

Algoritmo de asignación de visitas a puntos de venta a rutas y de optimización de las rutas resultantes

Laia Ferrer, Rafael Pastor, Alberto García, Albert Corominas

Instituto de Organización y Control de Sistemas Industriales. Universidad Politécnica de Cataluña. Av. Diagonal 647, 08028 Barcelona. laia.ferrer@upc.edu, rafael.pastor@upc.edu, alberto.garcia-villoria@upc.edu, albert.corominas@upc.edu

Resumen

Este trabajo estudia la optimización de las rutas a realizar por los comerciales de una empresa distribuidora de productos a grandes superficies. Los vendedores realizan rutas independientes entre sí, ya que el área geográfica y los clientes que visita cada vendedor están previamente definidos. La empresa necesita definir las rutas a seguir durante cierto horizonte temporal, en el cual cada cliente debe ser visitado un número determinado de veces. Además, se considera la posibilidad de realizar rutas múltiples de varias jornadas, para visitar clientes alejados del domicilio del vendedor. El objetivo del algoritmo es generar tantas rutas como jornadas laborales tenga el horizonte temporal, que posteriormente serán ordenadas y asignadas a un día concreto por los propios vendedores. Para resolver el problema se desarrolla un algoritmo por fases que acaba utilizando una búsqueda tabú para mejorar las rutas obtenidas en las fases anteriores. El algoritmo ha sido validado con datos reales proporcionados por la empresa que ha motivado este estudio. Las rutas obtenidas como resultado han sido calificadas como muy satisfactorias.

Palabras clave: problema de rutas de vehículos, restricciones periódicas, jornadas múltiples

1. Introducción

Este trabajo estudia el caso real de generación de las rutas a realizar por los comerciales de una empresa distribuidora de productos a grandes superficies. La empresa dispone de una serie de vendedores y necesita definir las rutas a seguir durante cierto horizonte temporal, en el cual cada cliente debe ser visitado un número determinado de veces. El objetivo del algoritmo a diseñar es generar tantas rutas como jornadas laborales tenga el horizonte temporal, que posteriormente serán ordenadas y asignadas a un día concreto por los propios vendedores. La empresa prefiere la asignación posterior de las rutas a los días de trabajo para poder adaptarse más fácilmente a posibles imprevistos. El área geográfica y los clientes que visita cada vendedor están previamente definidos. Así, como las rutas de los vendedores son independientes entre sí, el algoritmo se ejecuta para cada uno de ellos de forma independiente.

El problema a resolver, para cada vendedor, es el siguiente. Sea un vendedor que ha de visitar N clientes, para lo cual puede hacer hasta D rutas, una cada día del horizonte temporal. Sea t_{ij} los tiempos de desplazamiento simétricos entre clientes o entre un cliente y el domicilio del vendedor y tv_i el tiempo de visita al cliente i . El problema consiste en diseñar hasta D rutas minimizando una función coste, sabiendo que cada cliente i ha de ser visitado r_i veces en el horizonte temporal considerado y que, por lo tanto, debe ser incluido en r_i rutas. La duración de cada ruta, incluido el tiempo de desplazamiento y de visita, en general, esta limitada a una jornada laboral. Aún así, si una ruta acaba a más de cierta distancia del domicilio del

vendedor se puede pernoctar y hacer una ruta múltiple de M jornadas. Como máximo se pueden realizar R rutas múltiples.

El problema a estudiar es básicamente un Periodic Vehicle Routing Problem (PVRP), en el que se resuelven las rutas a seguir por una serie de vehículos durante un horizonte temporal, en el que cada cliente debe ser visitado un número determinado de veces con frecuencia regular (Chao *et al* (1995) y Cordeau *et al* (1997)). Aun así, como en el caso particular que aquí nos ocupa la frecuencia no condiciona la generación de rutas, sino que se considera en una fase posterior, el problema se puede modelizar como una variación del Distance-Constrained Vehicle Routing Problem (DVRP), en el que se diseñan las rutas a realizar por los vehículos para servir a unos clientes con restricciones de duración máxima de las rutas (Toth y Vigo (2002)). En este caso, la duración de las rutas puede ser de una jornada o de varias, sin que el número de cada este prefijado.

En resumen, este trabajo se propone estudiar una aplicación del DVRP simétrico, con clientes visitados r_i veces y con posibles rutas múltiples. El objetivo consiste en asignar las visitas que hay que realizar a los clientes a las rutas, simples o múltiples, de forma que se minimice el valor de una función objetivo parametrizada por el usuario que incorpora: i) el tiempo de desplazamiento de los vendedores; y ii) el exceso de tiempo cuando las jornadas superan las horas establecidas en la jornada laboral.

Para resolver el problema se desarrolla un algoritmo resolución por fases que acaba utilizando una búsqueda tabú para mejorar las rutas obtenidas en las fases anteriores. En el próximo apartado se presenta el algoritmo desarrollado y en el apartado 3 se resumen los resultados obtenidos.

2. Algoritmo de resolución por fases

Para resolver el problema se desarrolla un algoritmo resolución por fases que acaba utilizando una búsqueda tabú para mejorar las rutas obtenidas en las fases anteriores. En una primera fase, etapas 1 a 4, se define una solución inicial; en una segunda fase que coincide con la etapa 5 se realiza una optimización local; y, finalmente, en la etapa 6 se ejecuta el procedimiento metaheurístico.

- Etapa 1: Se diseñan rutas múltiples que incorporan clientes “lejanos” del domicilio del vendedor, pero cercanos entre ellos. Se generan también rutas simples que contienen clientes lejanos.
- Etapa 2: Se completan las rutas simples que finalizan en clientes “lejanos” intercalando algún cliente cercano entre el domicilio del vendedor y el último cliente lejano.
- Etapa 3: Se completan las rutas múltiples con clientes cercanos.
- Etapa 4: Se diseñan el resto de rutas simples de modo que el total de rutas incorpore cada cliente el número de veces establecido.
- Etapa 5: Se ejecuta un proceso de optimización local para mejorar las rutas obtenidas.
- Etapa 6: Se utiliza una búsqueda tabú como procedimiento metaheurístico para mejorar las rutas obtenidas en etapas anteriores.

Cada etapa, que se detalla a continuación, se desarrolla asumiendo las siguientes hipótesis de trabajo:

- Se supone que el número de veces que se visita todo cliente es menor o igual que la suma del número de rutas múltiples y simples admisible.
- Una solución con rutas muy cargadas, y que sobrepasan las jornadas admisibles, indica que la carga de visitas asignadas a dicho vendedor debe ser reducida.
- En una ruta múltiple, aunque ésta se componga de varias jornadas, no se repite la visita a un mismo cliente.

Una vez generadas las rutas, en un post-proceso final, las rutas que tengan un cliente inicial o final “lejano” se ordenaran de forma que se comience con el cliente “lejano” y se finalice en el cliente “cercano”.

2.1. Etapa 1: Diseño de rutas múltiples

Para todo cliente se conoce la provincia en la que está y, como se ha comentado, su distancia al domicilio del vendedor. Estos datos permiten conocer si hay clientes “lejanos” (para los cuales se podría pensar en una ruta múltiple) y, por tanto, el conjunto de clientes “lejanos” en cada provincia.

En esta primera etapa se trata de diseñar rutas múltiples (si son necesarias) que incorporen clientes “lejanos” del domicilio del vendedor, pero cercanos entre ellos. Se considera que los clientes “lejanos” que pertenecen a una misma provincia son cercanos entre ellos.

El esquema de la generación de las rutas lejanas en esta etapa es el siguiente:

1. Identificar la provincia más “lejana” al domicilio del vendedor, considerando los clientes que aun no han sido visitados. Con los clientes “lejanos” que quedan por visitar de la provincia más “lejana”, resolver un TSP.
2. Si se han visitado todos los clientes de la provincia, comprobar si hay algún cliente “lejano” de otra provincia que pueda incorporarse a la ruta parcial ya establecida: si existe alguno, ir a paso 3, sino ir a paso 5. Si no se han visitado todos los clientes de la provincia, ir a paso 5.
3. Con los nuevos clientes “lejanos” de la nueva provincia y con la secuencia parcial de visitas ya establecida, resolver un nuevo TSP.
4. Si se han incluido en la ruta todos los clientes “lejanos” que faltaban por visitar de la nueva provincia considerada, comprobar si hay algún cliente “lejano” de otra provincia (que tenga que ser visitado) que esté cerca de uno de los dos clientes extremo de la ruta parcial ya establecida; si existe alguno, ir a paso 3, sino ir a paso 5. Si no se han visitado todos los clientes de dicha provincia, ir a paso 5.

5. Fin de la ruta en construcción. La ruta resultante puede ser: una ruta múltiple con una carga de trabajo completa o incompleta (que será completada posteriormente) o una ruta simple (aunque lejana, ya que empieza y finaliza “lejos” del domicilio del vendedor).

2.2. Etapa 2: Arreglo de rutas simples que finalizan “lejos”

Las rutas simples creadas empiezan y acaban en clientes “lejanos” del domicilio del vendedor (por lo que deberían ser múltiples). Se trata de intercalar al menos un cliente “cercano” entre el domicilio del vendedor y el primer o el último cliente “lejano” que visita. Como las distancias son simétricas, si una ruta empieza en un cliente “cercano” y acaba en uno “lejano”, se puede cambiar el sentido de la ruta y empezar en un cliente “lejano” y acabar en uno “cercano” (lo que eliminaría la imposición de ser ruta múltiple ya que no acaba lejos del domicilio del vendedor).

1. Ordenar las rutas de mayor a menor tiempo de ruta.
2. Si hay rutas no tratadas, seleccionar la siguiente ruta de la lista, sino ir a paso 5.
3. Manteniendo invariable la ruta parcial de clientes “lejanos” ya establecida, resolver un TSP, con los clientes “cercaños” que todavía han de ser visitados. Si se ha incluido en la ruta algún cliente “cercano” ir a paso 2; sino ir a paso 4.
4. Para evitar que la ruta simple deba ser múltiple, se inserta un cliente “cercano” aunque se sobrepasen la jornada laboral: se inserta el más “central” (que tenga que ser visitado) en el camino de ida o de vuelta al domicilio del vendedor. Ir a paso 2.
5. Fin, ninguna de las rutas tendría porque ser múltiple (excepto si no hay suficientes clientes cercanos).

2.3. Etapa 3: Completar rutas múltiples

Se completan las rutas múltiples generadas hasta el momento con clientes cercanos que aun no han sido asignados y teniendo siempre en cuenta que no se supere el tiempo máximo de la ruta

1. Ordenar las rutas de mayor a menor tiempo de ruta.
2. Si hay rutas no tratadas, seleccionar la siguiente ruta de la lista, sino ir a paso 5.
3. Si la duración de la ruta es mayor que la jornada laboral ir paso 2.
4. Manteniendo invariable la ruta parcial de clientes “lejanos” ya establecida, resolver un TSP, con los clientes “cercaños” que todavía han de ser visitados y de una duración máxima igual a la jornada laboral. Ir a paso 2.
5. Fin, no se puede insertar ningún otro cliente en las rutas múltiples.

2.4. Etapa 4: Diseñar el resto de rutas

En esta etapa se diseñan nuevas rutas simples hasta que todas las visitas a los clientes están en alguna ruta.

1. Resolver un TSP, con todos los clientes que todavía han de ser visitados (recordar que hay clientes de más de una visita) y de una duración máxima de una jornada laboral, hasta que se hayan generado rutas para todas las jornadas laborales.
2. Ordenar los clientes que faltan por visitar alguna vez de mayor a menor distancia respecto a la residencia del vendedor e insertar el cliente en la ruta y posición de mejor valor de la función objetivo, siempre que dicho cliente no esté ya en esa ruta.
3. Fin, todas las visitas a los clientes están en alguna ruta.

2.5. Etapa 5: Optimización local

Para evitar trabajar con rutas simples que deberían ser múltiples, a partir de la esta quinta etapa se añade un sumando a la función objetivo que consiste en sumar un valor muy grande para cada falsa ruta simple (ruta simple que empieza y acaba en un cliente “lejano”).

La etapa 5 consiste en ejecutar, de forma secuencial, los siguientes tres pasos:

1. Realizar una optimización inicial de cada ruta. Si el número de clientes de la ruta, a parte del domicilio del vendedor, es pequeño, se hace una exploración de todas las rutas posibles y se selecciona la óptima. Si el número de clientes de la ruta es grande se realiza una optimización local con un dos intercambio en el orden de visita de toda pareja de clientes, considerando que uno de ellos es real y el otro real o fantasma.
2. El segundo paso consiste en un procedimiento de optimización local en el que se generan todas las soluciones vecinas de la solución en curso y se adopta, como nueva solución a tratar, la mejor de las vecinas, siempre que sea mejor que la solución en curso; el proceso finaliza cuando la mejor solución vecina no es mejor que la solución en curso. Las soluciones vecinas de la solución en curso se obtienen al intercambiar 2 clientes de dos rutas distintas. Las rutas vacías (sin clientes asignados) también deben ser consideradas como rutas entre las que realizar intercambios (en este caso, evidentemente, únicamente contienen clientes fantasma). Además, cada vez que se adopta una nueva solución en curso se realiza una optimización de las rutas modificadas (como la comentada en el paso 1).
3. El tercer paso consiste en ejecutar nuevamente el procedimiento de optimización local de la etapa anterior variando dos elementos. El primero es la forma de generar el vecindario: cuando se realiza un intercambio de un cliente en otra ruta, en lugar de colocarlo en la posición en la que estaba el cliente por el cual se cambia, se prueba a insertarlo en todas las ubicaciones posibles. El segundo cambio se refiere a la condición de finalización del algoritmo: el procedimiento finaliza cuando, en dos iteraciones consecutivas, la mejora del valor de la función objetivo no es significativo.

2.6. Etapa 6: Procedimiento metaheurístico

Antes de nada, cabe recordar que no se admite una solución en la que se visita dos veces a un mismo cliente en la misma ruta, ni una en la que existen rutas simples que deberían ser múltiples.

Formalización:

- \bar{X} , \bar{Z} : mejor solución hallada hasta el momento y su valor.
 X_{cur} , Z_{cur} : solución en curso y su valor.
 \bar{X}_{vec} , \bar{Z}_{vec} : mejor solución vecina de X_{cur} y su valor.
 \bar{X}_{vec}^{nt} : mejor solución vecina no tabú de X_{cur} .
Lista tabú: lista de propiedades prohibidas de las soluciones, es circular y variable en el tiempo.

Paso 0. Obtener solución inicial X_{cur} ; $\bar{X} = X_{cur}$ y $\bar{Z} = Z_{cur}$.

Paso 1. Hacer (hasta condición de fin):

- Generar todas las soluciones vecinas de X_{cur} .
- Si \bar{X}_{vec} cumple el nivel de aspiración: $\bar{X} = \bar{X}_{vec}$ y $\bar{Z} = \bar{Z}_{vec}$, $X_{cur} = \bar{X}_{vec}$; caso contrario $X_{cur} = \bar{X}_{vec}^{nt}$ y actualizar lista tabú (introducir propiedad de X_{cur} y eliminar, si procede, la más antigua).

Concreción de los parámetros:

- *Solución inicial*: la obtenida en la etapa 5.
- *Condición de fin*: tiempo de cálculo introducido por el usuario.
- *Nivel de aspiración*: la mejor vecina es mejor que la mejor solución hallada hasta el momento.
- *Reglas tabú*: cliente-i/ruta-r
- *Longitud de la lista*: 10
- *Vecindario*: idéntico vecindario que en la etapa 5.

La etapa, y el procedimiento, finaliza con una optimización de cada ruta, igual a la utilizada en la etapa 5.

3. Resultados

El algoritmo diseñado permite generar las rutas a seguir por un vendedor en las diferentes jornadas de trabajo de un cierto horizonte temporal, considerando el número de visitas a realizar a cada cliente y permitiendo hacer rutas múltiples equivalentes a varias jornadas de trabajo consecutivas. Se ha desarrollado un algoritmo que resuelve el problema por fases, primero generando rutas factibles y después mejorando las rutas obtenidas en la primera fase.

El algoritmo ha sido validado con datos reales proporcionados por la empresa que ha motivado este estudio. Las rutas obtenidas como resultado han sido calificadas como muy satisfactorias por los representantes de la empresa.

La aplicación desarrollada ha permitido a la empresa disponer de una herramienta de diseño de rutas, en función de los clientes asignados a un vendedor y el número de visitas a realizar a cada cliente, que ha sido utilizada en la negociación de los objetivos de los vendedores para el año 2006, así como en la realización de sus rutas de visitas.

Referencias

- Chao, I.M.; Golden, B.L.; Wasil, E. (1995). An improved heuristic for the periodic vehicle routing problem. *Networks*, Vol. 26, pp. 25-44.
- Cordeau, J.F; Gendreau, M.; Laporte, G. (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, Vol. 30, pp. 105-119.
- Toth, P.; Vigo, D. (2002). *The vehicle routing problem*. 1th ed. SIAM, Society for Industrial and Applied Mathematics, Philadelphia.