71.14 Modelos y Optimización I

El Problema del Viajante

Conceptos, Variaciones y Soluciones Alternativas

Lic. Silvia A. Ramos

"Según los Viajeros Perdidos, recorrer el mundo es la única forma de alcanzar la cultura y aún la sabiduría. La afirmación no parece muy consistente: la calle está llena de sujetos que han recorrido los cinco continentes, permaneciendo en la más inmaculada ignorancia. De cualquier modo, los kilómetros transitados y los países conocidos otorgan rango y jerarquía en este circulo".

"[...] Todo viajero es la mitad de sí mismo. No hay lugar en los aviones para llevar las cosas que lo completan. Esquemas, gestos, personas, vientos, olores, tapiales, saludos, colores, miradas, no caben en las valijas".

"[...] No está mal contemplar las catedrales góticas, los canales de Venecia o la Gran Muralla. Sí, está mal creer que esas contemplaciones darán sentido a la vida. Para encontrarse uno mismo no es necesario caminar mucho. Se los digo yo, que me he rastreado por todas partes y me encontré en el patio de mi casa., cuando ya era demasiado tarde".

Alejandro Dolina

Crónicas del Ángel Gris

La autora agradece la colaboración del Licenciado Sebastián Ceria de la Universidad de Carnegie Mellon (EEUU) en el desarrollo de casos reales para el problema del viajante y bibliografía adicional.

El problema del viajante

El problema del viajante siempre fue de gran atención por sus características especiales. Se trata de un problema muy sencillo de contar pero de resolución no tan fácil.

¿Por qué se llama así? Porque la enunciación más común de este problema habla de un viajante de comercio que tiene que partir de su casa para visitar una serie de clientes, ubicados cada uno en una ciudad distinta, antes de retornar finalmente a su casa. Sin embargo, podemos plantear muchos ejemplos similares: un corredor de una fábrica que debe recoger los pedidos de los clientes, un cadete que debe realizar trámites en varios bancos y volver a su oficina, etc. Volveremos a estos ejemplos más tarde pero ahora trataremos de plantear el problema teóricamente y para eso usaremos el enunciado clásico:

- Un viajante debe salir de la ciudad de origen y visitar N ciudades.
- Cada ciudad puede ser visitada una sola vez
- El costo asociado de ir de una ciudad a la otra es conocido (distancia, costo del pasaje, etc.).
- Al terminar el recorrido debe volver a la ciudad de origen.

El caso consiste en encontrar el orden en el cual debe visitar a cada una de las ciudades para minimizar la distancia recorrida (generalmente, esta distancia está asociada a un costo de transporte, si sólo minimizamos la distancia, esto quiere decir que para distancias iguales, suponemos costos iguales).

Este problema es de secuenciamiento con ciclo. Los problemas de secuenciamiento más comunes son las redes, pero en el caso del viajante la red se cierra y hay una ubicación de la que se parte que es recorrida dos veces, de allí el ciclo.

Entonces desarrollaremos una formulación del problema:

Un viajante de comercio tiene que partir de su casa, ubicada en la que numeraremos como ciudad cero, visitar n clientes, ubicados cada ellos en una ciudad distinta y volver a su casa, recorriendo la mínima distancia total.

A cualquier solución de este problema se la llama "tour" (sea óptima o no).

Supongamos que las ciudades a ser visitadas están numeradas 1,2... n. Utilizaremos las variables Y_{ij} con el siguiente significado:

$Y_{ij} = 1$ si el tour va directamente de la ciudad i a la j

$Y_{ij} = 0$ sino

El concepto "directamente" es muy importante porque quiere decir que la Y_{ij} sólo tomará valor si el viajante va desde la ciudad i a la j sin pasar por ninguna otra ciudad en el medio.

El objetivo será minimizar $\sum_{i=0}^{n}\sum_{j=0}^{n} C_{ij}Y_{ij}$ donde C_{ij} es la distancia entre las ciudades i y j.

Hay dos condiciones obvias que deben cumplirse:

- Exactamente una ciudad debe ser visitada inmediatamente después de la ciudad i.
- Exactamente una ciudad debe ser visitada inmediatamente antes de la ciudad j.

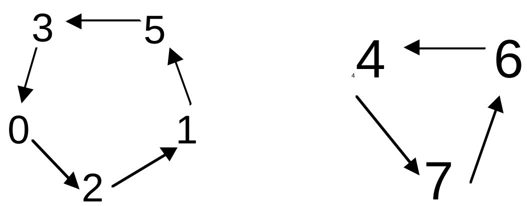
Estas condiciones se expresan del siguiente modo:

$$\sum_{j=0}^{n} Y_{ij} = 1 \qquad \forall i = 0,1,2,\ldots n \qquad i \neq j$$

$$\sum_{i=0}^{n} Y_{ij} = 1 \qquad \forall j = 0,1,2,\ldots n \qquad i \neq j$$

Pero estas restricciones no bastan para plantear el modelo y a continuación lo demostraremos:

Si tuviéramos que recorrer 7 ciudades podríamos obtener la siguiente solución:



Es simple comprobar que esta solución cumple con todas las restricciones del modelo.

La solución anterior incluye "subtour". Un subtour es un tour que no incluye la ciudad 0 o que no incluye todas las ciudades aunque empiece y termine en la cero. Debemos agregar restricciones extras para evitar los subtours. Estas restricciones, si bien son necesarias, complican un poco la resolución del modelo. Una forma de hacerlo será evitando que haya un subtour de dos ciudades (si voy de una ciudad a otra no vuelvo a la primera) de tres ciudades (si $Y_{ij} = 1$ e $Y_{jk} = 1$ implica que Y_{ki} no puede ser 1) y así evitando los subtours de 4,5 y 6 ciudades. En general hay que evitar los subtours desde 2 ciudades hasta (n-1) ciudades. Esto implicaría agregar muchísimas restricciones. Para dar un ejemplo, si quiero restringir que no haya subtour de 2 ciudades tendré que plantear 6 + 5 + 4 + 3 + 2 ecuaciones y todavía nos faltan agregar las restricciones de 3, 4, 5 y 6 ciudades (¡prueben y verán!) En la década del ‘60, Miller y Tucker desarrollaron una fórmula que plantea ecuaciones relacionando todas las ciudades de a pares (salvo la cero) y que evita los subtours. Para plantear las hay que agregar una variable U_i por cada ciudad distinta de la cero. Esa variable indica el orden en que la ciudad es visitada.

Las restricciones a agregar son de la forma:

$$U_i - U_j + nY_{ij} \leq n-1 \qquad \forall i, j = 1\ldots n \qquad i \neq j$$

Analicemos cuidadosamente el comportamiento de las variables que agregamos

- Las U_i son variables enteras: El significado de la variable indica su naturaleza entera pero no es necesario definirlas así. Al estar relacionadas con variables y coeficientes enteros, el algoritmo de resolución de Programación Lineal Entera dará valores enteros a dichas variables.
- Secuencia: dijimos que el conjunto de variables U_i definen una secuencia de visitas a las ciudades del problema. Por secuencia entendemos que el valor la variable asociada a la ciudad de partida de un tramo es menor que la variable asociada a la ciudad de llegada del mismo tramo.

Esto queda asegurado en los vínculos en los que $Y_{ij} = 1$. Si $Y_{ij} = 1$ significa que la ciudad i precede inmediatamente a la ciudad j.

Lo demostraremos con la relación:

$$U_i - U_j + n \cdot Y_{ij} \leq n - 1$$

Cuando los Y_{ij} son iguales a 1

$$U_i - U_j + n \leq n - 1$$

$$U_i - U_j \leq n - 1 - n$$

$$U_i - U_j \leq -1$$

$$U_i \leq U_j - 1$$

$$U_j \geq U_i + 1$$

Es decir, el valor de la variable de llegada debe ser mayor que la variable de partida en por lo menos una unidad. A continuación demostraremos que difieren exactamente en una unidad. A esto se le llama correlación unitaria.

- Correlación Unitaria: El significado de las variables U_i es el número de secuencia en el cual la ciudad i es visitada.

Por ejemplo, si $U_3 = 1$ y $U_5 = 2$ en este caso la ciudad 3 es visitada en primer lugar y luego se visita la ciudad 5.

Para que esto ocurra debe cumplirse que $U_j = U_i + 1$.

Por el momento, demos por cierto que el conjunto de restricciones que hemos agregado impide los subtours. En el conjunto de variables U_i habrá una U_l que indique que es la primera ciudad visitada y una U_k que indique cual es la última ciudad visitada antes de retornar a la ciudad de partida (ciudad 0).

Debemos recordar que el planteo agregado no involucra la ciudad 0 y desde la última ciudad sólo se puede llegar a la ciudad 0 (la última ciudad y la primera después de la cero no se relacionan). Por lo tanto $Y_{kl} = Y_{lk} = 0$

Si planteo:

$$U_k - U_l + nY_{kl} \leq n-1$$

Como $Y_{kl} = 0$

$$U_k - U_l \leq n-1$$

Es decir, la diferencia entre 2 ciudades en secuencia es unitaria, ya que, de ser de otra manera, habrá dos ciudades con el mismo número de secuencia, y eso, como vimos en el planteo anterior es imposible. Usando nuestro ejemplo, en el cual teníamos 7 ciudades, sabemos que U_l debe ser = 1 y U_k debe ser = 7.

$$U_k - U_l \leq 6$$

$$7-1 \leq 6$$

Como U_k - U_l = 6, U_k no puede ser mayor que 7, si U_l era igual a 1. Pero, ¿puede ser menor? Si U_k fuera igual a 3, el U_i de la segunda ciudad en ser visitada debería ser igual a 2, el de la tercera igual a 2, etc. Pero como existe la secuencia que demostramos antes, si el U_i de la segunda ciudad en ser visitada es igual a 2, el de la tercera (la siguiente) debe ser mayor o igual que 3, entonces, si ningún número de orden puede repetirse y el mayor debe ser menor o igual que n - 1 se concluye que hay que ir aumentando el valor de las U_i de a una unidad.

- Valor Inicial: En los ejemplos que hemos visto anteriormente la primera ciudad visitada vale 1 pero esto no es necesariamente cierto. Lo importante es que, si el viajante debe recorrer n ciudades y la primera ciudad tiene U_l = A, la última ciudad tendrá U_k = A + (n - 1). No depende de cuál sea el valor de partida (ni siquiera de si es entero o no) porque las propiedades de secuencia y correlación unitaria aseguran que los valores de U_i aumentarán exactamente de a una unidad desde U_l hasta U_k.
- Evitar subtours:

Veámoslo en nuestro ejemplo:

$$U_4 - U_6 + 7Y_{46} \leq 6$$

$$U_6 - U_7 + 7Y_{67} \leq 6$$

$$U_7 - U_4 + 7Y_{74} \leq 6$$

Según el subtour planteado, $Y_{46} = Y_{67} = Y_{74} = 1$

Supongamos la secuencia 1, 2, 3 para las ciudades 4, 6 y 7 (U_4 = 1, U_6 =2, U_7 = 3). Reemplazando en las inecuaciones anteriores

$$1-2+7 \leq 6 \qquad 6 \leq 6$$

$$2 - 3 + 7 \leq 6 \qquad 6 \leq 6$$

$$3 - 1 + 7 \leq 6 \qquad 9 \leq 6 \qquad \text{no cumple}$$

De igual manera, podemos demostrar que impide que se haga cualquier subtour que no incluya la ciudad 0.

La clave es que, al considerar un subtour, una de las restricciones reflejará la llegada a la primera ciudad partiendo de la última y la diferencia U_i - U_j será positiva y no cumplirá la condición. Entonces demostramos que al agregar esas condiciones se impide cualquier subtour que quiera plantearse.

¿Por qué no evita el tour completo? Porque incluye la ciudad 0, y como las U_i están definidas para cualquier ciudad salvo la cero, entonces, ésa es la única ciudad a la cual se puede llegar después de haber salido de ella, de tal modo que Y_{0l} = Y_{k0} = 1 donde l es la primera ciudad y k es la última. En el resto de los casos, esto no es posible.

Problema genérico del viajante

Ahora que comprendimos el problema clásico del viajante, podemos analizar el problema genérico que fue desarrollado por Miller, Tucker y Zemlin. El enunciado es el siguiente:

Se pide a un viajante que visite cada una de n ciudades. Sale de la "ciudad base" marcada con 0 (cero), para visitar cada una de las ciudades una sola vez, teniendo que regresar a la ciudad 0 finalmente. Durante sus viajes, debe retornar a la ciudad 0 exactamente "t" veces, incluyendo su regreso final y no debe visitar más de "p" ciudades en cada vez o vuelta. Se requiere encontrar un itinerario tal que minimice la distancia total recorrida por el viajante.

La solución sería plantear un modelo que minimice:

$$z = \sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} Y_{ij}$$

donde las d_{ij} son constantes que representan la distancia entre la ciudad i y la ciudad j

¿Cuántas Vueltas tiene que dar el viajante? ¿*t*? Seguramente que t. Y p $\geq$ n, sino no tiene solución.

Sea Yv_{ij} una bivalente que indica si voy de la ciudad i a la j en la vuelta v

1) $\sum_{i=0}^{t} Yv_{ij} = Y_{ij} \leq 1$ (sólo debo llegar una vez a cada ciudad)

Además, de cada ciudad debo llegar a una e ir a otra.

2) $\sum_{i=0}^{n} Y_{ij} = 1 \quad \forall j = 0,1,2,...n \quad i \neq j$ $\qquad \sum_{j=0}^{n} Y_{ij} = 1 \quad \forall i = 0,1,2,...n \quad i \neq j$

¿Cómo evito subtours por vuelta? En cada vuelta recorro p ciudades. Necesito agregar una U_i para cada ciudad en cada vuelta (en cada una de las vueltas habrá una segunda, tercera ciudad, etc., hasta p). Para eso defino las variables Uv_i (donde i indica la ciudad y v el número de vuelta).

3) $Uv_i - Uv_j + p \cdot Yv_{ij} \leq p - 1 \quad \forall i, j = 1...n \quad i \neq j$

Debo relacionar cada Uv_i con una bivalente Wv_i (así evito que la ciudad sea visitada en más de una vuelta).

4) $m \cdot Wv_i \leq Uv_i \leq M \cdot Wv_i$

5) $\sum_{v=1}^{n} Wv_i = 1 \quad \forall i = 1...n$ para que la ciudad i figure una vez sola (en una sola vuelta)

Como *t* es fija debo agregar:

6) $\sum_{i=1}^{n} Y_{i0} = t$

7) $\sum_{i=1}^{n} Y_{0i} = t$

(porque en cada una de las *t* vueltas salgo y vuelvo a la ciudad cero)

Aclaremos que en la última vuelta para las ecuaciones de tipo (3) habrá que cambiar *p* por la cantidad de ciudades visitadas (si ese número es menor que p).

Problema del viajante con límite de tiempo por ciudad y por itinerario completo con distintos medios de. transporte entre ciudades

Supongamos que el viajante del problema original debe realizar sus visitas a las ciudades con un límite de tiempo dado T_i (el tiempo límite para llegar a la ciudad i).

Además existe un tiempo límite para finalizar el viaje entero. Lo representaremos con T.

El tiempo de permanencia del viajante en cada ciudad no es necesariamente el mismo para todas. Sea P_i el tiempo de permanencia en la ciudad i.

Para viajar de una ciudad a la otra se dispone de distintos medios de transporte, cada uno con un tiempo y un costo asociado. Sea Vh_{ij} el tiempo de viaje desde la ciudad i hasta la ciudad j usando el medio de transporte h y sea Ch_{ij} el costo correspondiente a dicho viaje. También suponemos S medios de transporte.

Suponemos que las Vh_{ij}, Ch_{ij}, P_i, T_i, T se conocen de antemano y por lo tanto son constantes.

Para poder representar los viajes entre las distintas ciudades usaremos las Y_{ij}.

Sabemos que a cada ciudad se debe llegar solamente una vez:

$$\sum_{i=0}^{n} Y_{ij} = 1 \qquad \forall j = 0,1,2,\ldots n \qquad i \neq j$$

También sabemos que desde cada ciudad se debe salir exactamente hacia una de las demás:

$$\sum_{j=0}^{n} Y_{ij} = 1 \qquad \forall i = 0,1,2,\ldots n \qquad i \neq j$$

Además, debemos incluir las condiciones que evitan subtours:

$$U_i - U_j + n \cdot Y_{ij} \leq n-1 \qquad \forall i, \forall j = 1,2,\ldots n \qquad i \neq j$$

Podemos pasar entonces a las condiciones particulares de este problema.

Como hay varios medios de transporte, debemos agregar al problema las siguientes variables:

Y_{ijh}: vale 1 si el viajante va de i a j utilizando el medio de transporte h y vale cero sino.

En primer lugar, si se hace un viaje entre las ciudades i y j, debe hacerse en un único medio de transporte (y si no se hace el viaje, no debe usarse ningún medio de transporte para realizarlo):

$$\sum_{h=0}^{S} Yh_{ij} = Y_{ij} \qquad \forall i, \forall j = 1,2,\ldots n \qquad i \neq j$$

Como existe un límite de tiempo para poder llegar a cada una de las ciudades necesitamos saber, cuando estamos en una determinada ciudad, por cuáles ciudades ya hemos pasado (para poder calcular el tiempo acumulado hasta el momento, tanto de viaje como de estadía) Pero las Y_{ij} solamente nos permiten saber cuál fue la ciudad inmediatamente anterior a una determinada ciudad j y nosotros necesitamos saber cuáles son todas las ciudades por las cuales hemos pasado antes de visitar la ciudad j. Para esto necesitaremos agregar variables:

W_{ij}: vale 1 si la ciudad i se visitó con anterioridad a la ciudad j y vale cero sino.

La definición de las variables W_{ij} nos muestra que son variables indicativas, por lo tanto tenemos que agregar restricciones para que las W_{ij} actúen como las hemos definido.

En los valores de las U_i tendremos el orden en el cual se visita cada una de las ciudades, por lo tanto podemos usarlas para determinar el valor de las W_{ij}.

Para ejemplificar al plantearlas, supongamos que la ciudad j es la 5

$$U_5 \le U_i + M \cdot W_{i5}$$

Si la ciudad i se visitó con anterioridad a 5, U_5 es mayor que U_i, entonces necesitamos que W_{i5} sea igual a 1 para sumar M en el segundo miembro.

$$U_i \le U_5 + M \cdot (1 - W_{i5})$$

Si la ciudad i se visitó con posterioridad a 5, U_5 es menor que U_i, entonces necesitamos que W_{i5} sea igual a cero para sumar M en el segundo miembro.

Genéricamente, para darles valor a cada W_{ij}, con j fijo de 1 a n:

$$U_j \le U_i + M \cdot W_{ij} \quad \forall i, \forall j = 1,2,\ldots n \quad i \ne j$$

$$U_i \le U_j + M \cdot (1 - W_{ij}) \quad \forall i, \forall j = 1,2,\ldots n \quad i \ne j$$

Entonces ahora tenemos que averiguar, al llegar a cada ciudad, cuánto tiempo transcurrió antes de llegar a ella. Ese tiempo se divide en tiempo de estadía en las ciudades anteriores y tiempo de viaje.

Por ejemplo, calculemos el tiempo de estadía acumulado antes de llegar a la ciudad 4:

$$TEA_4 = P_1 W_{14} + P_2 W_{24} + P_3 W_{34} + P_5 W_{54} + \ldots + P_n W_{n4} +$$

Genéricamente, el tiempo de estadía acumulado antes de llegar a cualquier ciudad j, con j de 1 a n:

$$TEA_j = \sum_{\substack{i=1 \\ i \ne j}}^{n} P_i \cdot W_{ij}$$

También debemos averiguar el tiempo de viaje acumulado antes de llegar a cada ciudad j. Si queremos calcular el tiempo de viaje anterior a la ciudad 7 (que designaremos con la variable TVA_7), éste resultará como:

TVA_7 = TVINICIAL + $TVACUMHTA_7$

El TVINICIAL es el tiempo de viaje desde la ciudad 0 hasta la primera ciudad visitada. Es fácil de calcular porque solamente hay un Y_{h0j} que vale 1:

1

$$TVINICIAL = \sum_{i=1}^{n} V_{h0j} \cdot Y_{h0j}$$

Para calcular $TVACUMHTA_7$ (tiempo de viaje acumulado hasta llegar a 7) necesitamos considerar solamente las Y_{hij} donde la ciudad i fue visitada con anterioridad a 7. Por eso crearemos nuevas variables Z_{hij7} que valen 1 si Y_{hij} vale 1 y W_{i7} también vale 1:

$2\ Z_{hij7} \leq y_{hij} + W_{i7} \leq 1 + Zh_{ij7}$

Por lo tanto $TVACUMHTA_7 = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{h=1}^{S} V_{hij} \cdot Z_{hij7}$ ahora podemos calcular:

Genéricamente, para calcular el tiempo de viaje acumulado antes de cada ciudad k:

TVA_k = TVINICIAL + $TVACUMHTA_k$

Donde $TVACUMHTA_k$ se calcula como vimos en el caso de $TVACUMHTA_7$ para cada k=1...n.

El tiempo de llegada a cada ciudad no debe superar el límite establecido para llegar a ella:

$TVA_K \leq T_k$

Además, el tiempo total no debe superar T:

$$TUINICIAL + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{h=1}^{S} V_{hij} \cdot Y_{hij} + TUFINAL \leq T \qquad TUFINAL = \sum_{i=1}^{n} \sum_{h=1}^{S} V_{hi0} \cdot Y_{hi0}$$

Y el funcional debe minimizar el costo total del viaje:

$$Z(MIN) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{h=1}^{S} C_{hij} \cdot Y_{hij7}$$

Un problema más complicado de lo imaginado

Suponemos que después de todo lo que explicamos anteriormente acerca del problema del viajante no supondrán que se trata de un problema sencillo. Pero la complejidad no es sólo porque encontrar un algoritmo y una serie de ecuaciones sea difícil sino por su resolución.

Los especialistas en matemática clasifican a los problemas en cuanto a la dificultad que hay para resolverlos mediante la teoría de la complejidad. Esta teoría clasifica el problema de acuerdo con los ejemplos más difíciles que se puedan encontrar' de él. En el caso del viajante, según la cantidad de ciudades y la distribución de las mismas en el plano se puede complicar muchísimo. Tanto, que la teoría de la complejidad clasifica al problema del viajante como NP-Hard (NP = No Determinístico Polinomial).

Todos los problemas pertenecientes a esa clase son difíciles. Cualquier algoritmo que se conoce para resolver el problema lleva un número exponencial de pasos (aproximadamente 2^n) en el peor de los casos.

Esta clasificación es muy importante en el sentido de la resolución numérica, en cuanto a la cantidad de tiempo que llevará resolverlo exactamente. En la tabla siguiente se muestra un problema del viajante con determinada cantidad de ciudades, en base a la teoría de la complejidad.

Cantidad de ciudades	10	20	30	50	60
Tiempo que demora en ser resuelto	0,001 seg.	1 segundo	17,9 min.	35,7 años	366 siglos

¿Qué sucedería con una computadora 100 veces más rápida?

En este tipo de problemas, si antes podíamos resolver un caso con una cantidad N de ciudades, ahora con la nueva computadora, podríamos resolver un problema de tamaño "N + 9,97". La diferencia es tan escasa porque se trata de un problema NP-Hard.

Esto llevó a los matemáticos a la siguiente conclusión: "No se cree que exista un algoritmo ‘eficiente’ que pueda resolver todos los problemas del viajante de comercio".

Por algoritmo eficiente se entiende uno que demore una cantidad razonable de tiempo. Está claro que 366 siglos no es un tiempo razonable. Pero que las conclusiones anteriores no nos lleven a desesperar. Siempre se puede encontrar una solución aproximada (el tiempo anterior es el que lleva encontrar la solución óptima exacta para el problema más difícil que tenga 60 ciudades).

Una opción para resolver este problema es aplicar métodos de soluciones aproximadas (heurísticas) que consisten en encontrar soluciones que no estén garantizadas a ser óptimas pero que estén cerca del óptimo. Hay muchas heurísticas pero eso es tema para un desarrollo aparte. Sólo daremos un pequeño ejemplo para saber de qué se trata. Por lo general las heurísticas se utilizan para encontrar una solución inicial que proporcione un punto de partida para sucesivos mejoramientos. Cuanto más rudimentaria sea la heurística, más alejada del óptimo estará la solución encontrada pero más sencillo y rápido será haberla hallado.

Estos métodos se utilizan porque algunos problemas del viajante no son sencillos de resolver (independientemente de la cantidad de ciudades que tengan) y demoran muchísimo tiempo con los algoritmos de programación lineal (Simplex, etc.). Un ejemplo de heurística es el llamado algoritmo "goloso" que consiste en hacer en cada caso lo que es localmente mejor. Una implementación de este tipo de algoritmo podría ser la del vecino más próximo que consiste en cumplir los siguientes pasos:

1. Se empieza por un tour parcial que contenga sólo una ciudad
2. Se agrega una ciudad al tour, la ciudad elegida es la más cercana a la última que se había agregado.
3. Se repite el paso 2 hasta que todas las ciudades estén en el tour

Por lo general, la aplicación de estas heurísticas de construcción de tours muy sencillas, dan soluciones bastante alejadas del óptimo, pero existen heurísticas de mejoramiento que permiten acercar esa solución al óptimo, minimizando el costo lo más posible.

Aunque se apliquen heurísticas y no se decida aplicar el algoritmo directamente, igualmente el objetivo es minimizar la cantidad de trabajo. Para ahorrarse cargar las distancias entre ciudades es bueno analizar si es posible que una ciudad se comunique con otra en el caso óptimo. Así podemos eliminar casos. Para cada ciudad guardo una lista de tamaño fijo que tiene sólo los vecinos más próximos (en un número de 10, por ejemplo). Teniendo dos puntos puedo calcular la distancia euclidiana entre ellos, así que sólo guardo las coordenadas en un plano.

Los casos más difíciles de viajantes se llaman "clustering" o de nubes de puntos y se dan cuando hay conjuntos de ciudades separados y la densidad de cada nube es mucho mayor que la de otros sectores del plano en que están ubicadas. Estos son los problemas en los que se demora mucho el algoritmo de Simplex, por ejemplo.

Métodos para encontrar la. solución exacta .del problema .del viajante de comercio

1) Métodos Branch-and-Bound:

Estos métodos ven al problema del viajante como un problema combinatorio de programación lineal entera.

Ventaja: Son eficientes en problemas fáciles (aquellos en los cuales la ubicación de las ciudades es al azar) que tienen miles de ciudades.

Desventajas: Si el modelo proviene de la vida real (es más difícil) estos métodos tienen dificultades con problemas de 100 a 200 ciudades.

2) Métodos del tipo Branch-and-Cut

Utilizan técnicas de programación entera combinando Branch-and-Bound con planos de corte, para ir eliminando restricciones.

Ventajas: Son muy eficientes. Pueden resolver problemas difíciles de miles de ciudades. El récord es de 5.000 ciudades. En Estados Unidos hay 3 grupos que trabajan en la resolución exacta del problema del viajante:

- Padberg y Rinaldi
- Grotscher y otros
- Applegate y Cook

Desventajas: es de difícil implementación y es extremadamente dependiente del tipo de problema (para algunos funciona muy bien y para otros mucho peor, además, no se "aprende" de un problema para otro).

3) Otra opción que se plantea es no agregar las Ui que impiden la formación de subtours y, luego de una corrida inicial, se le agregan restricciones con las Ui que sean necesarias para romper los subtours que se hayan formado.

Ventajas: Simplifica inicialmente el problema y permite una solución completa.

Desventajas: Es muy trabajoso y depende de la complejidad del modelo planteado (en algunos casos habrá que agregar todas las Ui con lo que el problema demora mucho en poder resolverse).

Obsesiones viajeras

Cuando nos adentramos en el problema del viajante se nos empiezan a ocurrir variantes y las posibilidades son prácticamente infinitas.

Por ejemplo: Un viajante tiene que partir de una ciudad cero, recorrer 7 ciudades y terminar en una determinada ciudad 8. ¿Se podría aplicar con algunas variantes el problema del viajante? Para empezar es necesario recorrer todas las ciudades una vez, evitar subtours dentro del circuito de 7 ciudades, etc. Todas estas condiciones son las del problema del viajante. Estamos seguros de que encontrarán la forma de relacionar la última ciudad del circuito (sea cual fuere) con la ciudad 8 y la primera ciudad con la cero. El secreto está en la relación que tienen la ciudad 0 y la 8 pero les dejamos que lo descubran ustedes mismos.

También podemos relacionar dos circuitos de viajantes de la siguiente manera: Un viajante parte de la ciudad cero y recorre 6 ciudades. Desde la última de este circuito (a la que llamaremos L) debe partir para visitar otras 7 ciudades y llegar a la ciudad L de vuelta.

Como extensión del modelo se suelen utilizar ejemplos en los cuales tendremos varios viajantes de comercio. Se cuenta con un número de viajantes "k" que es un número fijo y todos deben salir y volver a la misma ciudad recorriendo entre todos todas las ciudades (qué ciudades recorre cada uno puede estar prefijado o se puede desarrollar un modelo que determine qué ciudades entran en el circuito de cada uno de los viajantes). El objetivo es encontrar un multi-tour que visite todas las ciudades y que tenga costo total mínimo.

Un caso bastante complicado es el del problema del viajante asimétrico que es aquél en el cual el costo de ir desde la ciudad A hasta la ciudad B no es el mismo que el de ir desde la ciudad B hasta la A. En este caso hay que guardar una cantidad muy grande de puntos. Por eso el récord resuelto de este tipo de problemas tiene 200 ciudades.

Una variante que se usa en la práctica es la del ruteo de vehículos. Se usa la idea básica del viajante de comercio para aplicarla al caso de un despacho de mercadería a clientes. Cada uno de los clientes tiene una demanda y se puede repartir una cantidad máxima de mercadería por viaje. Se trata de determinar la cantidad a llevar en cada viaje y los recorridos a .hacer en cada caso. Esta variación es muy aplicada por tratarse de un caso muy común y actualmente varios equipos de investigadores efectúan análisis acerca de ella. El récord de este problema resuelto tiene 50 ciudades.

¿Se les ocurren otras variantes? Si es así, les agradecemos que nos las sugieran.

Problemas. del viajante en la realidad

El caso del viajante de comercio ya es muy poco frecuente, y hace bastante tiempo que Arthur Miller se inspiró en esa profesión para su obra teatral. Entonces: ¿qué sentido tiene hablar de un problema que parece perimido? Es que la estructura se puede aplicar a otros casos, como hablamos mencionado anteriormente: un empleado que realiza trámites y luego vuelve a su oficina, un corredor que visita comercios diariamente y luego vuelve a la fábrica, etc.

Uno de los ejemplos prácticos de aplicación del problema del viajante se desarrolló en Atlanta, USA. Se aplicó el modelo para proveer almuerzos calientes a personas demasiado viejas o enfermas para salir de compras o cocinar. Como la comida debía distribuirse utilizando una única camioneta de reparto se utilizó el modelo del viajante de comercio para generar diariamente la ruta del vehículo de reparto, siempre tratando de que esta sea lo más corta posible. Además el sistema para generar la ruta debía ser barato, eficiente y de fácil mantenimiento, ya que el solicitante era una institución benéfica, que no tenía computadora, ni medios económicos para adquirir una. Para este caso, desarrollado en 1982, Bartholdi y Platzman usaron una heurística de curvas de llenado con el sector de la ciudad al que se debla servir. Luego de este mapeo, se consiguió implementar un ingenioso sistema de ficheros circulares, con el cual se hacia sencillo el agregar un nuevo beneficiario al sistema, manteniéndolo clasificado por dos vías: por orden alfabético y en relación con su posición en el mapeo de la ruta de la camioneta. Así es muy fácil agregar un nuevo "cliente" al servicio. Inclusive, más adelante se dividió el fichero en cuatro porque se consiguió hacer tres recorridos más (el caso de varios viajantes de comercio).

Otro caso práctico en el cual se aplicó viajante de comercio fue desarrollado por el griego Vangelis F. Magiron para el perforado de plaquetas de circuitos impresos. La tarea que se realizaba era la perforación de la plaqueta en las posiciones correspondientes a los "pins" de diversos componentes electrónicos, que luego serian soldados. La empresa contaba con una perforadora programable a la cual se la podía "entrenar" indicándole el ruteo a seguir y las perforaciones a realizar, luego la máquina repetía el programa definido en ese proceso. Se había detectado que la situación antes de la modelización representaba un "cuello de botella" en la producción ya que la máquina gastaba más tiempo en el movimiento lateral de la perforadora que en la perforación en si misma. Para corregir eso se desarrolló un modelo del viajante que permitió encontrar un ruteo eficiente de la cabeza perforadora que minimice el tiempo total utilizado. En este problema, como en los de la mayoría de los ejemplos del viajante de la vida real, se utilizaron heurísticas para su resolución. Primero se determinaron las coordenadas de cada uno de los puntos que habla que perforar y se tomaron las distancias rectilíneas, satisfaciendo la desigualdad triangular y luego se escribió un algoritmo que implementaba una heurística de construcción seguida de una heurística de mejoramiento intercambiando de a 3 ejes, borrando 3 ejes y agregando otros 3 (heurística del. 3-intercambio).

El algoritmo fue codificado en BASIc y primeramente tardaba 40 minutos con un tour de 100 orificios. El algoritmo se acercaba al óptimo con una diferencia del 15% en el peor de los casos y de un 10% promedio. Luego se programó en un lenguaje de màquina aumentando la velocidad del algoritmo para converger a una solución óptima. Tan exitoso fue el experimento que en junio de 1989 se comenzó a comercializar el sistema bajo el nombre de OPTODROME.

Como otro ejemplo de los sistemas reales podemos mostrar un tour para recorrer 48 capitales de los Estados Unidos que resolvió el equipo de Padberg/Rinaldi. Para obtener una solución se “mapeaban” las ciudades, colocándolas en un mapa y alimentando al sistema con las coordenadas de cada ciudad que son las que se consignan a continuación:

1	6734	1453
2	2233	10
3	5530	1424
4	481	841.
5	3082	1644
6	7608	4458
7	7573	3716
8	7265	1268
9	6898	1885
10	1112	2049
11	5468	2606
12	5989	2873
13	4706	2674
14	4612	2035
15	6347	2683
16	6107	669

17	7611	5184
18	7462	3590
19	7732	4723
20	5900	3561
21	4483	3369
22	6191	1110
23	5199	2182
24	1633	2809
25	4307	2322
26	675	1006
27	7555	4819
28	7541	3981
29	3177	756
33	7352	4506
31	7545	2801
32	3245	3305
33	6426	3173

34	4608	1198
35	23	2216
36	7248	3779
37	7762	4595
38	7392	2244
39	3484	2829
40	6271	2135
41	4985	143
42	1916	1569
43	7280	4899
44	7509	3239
45	13	2676
46	6807	2993
47	5185	3258
48	3323	1942

El tour óptimo fue el siguiente

NAME: att48.opt.tour

COMMENT : Optimum solution for att48

TYPE : TOUR

DIMENSION : 48

TouR_Section

1 – 8 – 38 – 31 – 44 – 18 – 7 – 28 - 6 - 37 - 19 - 27 – 17 – 43 – 30 – 36 – 46 – 33 – 20 – 47 - 21 – 32 – 39 – 48 – 5 – 42 – 24 – 10 – 45 – 35 - 4 – 26 - 2 – 29 – 34 – 41 – 16 22 - 3 – 23 – 14 – 25 – 13 – 11 – 12 – 15 – 40 – 9 - 1 (vuelve a la ciudad 1)

Luego, el mismo equipo trabajó en un problema de 532 ciudades

<u>BIBLIOGRAFIA</u>

1) Investigación de Operaciones (Conceptos generales) - Hillier - Lieberman
2) Programación Lineal (El problema genérico del viajante) - Saúl Gass
3) Introduction to Operations Research (Conceptos generales) – Ecker - Kupferschmid
4) A minimal technology routing system for meals on wheels (Problemas del viajante en la realidad) – Bartholdi – Platzman – Collins - Warden
5) The efficient drilling of printed circuit boards (Problemas del viajante en la realidad) – V. Magirou
6) Clases teóricas del problema del viajante (Conceptos generales) – María Alejandra Alcalde – Graciela Hadad