura:



Entonces, ¿existe una condición necesaria y suficiente para que un grafo mixto sea Euleriano?

Grafos mixtos Eule

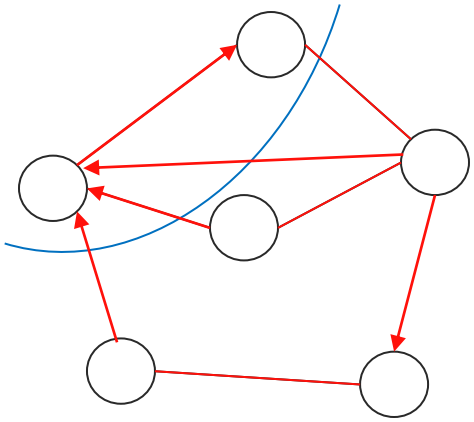
Ford y Fulkerson (1962)

G=(V,E,A) fuertemente conexo es Euleriano sii

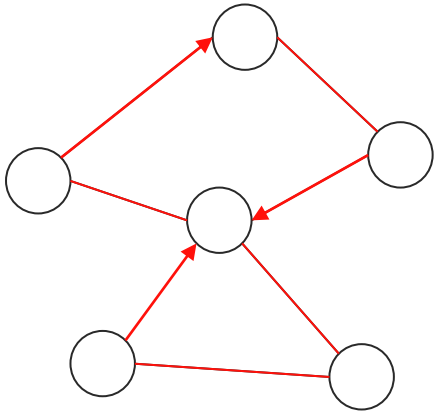
G es par, y

G es equilibrado, i.e. $\forall S \subset V$,

(arcos entran en S)-(arcos salen de S) $\leq$ (aristas entr



No equilibrado



Equilibrado

Grafos mixtos Eulerianos

¿Cómo podemos chequear si un grafo es equilibrado?

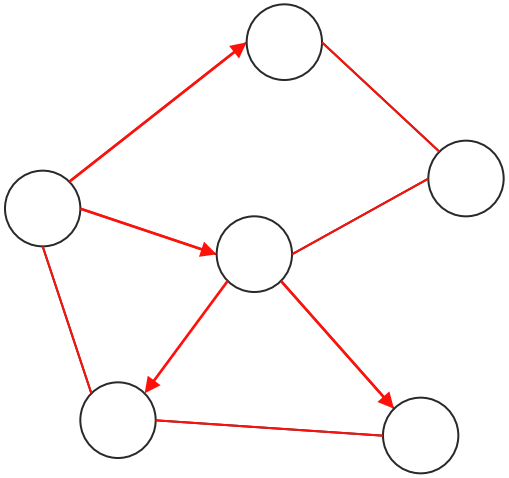
Nobert y Picard (1996) propusieron un algoritmo polinómico que encuentra una desigualdad de equilibrio violada si ésta existe.

El problema del cartero chino en un grafo mixto (MCPP)

Edmonds & Johnson (1973)

NP-difícil (Papadimitriou,1976)

Resoluble en tiempo polinómico si G es par (Edmonds & Johnson,1973)



$A^+(S)$ (A… s saliendo de (entr…

E(S) = a… S y V\S

MCPP: Algoritmos heurísticos

El algoritmo exacto para el caso en que *G* es par (llamado *Even MCPP Algorithm*) es la base para dos heurísticos para el caso general sugeridos por Edmonds & Johnson (1973) y desarrollados y mejorados por Frederickson (1979):

El algoritmo MIXED1 equivaldría a transformar primero *G* en un grafo par y aplicar a continuación el Even MCPP Algorithm.

MCPP: Algoritmos heurísticos

El algoritmo MIXED2 puede ser considerado como la versión inversa de MIXED 1. Primero resuelve un problema de flujo de coste mínimo en *G* para obtener un grafo simétrico. A continuación, resuelve el CPP (no dirigido) para obtener un grafo par (y simétrico).

MIXED1 y MIXED2 tienen un ratio del peor caso de 2, pero *Mixed Algorithm*, que consiste en aplicar ambos heurísticos y seleccionar el mejor tour obtenido, tiene un ratio del peor caso de 5/3.

Raghavachary & Veerasamy (1998) propusieron una modificación del Mixed Algorithm de Frederickson con un mejor ratio del peor caso de 3/2.

El problema del cartero chino en un grafo mixto (MCPP)

Formulación (Norbert & Picard, 1996)

x_e= copias de e a añadir a G para obtener un grafo Euleriano.

$$\text{Minimizar} \sum c_e x_e$$

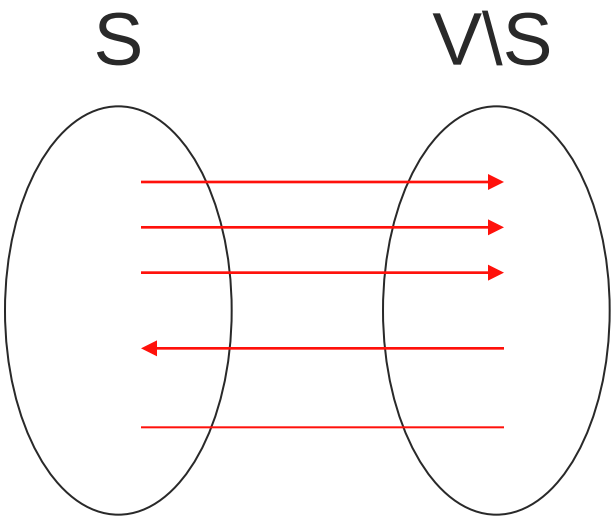
$$x(\delta(i)) \equiv |\delta(i)| \pmod 2, \quad \forall i \in V$$

$$x(A^+(S)) - x(A^-(S)) + x(E(S)) \geq u_S, \quad \forall S \subset V,$$

$$x_e \geq 0 \text{ y entero} \quad \forall e \in E \cup A,$$

donde $u_S = |A^-(S)| - |A^+(S)| - |E(S)|$

Desigualdades de equilibrio



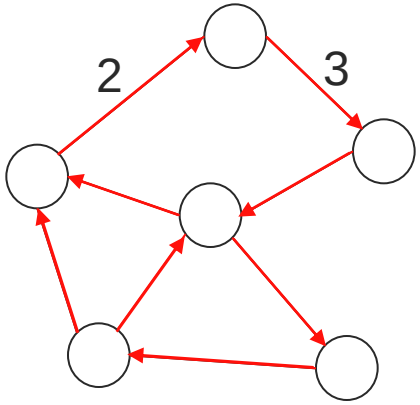
$u_S = |A^-(S)| - |A^+(S)| - |E(S)|$
$= 1 - 3 - 1 =$ “desequilibrio” de S

$(x(A^+(S))+3) - (x(A^-(S))+1) + (x(E(S))+1) \geq 0$

$(x(E(S))+1) \geq (x(A^-(S))+1) - (x(A^+(S))+3)$

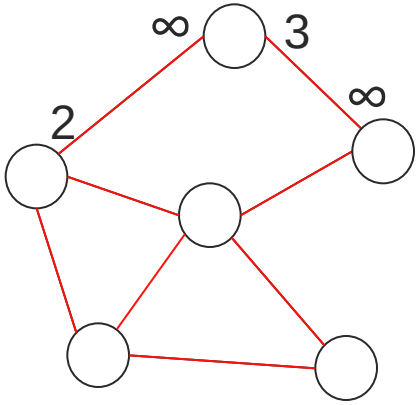
El Problema del Cartero con Viento

Problemas de rutas en grafos windy



Un grafo “windy” es un grafo no dirigido con costes asimétricos.

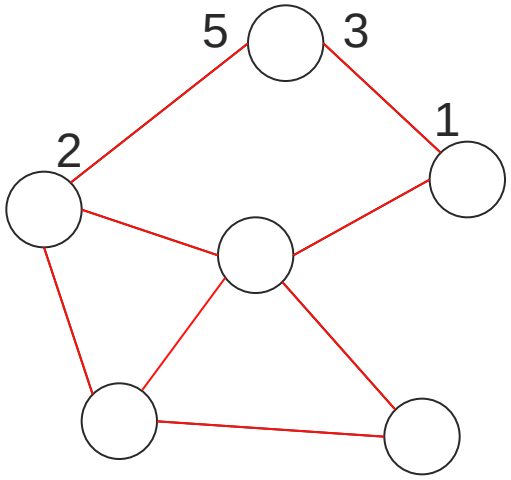
Los grafos no dirigidos, dirigidos y mixtos pueden ser considerados casos especiales de los grafos windy.



Por lo tanto, los PRA’s windy generalizan los correspondientes PRA’s definidos en grafos no dirigidos, dirigidos y mixtos.

El Problema del Cartero con Viento

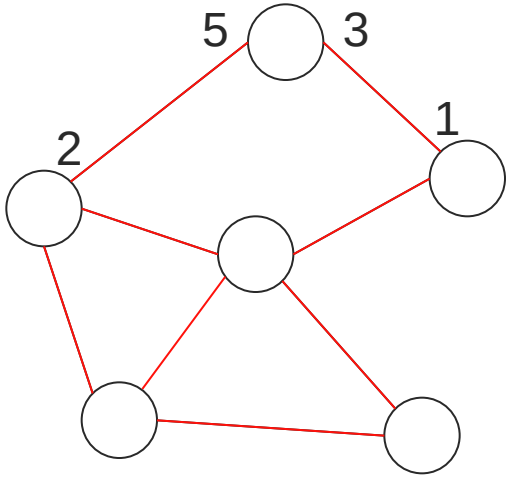
Minieka (1979)



(that the cost of traversing an edge is the same for either direction) “is hardly a good assumption when one direction might be uphill and the other downhill, when one direction might be with the wind and the other against the wind or when fares are different depending on direction”.

Dado un grafo windy G=(V,E), el WPP consiste en encontrar un tour de coste mínimo que atraviese todas las aristas de G al menos una vez.

El Problema del Cartero con Viento

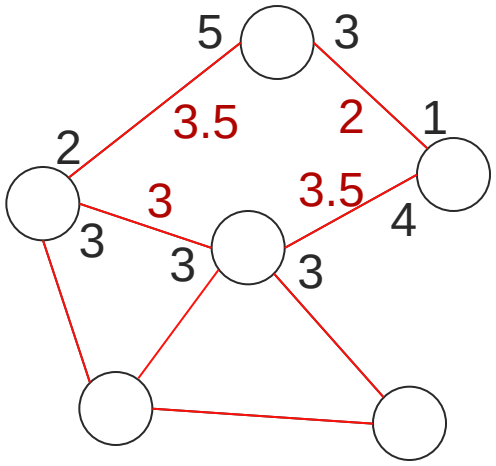


WPP es NP-difícil
(Brucker 1981 y Guan 1984)

Algunos casos particulares pueden ser resueltos en tiempo polinómico:

-Cuando las dos orientaciones de cada ciclo C en G tienen el mismo coste (Guan 1984), o
-Cuando G es par (Euleriano) (Win 1987)

El Problema del Cartero con Viento

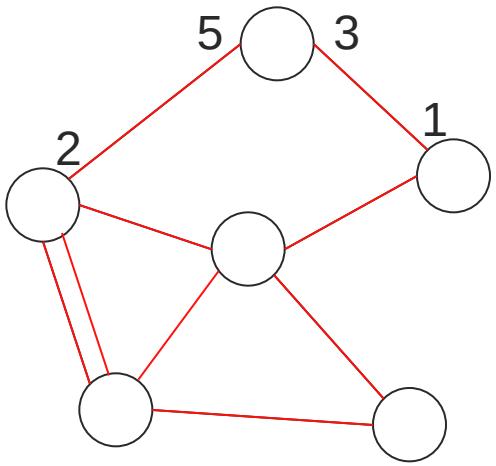


Cuando las dos orientaciones de cada ciclo C en G tienen el mismo coste (Guan 1984)

Algoritmo exacto (polinómico) de Guan:

1. Para cada arista e=(i,j) definir el coste $c_e=(c_{ij}+c_{ji})/2$
2. Resolver el CPP estándar con los costes c_e
3. Construir una orientación de la solución óptima del CPP.

El Problema del Cartero con Viento



Si G es par (Euleriano) (Win 1987)

(Basado en el algoritmo de Edmonds y Johnson para el MCPP en grafos pares, 1973)

Heurístico para grafos cualesquiera

Win (1989)

Dado un grafo windy conexo G=(V,E),

1. Añadir algunas aristas a G para obtener un grafo Euleriano (resolviendo un acoplamiento de coste mínimo en los vértices de grado impar, costes $c_e = \frac{c_{ij} + c_{ji}}{2}$)

2. Aplicar el algoritmo de Win para WPP's Eulerianos

Worst case ratio = 2

Formulación del WPP

Win (1987), Grötschel & Win (1992)

x_{ij} = nº de veces que (i,j) es recorrida desde i a j

$$\text{Minimizar } \sum_{(i,j)\in E}(c_{ij}x_{ij}+c_{ji}x_{ji})$$

$$x_{ij}+x_{ji} \geq 1, \ \forall (i,j)\in E \qquad (1)$$

$$\sum_{(i,j)\in \delta(i)}(x_{ij}-x_{ji})=0, \ \forall i\in V \qquad (2)$$

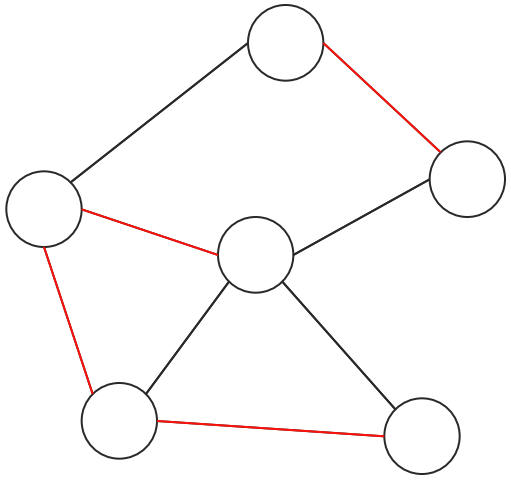
$$x_{ij}, x_{ji} \geq 0 \qquad (3)$$

$$x_{ij}, x_{ji} \text{ enteros} \qquad (4)$$

El Problema del Cartero Rural (RPP)

El Problema del Cartero Rural

Orloff (1974)



$G^R=(V,E_R)$ disconexo:

R-sets: $V_1, V_2, \ldots, V_p$

$\delta_R(S) = \delta(S) \cap E_R$

El Problema del Cartero Rural

Formulación del RPP (C. & Sanchis, 1994) :

x_e=copias of e a añadir a G^R para obtener un grafo Euleriano.

$$\text{Minimizar} \sum c_e x_e$$

$$x(\delta(S)) \geq 2, \quad \forall S \subset V,\ \delta_R(S)=\varnothing$$

$$x(\delta(i)) \equiv |\ \delta_R(i)\ | \ (\text{mod. } 2), \quad \forall i \in V$$

$$x_e \geq 0 \text{ y entera } \forall e \in E$$

El Problema de Rutas por Arcos con Capacidades (CARP)

On the Road to Efficiency (Hall & Partyka)

“The classic traveling salesman problem is a walk in the park compared to the vehicle routing problem….”

OR MS Today, June 1997

El CARP

Christofides (1973)

Golden and Wong (1981)

G = (V,E)

E_R : aristas que requieren servicio (requeridas)

c_{ij} coste de la arista (i,j),

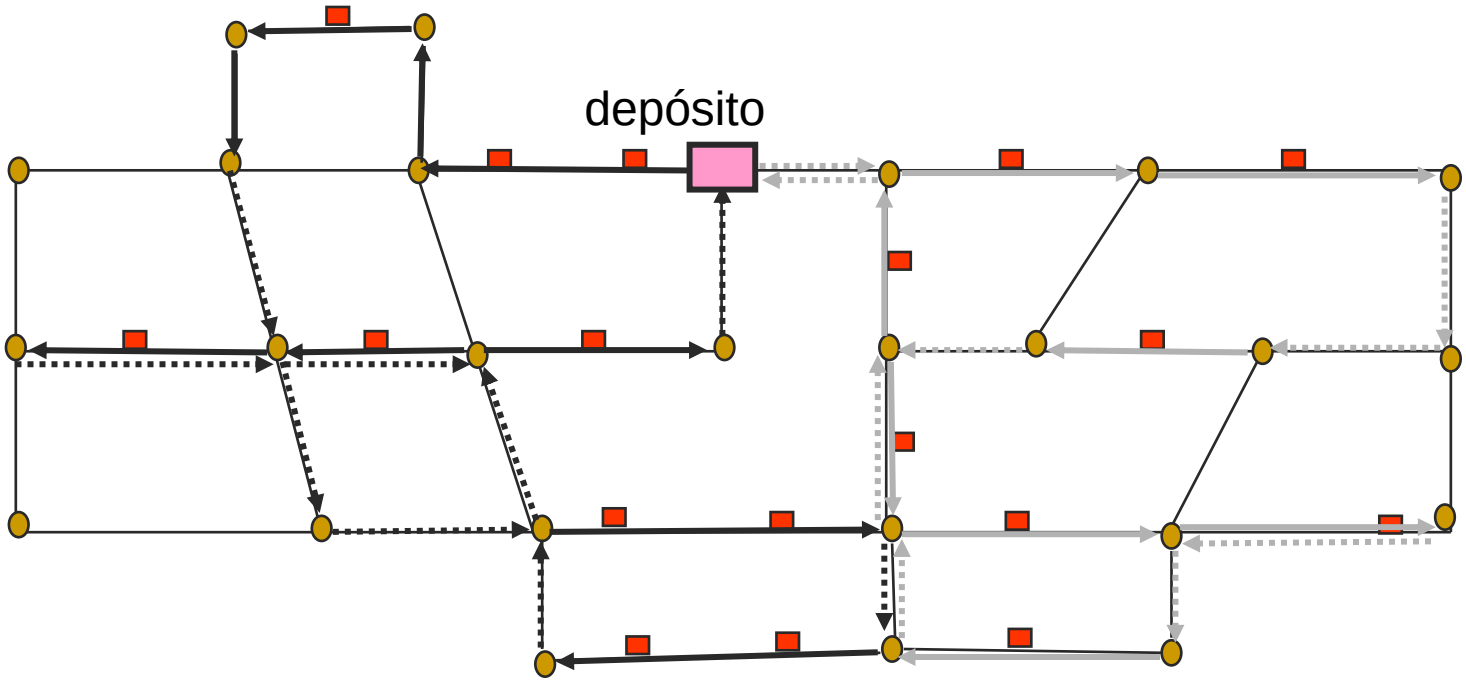
d_{ij} demanda de la arista (i,j),

C capacidad de los vehículos.

El CARP consiste en encontrar un conjunto de rutas con coste total mínimo tales que:

- cada arista de E_R esté en alguna ruta
- cada ruta pase por el depósito, y la demanda servida no exceda la capacidad del vehículo C

El CARP



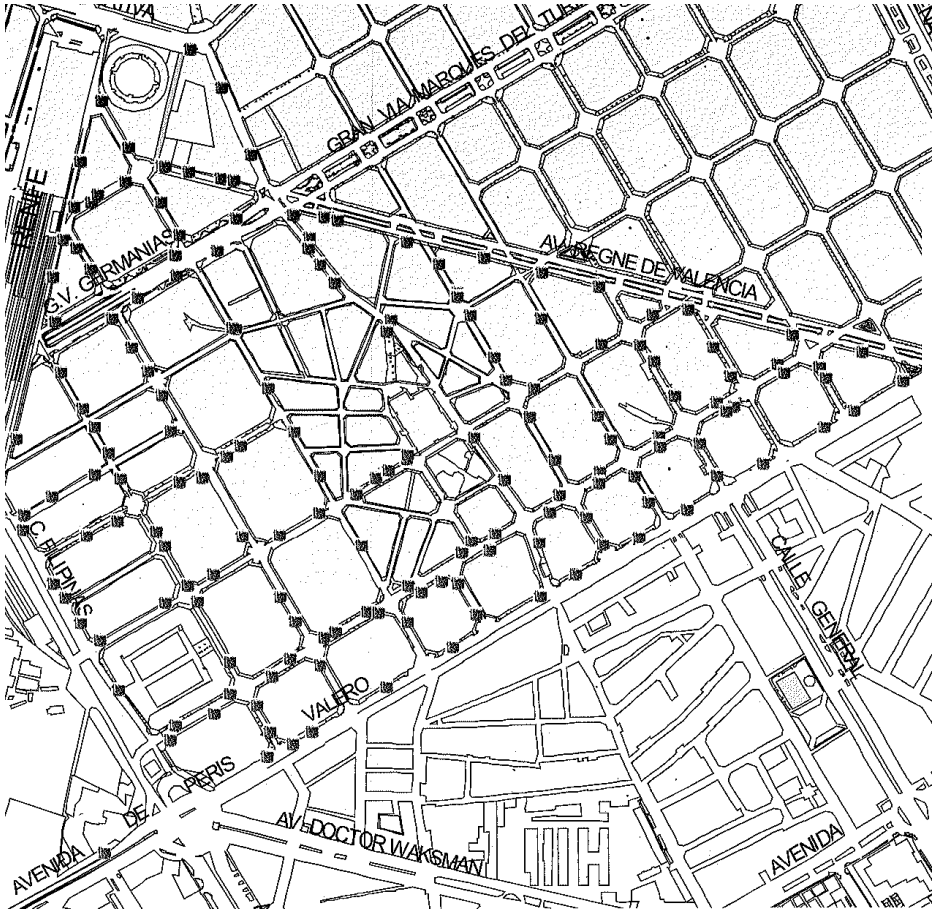
C = 15

sirviendo

sin servir

Aplicaciones

Recogida de la basura



AJUNTAMENT DE VALENCIA

Barrio 2.1 - Russafa

Limpieza de calles y recogida de la basura

Valencia (2002):

- Limpieza de calles (diaria): 1028 trabajadores, 11 camiones
- Riego de calles: 39 camiones
- Recogida de basura: 10584 contenedores+ 1660 (vidrio) + 1770 (cartón) + 1725 (plástico) + 30 (otros) 101 camiones

- Presupuesto 2007 : 130.107.449 Euros (18,23 %)

Limpieza de calles y recogida de la basura

(1972-1974) Cleveland:

- Reducciones importantes en el coste de la recogida de basura:

  de 1640 trabajadores a 850.
- Presupuesto: de 14.8 millones de dólares en (1970)

  a 8.8 millones en (1972):

CLARK & GILLEAN (1975)

Lectura de contadores

(1979) Beersheva (Israel):

8 Zonas (1 zona consiste en 42 vértices y 62 aristas)

- Reducción importante: de 24 a 15 tours en 1 zona
- Ahorro estimado: 40% en 1 zona.

STERN & DROR (1979)

Control de la nieve y el hielo

“According to Perrier et al. (2006a,b, 2007a,b, 2011), the importance of winter road maintenance is due to the magnitude of the expenditures associated to these operations, and to the indirect costs resulting from the loss of productivity and decreased mobility.
In the United States alone these operations consume over $2 billion yearly in direct costs.
In Japan and Europe snow removal expenditures are two to three times those of the United States”.

(Salazar-Aguilar, Langevin, Laporte, 2011)

Retirada de la nieve

La autopista 720 durante una tormenta de nieve en Montreal. The Montreal Gazette, 07/03/2011

(Synchronized Arc Routing for Snow Plowing Operations, Salazar-Aguilar, Langevin, Laporte, 2011)

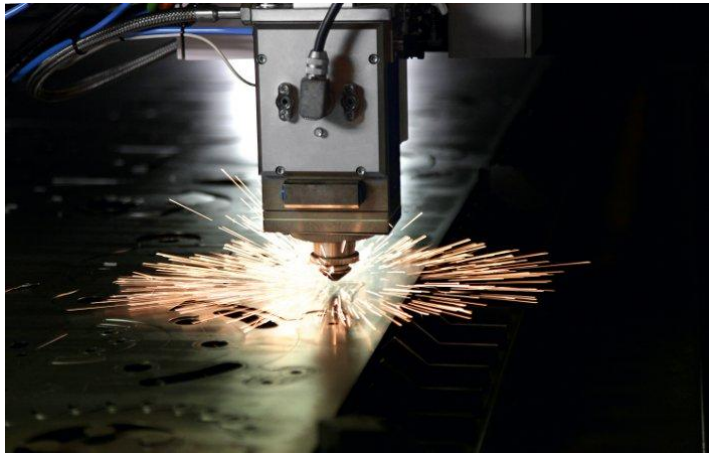
Retirada de la nieve

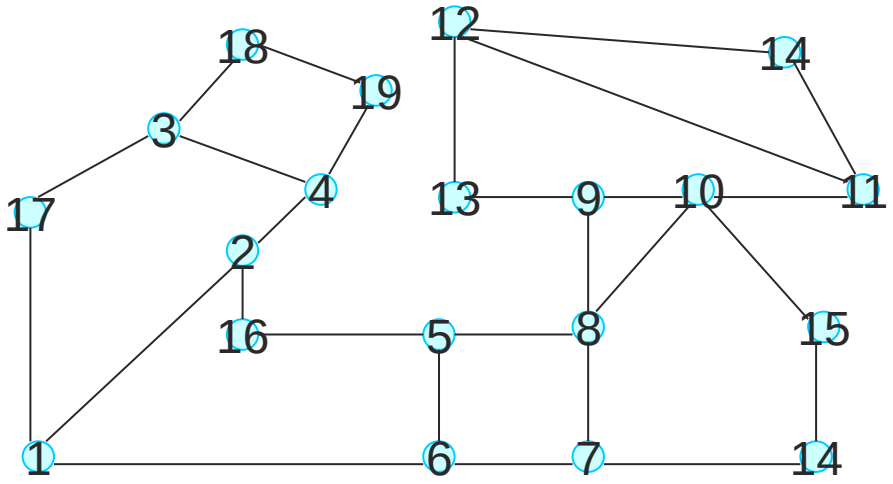
“Snow removal costs can be very high.
In Montreal the average cost of a 20 cm snow storm in 2010 was $17 million Canadian dollars.
Each year, the city has to clear 6,550 km of sidewalks and 4,100 km of streets.
On average, there are 65 weather events calling for response every winter.
Snow clearings performed in four stages: salting, plowing, removal, and disposal.
Plowing operations begin as soon as there is an accumulation of 2.5 cm of snow on the ground and continue as long as the storm lasts, ending about eight hours after the snow stops falling”.

Minimización de los puntos de perforación en corte con láser

Manber & Israni (1984) consideraron el problema de minimizar el número de puntos de perforación que se requiere en el proceso de corte con láser (y determinar el camino que debe sequir el taladro)

RPP no dirigido



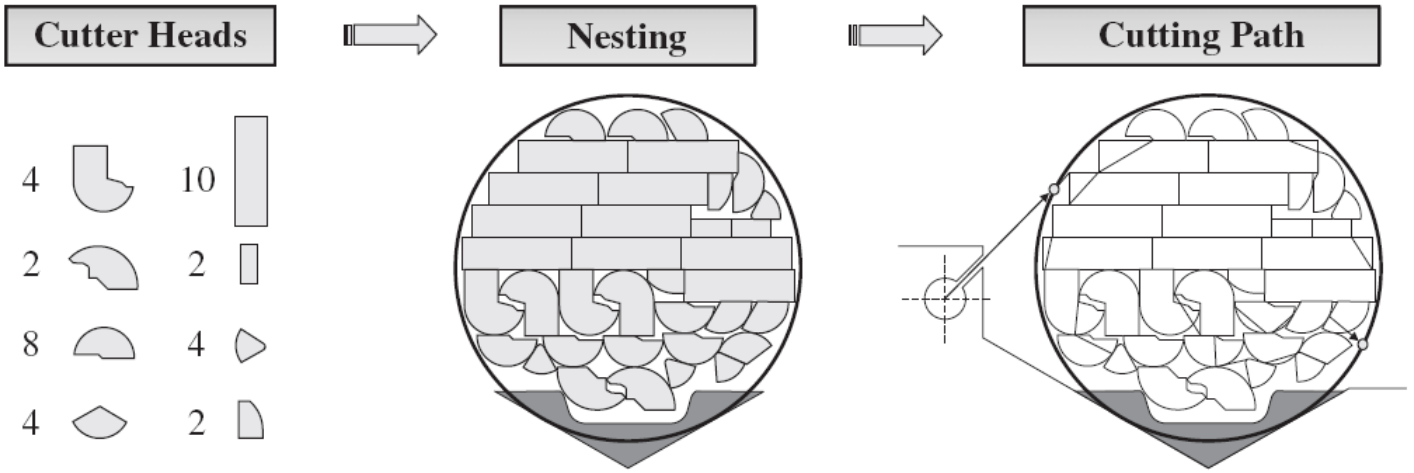


Problema del camino del corte

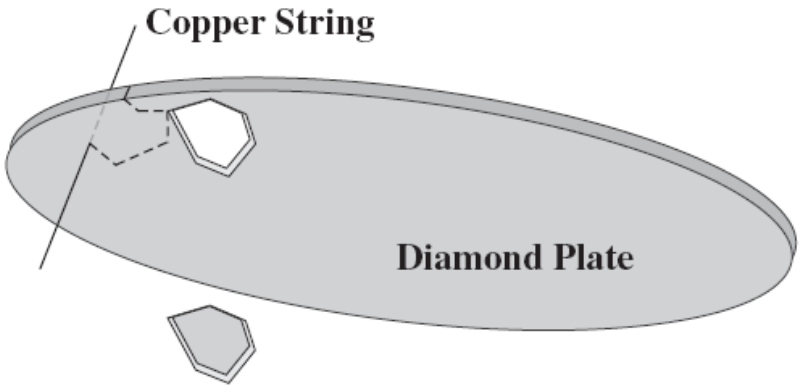
Moreira et al., 2007

Una gran compañía fabrica herramientas de gran precisión para cortar madera, plástico y otros materiales. El proceso de producción incluye el corte de cabezas que tienen que ser cortadas de caras placas circulares hechas de tungsteno con una delgada capa de diamante.

El problema consiste en encontrar el camino óptimo para el corte de las piezas.



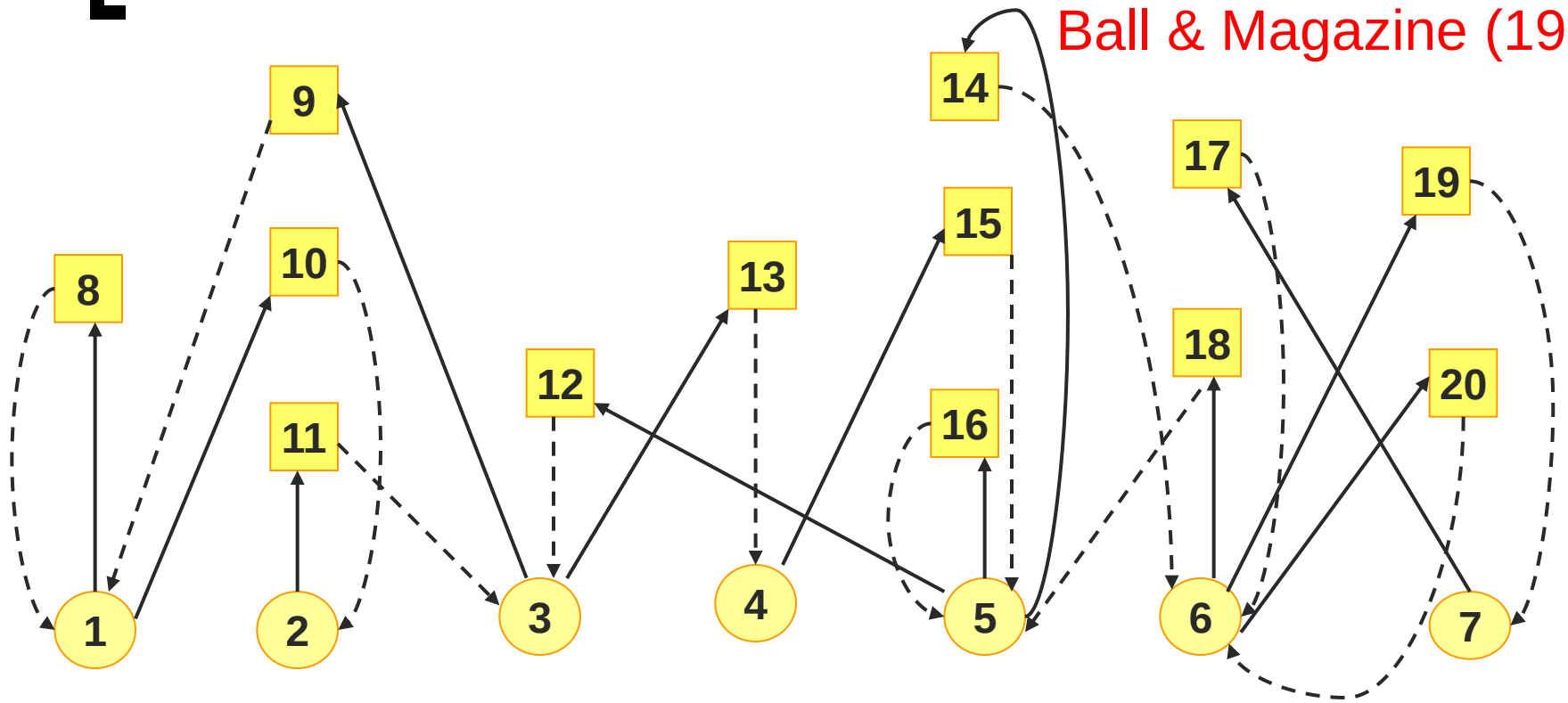
Problema del camino del corte



El corte se realiza por medio de una “cuerda de cobre electrificada”. Básicamente, la cuerda de cobre atraviesa el disco cortando las pequeñas piezas que van cayendo en un contenedor especial.

El disco tiene 10 cm de diámetro. La velocidad de la cuerda de cobre es constante: 1.5 mm/min.
Un disco cuesta aproximadamente 20h en ser cortado totalmente.

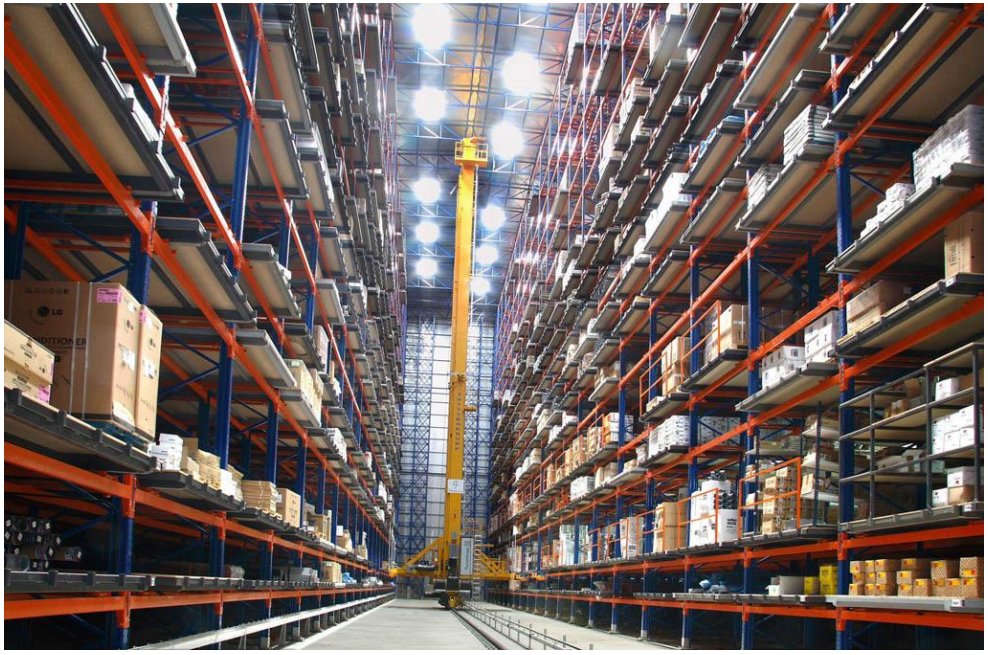
Secuencia de inserción óptima en circuitos impresos



El Problema de la Grúa (SCP)

Frederickson, Hecht & Kim (1978)

Una grúa debe, partiendo de una posición inicial, realizar una serie de movimientos y volver a la posición de partida. El objetivo es secuenciar los movimientos de la grúa de forma que se minimice el coste total.

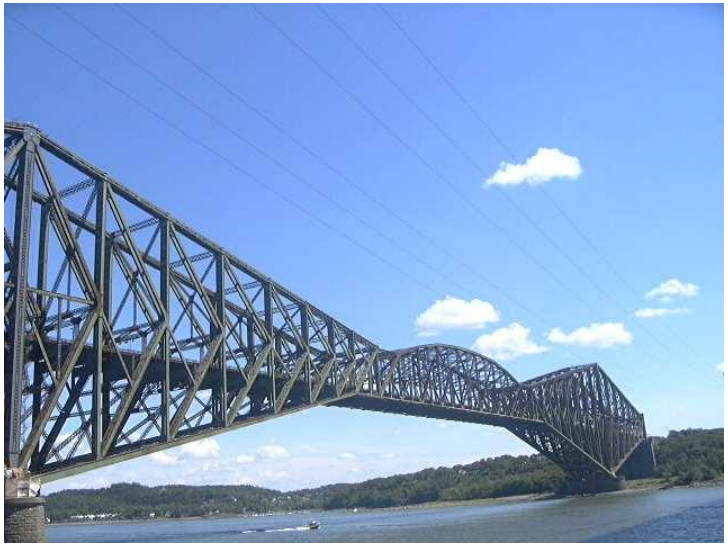


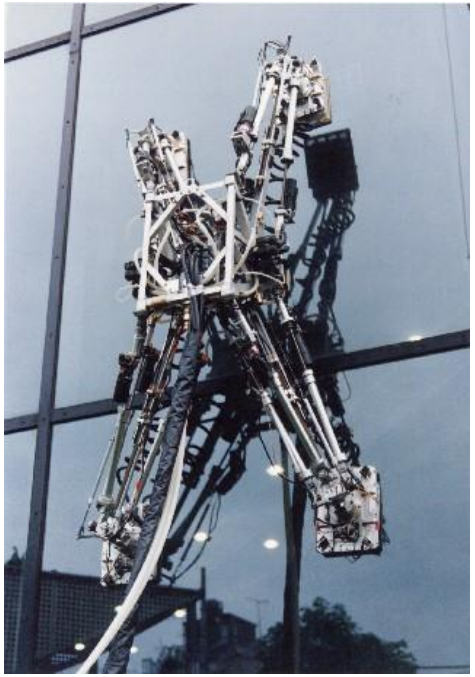
Inspección de estructuras 3D por robots escaladores

Un robot tiene que inspeccionar un conjunto de elementos de una estructura 3-D optimizando su consumo energético.

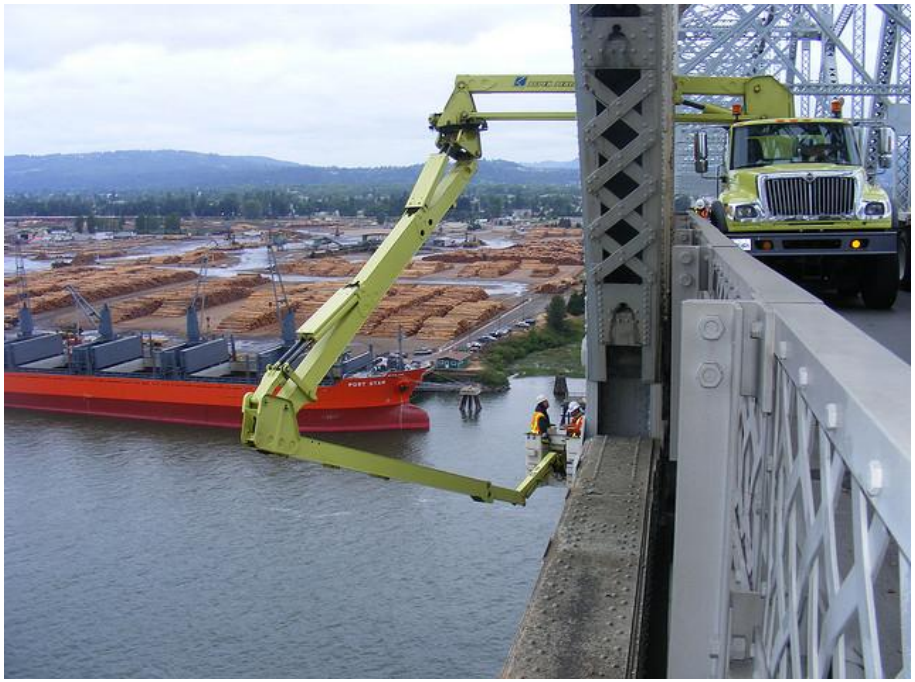
Queremos encontrar la ruta óptima para el robot:

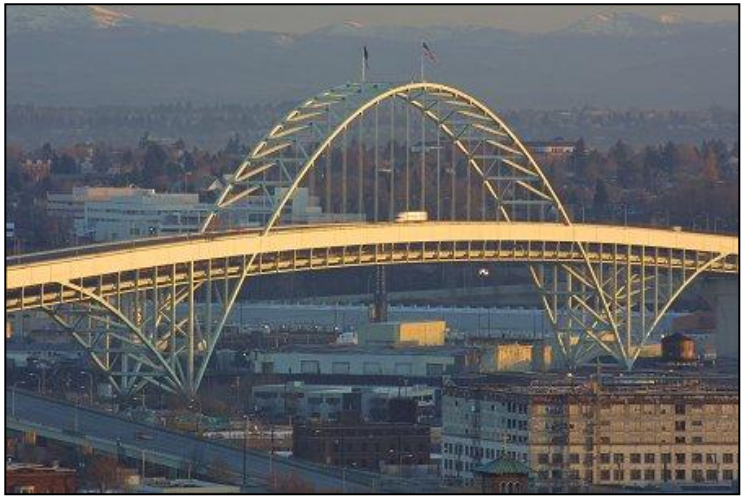
- Minimizando su consumo
- Maximizando su autonomía





Inspección de estructuras 3D por robots escaladores



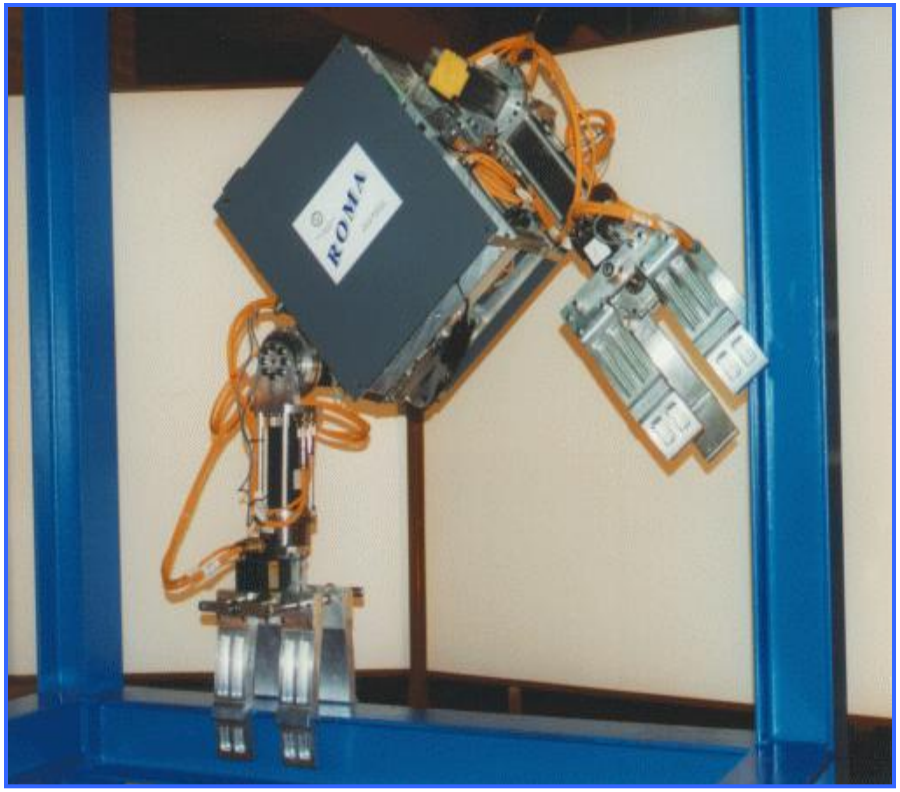


ROMA robot

(RObot Multifuncional Autoportante)

Area de Ingeniería de Sistemas y Automática de la Univ. Carlos III

- Autonomía: 3 horas
- Peso: 75 Kg
- Sistema de control inteligente (CPU, Ethernet por radio, cámara de TV, telémetro láser)



Modelización del problema

Queremos encontrar la ruta óptima para el robot:

- Minimizando su consumo
- Maximizando su autonomía

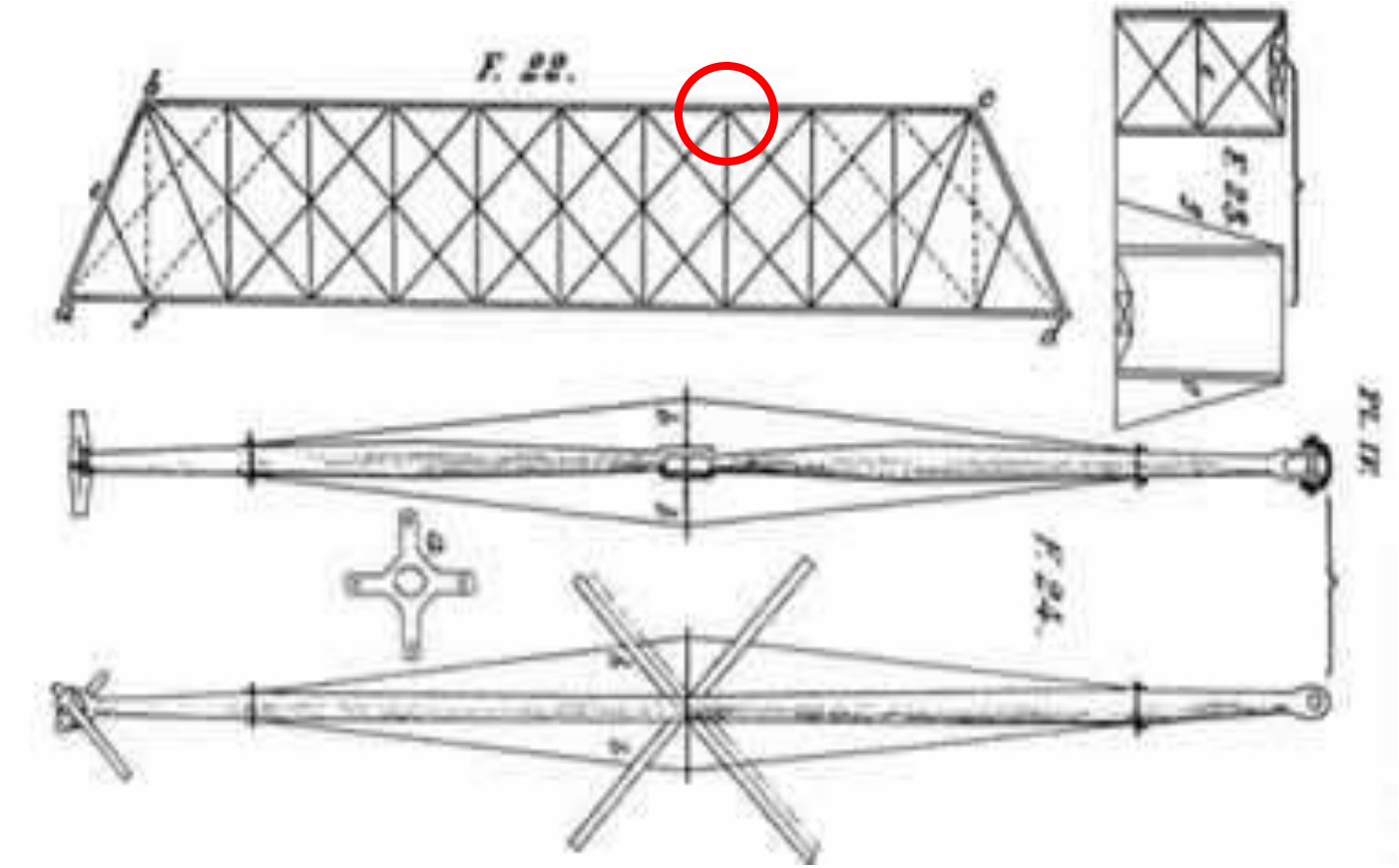
¿Qué necesitamos?

Información sobre el consumo energético del robot:

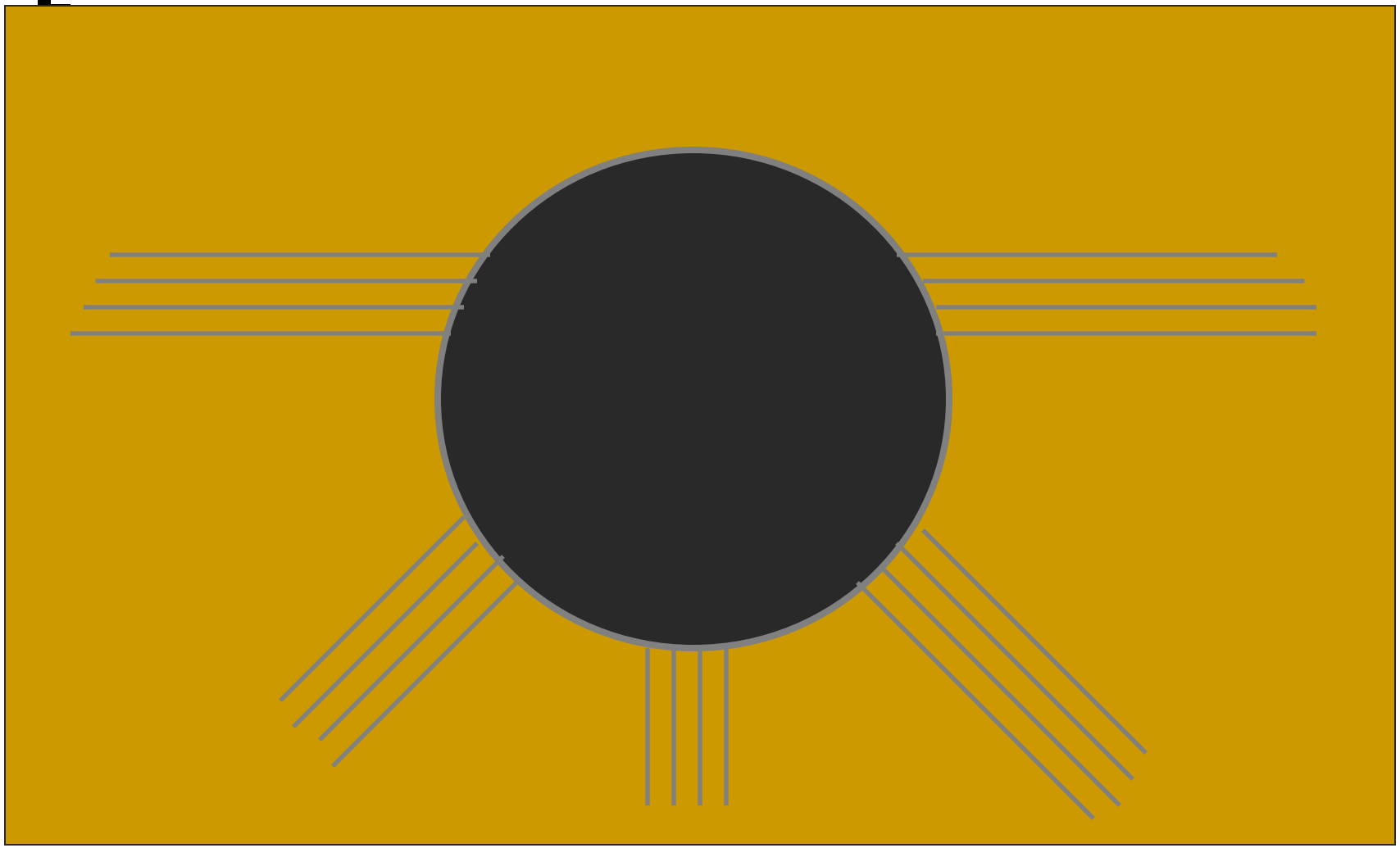
- Coste de recorrer un elemento (asimetría)
- Coste de atravesar una unión de vigas (asimetría)

Modelización de las uniones

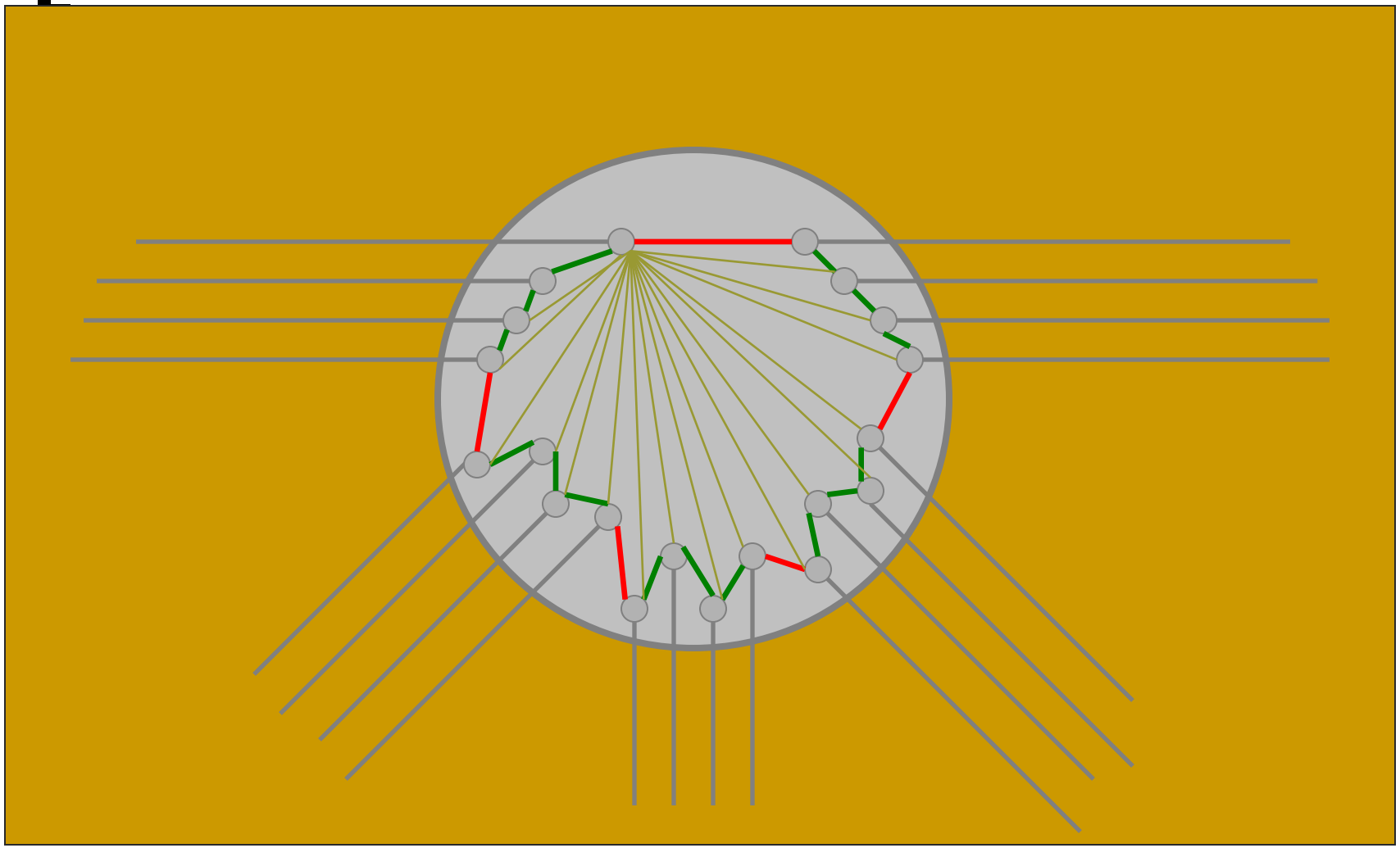
Modelización de las uniones



Modelización de las uniones



Modelización de las uniones



Características del modelo

- Grafo no dirigido con costes asimétricos
- Aristas “requeridas” y “no requeridas”

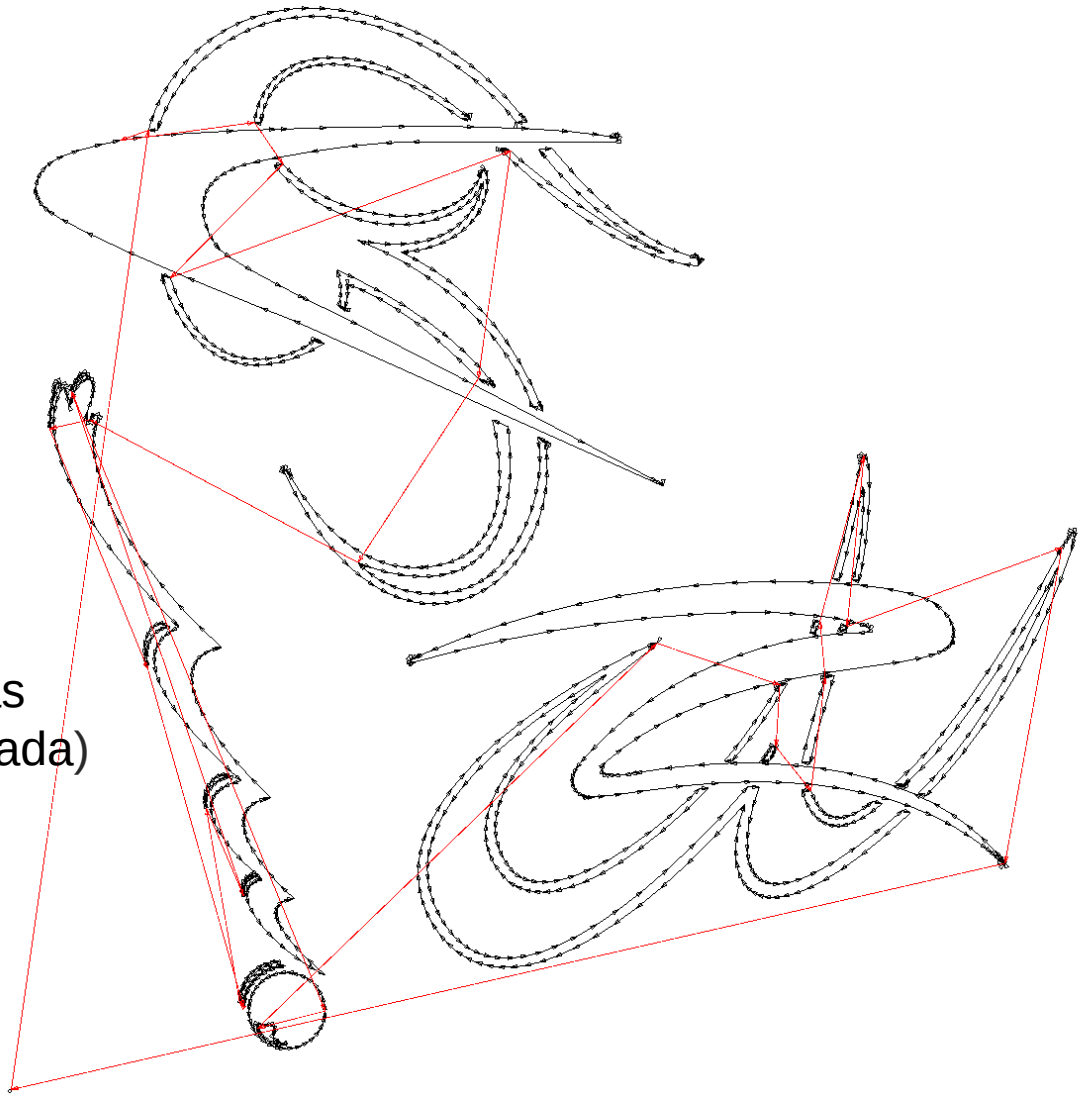
Windy Rural Postman Problem

Plotter de corte

Pegatinas

Plotter de corte

El diseño consiste en 'vectores‘ que necesitan ser cortados con la cuchilla bajada.



Flechas negras : aristas a recorrer (cortar)

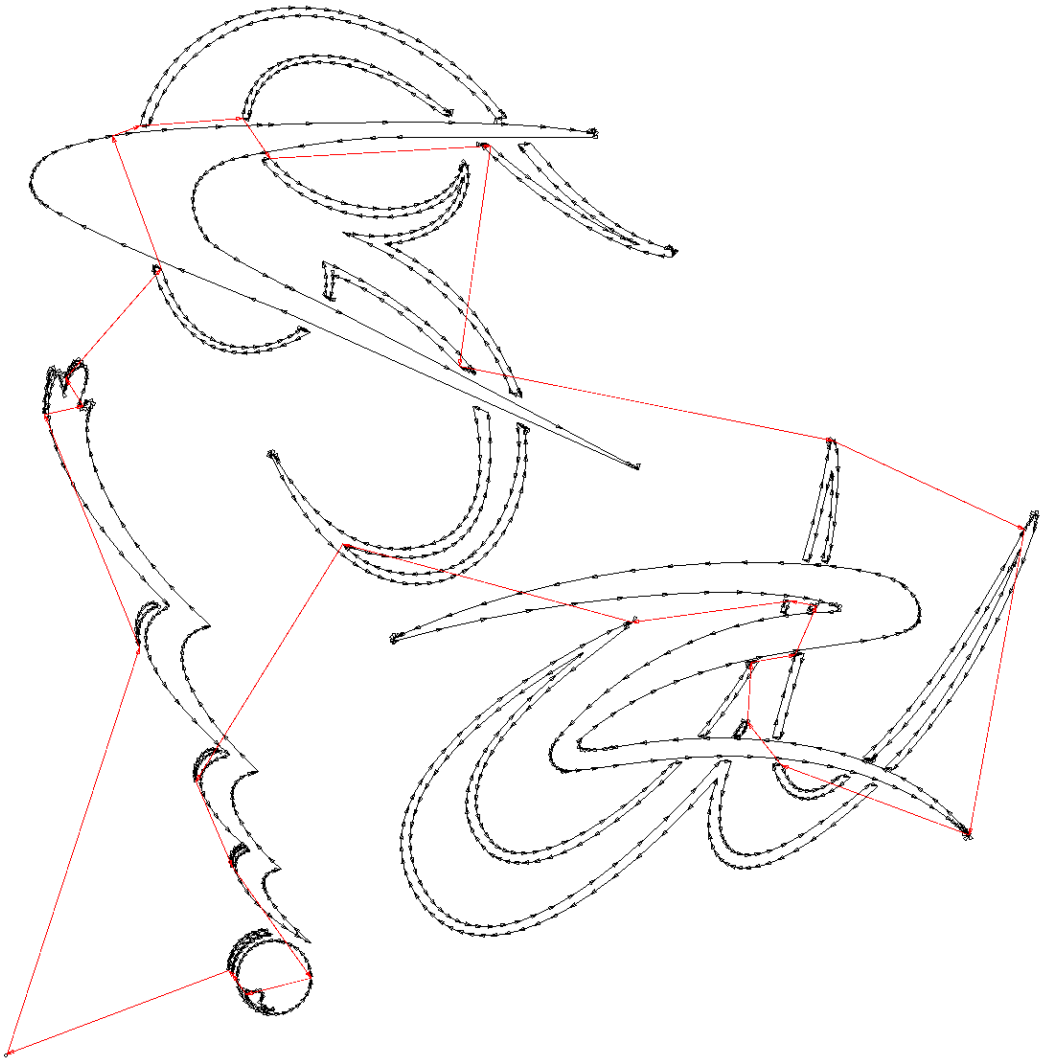
Flechas rojas : aristas no requeridas (movimientos con la cuchilla levantada)

Up time: 82564.29

Down time: 204545.60

Plotter de corte

Para algunos materiales hay una dirección de movimiento preferida (u obligada) para esos vectores (estirar el material en lugar de empujarlo).
Este es el aspecto 'windy‘ del problema.



Up time: 48520.55

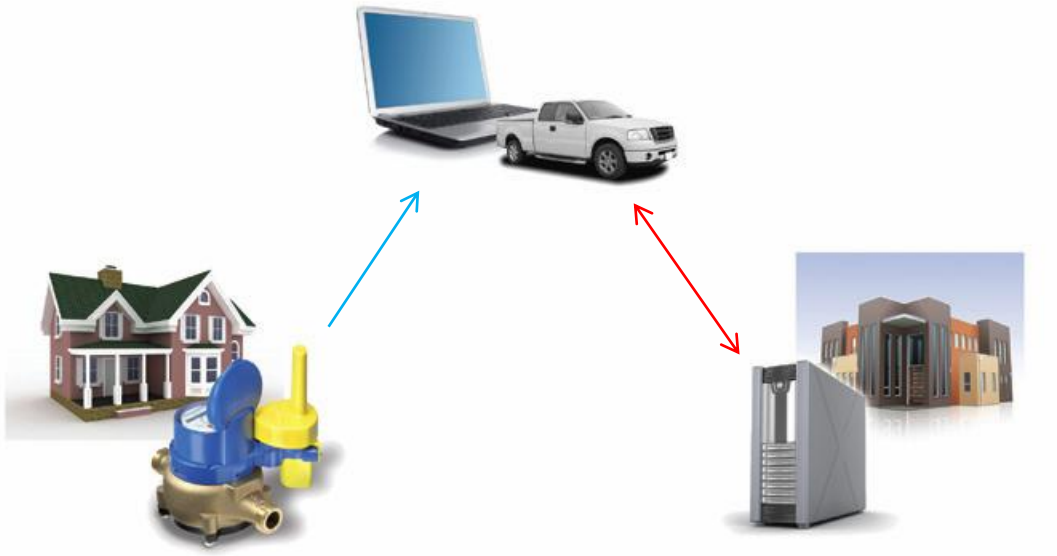
Down time: 204545.60

El Problema del Cartero Chino con beneficio máximo (MBCPP)

- En lugar de costes asociados con el recorrido de las calles, se pueden obtener beneficios al servirlas.
- Algunas calles necesitan ser servidas más de una vez.
- Sin embargo, los beneficios obtenidos pueden ser distintos.
- Ejemplo: retirada de la nieve.

El DRPP Generalizado (GDRPP)

Para leer a distancia la información de un contador sobre el consumo de un usuario, se instala en los vehículos un aparato de lectura que recoge los datos enviados por los contadores.
El lector no necesita visitar a cada usuario para recoger la información sino entrar en el rango de lectura del dispositivo.
GDRPP: Construir las rutas de coste mínimo para la lectura de los contadores.



Resolución exacta: la combinatoria poliédrica

Combinatoria poliédrica

- La mayoría de los problemas de rutas por arcos se pueden formular como PLE's.
- PLE es un problema NP-difícil, mientras que PL es resoluble en tiempo polinómico.
- Diversos métodos exactos han sido propuestos para la resolución de los PLE's:
  - programación dinámica (Bellman, 1957)
  - planos de corte (Gomory, 1958)
  - branch & bound (Land y Doig, 1961)
  - relajación lagrangiana (Geoffrion, 1974)

Entre ellos, los planos de corte y el b&b son los más usados

Combinatoria poliédrica

- En los 70’s se empezaron a buscar planos de corte más fuertes, que funcionaran mejor que los de Gomory.
- En los 80’s se obtuvieron buenos resultados “cortando primero” y “ramificando después”.
- En los 90’s se obtuvieron mejores resultados aún “cortando en cada nodo del árbol de b&b” (branch & cut).
- Branch & cut funciona bien. Es el método utilizado ahora por casi todos los paquetes de software de PLE.

Combinatoria poliédrica

- ¿Qué es un plano de corte fuerte?
- La respuesta está en estudiar el poliedro asociado a las soluciones del PLE.
- La idea apareció en trabajos de Balinski, Gomory y Edmonds.
- Después, Chvàtal, Padberg, Balas y otros desarrollaron y popularizaron el concepto: Combinatoria Poliédrica.

Combinatoria poliédrica

PLE: Max $\{cx: Ax \leq b, x \in Z_+^n\}$

La región posible de su RL es:

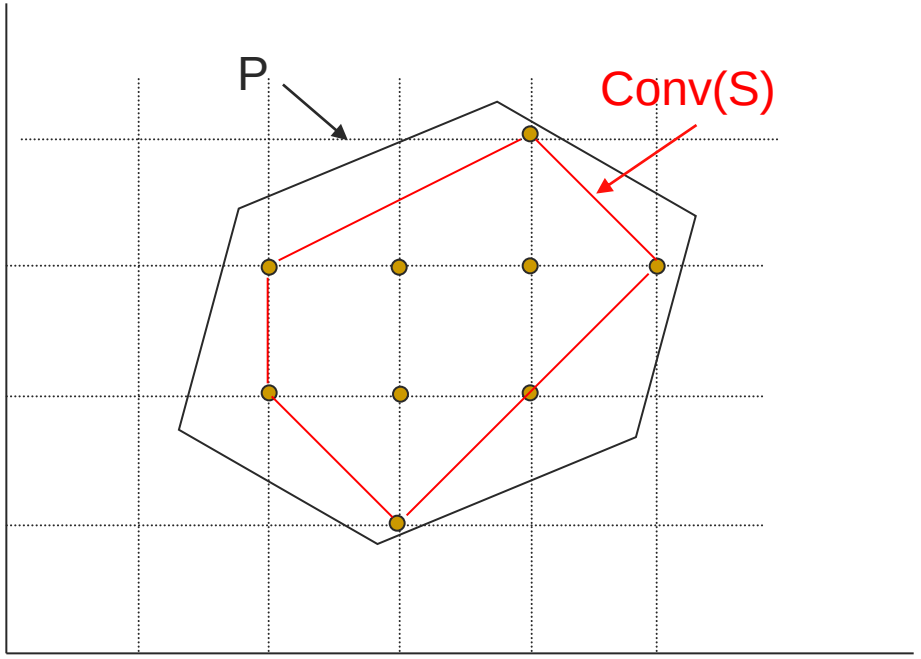
$P=\{x \in R_+^n : Ax \leq b\}$

El cjto de sol. posibles es:

$S=\{x \in Z_+^n : Ax \leq b\}$

La envoltura entera es: conv(S)

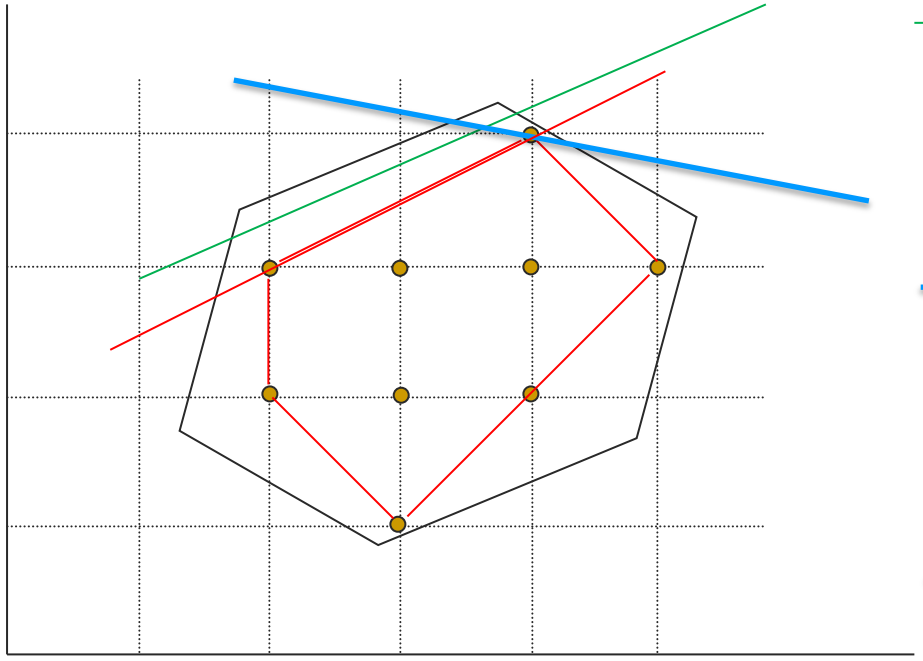
Obviamente: $conv(S) \subseteq P$.



Un plano de corte es una desigualdad que es válida para conv(S) pero no para P.

Los planos de corte más fuertes posible son aquellos que definen facetas de conv(S).

Combinatoria poliédrica



Un plano de corte débil

Un plano de corte más fuerte

El plano de corte más fuerte posible: define una faceta de conv(S)

Combinatoria poliédrica

- Desafortunadamente, el número de facetas suele crecer muy rápidamente con n.
- No podemos calcularlas todas.
- Sin embargo, para la mayoría de los problemas de optimización combinatoria que son resolubles polinómicamente, conocemos todas sus facetas.
- Para muchos problemas NP-difíciles conocemos muchas de sus facetas.
- No necesitamos conocer todas las facetas para resolver una instancia de un problema.
- Incluso si una desigualdad no define faceta puede ser un plano de corte útil.

Combinatoria poliédrica

PLE: Max $\{cx: x \in S\}$ es equivalente a

Max $\{cx: x \in conv(S)\}$,

pero éste es un problema que puede ser resuelto como un PL si

conocemos la descripción de conv(S) (o una parte de ella)

Pero, ¿cómo tratamos el número exponencial de desigualdades que podemos conocer?

Algoritmo de planos de corte

1. Sea (PL_0) el PL inicial. Hacer k = 0
2. Resolver (PL_k). Sea x^k su solución.
3. Si x^k es una solución possible del PLE: es óptima. Parar.
4. Encontrar desigualdades válidas para conv(S) que estén violadas por x^k (Problema de Separación)
    1. Si encontramos algunas, añadirlas a (PL_k) para obtener (PL_{k+1}). Hacer k := k+1 e ir a 2.
    2. En otro caso, Parar.

Algoritmo de planos de corte

Si el algoritmo acaba en (3): Solución óptima

Si acaba en (4.2) es porque o bien:

- no conocemos todas las desigualdades que definen conv(S), o (y)
- no sabemos cómo encontrar aquellas que están violadas por la solución lineal.

Entonces,

- cx^k es una (buena) cota superior y podemos ir a “ramificar”.

Ejemplos: B&C para el WPP

Win (1987), Grötschel & Win (1988):

WPP instancias con $|V| \in (52,264)$ y $|E| \in (78,479)$: 31/36

C., Oswald, Plana, Reinelt & Sanchis (2011):

WPP instancias con $|V| \in (500,3000)$ y $|E| \in (813,9085)$: 99/120.

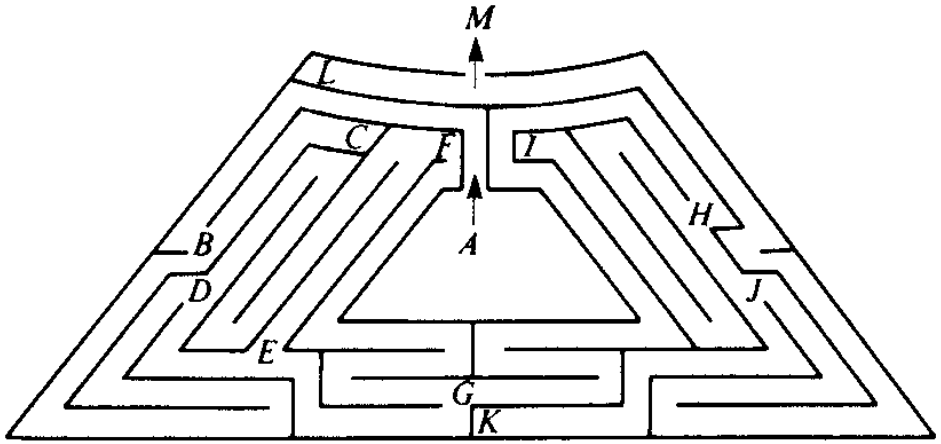
B&C para el RPP

El Branch-and-Cut de C., Plana and Sanchis (2007) es capaz de resolver instancias del RPP con hasta 1000 vértices, 3080 aristas y 200 R-sets en, aproximadamente, 1 hora de CPU en un Pentium IV a 2.8 GHz (sin utilizar un heurístico que proporcione buenas soluciones posibles iniciales)

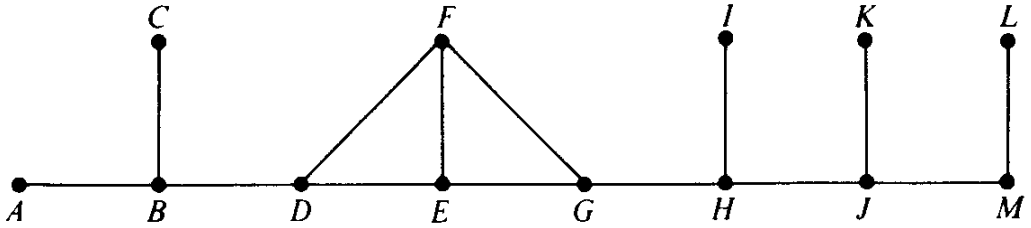
Gracias por su atención.

El laberinto de Hampton Court

Supongamos que tenemos un mapa del laberinto:



Podemos representarlo por medio de un grafo que muestre las opciones disponibles en cada intersección:



Laberintos

Desde (A) podemos alcanzar la salida (M) siguiendo el camino A B D E G H J M.

Pero, buscamos un camino que desde el centro de cualquier laberinto llegue a la salida.

¿Qué pasaría si encontráramos un camino que recorriera cada arista del grafo asociado?

Ese camino pasaría por los puntos de entrada y salida!!!

... Y siempre podemos encontrar un camino así.

Laberintos

Duplicando cada arista del grafo que representa el laberinto, obtenemos un nuevo grafo con todos sus vértices de grado par. Este nuevo grafo es Euleriano y, por lo tanto, tiene un tour que atraviesa cada arista original dos veces, lo que es suficiente para nuestros propósitos.

Obviamente nuestro argumento lo es de existencia

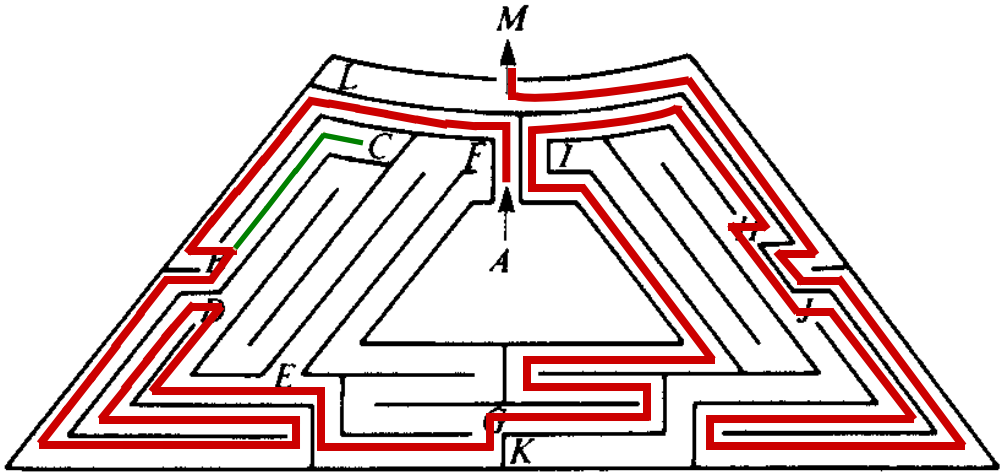
(pero no proporciona un método práctico para escapar de un laberinto).

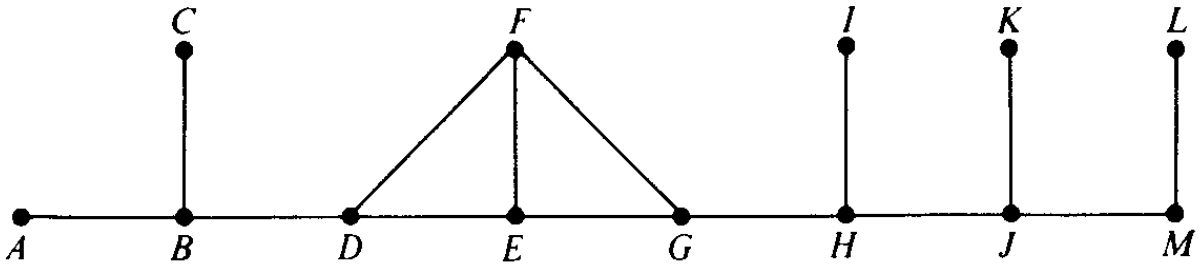
Laberintos

En 1895, Gaston Tarry publicó el mejor método para escapar de un laberinto. Está basado en la regla siguiente:

"*never traverse again a passage that took you to an intersection for the first time, unless there is no other alternative".*

Laberintos





Laberintos

Siguiendo la regla de Tarry en cada intersección, escaparemos de cualquier laberinto atravesando cada pasadizo a lo sumo 2 veces (una en cada dirección).

El único problema es “reconocer” cuál, entre los pasadizos que llevan a una intersección, es el que nos llevó allí por primera vez.

Pero Tarry también resolvió este problema:

Laberintos

1. Cuando atravieses un pasadizo por primera vez, pon 2 guijarros a su entrada y 1 o 3 guijarros a su salida, dependiendo de si habías cruzado antes, o no, por esa intersección.
2. Al entrar en un pasadizo con 1 guijarro a la entrada, pon otro guijarro ene el mismo sitio.

Así …

Laberintos

0 guijarros: Este pasadizo no ha sido atravesado en ninguna dirección. Podemos utilizarlo.

1 guijarros : El pasadizo ha sido atravesado hacia la intersección. Puede usarse en la dirección opuesta.

2 guijarros : El pasadizo ha sido recorrido desde esta intersección. No puede utilizarse en esta dirección de nuevo.

3 guijarros : Éste fue el pasadizo que nos trajo a esta intersección por primera vez. No debemos usarlo de nuevo a menos que no haya otros pasadizos con 0 o 1 guijarros.

Laberintos