1. ARTÍCULOS DE INVESTIGACIÓN OPERATIVA

LA COMBINATORIA POLIÉDRICA Y LOS PROBLEMAS DE RUTAS DE VEHÍCULOS

Angel Corberán[1] y José María Sanchis[2]

[1]Universitat de València. [2]Universidad Politécnica de Valencia

angel.corberan@uv.es, jmsanchis@mat.upv.es

Introducción

Consideremos muy brevemente las siguientes situaciones:

1. Un viajante tiene que visitar una serie de ciudades y luego volver a su ciudad de origen. ¿Cuál es la ruta que debe seguir el viajante de modo que la distancia total recorrida sea mínima?

2. Un taladro láser tiene que realizar n perforaciones en un tablero. Si se conocen las coordenadas de los puntos a perforar, ¿cuál debe ser el recorrido que debe realizar el taladro por los n puntos del tablero para minimizar la distancia total recorrida?

3. Un cartero tiene que pasar por determinadas calles de su ciudad para repartir el correo y luego volver a su oficina. Diseñar la ruta que debe seguir el cartero de modo que la distancia total recorrida sea mínima.

4. De una plancha hay que cortar ciertas figuras cuya posición en el tablero está ya determinada. ¿Cuál debe ser el recorrido que debe hacer el instrumento cortante para cortar todas las piezas de forma que la longitud total recorrida sea mínima?

Hay una gran cantidad de problemas del mundo real que pueden ser formulados en términos parecidos. Pensemos que la mayor parte de organismos públicos y empresas privadas tienen que plantearse problemas relacionados con el reparto del correo, la recogida de basuras, limpieza, inspección o mantenimiento de calles, carreteras, o redes eléctricas, distribución de todo tipo de productos, visitas a clientes, transporte de personal, etc.

Todos estos problemas reales pueden ser modelizados mediante un **grafo**. Recordemos que un grafo es un par $G=(V, E)$, donde V es un conjunto de vértices y E es un conjunto de pares (i,j) de vértices de V que llamaremos enlaces y que pueden ser aristas (si pueden ser recorridas en ambos sentidos) o arcos (si sólo pueden ser recorridos en un determinado sentido). En los ejemplos anteriores, cada ciudad a visitar o cada punto a perforar se representarían por un vértice y cada calle, cada línea a cortar, en general cada trayecto entre dos vértices distintos, se representa por un arco o por una arista, con un valor asociado que corresponda a su coste o longitud. Así, un grafo puede modelizar de modo cómodo y preciso cualquier red "real": redes de calles, de carreteras, de líneas férreas, de líneas eléctricas, informáticas, etc.

Al mismo tiempo, todos los problemas anteriores pueden formularse como **Problemas de Rutas**. Básicamente, estos problemas consisten en el diseño de rutas, o **tours**, sobre los vértices o los enlaces de un grafo que cumplan una serie de condiciones y que tengan un coste total mínimo. Si consideramos **solución posible** del problema a cualquier tour que cumpla todas las condiciones y **solución óptima** a aquella solución posible que tenga coste mínimo, los Problemas de Rutas pueden considerarse también como **Problemas de Optimización Combinatoria,** puesto que el número de soluciones posibles de estos problemas es finito (o numerable).

El objetivo ante tales problemas es diseñar **algoritmos** de resolución que, a partir de los datos numéricos de un ejemplo concreto (que llamamos **instancia**), produzca la solución óptima. Puesto que estamos interesados en resolver situaciones reales y éstas se modelizan con grafos con muchos vértices y enlaces, el número de soluciones posibles es tan grande que la evaluación de todas ellas es inviable. Así pues necesitamos algoritmos eficientes.

En **Teoría de la Complejidad Algorítmica** se acepta que los algoritmos polinómicos (aquellos en los que el número máximo de operaciones básicas a realizar depende polinómicamente del tamaño de la instancia) son eficientes, pues su tiempo de ejecución crece de un modo "razonable" a medida que crece el tamaño de la instancia. De este modo podemos decir que un problema es "fácil" si puede ser resuelto mediante un algoritmo polinómico y que es "difícil" en otro caso. Para la mayor parte de los Problemas de Rutas (y de Optimización Combinatoria en general), no sólo no se conoce ningún algoritmo que sea polinómico sino que, si la conjetura "***P≠NP***" es cierta, tales algoritmos no existen. Estos problemas se conocen como ***NP-difíciles***.

Aunque no sean polinómicos, es importante el desarrollo de algoritmos exactos para problemas NP-difíciles. Nótese que el estudio de la complejidad algorítmica se hace sobre el comportamiento en el peor caso. Es decir, para un problema NP-difícil lo que crece exponencialmente con el tamaño de las instancias es el número *máximo* de operaciones necesarias para resolver *cualquier* instancia de ese tamaño. Pero ocurre que algunas instancias concretas de gran tamaño sí se resuelven en un tiempo razonable. Además, resuelven todas las instancias de hasta un cierto tamaño que cada vez es más grande, sobre todo desde los años 80 en los que la llamada **Combinatoria Poliédrica** ha adquirido un gran auge en el diseño de algoritmos de resolución de Problemas de Optimización Combinatoria. Estos algoritmos se basan en el estudio del **poliedro** definido por la envoltura lineal convexa del conjunto de soluciones posibles y serán presentados con mayor detalle a continuación.

Combinatoria Poliédrica

Sea E un conjunto finito con un coste c_e asociado a cada elemento $e \in E$ (en los Problemas de Rutas, E es el conjunto de enlaces del grafo, que tienen asociado un coste o longitud) y sea un conjunto $\mathcal{F}$ (finito o infinito numerable) de familias de elementos de E, llamadas soluciones posibles (en los Problemas de Rutas son los tours sobre los enlaces del grafo, que pueden ser considerados como una familia de enlaces). El Problema de Optimización Combinatoria (abreviadamente P.O.C.) consiste en encontrar la solución posible de $\mathcal{F}$ de coste mínimo.

Para cada solución posible F de $\mathcal{F}$ definimos su vector de incidencia $x^F = \{x^F{}_e,\ e \in E\} \in Z^{|E|}$, donde $x^F{}_e$ denota el número de veces que e aparece en F. Así, definimos el conjunto convexo $P_\mathcal{F}$= conv $\{x^F: F \in \mathcal{F}\}$.

Para la mayoría de los P.O.C. se puede demostrar que $P_\mathcal{F}$ es un poliedro. Claramente, cada solución posible $F \in \mathcal{F}$ corresponde a un punto de $P_\mathcal{F}$ y cada vértice de $P_\mathcal{F}$ corresponde a una solución posible en F. Por lo tanto, podemos resolver el P.O.C. original resolviendo el problema de Programación Lineal (PL) siguiente:

$$\text{Min } cx$$

$$x \in P_\mathcal{F}$$

Sabemos que un poliedro P puede describirse por un sistema de desigualdades lineales, de forma que cada desigualdad induce una faceta de P (una **faceta** del poliedro P es una cara propia no vacía de P que sea maximal respecto a la inclusión de conjuntos y se corresponde con la idea coloquial de cara de un poliedro). Para poder abordar el problema anterior con las técnicas de la Programación Lineal, necesitamos conocer el sistema lineal que describe el poliedro $P_\mathcal{F}$. Así, el primer ingrediente de estas técnicas es la descripción de las desigualdades concretas que inducen faceta del poliedro asociado al problema que queremos resolver.

Una vez hecho ese estudio teórico del poliedro, aparecen tres inconvenientes importantes. El primero es que el número de desigualdades que lo describen crece exponencialmente con el tamaño de la instancia. Así, generar y almacenar todas las desigualdades del PL anterior es tan ineficiente como la enumeración exhaustiva de todas las soluciones posibles del P.O.C. Una alternativa es comenzar con un subconjunto pequeño de tales desigualdades e ir añadiendo otras a medida que las necesitamos. De este modo, en lugar de generar la descripción completa del poliedro, sólo se genera un número limitado de desigualdades de una zona del poliedro "cercana" al óptimo. Ésta es básicamente la idea detrás de los *algoritmos de planos de corte*:

Algoritmo de Planos de Corte

1. Sea PL_0 una relajación lineal inicial del P.O.C. Hacer k=0.

2. Resolver PL_k. Sea x^k una solución óptima de PL_k.

3. Resolver el problema de identificación de facetas para x^k y P_F.

   3.1 Si se encuentran desigualdades violadas por x^k, definir PL_{k+1} como PL_k más estas desigualdades. Hacer k:=k+1 e ir al paso 2.

   3.2 Si no se encuentra ninguna desigualdad violada, parar.

El núcleo de este proceso es el paso 3, en el que se resuelve el siguiente problema:

Problema de Identificación de Facetas: dado un punto x* y un poliedro P_F, encontrar una desigualdad lineal $f\,x \leq f_0$ que defina una faceta de P_F que sea violada por x*, o bien demostrar que tal desigualdad no existe (pues $x^* \in P_F$).

El método anterior termina en 3.2 en una solución óptima del P.O.C. siempre que conozcamos todas las desigualdades que inducen faceta del poliedro. Esto nos lleva al segundo inconveniente: si la conjetura "**P≠NP**" es cierta, no se puede obtener un conocimiento completo del sistema lineal que represente el poliedro P_F asociado a un problema NP-difícil.

Sin embargo, un conocimiento *parcial* pero suficientemente amplio de dicho sistema lineal es muy útil. En primer lugar, en algunas instancias dicha descripción parcial sí puede ser suficiente para obtener una solución óptima en el paso 3.2. Por otro lado, si terminamos en 3.2 sin tener una solución óptima podemos acudir a un *Branch and Bound* pero con mayor probabilidad de alcanzar el óptimo, ya que tenemos una relajación lineal mucho más ajustada. Y en cualquier caso, aunque no alcancemos la solución óptima, siempre obtendremos una cota inferior al coste óptimo del P.O.C. muy útil para valorar la bondad de las soluciones obtenidas mediante algoritmos heurísticos.

El tercer inconveniente proviene de que si un P.O.C. es NP-difícil, el Problema de Identificación de Facetas asociado a éste es también NP-difícil. De hecho, puede demostrarse que:

***Teorema**: El P.O.C Min $\{cx: x\in P_F\}$ puede ser resuelto por un algoritmo polinómico para cualquier vector entero c si, y sólo si, el Problema de Identificación de Facetas puede resolverse por un algoritmo polinómico para cualquier vector racional x*.*

Así, la destreza en detectar desigualdades violadas por la solución x^k del PL_k actual es lo que determina la eficiencia del algoritmo de planos de corte. En la práctica, las desigualdades conocidas que definen faceta se dividen en diferentes familias y se aborda el problema de identificación de facetas para cada una de estas familias con algoritmos específicos. Es decir, el paso 3 en el esquema del algoritmo de planos de corte está constituido por un conjunto de diferentes subrutinas que son ejecutadas cuando es necesario. Para una familia dada de desigualdades, un *algoritmo de separación exacto* es una rutina que toma la solución x^k y encuentra una o más desigualdades violadas de esa familia (si existe alguna). Un *algoritmo de separación heurístico* es similar excepto que puede fallar en la detección de una desigualdad violada de esa familia. Para agilizar el paso 3 es frecuente diseñar uno o varios heurísticos rápidos para cada familia de desigualdades, y sólo se intenta el algoritmo exacto de separación (si está disponible) cuando los heurísticos fallan.

Esta aproximación ha llevado a resultados importantes en un gran número de Problemas de Optimización Combinatoria. Como ejemplo quisiéramos destacar que los resultados obtenidos en el TSP y el CVRP son impresionantes. También en otros problemas no tan conocidos, pero tan difíciles o más que los mencionados y que se han resistido durante décadas a los esfuerzos de los investigadores.

Una clasificación de los Problemas de Rutas

Consideremos de nuevo los problemas del viajante y del cartero expuestos en la Introducción. Nótese que mientras que en el primero la demanda de servicio tiene lugar en los vértices del grafo (ciudades), en el segundo la demanda tiene lugar en los enlaces (calles). Esta diferencia en la localización de la demanda proporciona el principal criterio para la clasificación de los Problemas de Rutas: Problemas de Rutas por **Vértices** y Problemas de Rutas por **Arcos** (Enlaces).

Frecuentemente, las zonas de demanda son tan grandes que no pueden ser atendidas por un solo vehículo. En estos casos, se dispone de una flota de vehículos y el problema que se plantea es diseñar una ruta para cada vehículo, de modo que se satisfaga la demanda global. De este modo, también se pueden clasificar los problemas de rutas según que sean una o varias las rutas a diseñar.

Otro criterio de clasificación de los problemas de rutas se basa en las características del grafo sobre el que están definidos, es decir, según que el grafo sea **no dirigido** (todos sus enlaces son aristas, que pueden recorrerse en ambos sentidos), **dirigido** (todos sus enlaces son arcos, que sólo pueden recorrerse en un sentido), **mixto** (con aristas y arcos simultáneamente) o un grafo "***windy***" (un grafo no dirigido en el que el coste de atravesar una arista *(i,j)* desde *i* a *j* puede ser diferente al de atravesarla de *j* a *i*).

A modo de resumen, la tabla siguiente presenta una clasificación de los Problemas de Rutas básicos, donde cada problema puede ser considerado sobre un grafo no dirigido, dirigido, mixto o windy.

	Demandas en los vértices	Demandas en los enlaces
1 vehículo	TSP Problema del Viajante GTSP Problema del Viajante Gráfico	CPP Problema del Cartero Chino RPP Problema del Cartero Rural
	GRP Problema General de Rutas	
k vehículos	CVRP Problema de Rutas de Vehículos con Capacidades	CARP Problema de Rutas por Arcos con Capacidades

El **Problema del Viajante** (Traveling Salesman Problem, TSP) es sin duda el problema de rutas de vehículos que más ha sido estudiado en la literatura científica. El planteamiento clásico de este problema consiste en construir un grafo completo con un vértice representando a cada una de las ciudades y una arista *(i,j)* entre cada par de vértices *i*, *j* con un coste asociado igual a la distancia entre las respectivas ciudades:

Dado un grafo completo G = (V,E) con costes no negativos c_e *,* $e \in E$*, encontrar el tour de coste mínimo que pase exactamente una vez por cada vértice del grafo.*

El **Problema del Viajante Gráfico** (Graphical Traveling Salesman Problem, GTSP) es una variante del TSP introducida por Fleischmann (1985) y Cornuèjols, Fonlupt y Naddef (1985) en la que el grafo no tiene por qué ser completo y la ruta debe pasar al menos una vez (no necesariamente sólo una vez) por cada vértice:

Dado un grafo G = (V,E) con costes no negativos c_e*,* $e \in E$*, encontrar un tour en G de coste mínimo que pase por cada vértice* $v \in V$

al menos una vez.

Las ventajas de esta formulación frente a la del TSP son que el poliedro de soluciones es de dimensión completa y que son necesarias menos variables, pues se trabaja directamente sobre el grafo que modeliza la red viaria, que dista mucho de ser un grafo completo. Si el grafo es dirigido, tenemos el Graphical Asymmetric Traveling Salesman Problem (GATSP, Chopra y Rinaldi 1996). Si el grafo es mixto o windy, el problema es equivalente al GATSP.

En cuanto a los Problemas de Rutas por Arcos, tenemos el **Problema del Cartero Chino** (Chinese Postman Problem, CPP), planteado por el matemático chino Meigu Guan (1962):

Dado un grafo $G = (V,E)$ con costes no negativos c_e, $e \in E$, encontrar un tour de coste mínimo que recorra cada enlace $e \in E$ al menos una vez.

Este problema es resoluble polinómicamente en un grafo no dirigido. También lo es en el caso dirigido y se les puede considerar, por lo tanto, problemas resueltos. En cambio, el CPP sobre un grafo mixto es un problema NP-difícil. También lo es la versión windy, el llamado **Problema del Cartero con Viento** (Windy Postman Problem, WPP), introducido por Minieka (1979).

El **Problema del Cartero Rural** (Rural Postman Problem, RPP) es una generalización del problema anterior en el sentido de que aquí no es necesario pasar por todos los enlaces del grafo sino solamente por un subconjunto de ellos que llamamos enlaces requeridos:

Dado un grafo $G = (V,E)$ con costes no negativos c_e, $e \in E$ y dado $E_R \subseteq E$ un subconjunto no vacío de enlaces de G que llamaremos requeridos, encontrar un tour en G que recorra cada enlace de E_R al menos una vez y que tenga coste mínimo.

Propuesto por Orloff (1974), el RPP es NP-difícil tanto sobre un grafo no dirigido como dirigido (DRPP), mixto (MRPP) o windy (WRPP).

El **Problema General de Rutas** (General Routing Problem, GRP) fue también definido por primera vez por Orloff (1974) y es el caso más general de problemas de rutas con un único vehículo, pues la demanda se puede encontrar tanto en las aristas como en los vértices del grafo:

Dado un grafo $G = (V,E)$ con costes no negativos c_e, $e \in E$, dados un subconjunto de enlaces requeridos $E_R \subseteq E$ y un subconjunto de vértices requeridos $V_R \subseteq V$, encontrar un tour en G de coste mínimo que recorra cada enlace requerido y cada vértice requerido al menos una vez.

Este problema es una generalización de los anteriores pues, como casos particulares, si $E_R=E$ resulta el CPP, si $V_R=\emptyset$ tenemos el RPP y si $E_R= \emptyset$ y $V_R=V$ el GTSP. Por lo tanto, el GRP es también un problema NP-difícil, así como sus versiones dirigida (DGRP), mixta (MGRP) y windy (WGRP).

Obviamente, el WGRP generaliza al GRP (si $c_{ij} = c_{ji}$), pero también al DGRP, ya que cada arco (i,j) con coste c puede ser modelizado con una arista con costes $c_{ij} = c$ y $c_{ji} = \infty$, y por lo tanto también al MGRP. Así, el WGRP generaliza a todos los Problemas de Rutas por Arcos con un sólo vehículo definidos sobre cualquier tipo de grafo y también a algunos de Rutas por Vértices.

Respecto de los problemas con varios vehículos, tenemos el **Problema de Rutas por Arcos con Capacidades** (Capacitated Arc Routing Problem, CARP), introducido por Golden y Wong (1981), en el que cada enlace (i,j), además de un coste que representa su longitud o el tiempo de ser atravesada, tiene asociada una *demanda* q_{ij} no negativa. Se dispone de una flota de vehículos de capacidad W con base en un determinado vértice (depósito). El objetivo es diseñar una ruta para cada vehículo de modo que, conjuntamente, pasen por cada enlace (i,j) con $q_{ij} >0$, que la suma de las demandas atendidas por cada vehículo no exceda su capacidad W y que la suma de los costes de todas las rutas sea mínima.

El **Problema de Rutas de Vehículos con Capacidades** (Capacitated Vehicle Routing Problem, CVRP), es similar al CARP excepto que ahora es cada vértice (cliente) i quien tiene asignado una demanda q_i no negativa. Tiene como caso particular al TSP y es un problema tremendamente difícil pero con un gran número de aplicaciones prácticas.

En todos los problemas anteriores, pueden exigirse **otras condiciones** adicionales. Algunas de ellas, como ciertas relaciones de precedencia entre servicios, pueden estar determinadas por la propia naturaleza de los servicios a realizar o por simple preferencia de los clientes. También más restrictivos son los problemas con "time windows", o ventanas de tiempo, en los que cada servicio debe ser realizado entre unos determinados márgenes temporales. A veces, determinados movimientos del vehículo en el grafo están penalizados o prohibidos, como ocurre a menudo en algunos giros de unas calles a otras adyacentes que están prohibidos por las reglas del tráfico urbano.

Nuestro trabajo

Fue Nicos Christofides quien nos introdujo, hacia 1980, en el apasionante mundo de los Problemas de Rutas de Vehículos, concretamente en los de Rutas por Arcos. Desde entonces la mayor parte de nuestra investigación se ha centrado en el estudio y resolución de estos problemas y, entre otras herramientas, hemos utilizado la Combinatoria Poliédrica. Como se ha visto, esto supone el estudio de los poliedros asociados a esos problemas, el diseño de algoritmos heurísticos y exactos de separación y su inclusión en procedimientos de resolución basados en el método de relajación para los P.O.C. descrito anteriormente. Creemos que hemos conseguido algunos resultados interesantes concernientes a la descripción de los poliedros asociados con el CVRP (por lo que respecta a los Problemas de Rutas por Vértices) y con varios Problemas de Rutas por Arcos. Concretamente con el CPP, el RPP y el GRP definidos sobre grafos no dirigidos, dirigidos, mixtos y windy. También hemos intentado contribuir a la resolución aproximada y exacta de estos problemas. Los resultados obtenidos han sido bastante buenos y, además de su valor por sí mismos, han confirmado la validez de la Combinatoria Poliédrica como área de trabajo y estrategia de resolución. Creemos que no es éste lugar para detalles, por eso no los habrá hallado el lector. Aquél que desee una mayor información quizás encuentre interesantes los trabajos de Benavent, Corberán y Sanchis (2000) y Corberán, Plana y Sanchis (2005).

Agradecimientos: Los autores desean agradecer la ayuda proporcionada por el Ministerio de Ciencia y Tecnología y la Generalitat Valenciana a través de los proyectos TIC2003-05982-C05-01 y GRUPOS03/189, respectivamente.

Referencias

E. Benavent, A. Corberán y J.M. Sanchis (2000): "Linear Programming Based Methods for Solving Arc Routing Problems". In M. Dror (Ed.): *Arc Routing: Theory, Solutions and Applications*. Kluwer Academic Publishers.

S. Chopra y G. Rinaldi (1996): "The Graphical Asymmetric Traveling Salesman Polyhedron: Symmetric Inequalities". *SIAM J. Discrete Math*. 9, 602-624.

A. Corberán, I. Plana y J.M. Sanchis (2005): "On the Windy General Routing Polyhedron". Technical Report. Department of Statistics and OR, University of Valencia (Spain).

G. Cornuèjols, J. Fonlupt y D. Naddef (1985): "The traveling salesman problem on a graph and some related integer polyhedra". *Mathematical Programming* 33, 1-27.

B. Fleischmann (1985): "A cutting-plane procedure for the traveling salesman problem on a road network". *European Journal of Operational Research* 21, 307-317.

B. Golden y R. Wong (1981): "Capacitated Arc Routing Problems". *Networks* 11, 305-315.

M. Guan (1962): "Graphic Programming using odd and even points". *Chinese Mathematics* 1, 273-277.

E. Minieka (1979): "The Chinese Postman Problem for Mixed Networks". *Management Science* 25, 643-648.

C. Orloff (1974): "A fundamental problem in vehicle routing". *Networks* 27, 95-108.