

## LA COMBINATORIA POLIÉDRICA Y EL PROBLEMA DEL VIAJANTE. APLICACIÓN AL CASO DE CIENTO TRES CIUDADES ESPAÑOLAS.

RAMÓN ÁLVAREZ-VALDÉS, ÁNGEL CORBERÁN, JOSÉ M. TAMARIT  
UNIVERSIDAD DE VALENCIA

*El trabajo resume los resultados de la aplicación de la Combinatoria Poliédrica al Problema del viajante (TSP): definición del poliedro, dimensión, desigualdades válidas, facetas. Estos resultados se aplican al caso concreto de encontrar el circuito para el TSP de coste mínimo que recorre ciento tres ciudades españolas.*

*Se trata de un proceso interactivo en el que, para cada solución de la relajación lineal del problema, obtenida mediante la aplicación de un código comercial de Programación Lineal, se identifican las restricciones violadas y se incorporan manualmente al problema relajado en la iteración siguiente. En nuestro caso, la séptima resolución del problema lineal, al que se habían incorporado 68 nuevas restricciones, proporcionó la solución óptima.*

**Keywords:** TRAVELLING SALESMAN PROBLEM; FACETS; CUTTING PLANES; LINEAR PROGRAMMING.

### 1. INTRODUCCION.

El problema del viajante (TSP) puede plantearse de la forma siguiente: un viajante debe visitar cada una de las ciudades de su zona exactamente una vez y volver a su punto de partida. Conocidos los costes de desplazamiento entre cualquier par de ciudades, ¿cómo debe realizar su itinerario de forma que visite cada ciudad exactamente una vez y que el coste total del circuito sea mínimo?. En términos de Teoría de Grafos, el problema es: dado un grafo  $G=(V,E)$ , donde  $V$  es un conjunto de vértices y  $E$  un conjunto de arcos o aristas, con costes no negativos asociados, encontrar un circuito de coste mínimo que pase exactamente una vez por cada vértice de  $V$ . En el presente trabajo nos referiremos exclusivamente al TSP simétrico, en el que el grafo  $G$  es no dirigido ( $c_{ij}=c_{ji}, \forall (i,j) \in E$ ).

Se trata de un problema clásico de la Optimización Combinatoria y es uno de los que mayor atención ha recibido durante más de treinta años. El TSP pertenece a la clase NP-completo (/11/) y por tanto parece muy improbable que se encuentre un algoritmo -

que lo resuelva en tiempo polinomial.

Desde 1954, numerosos investigadores han desarrollado algoritmos tanto heurísticos como exactos para resolver problemas de tamaño cada vez mayor. La razón de este interés es doble: por una parte, el gran número de problemas reales en los que el TSP da la solución o aparece como subproblema (problemas de rutas de vehículos, secuenciación,...) Por otra parte, técnicas desarrolladas inicialmente para este problema han sido luego adaptadas con éxito a otros problemas de Optimización Combinatoria, como ocurrió con la Relajación Lagrangiana desarrollada por Held y Karp /10/ para el TSP y cuyo uso se extendió posteriormente a otros muchos problemas. El mismo término "Branch and bound" fué acuñado por Little et al. /15/ en relación con su algoritmo para el TSP.

Damos ahora un breve resumen de los algoritmos heurísticos y exactos aparecidos en los últimos años. Muchos algoritmos heurísticos

- Ramón Alvarez-Valdés Olaquibel - Angel Corberán Salvador - José Manuel Tamarit Goerlich - Universidad de Valencia - Facultad de Matemáticas - Dep. de Estadística e Investigación Operativa. - Dr. Moliner, 50 BURJASOT Valencia.

- Article rebut el juliol de 1985.

han sido propuestos para la resolución aproximada del TSP. Entre todos ellos destacaremos el algoritmo de Christofides /2/ que es el que asegura para cualquier ejemplo del problema un menor alejamiento respecto del óptimo y que es un algoritmo polinomial basado en el cálculo de un árbol generador de mínimo peso y un acoplamiento mínimo, y el algoritmo de Lin y Kernighan /14/, basado en técnicas de búsqueda local, que propone, partiendo de una solución posible inicial, ir realizando intercambios de aristas que conduzcan a soluciones progresivamente mejores.

Los algoritmos exactos se dividen en dos grandes grupos: algoritmos basados en la Programación lineal y algoritmos basados en técnicas de branch and bound. Dantzig et al. /4/ fueron los primeros que propusieron un algoritmo exacto para la resolución del TSP, usando la programación lineal de forma iterativa e introduciendo en cada iteración nuevas restricciones incumplidas por la solución lineal, hasta obtener la solución entera. Sin embargo, este tipo de algoritmos cayó en desuso en los años 60 debido al desarrollo de técnicas de branch and bound desde las propuestas por Little et al. /15/ y Held y Karp /10/ hasta las más recientes de Balas y Christofides /1/, cuyo algoritmo es el mejor hasta la fecha para el TSP asimétrico. Los métodos de branch and bound, que han demostrado ser muy eficientes en problemas de tamaño medio, parecen difícilmente aplicables a problemas de mayor tamaño, debido a su alto coste computacional. Esta razón, junto con el gran desarrollo de la Combinatoria Poliédrica durante la última década, ha propiciado el redescubrimiento de las técnicas de planos de corte, fundamentados ahora en el conocimiento de la estructura del poliedro de soluciones posibles. En este terreno cabe citar los trabajos de Grotschel y Padberg /7/, Grotschel /8/, Crowder y Padberg /3/, Padberg y Hong /16/ y Grotschel y Padberg /9/, que resuelven problemas de hasta 300 vértices.

Nuestro trabajo presenta la resolución heurística y exacta de un problema real de 103 ciudades españolas. El algoritmo heurístico utilizado parte de una solución inicial proporcionada por el algoritmo de Christofides /2/ y la mejora siguiendo las ideas del algoritmo de Lin y Kernighan /14/ y aparece

descrito en la Sección 3. En la Sección 4 se resumen las ideas básicas de Combinatoria Poliédrica que han sido necesarias en el presente trabajo y se describe el algoritmo utilizado, que sigue la línea de los trabajos de Dantzig et al. /4/ y Grotschel /8/. La Sección 5 presenta el problema en estudio y su resolución mientras que la Sección 6 recoge las conclusiones.

Se trata de una técnica nueva, todavía poco probada. El presente trabajo incluye las ideas en las que se basa y presenta el proceso completo de su aplicación sobre un problema concreto, con lo que se ilustra su manejo y se aporta una nueva experiencia computacional a las ya existentes en la literatura. (1)

## 2. NOTACION.

En esta Sección seguiremos básicamente la notación utilizada por Grotschel /8/.

Consideraremos grafos  $G=(V,E)$  no dirigidos, sin bucles ni aristas múltiples. El conjunto de vértices  $V$ , que representa las ciudades será  $\{1,2,\dots,n\}$ . Las aristas del conjunto  $E$  se denotan  $(i,j)$ ,  $i \neq j$ . Un grafo con  $n$  vértices es completo si  $E=\{(i,j), \forall i,j \in V, i \neq j\}$  y se denota  $K_n=(V,E)$ . Un conjunto  $C$  de  $k$  aristas,  $k \geq 3$ , tal que  $C=\{(v_1,v_2), (v_2,v_3), \dots, (v_k,v_1)\}$ ,  $v_i \neq v_j$  es un ciclo de longitud  $k$ . Los ciclos de longitud  $n$  se denominan circuitos o tours y los ciclos de longitud menor que  $n$ , subcircuitos o subtours.

Si  $C_1, C_2, \dots, C_r$  son ciclos tales que cada vértice está exactamente en un ciclo, la unión de los ciclos es un acoplamiento de cardinalidad 2. Por tanto, cada tour es un acoplamiento de cardinalidad 2.

Para los subconjuntos  $W \subset V$  y  $F \subset E$  denotamos:

$$V(F) = \{i \in V: i \text{ está en una arista de } F\}$$

$$E(W) = \{(i,j) \in E, i \in W, j \in W\}$$

Si  $\{x_e, e \in E\}$  es un conjunto de variables indexadas por  $E$ ,  $F \subset E$  y  $W \subset V$  escribiremos:

$$x(F) = \sum_{e \in F} x_e \quad \text{y} \quad x(W) = x(E(W)).$$

### 3. EL ALGORITMO HEURISTICO.

Como se ha mencionado en la introducción, hemos utilizado un algoritmo que sigue las ideas del algoritmo de intercambio de Lin y Kernighan /14/ para hallar soluciones progresivamente mejores a partir de una solución inicial obtenida por el algoritmo de Christofides /2/.

Dado un grafo  $G=(V,E)$ , el algoritmo de Christofides consta, básicamente, de las siguientes etapas:

- a) Encontrar un árbol generador de mínimo peso,  $S$ , en  $G$ .
- b) Identificar todos los vértices de grado impar en  $S$ . Resolver un acoplamiento -- perfecto de mínimo peso en el subgrafo inducido por  $S$  en  $G$ . El grafo formado por las aristas del árbol y del acoplamiento es euleriano y por lo tanto existe un circuito que, partiendo de cualquier vértice, atraviesa exactamente una vez todas las aristas.
- c) Transformar el circuito euleriano en un circuito para el TSP.

La ventaja de este algoritmo es que asegura la obtención de una solución posible en tiempo polinomial y que el valor de esta solución nunca excede en más de un 50% el valor de la solución óptima.

Esta solución puede ser mejorada fácilmente siguiendo el esquema de intercambio basado en el trabajo de Lin y Kernighan /14/:

- a) Partir del circuito para el TSP obtenido anteriormente.
- b) Mejorar el circuito utilizando un  $k$ -cambio (para un  $k$  dado, un  $k$ -cambio es la eliminación de  $k$  aristas del circuito y su reemplazamiento por otras  $k$  aristas que formen un nuevo circuito para el TSP con coste menor).
- c) Continuar la etapa anterior hasta que no pueda obtenerse una mejora adicional.

La solución así obtenida recibe el nombre de  $k$ -óptimo y es un óptimo local, no necesariamente un óptimo global. En general, cuanto

mayor sea el valor de  $k$  en el procedimiento descrito, mejor será la solución, pero el coste computacional crece rápidamente con el valor de  $k$ . Así pues, se hace necesario un equilibrio entre la proximidad al óptimo y el coste computacional, que depende en gran medida de la intuición y del estudio empírico del algoritmo. Estudios realizados por Lin /13/, Lin y Kernighan /14/ y Golden et al. /6/, sugieren que las soluciones 3-óptimas son mejores que las 2-óptimas, pero que las 4-óptimas no son lo suficientemente mejores que las 3-óptimas para justificar el incremento del coste computacional.

### 4. UN ALGORITMO EXACTO BASADO EN LA COMBINATORIA POLIEDRICA.

Esta sección resume los resultados de la aplicación de la Combinatoria Poliédrica al problema del viajante, así como su utilización en la construcción de un algoritmo eficiente para su resolución.

#### 4.1.- El poliedro del TSP y poliedros asociados.

El problema del viajante puede ser formulado como sigue: a cada arista  $e \in E$  asociamos una variable  $x_e$  y a cada circuito  $T \in E$  le asociamos un vector de incidencia  $x^T$  tal que

$$x_e^T = \begin{cases} 1 & \text{si } e \in T \\ 0 & \text{en caso contrario} \end{cases}$$

Como  $|E| = \frac{1}{2}n(n-1) = m$ ,  $x^T \in \mathbb{R}^m$ .

El poliedro del viajante será la envoltura convexa de los vectores de incidencia de todos los circuitos en  $K_n$ :

$$Q_T^n = \text{conv} \{x^T \in \mathbb{R}^m / T \text{ es un circuito de } K_n\}$$

Por tanto, a cada vértice de  $Q_T^n$  le corresponde un circuito en  $K_n$  y viceversa. Si tuviéramos una caracterización completa de  $Q_T^n$  en términos de igualdades y desigualdades lineales, el problema se podría resolver mediante la programación lineal, pues se reduciría a:  $\min cx; x \in Q_T^n$ .

Para conseguir el mayor acercamiento posible a la caracterización lineal de  $Q_T^n$  vamos a ir definiendo una serie de poliedros cuya caracte-

terización es conocida y que contienen a  $Q_T^n$ .

Por su definición,  $Q_T^n$  está contenido en el hiper-cubo unidad

$$\{x \in \mathbb{R}^m / 0 \leq x_e \leq 1, \forall e \in E\}$$

Además, todo circuito T ha de cumplir que cada vértice sea incidente con exactamente dos aristas de T, con lo que el circuito ha de satisfacer el sistema  $Ax=2e_n$ , donde A es la matriz de incidencia vértices-aristas de  $K_n$  y  $e_n$  un vector columna de unos. Esto supone que

$$Q_T^n \subset Q_A^n = \{x \in \mathbb{R}^m / Ax=2e_n, 0 \leq x_e \leq 1, \forall e \in E\}.$$

El poliedro  $Q_A^n$  contiene estrictamente a  $Q_T^n$ . Sus vértices incluyen, además de los correspondientes a los vectores de incidencia de los circuitos, los vectores de incidencia de los subtours y vértices fraccionarios en los que  $0 < x_e < 1$  para algunas  $e \in E$ . Para obtener un poliedro más cercano a  $Q_T^n$  hemos de eliminar los vértices de los subtours y los vértices fraccionarios. Para ello utilizaremos planos de corte.

#### 4.2.- Desigualdades válidas y facetas del poliedro $Q_T^n$

Una desigualdad  $ax \leq a_0$  se dice que es válida para  $Q_T^n$  si todos los puntos del poliedro la cumplen, es decir, si  $Q_T^n \subset \{x \in \mathbb{R}^m / ax \leq a_0\}$

Una desigualdad válida  $ax \leq a_0$  define una cara propia de  $Q_T^n$  si algunos puntos del poliedro, pero no todos, la satisfacen como igualdad, es decir

$$\emptyset \neq Q_T^n \cap \{x \in \mathbb{R}^m / ax = a_0\} \neq Q_T^n$$

Una desigualdad válida define una faceta si define una cara propia maximal, es decir, no contenida en ninguna otra. Dos facetas definidas por  $ax \leq a_0$  y  $bx \leq b_0$  son equivalentes si

$$Q_T^n \cap \{x \in \mathbb{R}^m / ax = a_0\} = Q_T^n \cap \{x \in \mathbb{R}^m / bx = b_0\}$$

Si K es el número de diferentes clases de facetas equivalentes de  $Q_T^n$  y elegimos de cada clase exactamente una desigualdad --  $a^i x \leq a_0^i$ ,  $i=1, \dots, K$ , junto con las igualdades

$Ax=2e_n$  que todo punto de  $Q_T^n$  debe cumplir, entonces

$$Q_T^n = \{x \in \mathbb{R}^m / Ax = 2e_n, a^i x \leq a_0^i, i=1, \dots, K\}$$

Esta caracterización lineal de  $Q_T^n$  es no redundante, es decir, si quitamos alguna igualdad de  $Ax=2e_n$  o alguna desigualdad  $a^i x \leq a_0^i$ , el poliedro de la derecha ya no es  $Q_T^n$ .

Esto significa que las facetas son los mejores planos de corte posibles, pues sólo el conocimiento de, al menos, un elemento de cada clase de facetas de  $Q_T^n$  proporciona una caracterización lineal completa y no redundante. Por tanto, a la hora de añadir planos de corte, deberíamos usar desigualdades que definen facetas. Las desigualdades válidas que no definen facetas no bastan para delimitar el poliedro de interés y su efecto es incierto, aunque en algunos casos puedan ser suficientes para obtener la solución óptima del problema entero.

La importancia de las facetas justifica su caracterización y el desarrollo de métodos de obtención. La caracterización de una faceta viene dada por el siguiente teorema de la Teoría de Poliedros:

**TEOREMA:** Sea  $P \subset \mathbb{R}^n$  un poliedro y supongamos que A es una matriz  $m \times n$  y  $b \in \mathbb{R}^m$ , tal que la envoltura afin de P  $\text{aff}(P) = \{x \in \mathbb{R}^n / Ax=b\}$ .

Sea F una cara no vacía de P. Las siguientes afirmaciones son equivalentes:

- a) F es una faceta de P.
- b) F es una cara maximal propia de P.
- c)  $\dim(F) = \dim(P) - 1$ .

d) Existe una desigualdad válida  $cx \leq c_0$  para P con las siguientes propiedades:

- (d<sub>1</sub>)  $F = \{x \in P / cx = c_0\}$ .
- (d<sub>2</sub>) Existe  $\bar{x} \in P / c\bar{x} < c_0$ , es decir, la desigualdad es propia.
- (d<sub>3</sub>) Si otra desigualdad  $dx \leq d_0$  válida para P satisface  $F \subset \{x \in P / dx = d_0\}$ , entonces existe un escalar  $\alpha > 0$ , y un vector  $\lambda \in \mathbb{R}^m$ , tales que  $d = \alpha c + \lambda A$  y  $d_0 = \alpha c_0 + \lambda b$ .

En las condiciones (c) y (d) se basan los mé-

todos usados para probar que una desigualdad válida  $cx \leq c_0$  define una faceta. El método directo, basado en (c), consiste en probar que la  $\dim(F)$  es  $\dim(P)-1$ , construyendo un conjunto de  $k=\dim(P)$  vectores  $x_1, x_2, \dots, x_k$ , generalmente vértices de  $P$ , que satisfacen  $cx_i = c_0$  y son afinmente independientes. El método indirecto, basado en (d), supone la existencia de una desigualdad válida  $dx \leq d_0$  tal que  $F \subset \{x \in P / dx = d_0\}$  y a partir de la estructura de la matriz  $A$  y las propiedades de los puntos de  $P$  que satisfacen  $cx = c_0$ , obtiene  $\alpha$  y  $\lambda$  de forma que  $d = \alpha c + \lambda A$  y  $d_0 = \alpha c_0 + \lambda b$ , con lo que  $cx \leq c_0$  es una faceta.

El método directo precisa como requisito indispensable el conocimiento de la dimensión del poliedro. En el caso del viajante  $\dim(Q_T^n) = \frac{1}{2}n(n-3)$ , es decir, el número de componentes del vector de incidencia  $\frac{1}{2}n(n-1)$  menos el número de igualdades existentes en la definición del poliedro  $n$  ( $/12/$ ).

Los planos de corte que se detallan a continuación se han obtenido de forma que eliminen los vértices de los subtours y algunos de los vértices fraccionarios de  $Q_A^n$  ya mencionados y cuentan, además, con el respaldo teórico de ser desigualdades válidas que definen facetas de  $Q_T^n$ , con lo que su inclusión nos permite avanzar en la caracterización lineal del poliedro  $Q_T^n$ .

Para eliminar los vértices correspondientes a los subtours, Dantzig et al. /4/, introdujeron las restricciones de eliminación de subtours:

$$x(W) \leq |W| - 1, \quad \forall W \subset V, 2 \leq |W| \leq n-1 \quad (1)$$

Estas restricciones son desigualdades válidas para  $Q_T^n$  y definen facetas  $\forall W \subset V, 3 \leq |W| \leq n-3$ . Además, las restricciones  $x(W) \leq |W| - 1$  y  $x(W') \leq |W'| - 1$  definen facetas equivalentes si y sólo si  $W = V - W'$  ( $/7/$ ).

Si definimos  $Q_S^n = \{x \in Q_A^n / x \text{ satisface (1)}\}$  es obvio que  $Q_S^n$  no tiene otros vértices enteros aparte de los vectores de incidencia de los circuitos, por lo que  $Q_T^n = \text{conv}\{x \in Q_S^n / x \text{ entero}\}$ .

Veamos ahora algunas restricciones que eliminan vértices fraccionarios. Edmonds /5/ introdujo las restricciones de acoplamiento de

cardinalidad 2:

$$\sum_{i=0}^k x(W_i) \leq |W_0| + \frac{1}{2}(k-1) \quad (2)$$

donde

$$W_0, \dots, W_k \subset V, |W_0 \cap W_i| = 1, |W_i| = 2, i = 1, \dots, k, k \geq 1 \text{ e impar.}$$

Si definimos  $Q_{2M}^n =$

$$\begin{aligned} &= \text{conv}\{x \in R^n / M \text{ es un acoplamiento de cardinalidad 2 en } K_n\} \\ &= \text{conv}\{x \in Q_A^n / x \text{ entero}\} \end{aligned}$$

Edmonds demostró que  $Q_{2M}^n = \{x \in Q_A^n / x \text{ verifica (2)}\}$ , con lo que sus restricciones eliminan los vértices fraccionarios de  $Q_A^n$  sin añadir otros nuevos.

Sin embargo, aunque por separado (1) elimina los vértices de los subtours y (2) los vértices fraccionarios de  $Q_A^n$ , incorporadas conjuntamente el poliedro  $Q_S^n \cap Q_{2M}^n$  tiene nuevos vértices fraccionarios, producidos por la intersección de los subespacios definidos por las restricciones (1) y (2).

Una clase más amplia de desigualdades que incluye las restricciones de acoplamiento como caso particular son los "peines": dados  $W_0, W_1, \dots, W_k \subset V$ , tales que  $|W_0 \cap W_i| \geq 1, i = 1, \dots, k; |W_i - W_0| \geq 1, i = 1, \dots, k; W_i \cap W_j = \emptyset, \forall i, j, k \geq 3$  e impar

$$\sum_{i=0}^k x(W_i) \leq |W_0| + \sum_{i=1}^k (|W_i - 1|) - (k+1)/2 \quad (3)$$

El conjunto  $W_0$  es el "mango" y los conjuntos  $W_i$  los "dientes" del peine.

Estas desigualdades definen facetas de  $Q_T^n$  y, además, dos peines con conjuntos de vértices  $(W_0, W_i, i = 1, \dots, k), (W'_0, W'_j, j = 1, \dots, h)$  definen facetas equivalentes si y sólo si  $h = k, W_0 = V - W'_0, W_i = W'_i, i = 1, \dots, k$  ( $/7/$ ).

Si definimos el poliedro  $Q_C^n =$

$= \{x \in Q_A^n / x \text{ satisface (1) y (3)}\}$ , entonces  $Q_T^n \subset Q_C^n \subset Q_S^n \cap Q_{2M}^n$ , y estas inclusiones son estrictas para  $n \geq 8$ . Aunque las facetas expuestas no caracterizan totalmente  $Q_T^n$ , el poliedro  $Q_C^n$  es una buena aproximación y parece apropiado su uso como relajación del conjunto de soluciones posibles del TSP, especialmente porque la estructura de sus facetas es bas-

tante simple. Sin embargo, el problema de programación lineal:

$$\min cx; \quad x \in Q_C^n$$

no puede ser resuelto usando directamente un código de programación lineal, debido al número de restricciones de los tipos (1) y (3), tal como puede apreciarse en la Tabla 1.

TABLA 1

Relación entre el número de restricciones y el tamaño del problema

Nº vértices	Nº subtours	Nº peines
6	25	60
10	501	8841970
20	$5 \times 10^5$	$1.5 \times 10^{18}$
50	$5 \times 10^{14}$	$10^{60}$

Esta dificultad puede ser superada incorporando de forma iterativa las restricciones que realmente son necesarias en cada momento, como describiremos en el apartado siguiente. Este proceso no asegura teóricamente la obtención de una solución óptima entera, ya que el poliedro  $Q_T^n$  no está totalmente caracterizado, por lo que podría ser necesario en algún caso acudir a un proceso de branch and bound que, partiendo de la solución lineal obtenida, consiga la solución entera óptima.

4.3. Esquema del algoritmo basado en planos de corte.

Nuestro problema es:  $\min cx; \quad x \in Q_T^n$ . Comenzamos resolviendo un problema lineal más sencillo:  $\min cx; \quad x \in Q_A^n$ , es decir,  $\min cx; \quad s.a. \quad Ax=2e_n$ , y vamos añadiendo restricciones que se violan según el esquema siguiente:

ETAPA 0: Inicialización.  
 Hacer  $k=0$ ;  $Q_0=Q_A^n$ ;  $PL_0=\{\min cx; x \in Q_0\}$

ETAPA 1: Resolución de  $PL_k$  mediante un código de programación lineal, obteniendo la solución óptima lineal  $x^k$ .

ETAPA 2: Comprobación de la solución  $x^k$ . Si corresponde a un circuito, fin ( $x^k$  es la solución óptima del TSP). En otro caso, ir a la etapa 3.

ETAPA 3: Identificación de los subtours y de los peines que se violan en la solución  $x^k$ . Si no hay ninguno, ir a la etapa 5. En otro caso, ir a la etapa 4.

ETAPA 4: Hacer  $k=k+1$ .  
 Incorporación de las restricciones encontradas en la etapa 3 ( $Bx \leq b$ ) al problema lineal, obteniendo  $Q_k = Q_{k-1} \cup \{x / Bx \leq b\}$  y así  $PL_k = \{\min cx; x \in Q_k\}$ .  
 Volver a la etapa 1.

ETAPA 5: Iniciar un proceso de branch and bound para encontrar la solución entera óptima, fijando alternativamente a 0 y a 1, en cada nivel, una de las componentes no enteras de la solución  $x^k$ .

5. SOLUCION DE UN PROBLEMA DE 103 CIUDADES ESPAÑOLAS.

Tal como quedó expuesto en la introducción, nuestro objetivo es probar la validez del algoritmo heurístico descrito y la posibilidad de resolver exactamente el problema a partir del conocimiento de la estructura del poliedro de sus soluciones posibles  $Q_T^n$ . Para ello vamos a aplicar los algoritmos de las secciones 3 y 4 a un problema real.

5.1.- Descripción del problema

Aunque el problema del viajante tiene muchas aplicaciones en otros campos, parece apropiado y es ya común en la literatura el usar como ejemplos de aplicación de nuevas técnicas de resolución problemas de ciudades y redes de carreteras. Así, el trabajo de Dantzig et al. /4/ de 1954, resolvía el problema para 49 ciudades de Estados Unidos, mientras Grötschel en 1980 lo resolvía para 120 ciudades de la República Federal Alemana /8/. Por ello nos decidimos por un problema de ciudades y carreteras españolas. La elección más natural es la de construir el circuito de longitud mínima entre las 47 capitales de provincia peninsulares. Este primer problema es relativamente sencillo y su resolución no ilustra suficientemente la potencia del método que se intenta exponer. - Por ello, hemos ampliado el número de ciudades

a 103, con lo que nos encontramos ante un problema de gran dimensión (el número de variables es ahora 5253) que puede ser un buen banco de pruebas para los algoritmos propuestos. El criterio de inclusión de las 56 ciudades no capitales es, básicamente, el de su población, aunque con ciertas matizaciones. Así, se han excluido las ciudades de los cinturones industriales de Madrid, Barcelona, Bilbao y Valencia, cuya excesiva proximidad a las grandes capitales suponía aumentar el número de ciudades sin por ello aumentar la dificultad del problema. Por el contrario, se han incluido otras ciudades de población relativamente pequeña (Jaca, Ciudad Rodrigo, Seu d'Urgell,...) que resultaban de interés por otros factores como proximidad a frontera, representación de regiones poco pobladas, ... Las distancias entre las ciudades han sido obtenidas a partir del mapa de carreteras del MOPU (sin que se pueda garantizar la total precisión). Se han utilizado carreteras nacionales y provinciales, recurriendo a las carreteras comarcales sólo en casos muy limitados en los que su uso nos pareció imprescindible. La lista de las 103 ciudades aparece en el Apéndice A.

5.2.- Una solución posible inicial

Antes de resolver el problema exactamente, nos planteamos la obtención de una buena solución posible en base a dos objetivos: en primer lugar, conseguir una "buena" estimación del coste del circuito óptimo y, en segundo lugar, la reducción del tamaño del problema, por medio de la eliminación de variables que pudieran fijarse a 0 ó a 1.

La idea de la reducción del tamaño del problema fue ya descrita por Dantzig et al. -- / 4/, y consiste básicamente, en lo siguiente: representamos por  $\bar{c}_j$  el coste reducido de una variable  $x_j$  en la solución óptima del P.L. y sea  $d$  la distancia entre el valor de una solución posible y el valor óptimo de la función objetivo del problema lineal. Todas aquellas variables no básicas  $x_j$  que toman el valor 0 en la solución óptima del PL, para las que  $\bar{c}_j > d$ , pueden ser fijadas a 0, -- pues su incorporación a la solución del TSP produciría un aumento en el coste que lo haría superior al de la solución posible ya conocida. Análogamente, aquellas variables

no básicas  $x_j$  que toman el valor 1 en la solución óptima del PL y para las que  $\bar{c}_j \leq -d$ , pueden ser fijadas a 1.

La solución posible se obtuvo utilizando el algoritmo descrito en la sección 3. La primera parte del algoritmo heurístico produjo un circuito euleroiano de 9579 km. que se muestra en la Figura 1. La Figura 2 representa la transformación del circuito euleroiano en un circuito para el TSP de 9236 km. de longitud y, finalmente, la Figura 3 muestra el -- circuito mejorado según la segunda parte del algoritmo descrito y cuya longitud de 8307 km. representa una desviación del 4.2% respecto de la solución óptima del problema.

5.3.- Solución exacta del problema mediante el algoritmo de planos de corte.

Comenzamos con el poliedro  $Q_A^{103}$ , es decir, con el problema lineal:

$$\min cx ; \text{ s.a. } Ax = 2e_n$$

Este problema tiene 5253 variables y 103 restricciones (una por ciudad) y fue resuelto por el código FMPS de UNIVAC. La solución obtenida  $x^1$  tiene un valor de 7421.5 km. Es una solución no entera en la que 26 variables tomaron el valor 0.5 y 90 el valor 1. Se trata, pues, de una solución fácil de representar gráficamente, tal como aparece en la Figura 4. A partir de esta representación se identifican los subtours y los peines que se violan. En  $x^1$  existían 25 subtours y 1 peine con tres dientes. El peine aparece en la Figura 5. Estas restricciones se incorporan a las originales para resolver un nuevo problema lineal con las mismas variables y 129 restricciones. La segunda solución  $x^2$  tiene un valor de 7683 km, y se procede de igual forma obteniendo sucesivamente soluciones con valores de 7873.5, 7948.67, 7959.72, 7966.87 y, finalmente, 7970 km. que corresponden a la solución  $x^7$  que es la solución entera del TSP. La descripción completa de las restricciones añadidas en cada iteración aparece en el Apéndice B, mientras que la Figura 6 representa la solución óptima.

En total fueron, pues, necesarias siete iteraciones con un total de 68 restricciones -- adicionales, de las que 47 correspondían a

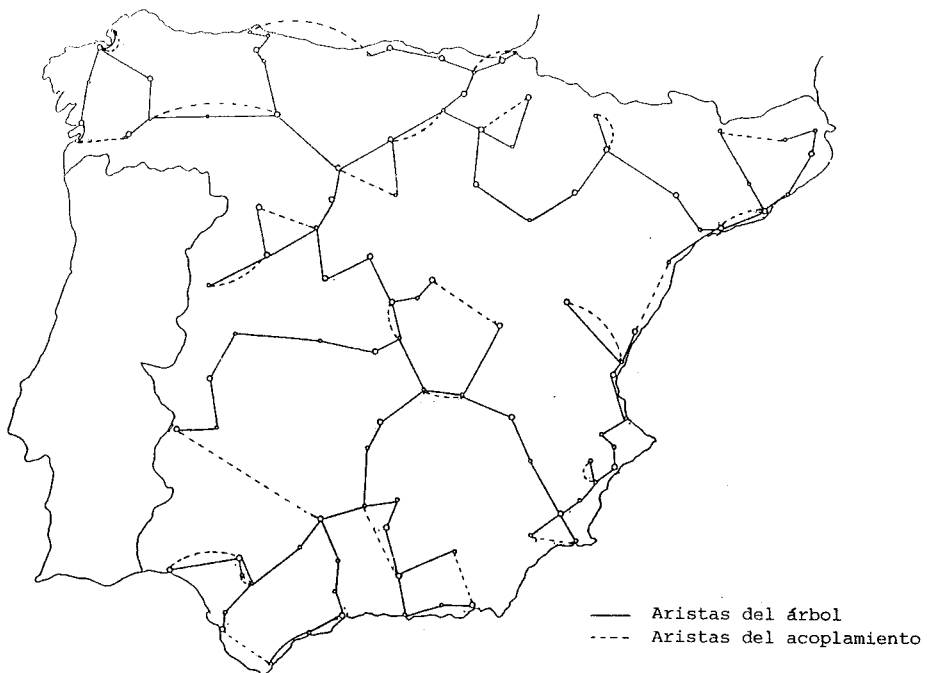


Figura 1: Circuito euleriano

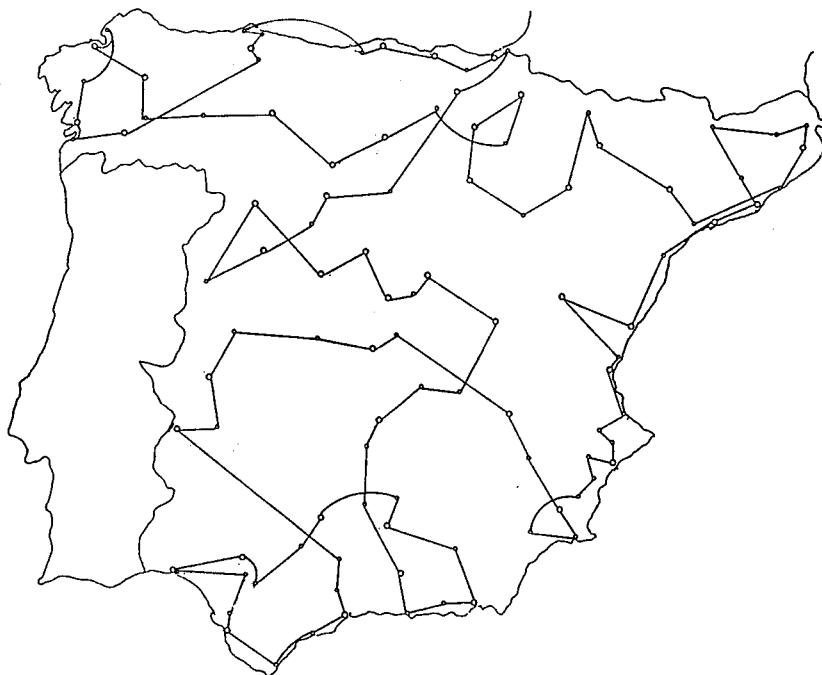


Figura 2: Circuito para el TSP.



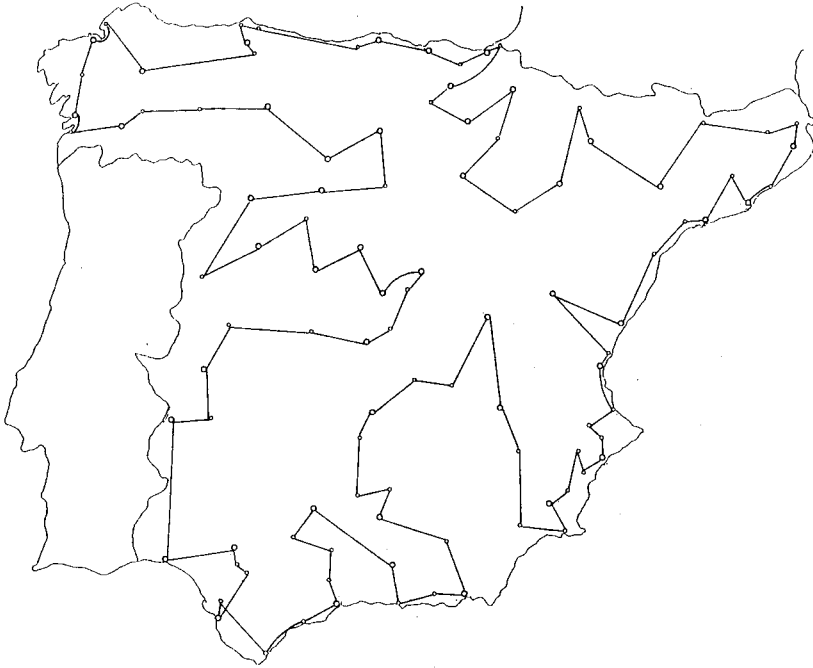


Figura 3: Soluci3n posible inicial

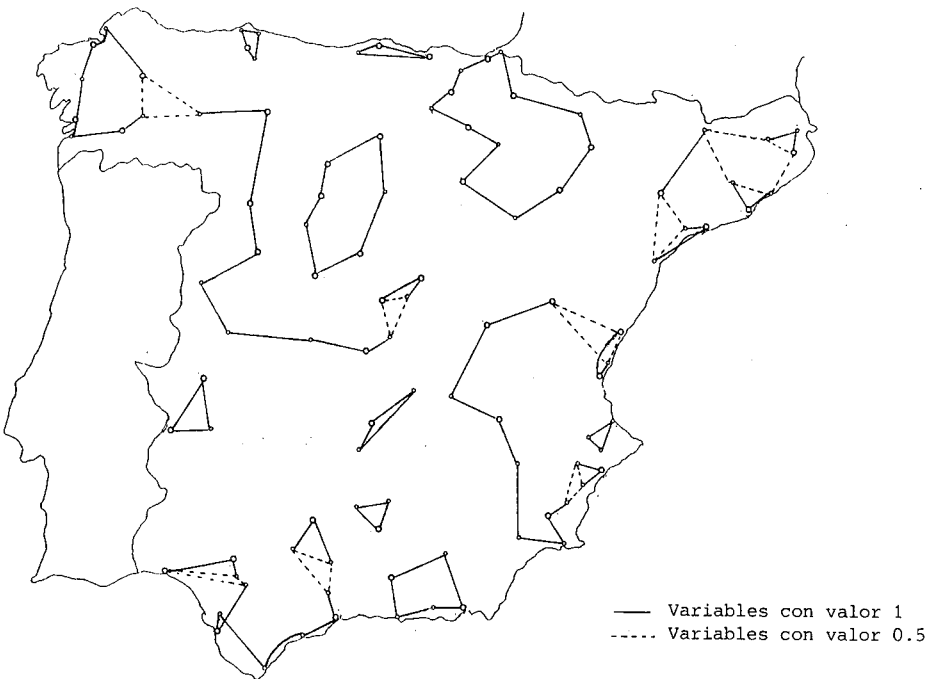


Figura 4: Primera soluci3n lineal  $x^1$

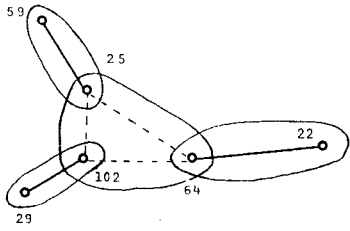


Figura 5a: Peine (1)

— Variables con valor 1  
 - - - Variables con valor 0.5

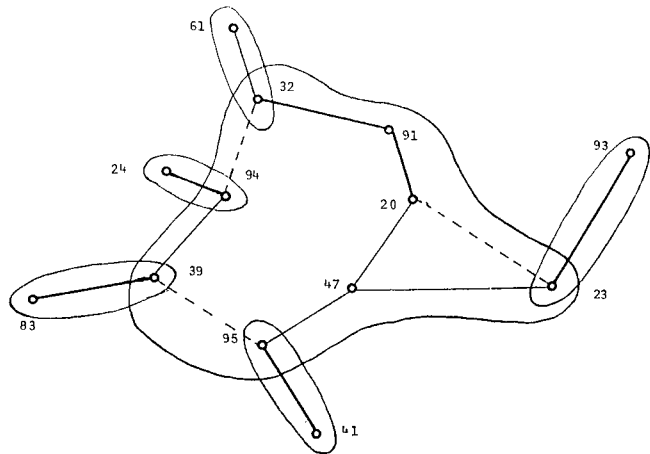


Figura 5b: Peine (10)

— Variables con valor 1  
 - - - Variables con valor 2/3  
 . . . Variables con valor 1/3

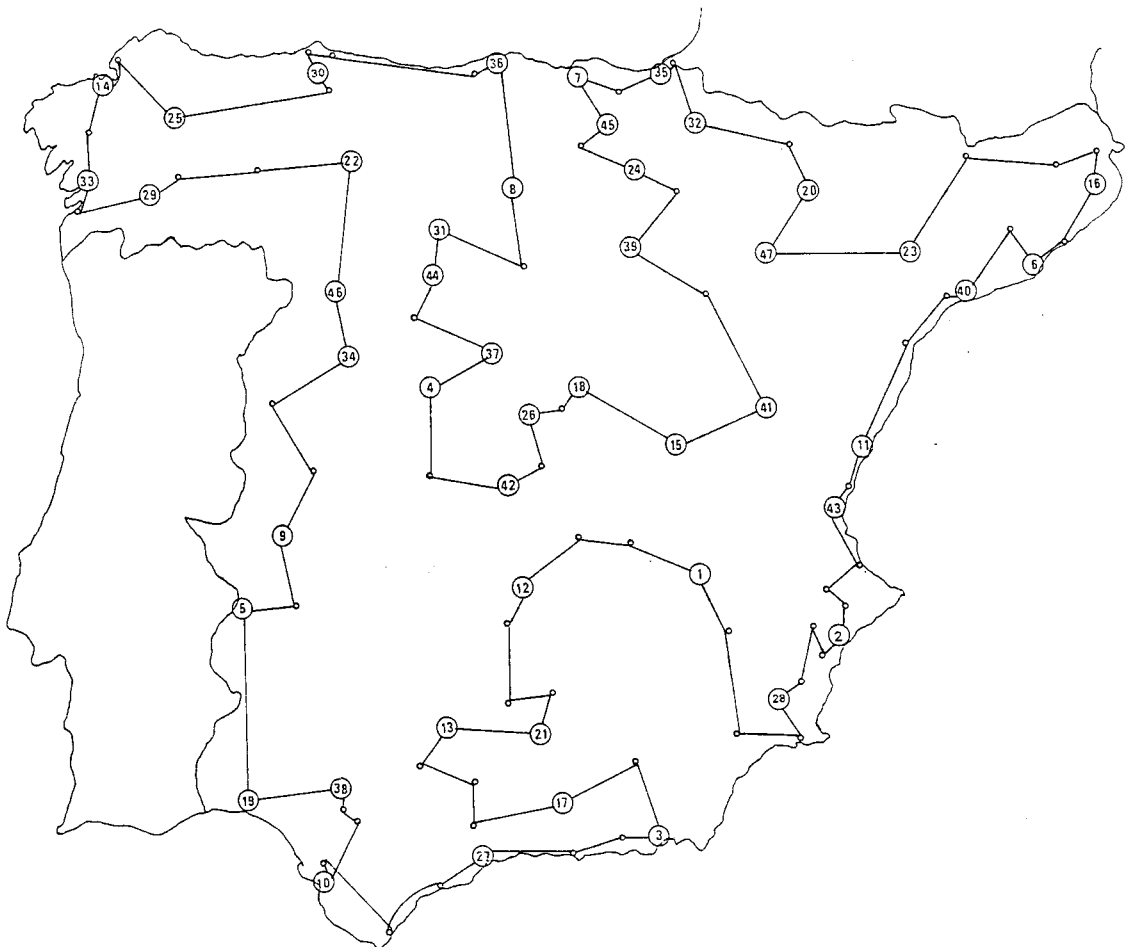


Figura 6: Solución óptima para el TSP ( $x^7$ ).

subtours y 21 a peines. Así pues, el poliedro definido por la intersección de estos 68 semiespacios con  $Q_A^{1,0,3}$  contiene a  $Q_T^{1,0,3}$  y tiene la misma solución óptima.

Las variables  $x_{i,j}$  (que valen 1 si el circuito utiliza la arista  $(i,j)$  y 0 en caso contrario) fueron acotadas superiormente por 1, y, así como en  $x^1$  sólo toman valores 1,0,0,5, circunstancia que se repite en  $x^2$  y  $x^3$ , pueden tomar cualesquiera valores racionales. Así, en  $x^4$  hay variables con valores 1/3 y 2/3 y en  $x^6$  con valores 1/8, 2/8, ..., 7/8. Este hecho dificulta su representación gráfica, que difícilmente puede ser monocromática, pero no es un obstáculo que impida la detección de nuevas restricciones no satisfechas por las correspondientes soluciones.

En la serie de soluciones óptimas de los problemas lineales puede observarse, como ya se señaló Grötschel /8/, que el incremento en el valor de la función objetivo es mucho mayor en las primeras iteraciones (261.5 km. de  $x^1$  a  $x^2$ , 190.5 de  $x^2$  a  $x^3$ ) que en las últimas (apenas 3.12 km. de  $x^6$  a  $x^7$ ). Si el tiempo de resolución es limitado o, simplemente, una solución cuasióptima fuera suficiente,  $x^3$ , por ejemplo, está a menos de 100 km. -- (1.25%) del óptimo y a 433 km. (5.2%) de la solución heurística conocida, lo que le confiere a ésta la garantía de una desviación respecto del óptimo menor del 5.2%.

Como ya hemos mencionado, los siete problemas lineales con 5253 variables, 5253 cotas superiores, 103 igualdades y hasta 68 desigualdades, fueron resueltos utilizando el código comercial FMPS de UNIVAC, en un ordenador UNIVAC 1100/60 del Centro de Cálculo de la Universidad Politécnica de Valencia. Los tiempos de CPU necesarios en cada ejecución oscilaron entre los 0.9 y 2.5 minutos y el número de iteraciones entre 238 y 521, ambos aumentando lentamente, pero no de forma monótona, de la primera a la última ejecución.

El número de variables podría haber sido reducido tal como se comentó en la sección anterior. En nuestro caso, a partir de la primera solución lineal, cuya distancia a la solución proporcionada por el heurístico es de 885.5 km., podrían haberse eliminado del orden del 5% de las variables. A partir de la segunda, para la que la distancia es de

624 km., la reducción hubiera sido del orden del 18%. Estas reducciones aumentan progresivamente hasta la séptima ejecución, en la que, a partir de la sexta solución lineal, a 340.1 km. de la solución posible, la reducción en el número de variables hubiera sido del orden del 36%. En la práctica, no se utilizaron estas reducciones ya que el número de variables y restricciones totales no supusieron ningún problema para el código y ordenador utilizados, ni en cuanto a la memoria, que nunca sobrepasó los 50 kb que se asignaban previamente a la ejecución del programa en cada iteración, ni en cuanto a los tiempos de resolución.

Para simplificar la elaboración del input para el FMPS se diseñó un programa auxiliar -- que, leyendo la matriz de las distancias y las restricciones adicionales en cada iteración en la forma en la que aparecen en el Apéndice B, define las variables, cotas superiores y el sistema total de restricciones -- en cada caso. Cada iteración no supone más que la introducción de las nuevas restricciones en el fichero correspondiente y la ejecución de este programa auxiliar, tras lo cual el código FMPS obtiene la solución óptima lineal correspondiente.

## 6. CONCLUSIONES.

Los resultados de la sección anterior ilustran claramente la importancia que para la resolución de problemas difíciles de Optimización Combinatorial, como el TSP, de gran dimensión, tiene el conocimiento del poliedro de soluciones posibles del problema. Este conocimiento se traduce en la posibilidad de disponer de planos de corte eficaces, lo que a su vez permite utilizar un algoritmo -- basado en la programación lineal en el que -- la introducción de dichos planos de corte -- lleve rápidamente hacia la solución óptima. Esto tiene especial importancia en nuestros días en los que los códigos de programación lineal están muy perfeccionados y son utilizables en todo tipo de ordenadores. Si además se cuenta, como en este caso, con un algoritmo heurístico que produce buenas soluciones posibles, el esquema se refuerza y la obtención de la solución se simplifica.

En el caso concreto del problema del viajante,

esta técnica produce excelentes resultados. Para este problema, la investigación está centrada en la automatización de la etapa 3 del algoritmo, es decir, el reconocimiento y generación automáticos de las restricciones que se violan. Existen algoritmos exactos -- que en tiempo polinomial reconocen los subtours y las restricciones de acoplamiento no satisfechas y algoritmos heurísticos para los peines en general, con lo que la resolución completa del problema, con el recurso, en algunos casos, al branch and bound, puede ser obtenida de forma automática.

Otros problemas de Optimización Combinatorial están siendo abordados también desde esta perspectiva del estudio de sus poliedros y del desarrollo de algoritmos de planos de corte. Los resultados que se están obteniendo indican que el uso de estas técnicas puede conducir a la solución óptima de problemas de dimensiones sensiblemente mayores de las que las técnicas actuales permiten resolver.

## 7. NOTAS.

(1) Con posterioridad a la revisión de este trabajo nos ha llegado el libro, de reciente publicación, "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization", editado por E.L.Lawler, J.K.Lenstra, A.H.G. Rinnooy Kan y D.B.Shmoys (John Wiley 1985).

Como su subtítulo indica, el libro revisa todas las técnicas de la Optimización Combinatoria a través de su aplicación al Problema del Viajante. En particular, provee una excelente explicación de las técnicas desarrolladas en el presente trabajo.

## 8. BIBLIOGRAFIA.

- /1/ BALAS, E. y CHRISTOFIDES, N.: "A Restricted Lagrangean Approach to the Travelling Salesman Problem", Math. Prog. 21, 19-46 (1981).
- /2/ CHRISTOFIDES, N.: "Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem". Technical Report TR-GS1A, Carnegie-Mellon, Pittsburgh, PA. (1976).

- /3/ CROWDER, H. y PADBERG, M.W.: "Solving Large-Scale Symmetric Travelling Salesman Problems to Optimality". Man.Science 26, 495-509. (1980).
- /4/ DANTZIG, G.B., FULKERSON, D.R. y JOHNSON S.M.: "Solution of a Large-Scale Travelling Salesman Problem". Operations Res. 2, 393-410. (1954).
- /5/ EDMONDS, J.: "Maximum Matching and a Polyhedron with 0,1 vertices", J. Res. National Bureau of Standards, Sect. B, 69, 125-130. (1965).
- /6/ GOLDEN B., BODIN, L., DOYLE, T. y STEWART Jr., W.: "Approximate Traveling Salesman Algorithms". Operations Res. 28, 694-711. (1980)
- /7/ GROTSCHTEL, M. y PADBERG, W.: "On the Symmetric Travelling Salesman. Part I: Inequalities. Part II: Lifting Theorems and Facets", Math. Prog. 16, 265-280 y 281-302. (1979).
- /8/ GROTSCHTEL, M.: "On the Symmetric Travelling Salesman Problem: Solution of a 120-city problem". Math. Prog. Study 12, 61-77. (1980).
- /9/ GROTSCHTEL, M. y PADBERG, W.: "Polyhedral Aspects of the Travelling Salesman Problem I: Theory. Part II: Computation". Technical Reports No 82212 y 82214- OR, Institut für Ökonometrie und Operations Research, Bonn. (1983).
- /10/ HELD, M. y KARP, R.M.: "The Travelling Salesman and Minimum Spanning Trees, Part I and Part II". Operations Res. 18, 1138-1162 y Math. Prog. 1, 6-26. (1970-1971).
- /11/ KARP, R.M.: "Reducibility among Combinatorial Problems" en Complexity of Computer Computations". Miller and Thatcher, eds. Plenum Press. New York, 85-103. (1972).
- /12/ KUHN, H.W.: "On Certain Convex Polyhedra" Bull of the American Math. Soc. 61, 557-558. (1955)
- /13/ LIN, S.: "Computer Solutions of the Travelling Salesman Problem". Bell System Tech. J. 44, 2245-2269. (1965).

- /14/ LIN, S. y KERNIGHAN, B.W.: "An Effective Heuristic Algorithm for the Travelling Salesman Problem". Operations Research 21, 498-516. (1973).
- /15/ LITTLE, J.D.C., MURTY, K.G., SWEENEY, D. W. y KAREL, C.: "An Algorithm for the Travelling Salesman Problem". Operations Research 11, 972-989. (1963).
- /16/ PADBERG, M.W. y HONG, S.: "On the Symmetric Travelling Salesman Problem: A Computational Study", Math. Prog. Study 12, 78-107. (1980).

Elda 96	Santander 36
El Ferrol 59	Santiago 60
Figueres 87	Segovia 37
Gandía 76	Seu d'Urgell 93
Gijón 52	Sevilla 38
Girona 16	Soria 39
Granada 17	Talavera 74
Guadalajara 18	Tarragona 40
Hellín 78	Teruel 41
Huelva 19	Toledo 42
Huesca 20	Torrelavega 68
Irún 61	Tortosa 73
Jaca 91	Utrera 70
Jaén 21	Valencia 43
Jerez 49	Valladolid 44
	Vigo 53
	Villarrobledo 78
	Vitoria 45
	Zamora 46
	Zaragoza 47

### 9. APENDICE A.

Lista alfabética de las 103 ciudades, con el código con el que aparecen en las figuras y en las restricciones.

Albacete 1	León 22
Alcalá de Henares 50	Linares 63
Alcázar de San Juan 85	Lleida 23
Alcoy 54	Logroño 24
Algeciras 57	Lorca 66
Alicante 2	Lucena 86
Almería 3	Lugo 25
Andújar 92	Madrid 26
Antequera 98	Málaga 27
Aranda de Duero 83	Manresa 55
Aranjuez 103	Marbella 65
Avila 3	Mataró 56
Avilés 67	Medina del Campo 100
Badajoz 5	Mérida 81
Barcelona 6	Mieres 97
Baza 90	Miranda del Ebro 82
Bilbao 7	Monforte de Lemos 102
Burgos 8	Motril 89
Cáceres 9	Murcia 28
Cádiz 10	Olot 88
Calahorra 94	Onteniente 77
Calatayud 95	Orense 29
Cartagena 50	Orihuela 101
Castellón 11	Oviedo 30
Ciudad Real 12	Palencia 31
Ciudad Rodrigo 99	Pamplona 32
Córdoba 13	Plasencia 84
La Coruña 14	Ponferrada 64
Cuenca 15	Pontevedra 33
Dalías 80	Puertollano 58
Dos Hermanas 69	Reus 72
Ecija 71	Sagunto 75
Eibar 62	Salamanca 34
Elche 48	San Sebastián 35

### 10. APENDICE B.

Relación de los subtours y los peines que se violan en cada iteración.

#### ITERACION 1

- Subtours: (1)  $W=(25,102,29,53,33,60,14,59)$ .  
 (2)  $W=(30,97,52,67)$   
 (3)  $W=(7,36,68)$   
 (4)  $W=(8,31,44,100,4,37,83)$   
 (5)  $W=(82,24,94,39,95,47,20,91,32,61,35,62,45)$   
 (6)  $W=(9,5,81)$   
 (7)  $W=(26,50,18)$   
 (8)  $W=(58,12,85)$   
 (9)  $W=(92,63,21)$   
 (10)  $W=(13,71,86)$   
 (11)  $W=(19,38,69)$   
 (12)  $W=(17,89,80,3,90)$   
 (13)  $W=(77,76,54)$   
 (14)  $W=(2,48,96)$   
 (15)  $W=(11,75,43)$   
 (16)  $W=(40,72,73)$   
 (17)  $W=(55,56,6)$   
 (18)  $W=(88,87,16)$   
 (19)  $W=(13,71,86,98)$   
 (20)  $W=(101,48,2,96)$   
 (21)  $W=(41,11,75,43)$   
 (22)  $W=(93,55,6,56,16,88,87)$   
 (23)  $W=(23,72,40,73)$   
 (24)  $W=(19,38,69,70)$   
 (25)  $W=(26,18,50,103)$
- Peines: (1)  $W_0=(25,102,64)$ ,  $W_1=(25,29)$ ,  $W_2=(102,29)$ ,  
 $W_3=(64,22)$

ITERACION 2

- Subtours: (26)  $W=(59,25,64,102,29,53,33,60,14)$   
 (27)  $W=(46,34,99)$   
 (28)  $W=(84,81,5,9)$   
 (29)  $W=(13,86,98,27,65,57,49,10,69,38,19,70,71)$   
 (30)  $W=(80,3,89)$   
 (31)  $W=(1,78,96,101,28,51,66,90,17,21,63,92,58,12,85,79)$   
 (32)  $W=(1,78,96,101,28,51,66,90,17,21,63,92,58,12,85,79,48)$   
 (33)  $W=(23,93,88,87,16,56,6,55)$   
 (34)  $W=(23,93,88,87,16,56,6,55,72,40)$   
 (35)  $W=(23,93,88,87,16,56,6,55,72,40,73)$   
 (36)  $W=(61,35,62)$   
 (37)  $W=(68,36,7,45,82,24,94,32,91,20,47,95,39,83,8)$   
 (38)  $W=(67,52,30,97,22)$
- Peines: (2)  $W_0=(48,96,101)$ ,  $W_1=(48,2)$ ,  $W_2=(96,78)$ ,  $W_3=(101,28)$   
 (3)  $W_0=(41,11,75)$ ,  $W_1=(11,73)$ ,  $W_2=(41,15)$ ,  $W_3=(75,43)$

ITERACION 3

- Subtours: (39)  $W=(14,22,25,29,30,33,52,53,59,60,64,67,97,102)$   
 (40)  $W=(1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,23,24,$   
 $26,27,28,31,32,34,35,36,37,38,39,40,41,42,43,44,45,46,47,$   
 $48,49,50,51,54,55,56,57,58,61,62,63,65,66,68,69,70,71,72,$   
 $73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,$   
 $92,93,94,95,96,98,99,100,101,103)$
- Peines: (4)  $W_0=(8,83,31)$ ,  $W_1=(8,68)$ ,  $W_2=(83,37)$ ,  $W_3=(31,44)$   
 (5)  $W_0=(13,58,92)$ ,  $W_1=(13,71)$ ,  $W_2=(58,12)$ ,  $W_3=(92,63)$   
 (6)  $W_0=(7,62,45)$ ,  $W_1=(7,36)$ ,  $W_2=(45,82)$ ,  $W_3=(62,35)$

ITERACION 4

- Subtours: (41)  $W=(95,41,15,18,50,39,83,8,31,44,100,37,4,74,42,103,26)$   
 (42)  $W=(28,51,101)$   
 (43)  $W=(3,66,90,17,89,80)$   
 (44)  $W=(1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,18,19,20,21,22,23,24,$   
 $25,26,27,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,$   
 $45,46,47,48,49,50,52,53,54,55,56,57,58,59,60,61,62,63,64,$   
 $65,67,68,69,70,71,72,73,74,75,76,77,78,79,81,82,83,84,85,$   
 $86,87,88,91,92,93,94,95,96,97,98,99,100,102,103)$   
 (45)  $W=(1,2,5,6,7,9,10,11,12,13,14,16,19,20,21,22,23,24,25,27,29,$   
 $30,32,33,34,35,36,38,40,43,45,46,47,48,49,52,53,54,55,56,$   
 $57,58,59,60,61,62,63,64,65,67,68,69,70,71,72,73,75,76,77,$   
 $78,79,81,82,84,85,86,87,88,91,92,93,94,96,97,98,99,102)$
- Peines: (7)  $W_0=(20,23,47)$ ,  $W_1=(20,91)$ ,  $W_2=(23,93)$ ,  $W_3=(47,95)$   
 (8)  $W_0=(7,62,35,45)$ ,  $W_1=(35,61)$ ,  $W_2=(45,82)$ ,  $W_3=(7,36)$   
 (9)  $W_0=(7,62,35,61,32,94,24,82,45)$ ,  $W_1=(7,36)$ ,  $W_2=(94,34)$ ,  $W_3=(32,91)$   
 (10)  $W_0=(32,91,20,23,47,95,39,94)$ ,  $W_1=(32,61)$ ,  $W_2=(94,24)$ ,  $W_3=(39,83)$   
 $W_4=(95,41)$ ,  $W_5=(23,93)$

ITERACION 5

- Peines: (11)  $W_0=(90,3,66)$ ,  $W_1=(90,17)$ ,  $W_2=(3,80)$ ,  $W_3=(66,51)$   
 (12)  $W_0=(66,3,90,80,89,17,98,27)$ ,  $W_1=(66,51)$ ,  $W_2=(98,86)$ ,  $W_3=(27,65)$   
 (13)  $W_0=(13,58,92)$ ,  $W_1=(13,71)$ ,  $W_2=(58,12)$ ,  $W_3=(92,63,21)$   
 (14)  $W_0=(78,96,101)$ ,  $W_1=(78,1)$ ,  $W_2=(96,48)$ ,  $W_3=(101,28)$   
 (15)  $W_0=(7,62,45)$ ,  $W_1=(7,36)$ ,  $W_2=(45,82)$ ,  $W_3=(62,35,61)$   
 (16)  $W_0=(66,90,3,80,89,17,21,63,92,58,13,71,86,98,27,65,57,49,10,$   
 $70,69)$ ,  $W_1=(66,51)$ ,  $W_2=(58,12)$ ,  $W_3=(69,38)$

**Questió 6** - V. 9, n.º 3 (setembre 1985)

ITERACION 6

Subtours: (46)  $w = (22, 25, 59, 14, 60, 33, 53, 29, 102, 64)$

(47)  $w = (30, 67, 97)$

Peines: (17)  $w_0 = (78, 96, 28, 101)$ ,  $w_1 = (78, 1)$ ,  $w_2 = (96, 48)$ ,  $w_3 = (28, 101, 51)$

(18)  $w_0 = (38, 69, 70, 49, 10, 57, 65, 27, 98, 86, 71, 13, 92, 63, 21, 17, 89, 80, 3, 90, 66)$ ,  $w_1 = (38, 19)$ ,  $w_2 = (92, 58)$ ,  $w_3 = (66, 51)$

(19)  $w_0 = (66, 78, 96, 101, 28, 51)$ ,  $w_1 = (66, 90)$ ,  $w_2 = (78, 1)$ ,  $w_3 = (96, 48)$

(20)  $w_0 = (22, 46, 100, 44, 31)$ ,  $w_1 = (22, 64)$ ,  $w_2 = (46, 34)$ ,  $w_3 = (100, 44, 31, 8, 83, 37, 4)$

(21)  $w_0 = (52, 68, 36, 7, 45, 62)$ ,  $w_1 = (52, 67)$ ,  $w_2 = (45, 82)$ ,  $w_3 = (62, 35)$