Cómo Programar las Vacaciones: El Problema del Viajante

Javier Soria

Universidad de Barcelona

[Universidad de La Rioja, 1 de junio de 2007]

Resumen: Buscamos cómo optimizar (tiempo, distancia, coste económico), un recorrido cíclico, que enlace una serie de puntos prefijados.

¿Qué ruta es la mejor para visitar todas las bodegas riojanas, o los Paradores Nacionales, o los castillos Cátaros de Francia, ...?

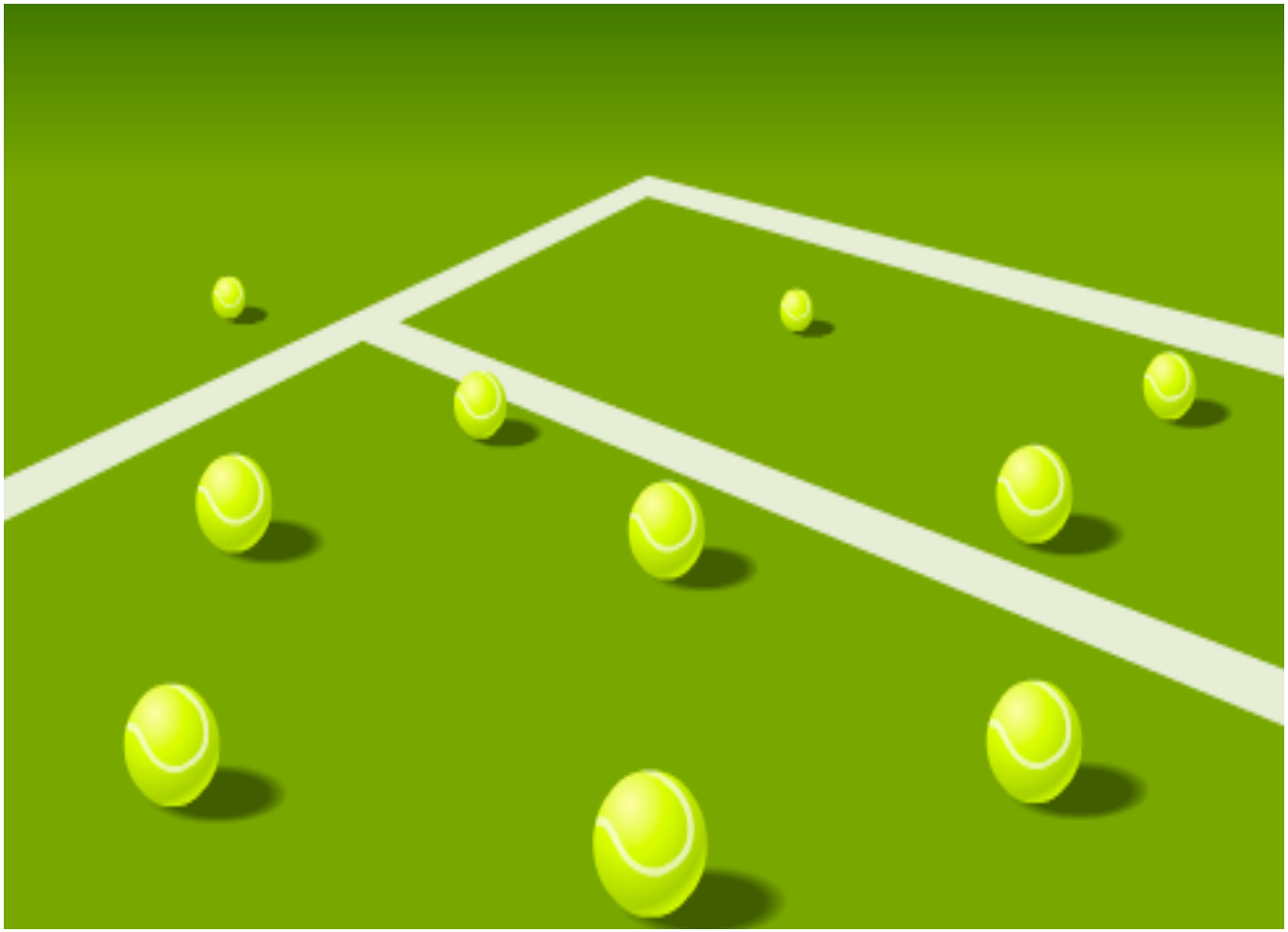
¿Cómo hemos de abrocharnos las zapatillas para usar el cordón más corto?

- **En las próximas vacaciones queremos hacer un recorrido turístico en coche por todas las capitales de las provincias limítrofes con La Rioja, o por todas las de la España peninsular, o por todas las capitales europeas, de manera que el tiempo que empleamos en viajar sea el menor posible.**

Recorrido con 6 ciudades.

Recorrido con 47 ciudades.

- **Supongamos que hemos de visitar a los clientes de nuestra empresa, para venderles nuestro nuevo producto, y queremos hacerlo de manera que el coste total del viaje sea el menor posible.**



¿Cuál es el recorrido más corto para recoger todas las pelotas?

¿Qué significa encontrar la solución de un problema que sólo tiene un número finito de casos?

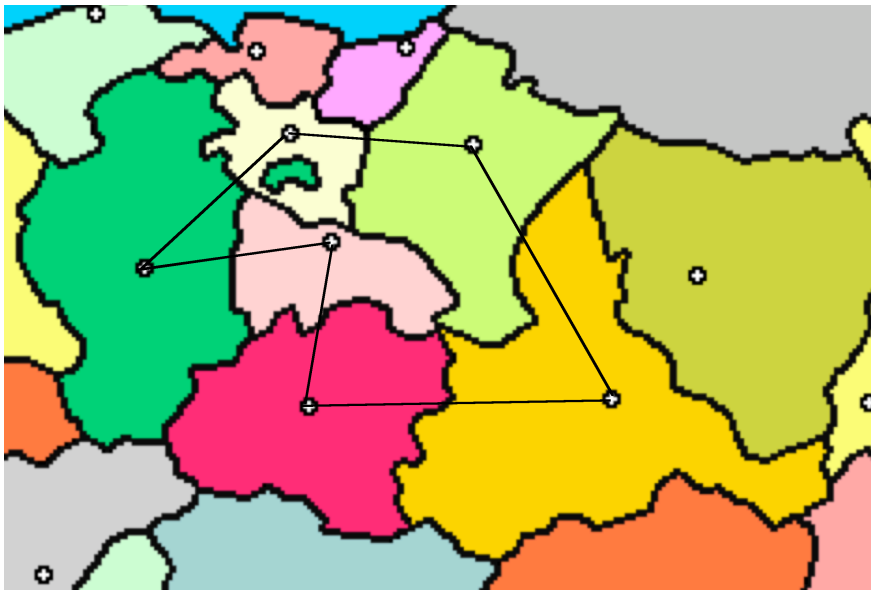
Método exhaustivo: Lo más natural sería escribir una lista con todas las posibilidades, y elegir la mejor...

En el caso del Problema del Viajante, con sólo 50 ciudades, si recubriéramos la Tierra con los ordenadores más rápidos que existen actualmente, con una velocidad de procesamiento de

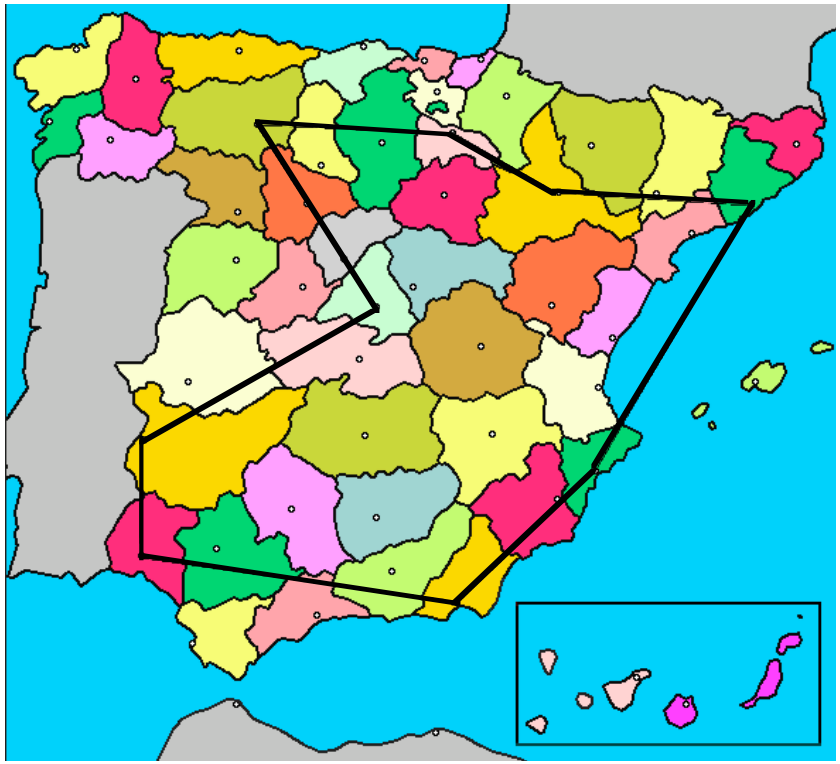
10^{36} = un billón de billones de billones de operaciones por segundo,

y suponiendo que cada uno ocupara una superficie de un kilómetro cuadrado, se necesitaría un tiempo comparable a la edad del Universo (quince mil millones de años) para poderlo calcular.

Algunos ejemplos usando el método exhaustivo:

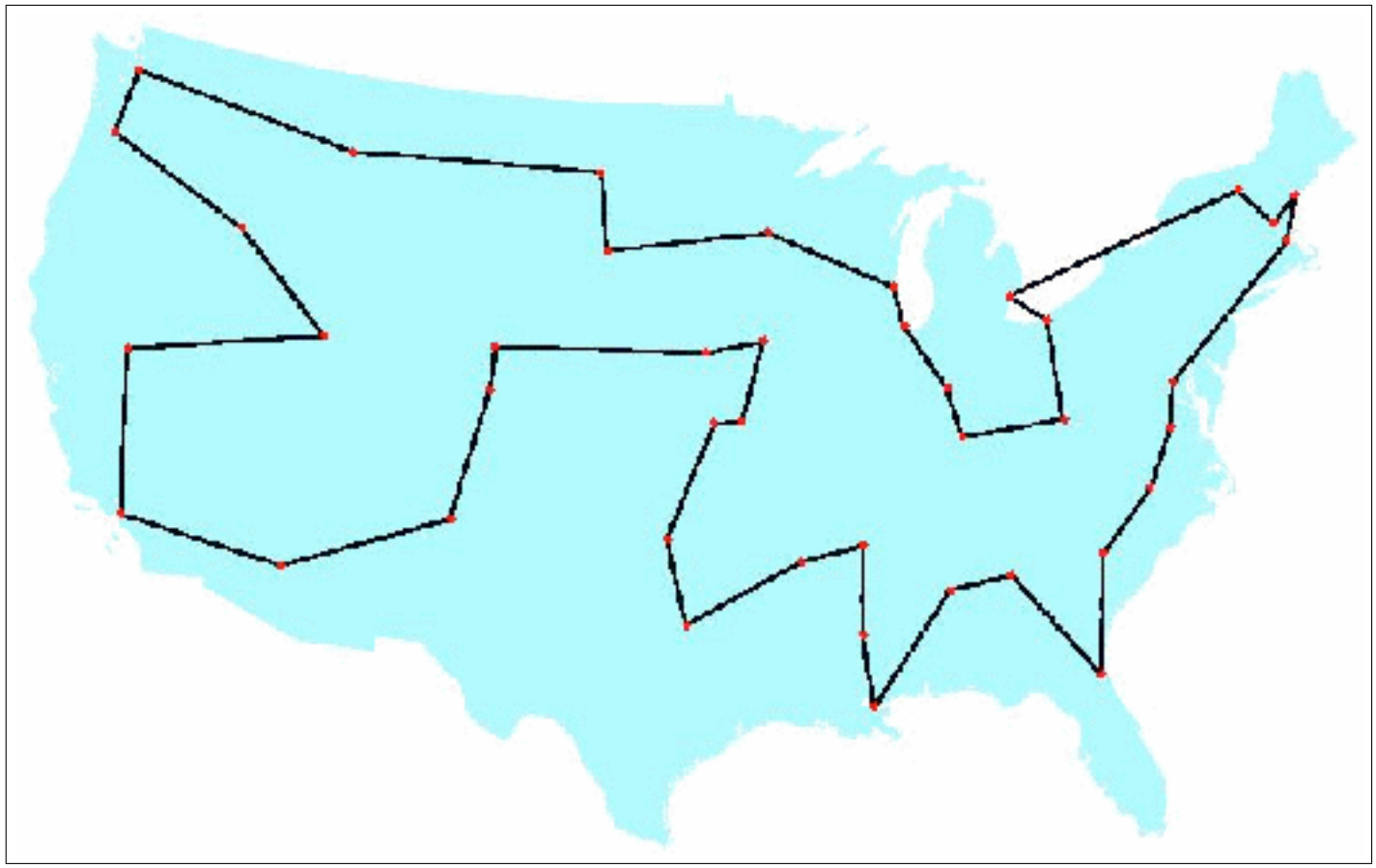


Logroño - Burgos - Vitoria - Pamplona - Zaragoza - Soria - Logroño
Longitud: 759 Km
Tiempo de cálculo: 0,2 segundos (iMac PowerPC G4 700MHz)
Número de ciudades: 6. Número de casos: 60.

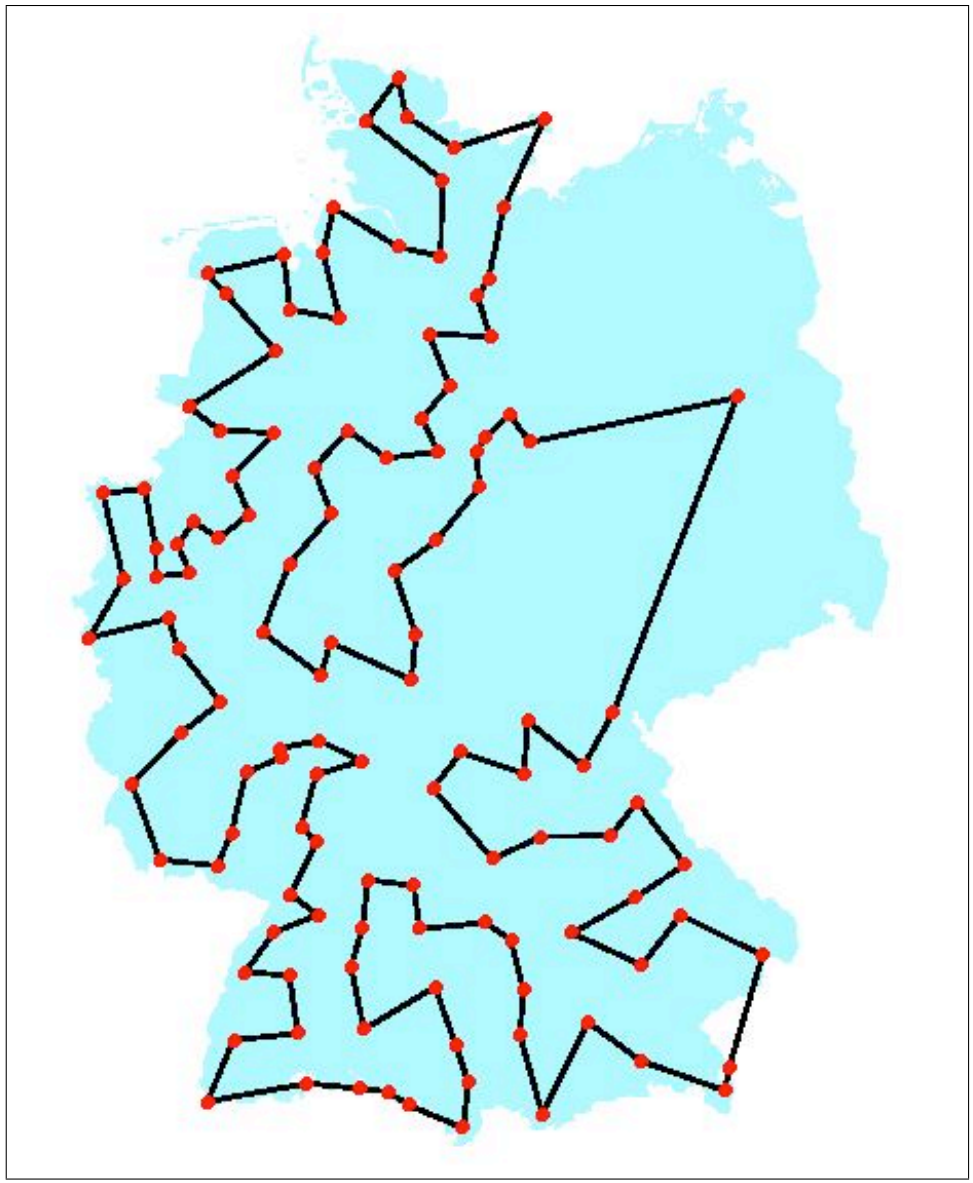


Logroño - León - Madrid - Badajoz - Huelva - Almería
Alicante - Barcelona - Zaragoza - Logroño
Longitud: 3.167 Km
Tiempo de cálculo: 1 minuto y 50 segundos (iMac PowerPC G4 700MHz)
Número de ciudades: 9. Número de casos: 20.160.

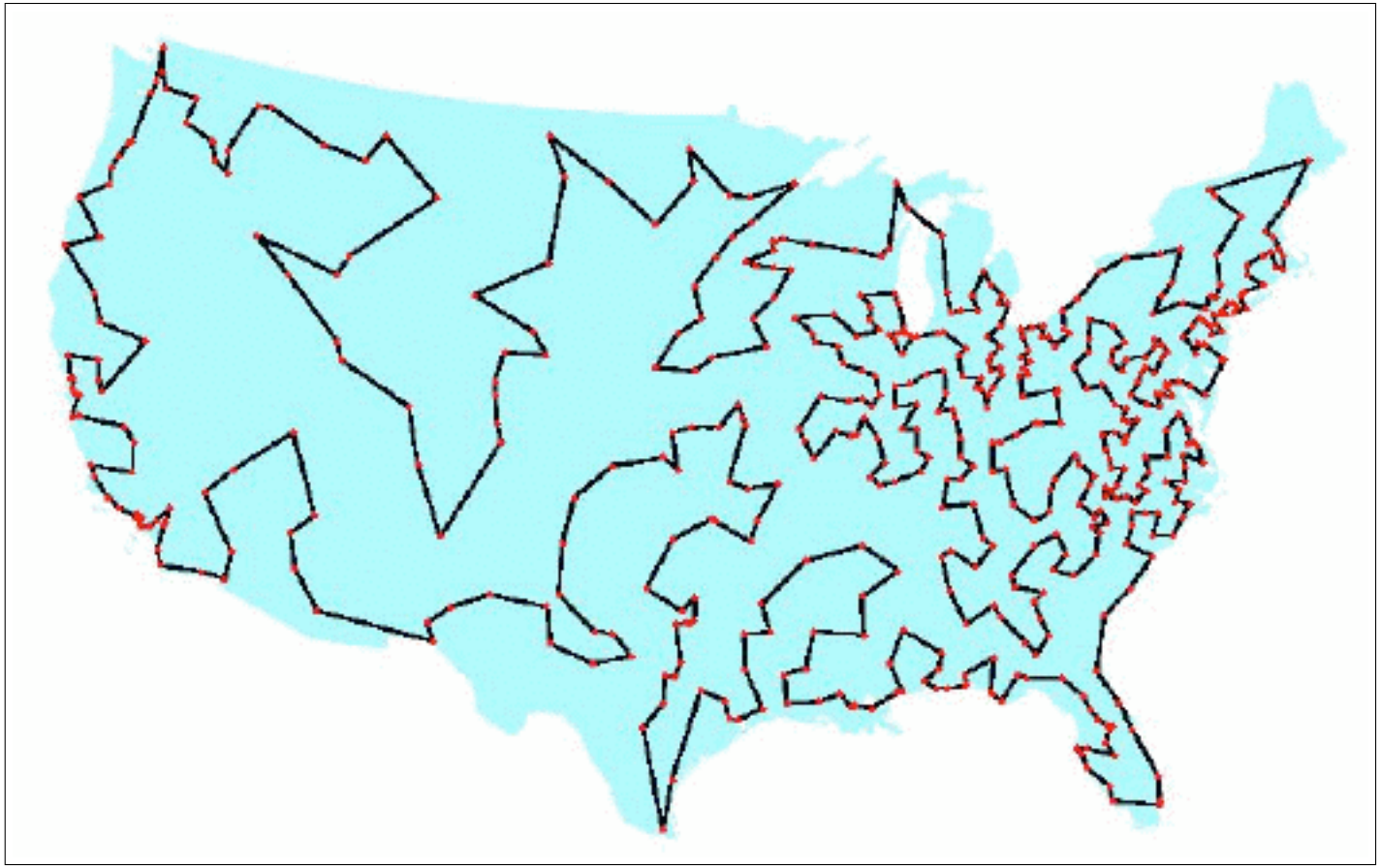
Hitos Históricos.



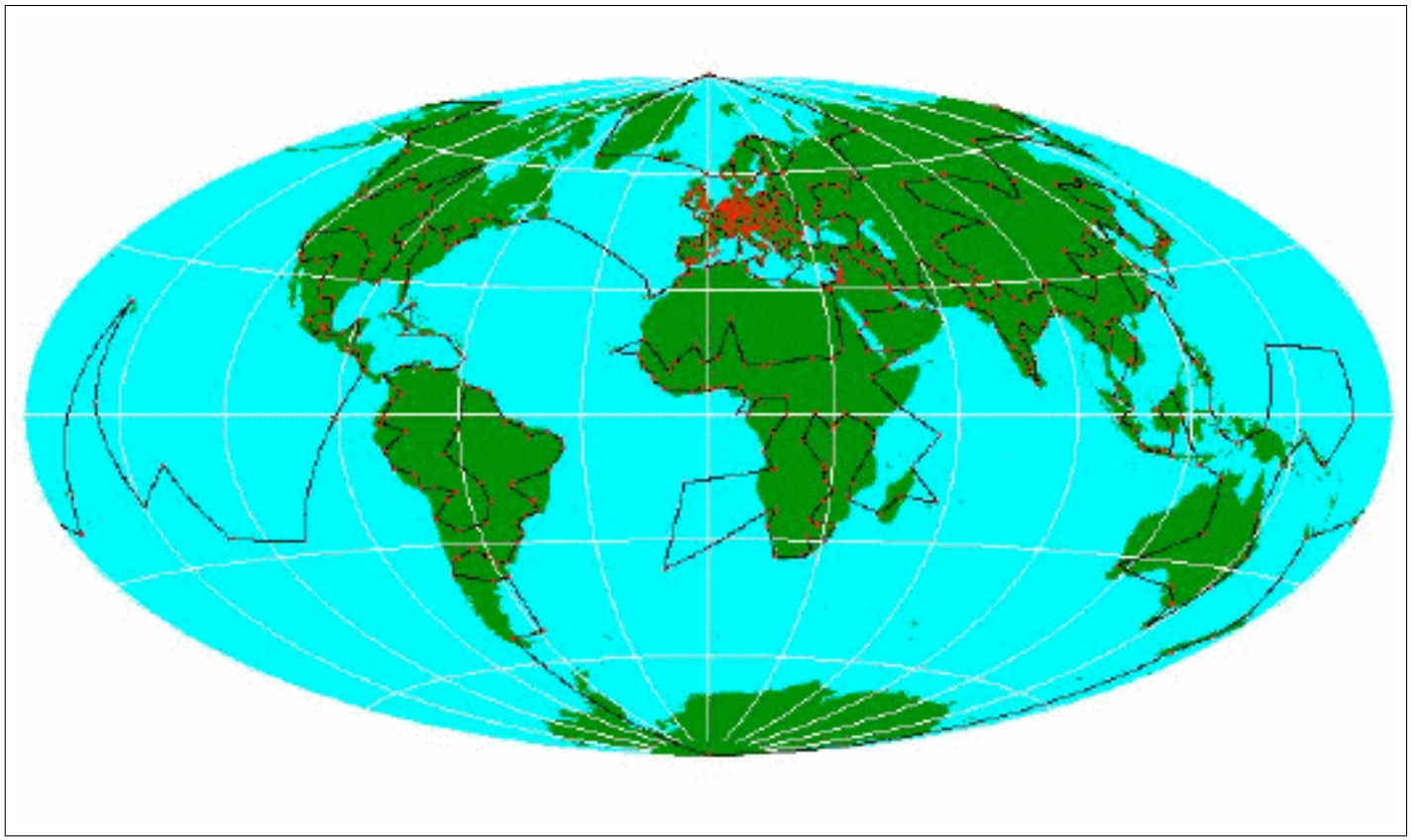
1954 (G. Dantzig, R. Fulkerson y S. Johnson): 49 ciudades de EEUU.



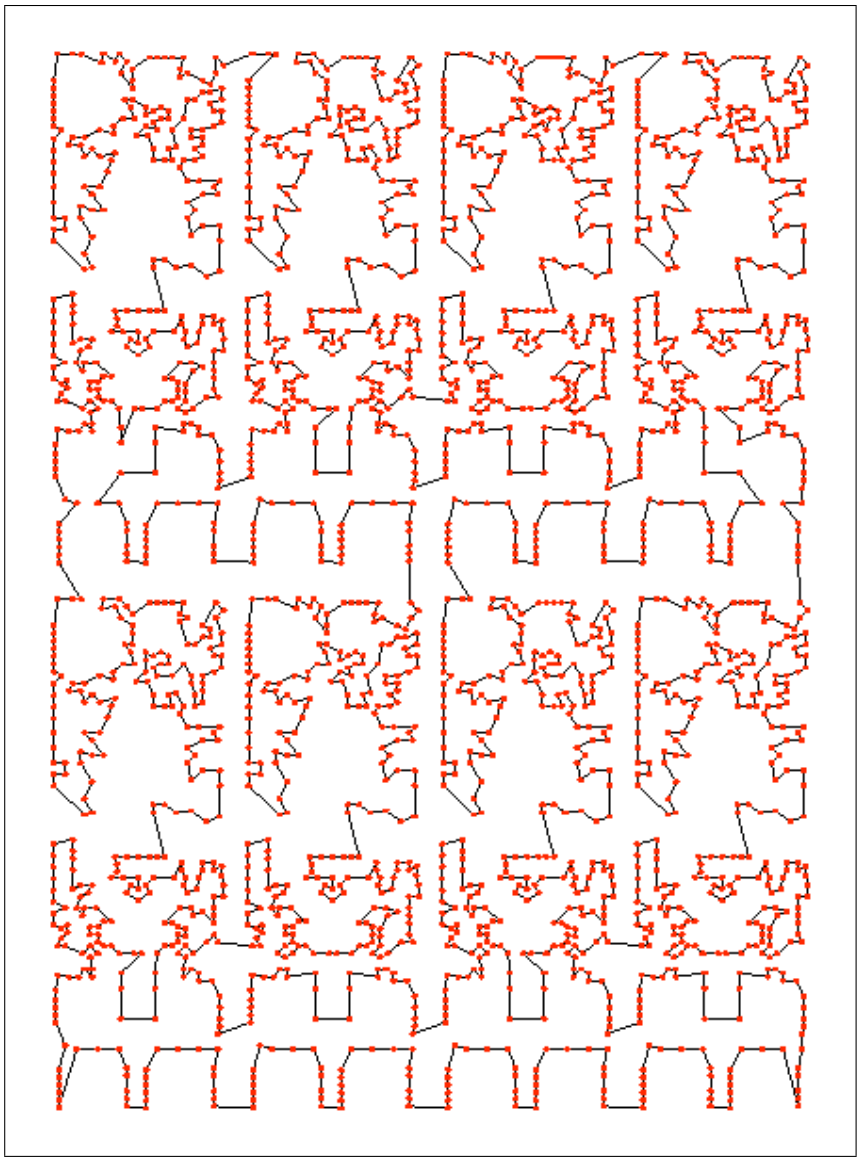
1977 (M. Grötschel): 120 ciudades de Alemania.



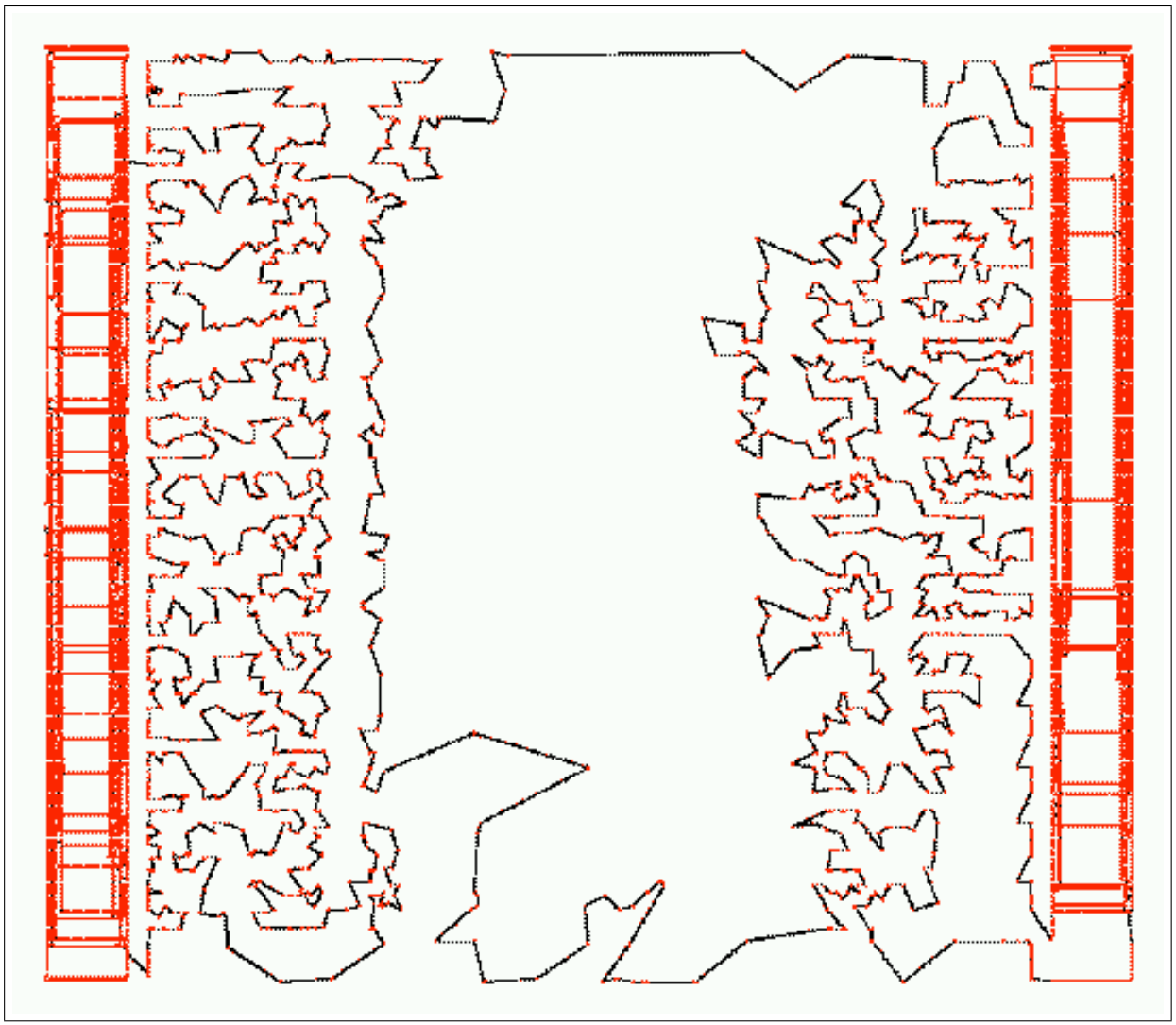
1987 (M. Padberg y G. Rinaldi): 532 empresas de AT&T en EEUU.



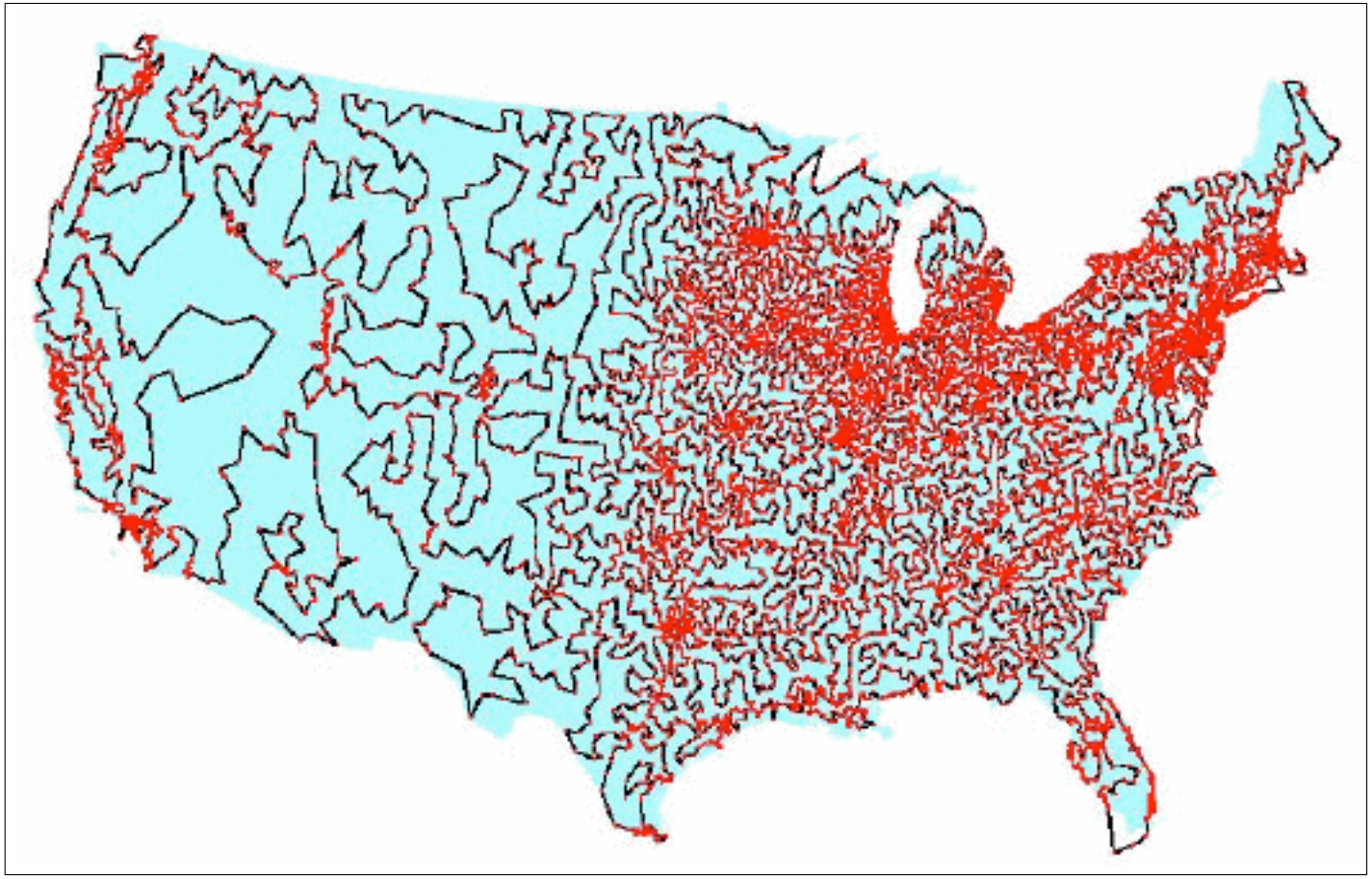
1987 (M. Grötschel y O. Holland): 666 lugares de interés en el mundo.



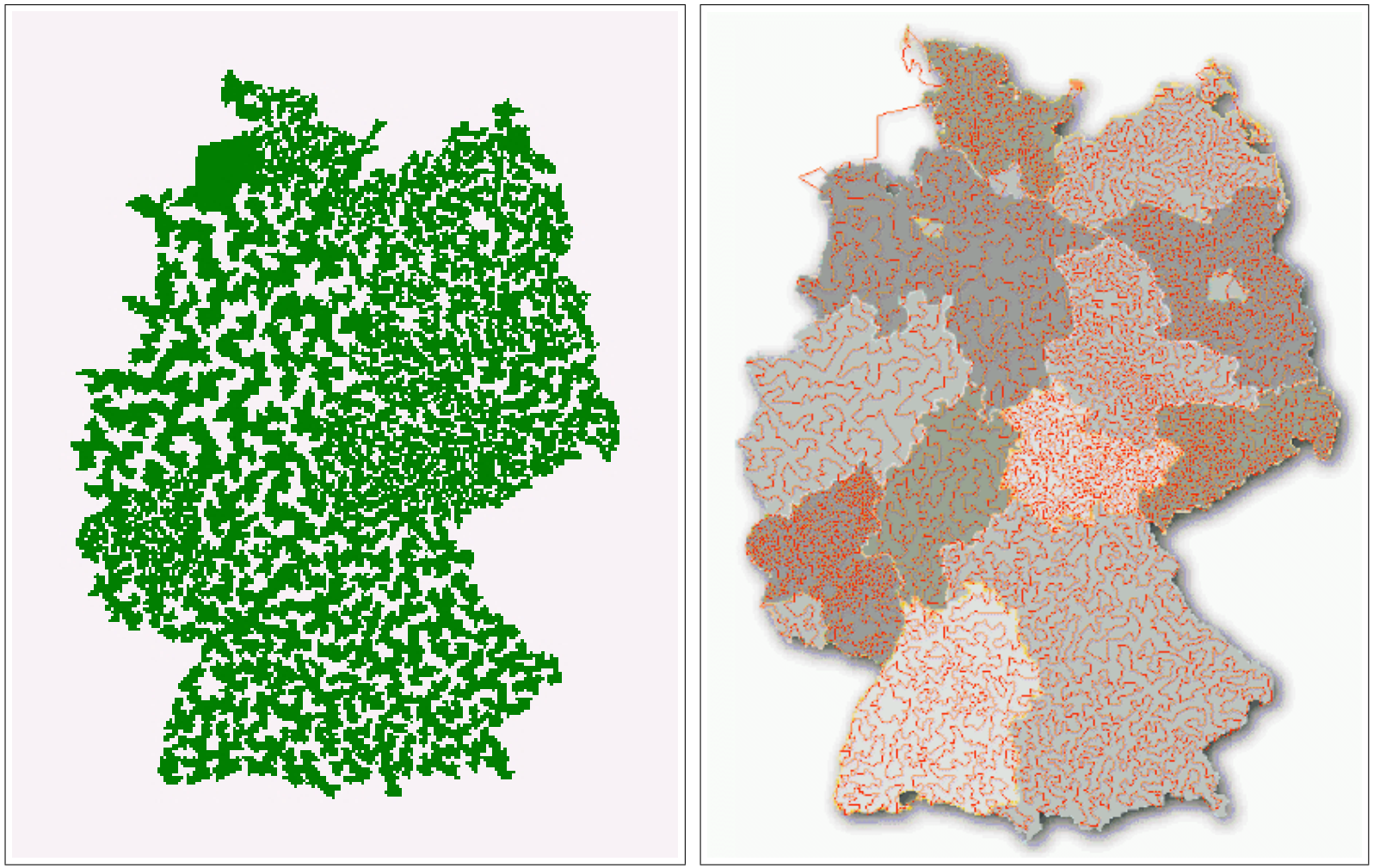
1987 (M. Padberg y G. Rinaldi): 2.392 puntos (Tektronics Incorporated).



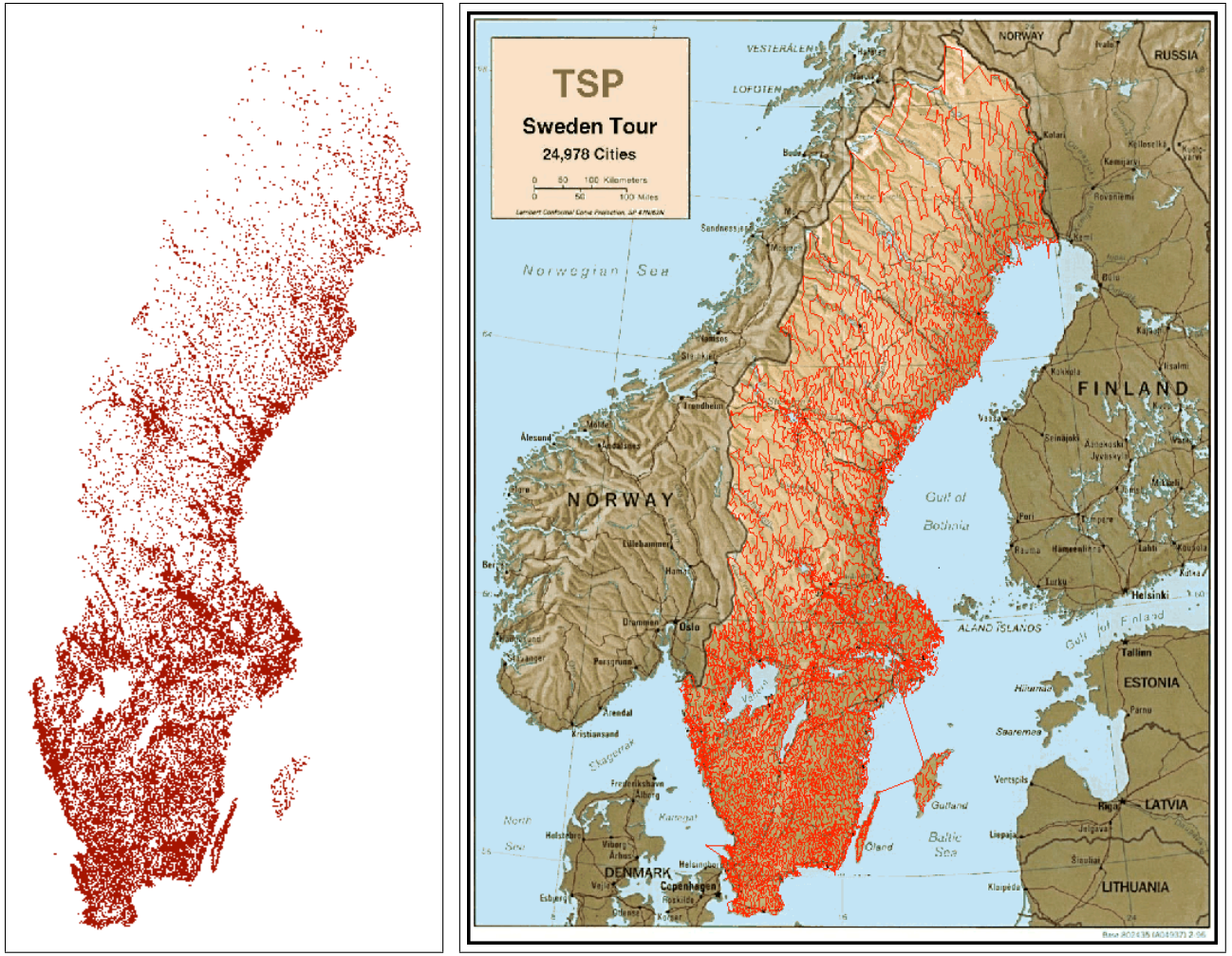
1994 (D. Applegate, R. Bixby, V. Chvátal y W. Cook): 7.397 puntos (problema de programación lógica, AT&T).



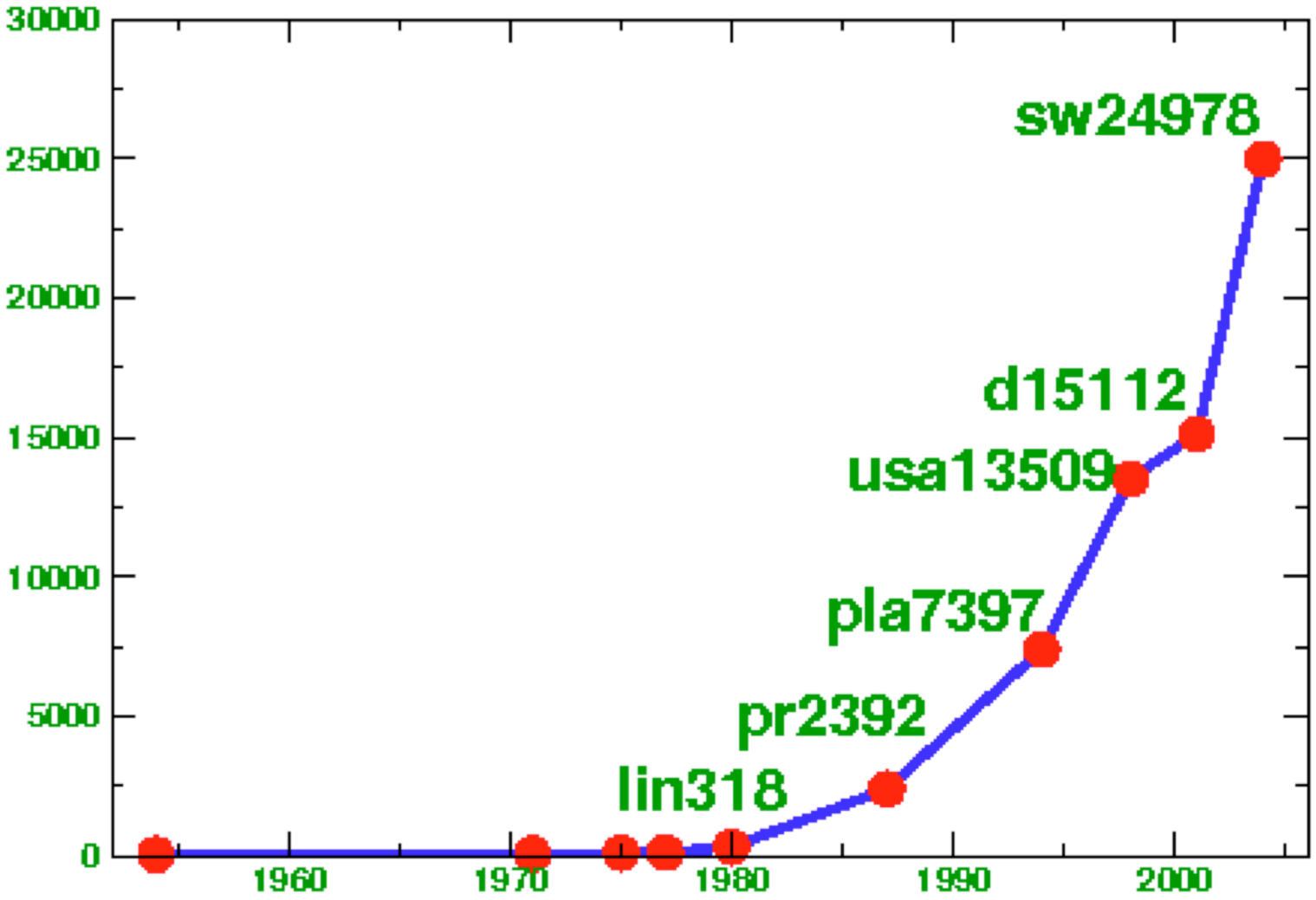
1998 (D. Applegate, R. Bixby, V. Chvátal y W. Cook): 13.509 ciudades de EEUU con una población superior a 500 habitantes.



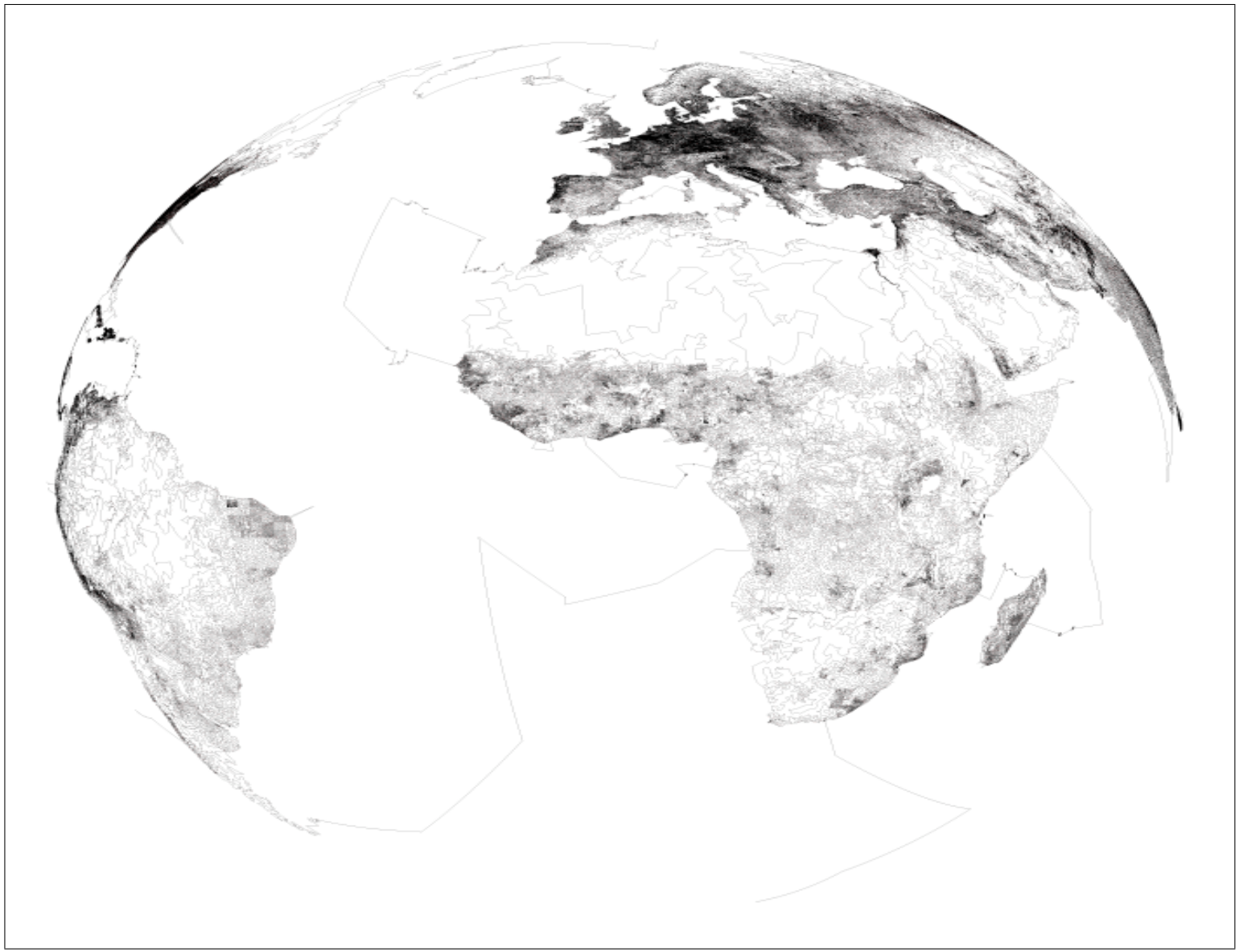
2001 (D. Applegate, R. Bixby, V. Chvátal y W. Cook):
15.112 ciudades de Alemania.
Tiempo de cálculo: 22,6 años de CPU (Compaq EV6 Alpha 500 MHz).



2004 (D. Applegate, R. Bixby, V. Chvátal, W. Cook y K. Helsgaun):
24.978 ciudades de Suecia.
Tiempo de cálculo: 84,8 años de CPU (Intel Xeon 2,8 GHz).



Evolución en la resolución del Problema del Viajante
("Traveling Salesman Problem", `http://www.tsp.gatech.edu`).



Solución aproximada (0,076 % de error) con 1.904.711 ciudades del mundo.

Algoritmos. Complejidad Computacional.

El número de casos posibles para recorrer una colección de N puntos, sin considerar el sentido o el punto de partida, es $(N-1)!/2$, que sabemos es un número inimaginable para ser tratado directamente:

Si $N = 50$ obtenemos

$$30414093201713378043612608166064768844377641568960512000000000$$

casos.

¿Existe un método que nos permita encontrar dicha solución de manera **efectiva** en un tiempo razonable? En general se entiende que un algoritmo es efectivo si el número de procesos que hay que llevar a cabo es una **función polinómica** de las variables empleadas.

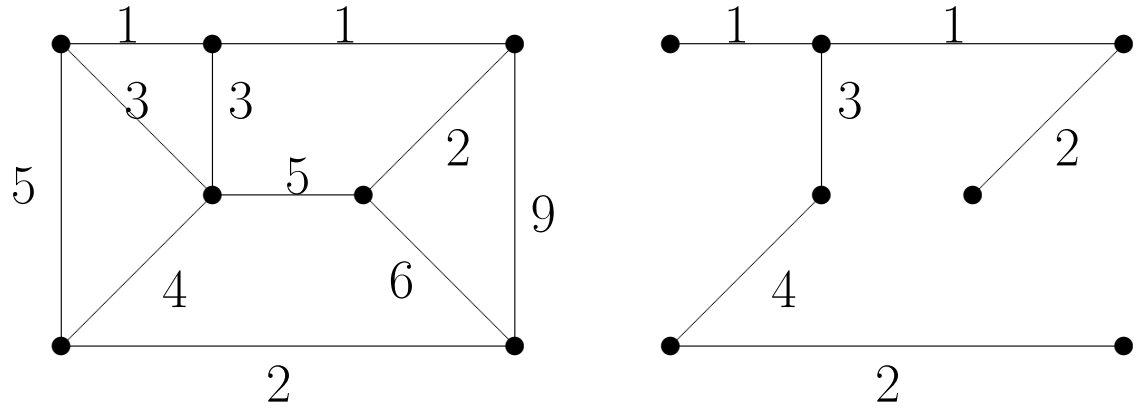
Problemas para los que se conoce un algoritmo polinómico:

- Cálculo del máximo común divisor de 2 números (algoritmo de Euclides).
- Determinar si un número es primo (Agrawal, Kayal y Saxena, 2002).

¿Muchos casos = Muy difícil? ... No siempre

Redes Viarias: Árboles expansivos o abarcadores.

El problema de hallar el mínimo conjunto de aristas que conecte todos los vértices, minimizando las distancias, es similar al del Viajante. De hecho el número posible de soluciones que conectan los vértices **(árbol abarcador)** es incluso mayor: $N^{(N-2)}$. Sin embargo existe un método efectivo (con complejidad polinómica N^2) llamado **algoritmo de Kruskal,** que calcula la solución en todos los casos:



La complejidad no depende solamente del número de casos, sino de la estructura del problema.

Desgraciadamente, el Problema del Viajante es uno de los enunciados elementales en matemáticas, para el que **no se conoce un algoritmo efectivo** (y de hecho se sospecha que no existe tal método).

Otro ejemplo para el que no se sabe si hay un algoritmo polinómico es el de **factorizar un número compuesto** (la no existencia en este caso es fundamental para la seguridad de los métodos de encriptación RSA en las transacciones bancarias).

La resolución de cualquiera de estos problemas está premiado con un millón de dólares por la Fundación Clay de EEUU.

Problema del Milenio: $¿\mathcal{P} = \mathcal{NP}?$

Si la solución es fácil de comprobar, ¿es fácil de calcular?

- **Factorizar:**
  714483485767302083604026045230246586639073114484890246696933169889355932873228786661634819501762200375934783471059374226865019918944197887960884221379660262521156659557083208942126709951203605759980226571172728378570812074142889673497031221126125922573026219220875239756337794139898468677537893345135166717089058642349906588344033143667817673079888613687065420352989056540989729277461695207650958281155907691913475426034947299994980085507164975597727750000478799669913170489801936798146428789679881887202030478277566819257212859959580410511 37
  (“imposible”).
- **El número anterior es divisible por:**
  686479766013060971498190079908139321726943530014330540939446345918554318339765605212255964066145455497729631139148085803712198799971664381257402829111505715 1
  (trivial).

Factorizar es un problema $\mathcal{NP}$. ¿Es $\mathcal{P}$? ¡Ojalá que no!

Ejemplo de un algoritmo efectivo:

Calcular $a^2 + b^2 + 2ab$.

Observamos que el número de procesos (multiplicaciones) que hay que efectuar, para calcular directamente la solución, es 3.

Si embargo, si usamos el "algoritmo del binomio" y cambiamos a^2+b^2+2ab por $(a+b)^2$, observamos que las multiplicaciones se han reducido a 1.

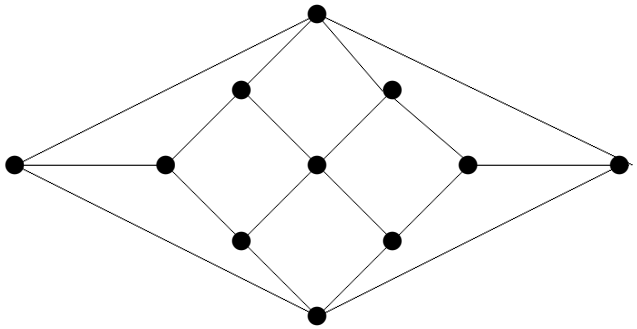
Caso concreto: Tomamos 2 números aleatorios entre 1 y $10^{20,000,000}$.

	$a^2 + b^2 + 2ab$	$(a+b)^2$
Tiempo	$26,1905$	$7,7097$

Un conocimiento del problema a resolver nos puede aportar una mejora sustancial en el algoritmo empleado.

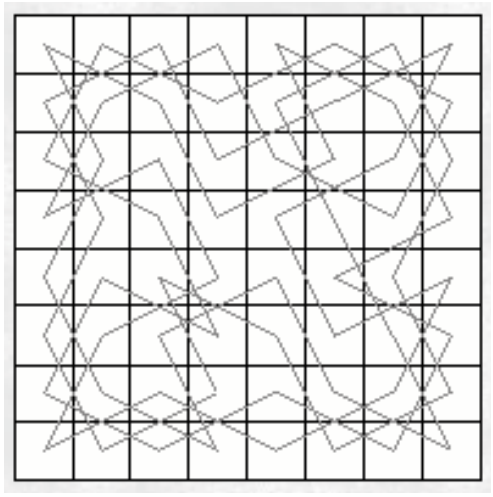
La solución del Problema del Viajante es, por la propia definición, un ciclo del grafo determinado por los puntos y todas las aristas posibles.

Los ciclos de un grafo que recorren todos los vértices sin repetir las aristas se denominan **ciclos de Hamilton**. Es nuevamente un problema sin resolver determinar cuándo un grafo admite un ciclo de Hamilton o no.



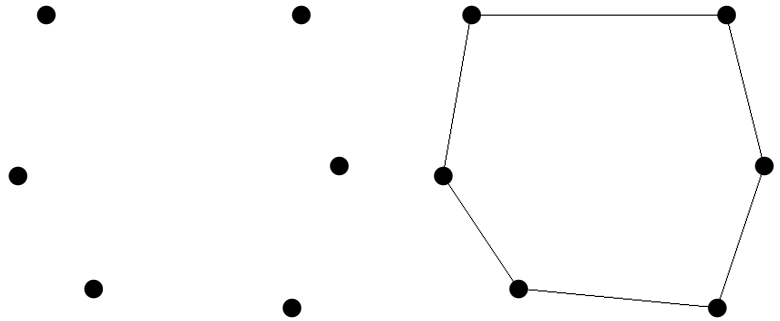
Grafo de Herschel

Otro ejemplo clásico de grafo de Hamilton es el del **Tablero de Ajedrez:** ¿Puede un caballo recorrer todas las casillas de un tablero de ajedrez, de dimensión $n \times n$, de manera cíclica? La respuesta es positiva si, y sólo si, $n \geq 6$ y n es un número par.

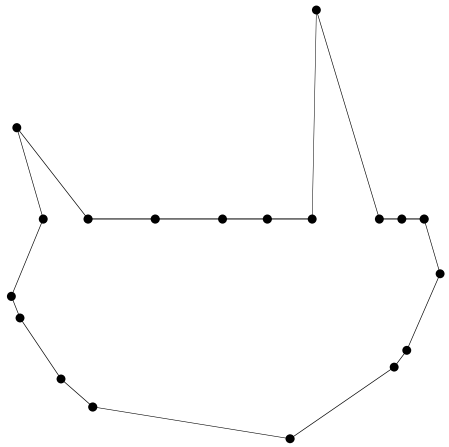


Resultados conocidos sobre el Problema del Viajante:

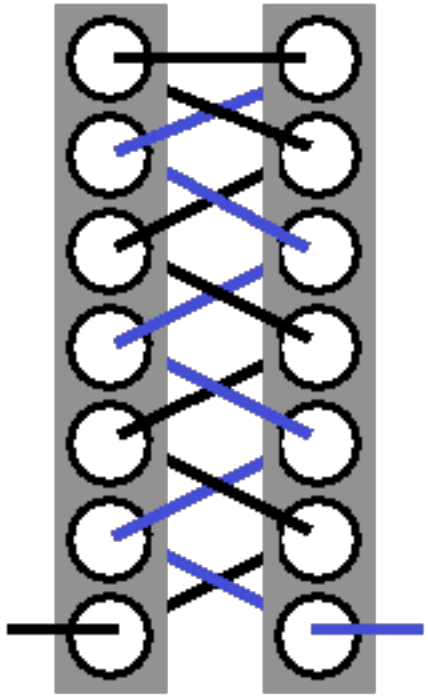
- La solución no puede tener cruces (L.V. Quintas y F. Supnick, 1965).
- Si los puntos forman los vértices de un polígono convexo, el polígono es la solución.



- Si los puntos del interior de la envolvente convexa están sobre una recta (V. Deineko, R. van Dal y G. Rote, 1992):



■



La manera más corta (y única) de acordonar unos zapatos es mediante el lazo en forma de zig-zag (J.H. Halton, 1995).

Desde la resolución en 1954 del Problema del Viajante para 49 ciudades de EEUU (G. Dantzig, R. Fulkerson y S. Johnson) se han desarrollado métodos cada vez más rápidos de cálculo, principalmente basados en el método del **Símplex de la Programación Lineal** (G. Dantzig):

El objetivo es **calcular el máximo (o el mínimo) de una función lineal**

$$F(x) = Ax = a_1x_1 + a_2x_2 + \cdots + a_nx_n,$$

siendo x un vector de $\mathbb{R}^n$ y A una matriz fila, **restringiendo la variable a un poliedro,** que se puede describir como la región de $\mathbb{R}^n$ determinada por las desigualdades:

$$Bx \leq c,$$

B es una matriz de dimensión $m \times n$ y c es un vector m-dimensional (es decir, en total tenemos m desigualdades lineales).

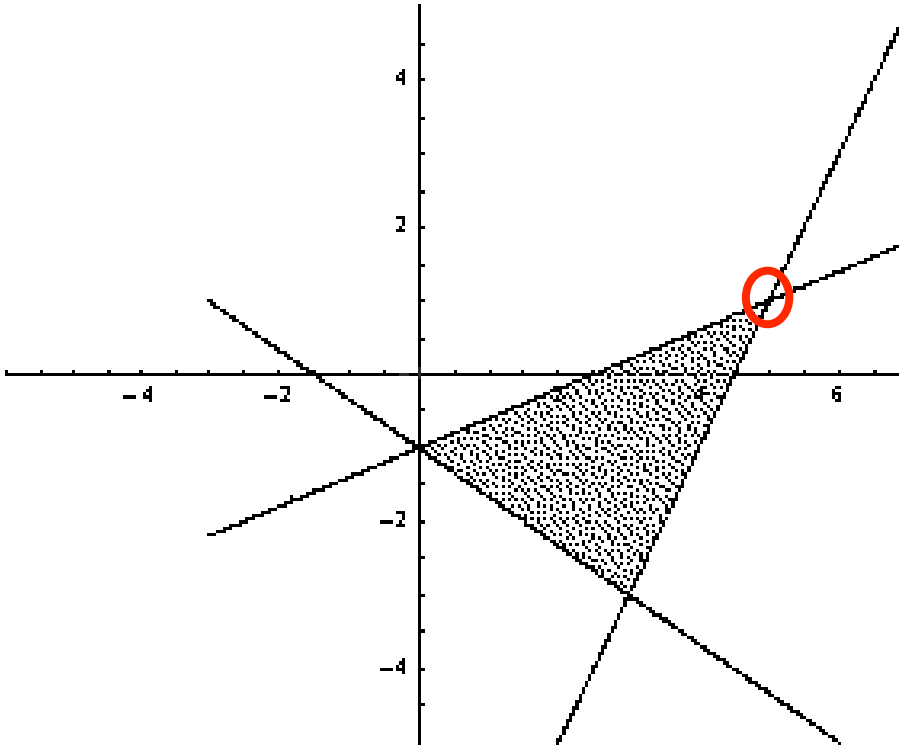
La solución se alcanza en uno de los vértices del poliedro

Calcular el máximo de $F(x, y) = 2000x + 5000y$, sujeta a las restricciones:

$$\begin{cases} -2x - 3y & \leq 3 \\ -2x + 5y & \leq -5 \\ 2x - y & \leq 9. \end{cases}$$

Calculamos los valores en los 3 vértices que se obtienen y observamos que la solución se alcanza en el punto $(5, 1)$ donde la función vale

$$F(5, 1) = 15{,}000.$$



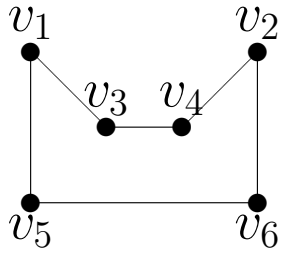
El método del Símplex es un algoritmo que permite obtener la solución del problema de Programación Lineal, de manera efectiva.

¿Cómo se reescribe el Problema del Viajante como un problema de Programación Lineal?

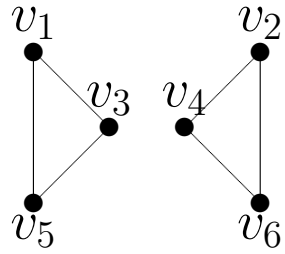
Numeramos TODAS las aristas que unen los N puntos. Hay un total de $\alpha_N = N(N-1)/2$: Si $N = 3$, hay 3. Si $N = 4$, hay 6, etc.

Si una arista se escoge, ponemos un 1 y si no, un 0. Una selección de un ciclo de N aristas es una colección de ceros y unos.

PROBLEMA: ¿Cómo reconocemos si una colección de ceros y unos representa un ciclo?



(0,1,0,1,0,0,1,0,1,1,0,0,0,0,1)



(0,1,0,1,0,0,1,0,1,0,1,0,0,1,0)

Si c_j es la longitud de la arista j-ésima, la longitud de un ciclo vendrá dada por la función lineal:

$$F(x_1, \cdots, x_{\alpha_N}) = c_1 x_1 + \cdots + c_{\alpha_N} x_{\alpha_N}.$$

Nuestro objetivo es minimizar F, restringida a la colección de todos los ciclos posibles (en realidad al menor poliedro que los contiene).

Escribir todas las condiciones que determinan este poliedro tiene una magnitud de complejidad computacional enorme, **comparable al método exhaustivo.**

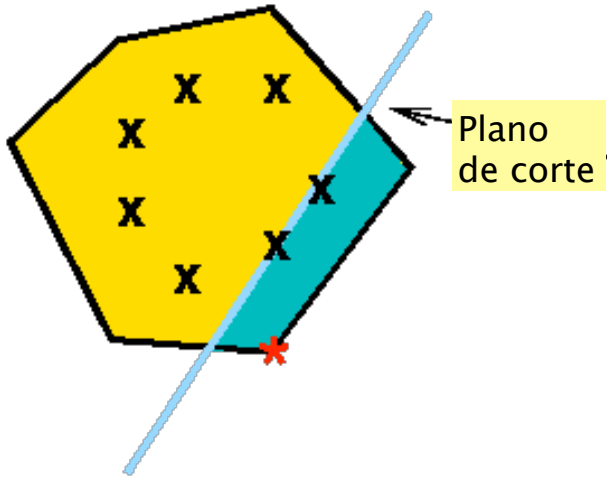
La estrategia es comenzar por un **poliedro sencillo** aunque mayor que el estrictamente necesario.

Si el método del Símplex nos da una solución que es un ciclo, SEGURO que será la solución del Problema del Viajante.

Si no es un ciclo, hemos de elegir un poliedro más pequeño que contenga a todos los ciclos, pero NO a la solución obtenida (esto se conoce como el **Método de los Planos de Corte).** Repetimos el proceso, cruzamos los dedos, y confiamos que en algún momento (de nuestra Era) esto funcione...

Este algoritmo está implementado en el programa CONCORDE:

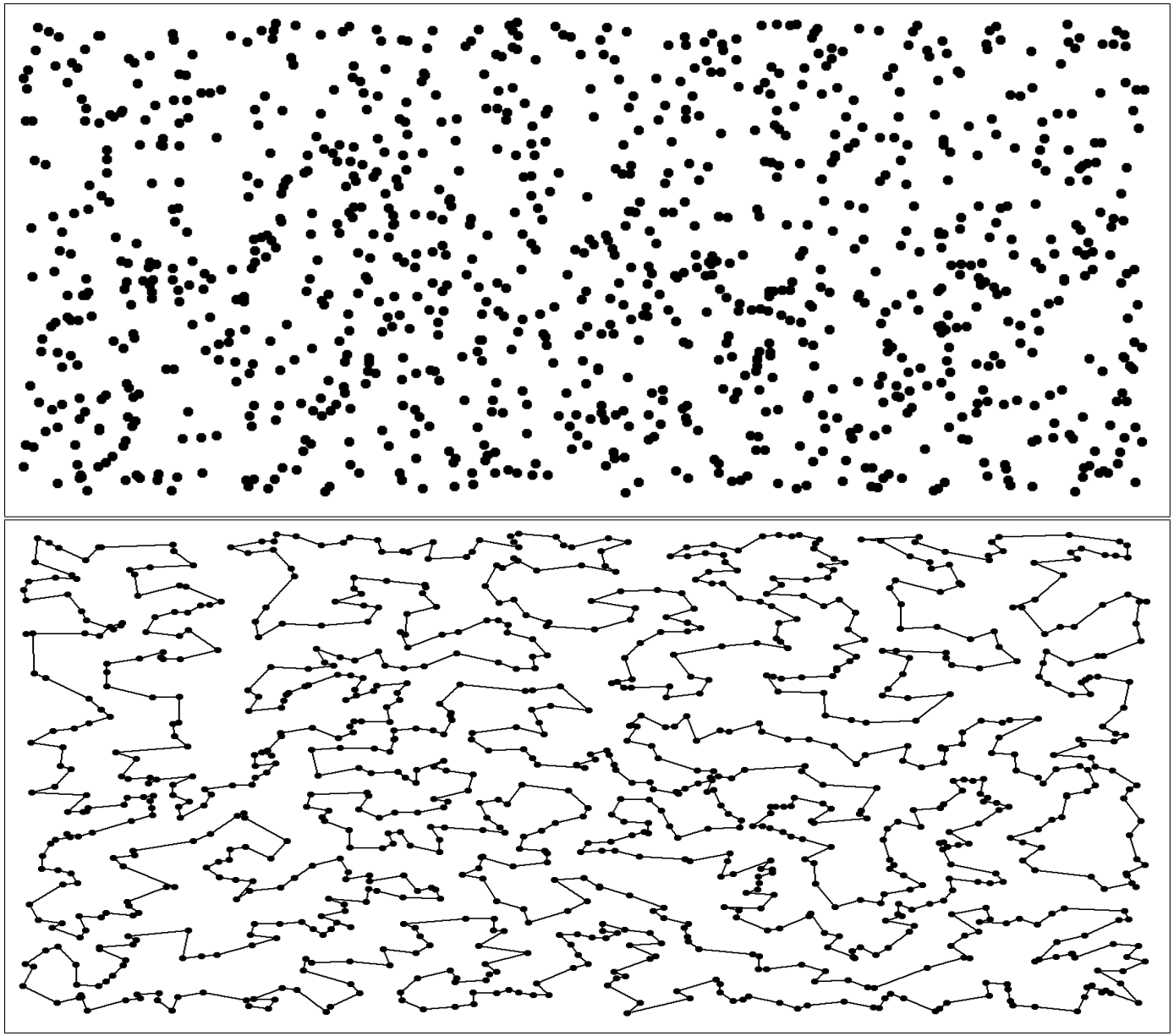
`http://www.tsp.gatech.edu/concorde.`



CONCORDE ha sido usado con éxito en 107 de los 110 problemas que se establecieron en los años 1990, como retos a superar.

¿Cuál es la solución del Problema del Viajante?

Solución (en menos de 1 segundo).



Solución con 1.000 puntos aleatorios (7 minutos y 30 segundos).