

DESARROLLO DE UN NUEVO ALGORITMO HEURÍSTICO PARA LA SOLUCIÓN DEL PROBLEMA DE PLANIFICACIÓN DE RUTAS CON VENTANAS DE TIEMPO

Eduardo Guillén Solórzano

Susana Barbeito Roibal

Manuel Martínez Carballo

Universidade da Coruña

RESUMEN

En este trabajo se presenta un nuevo algoritmo para la resolución del conocido problema de planificación de rutas de transporte sujetas a ventanas de tiempo. Este problema se conoce en la literatura como el VRPTW o Vehicle Routing Problem with Time Windows, y para su solución se han desarrollado multitud de métodos y técnicas exactos y heurísticos. Si bien los primeros requieren una elevada inversión en tiempos de cálculo, los segundos aportan soluciones aproximativas en menores tiempos de cálculo. En este trabajo se presenta un nuevo algoritmo de construcción de rutas de forma aproximativa. Las soluciones arrojadas por el algoritmo mejoran en algunos casos las aportadas por otros autores con métodos similares, aunque nuestro modelo también tiene algunas limitaciones que se comentan en este trabajo.

1. INTRODUCCIÓN

El problema del VRPTW consiste en el establecimiento de un número de rutas de transporte que visiten un conjunto de clientes dispersos en el espacio en unos momentos determinados del horizonte de planificación. Cada uno de estos clientes requiere una cierta cantidad demandada que ha de ser transportada en un vehículo con capacidad limitada. Para realizar la distribución se dispone de una flota de vehículos homogéneos dispuestos en un almacén central. El objetivo último es minimizar la distancia total recorrida por el conjunto de vehículos, sin violar ninguna de las restricciones del problema.

2. MÉTODO DE CÁLCULO

En el método presentado en este trabajo se recoge la aplicación de una serie de reglas para la obtención de la solución final de una manera aproximativa. Estas reglas se han integrado dentro de un algoritmo heurístico de solución denominado ALADIN, en el que se utilizan reglas básicas para la construcción de las rutas. El objetivo es resolver el problema de forma aproximativa en un tiempo muy reducido, de manera que la solución se devolverá posteriormente a la Interfaz de la aplicación desarrollada para la implementación del mismo para su representación. Los detalles del algoritmo se recogen en las siguientes líneas. La notación del mismo es la siguiente:

N	Número total de clientes
i	Índice de los clientes: $i = 1, 2, \dots, N$
x_i	Abscisa de la posición del cliente i
y_i	Ordenada de la posición del cliente i
ts_i	Tiempo de servicio del cliente i .
V	Número total de vehículos disponibles
k	Índice de los vehículos. $k=1, \dots, V$
q	Capacidad de los vehículos en consideración
d_i	Demanda del cliente i

- ta_i Momento de apertura del cliente i
- tc_i Momento de cierre del cliente i
- ta_0 Momento de apertura del depósito central
- tc_0 Momento de cierre del depósito central
- d_{ij} Distancia entre i y j .
- t_{ij} Tiempo de desplazamiento entre el cliente i y el cliente j , equivalente a la distancia
- te_{ki} Tiempo de espera antes de visitar el nodo i por el vehículo k .
- LR_k Lista de nodos que pertenecen a la ruta k . Se trata de un vector de nodos ordenados que indican las sucesivas visitas que realiza el vehículo k .
- tr_k Momento en el que se encuentra el vehículo k . Esta variable será igual a la suma de los tiempos de desplazamiento entre los nodos que visita este vehículo k , más los tiempos de servicio de los nodos de la ruta k , más las esperas que se han de realizar antes de visitar cada uno de los nodos de la ruta k . Los tiempos de desplazamiento de los nodos será la suma de t_{pq} cuando $q > p$
- c_k Carga total del vehículo k . La carga total del vehículo k es la suma de las demandas de los nodos que ha visitado.
- CR_k La cabeza de la ruta k se corresponde con el nodo en el que está ubicado el vehículo k .
- D_k Distancia total recorrida por el vehículo k . Esta variable es igual a la suma de los desplazamientos entre los nodos que visita este vehículo.
- E_k Espera total que realiza el vehículo k .

En los siguientes apartados se detallan las reglas de decisión que subyacen del algoritmo desarrollado en este trabajo para la resolución del VRPTW.

3. DESCRIPCIÓN DEL ALGORITMO

En este apartado se analiza con mayor detalle la secuencia de resolución del algoritmo integrada en el software de solución. Cada paso se describe a continuación en los siguientes apartados.

Generación de la lista L1 de nodos semilla

El primer paso es la generación de la lista inicial de candidatos a ser servidos por los vehículos que recorrerán las R primeras rutas diseñadas, esta primera lista la llamamos LI , y contendrá exactamente R nodos semilla. LI será la lista de nodos candidatos a ser servidos inicialmente por el problema. Tal y como se comentaba al principio, los primeros nodos que han de ser servidos derivan de la consideración de sus ventanas de tiempo. Por ello el algoritmo se ha de centrar en el análisis de las ventanas de tiempo de estos nodos, y en especial en el momento de cierre de los mismos. Este dato es crítico porque marca el plazo límite para visitar estos nodos, por lo que nunca se puede violar esta restricción.

Asignación y adición directa a una ruta sin espera

Para aquellos nodos i donde la holgura es menor que $\min ts_i$ (mínimo tiempo de servicio entre todos los nodos del problema) entonces se generan las correspondientes rutas entre el depósito y estos nodos. Para ello se procederá a computar una espera en caso de que la entrada no sea directa en el nodo, caso contrario se entrará directamente a realizar el servicio. Así las rutas generadas inicialmente se actualizan de la siguiente manera con los nodos i que implican asignaciones directas:

$$LR_k = [0, i]$$

$$CR_k = [i]$$

$$c_k = d_i$$

$$tr_k = t_{0i} + ts_i + te_{ki}$$

$$D_k = t_{0i}$$

$$E_k = te_{ki}$$

En este caso la espera sería $te_{ki} = [ta_i - (t_{0i})]^+$ Seguidamente se elimina el nodo i de la lista LI . Para el resto de los nodos de LI se estudia uno a uno comenzando por aquellos con menor holgura, para analizar las posibles inserciones de nodos intermedios entre el depósito central y cada uno de los nodos semilla en consideración.

Generación de la lista L2 de nodos para la inserción simple

Se selecciona el candidato i con menor holgura de entre aquellos que no se han asignado directamente. Después de comprobar su holgura verificamos si ésta es mayor que $\min ts_i$ y si la holgura supera en 1 ó 2 veces este mínimo tiempo de servicio. Si su holgura supera el valor de un solo tiempo de servicio, entonces se genera la lista $L2$ de posibles nodos intermedios (nodos m) para realizar inserciones simples. Los nodos en camino son aquellos cuyo servicio permiten llegar a tiempo al nodo en consideración, y además no suponen un gran desplazamiento (parámetro β), e igualmente no violen las restricciones de carga total del vehículo, así como del tiempo de llegada estimado de regresar al depósito central. Las restricciones aplicables son las siguientes:

Restricción de apertura del nodo m . El nodo ha de estar abierto en el momento en que llega el vehículo en cuestión. Por ello se establece la siguiente restricción: $ta_m \leq t_{0m}$. En esta restricción se indica que el momento de apertura de m ha de ser anterior a la llegada del vehículo desde 0 hasta m .

Restricción de llegada a tiempo al nodo i en cuestión. $ta_m + ts_m + t_{mi} \leq tc_i$
Dicho de otra forma, los nodos m en consideración como candidatos a ser escalas intermedias son aquellos que permiten llegar al nodo i en cuestión antes de que éste cierre. Para ello se contempla un incremento en el tiempo de viaje equivalente al tiempo de desplazarse hasta ese nodo m y realizar su servicio, a partir de su apertura, y el desplazamiento de m a i .

Restricción de tiempos de desplazamiento totales. Además es necesario comprobar que la visita de los nodos m en cuestión me permite llegar a tiempo al nodo i que estoy considerando por eso se establece una restricción temporal consistente en el cómputo de todos los desplazamientos y tiempos de servicio hasta el nodo i . $t_{0m} + ts_m + t_{mi} \leq tc_i$ Suponiendo que la visita del nodo m en consideración permita llegar antes del momento de cierre, esto provocará un tiempo que se computa como nueva holgura del nodo i , de manera que la inecuación anterior resultaría en la siguiente ecuación. $t_{0m} + ts_m + t_{mi} + nh_i = tc_i$ donde nh_i ha de ser mayor o igual que 0 .

Restricción de desplazamiento geográfico máximo autorizado. Es necesario establecer una restricción que evite grandes desplazamientos, y que depende del

parámetro β seleccionado. En este sentido solamente se permitirán aquellos desplazamientos para realizar las visitas de los nodos intermedios m que supongan un máximo de β veces el desplazamiento original entre el depósito y el nodo i . Esto es

$$t_{0m} + t_{mi} \leq \beta \cdot t_{0i}$$

Asignación y adición directa a una ruta k con espera

Si la lista $L2$ está vacía, entonces la opción es esperar y visitar después de la espera el nodo i en consideración. Para ello se amplía una ruta k comenzando en 0 y visitando el nodo i en cuestión, aunque esperando un tiempo equivalente a te_{ki} , y que implica una espera total de la ruta E_k . Se amplía una de las rutas inicializadas al comienzo de la planificación, actualizando todos los valores de la siguiente manera:

$$LR_k = [0, i]$$

$$CR_k = [i]$$

$$c_k = d_i$$

$$tr_k = t_{0i} + ts_i + te_{ki}$$

$$D_k = t_{0i}$$

$$E_k = te_{ki}$$

En este caso la espera sería $te_{ki} = [ta_i - (t_{0i})]^+$. A continuación se elimina el nodo i de la lista $L1$

Si la lista $L2$ no está vacía entonces el algoritmo procede a la generación de la lista de posibles inserciones dobles de nodos intermedios, $L3$.

Generación de la lista de inserciones dobles $L3$

Se comprueba si existe posibilidad de realizar una segunda parada antes de realizar el servicio al nodo i en estudio. Para ello se genera una segunda lista de dobles paradas denominada $L3$. La lista $L3$ está formada por pares de nodos de $L2$ que permiten la realización de dos paradas antes del servicio del nodo i en consideración. Para ello se seleccionan de $L2$ el conjunto n_2 de pares de nodos m y l tales que:

Restricción de apertura de los nodos m y l . Cumplen la restricción temporal expuesta para $L2$, o lo que es lo mismo, se trata de nodos que están abiertos para realizar la escala intermedia. Estas restricciones son las siguientes: $ta_m \leq t_{0m}$, y $ta_l \leq t_{0m} + ts_m + t_{ml}$

Restricción de llegada a tiempo al nodo i en cuestión. Cumplen la restricción de llegada a tiempo al nodo i , en caso de visita de ambas escalas intermedias de los nodos m y l . $t_{0m} + ts_m + t_{ml} + ts_l + t_{li} \leq tc_i$ Incluyendo la variable de holgura para deshacer la inecuación, la expresión anterior resultaría en $t_{0m} + ts_m + t_{ml} + ts_l + t_{li} + nh'_i = tc_i$ donde nh'_i ha de ser mayor o igual que 0 .

Restricción de desplazamiento geográfico máximo autorizado. Es necesario comprobar que no implican un gran desplazamiento, por lo que el desplazamiento total ha de ser menor o igual que γ veces el valor de β veces la distancia original. Matemáticamente: $t_{0m} + t_{ml} + t_{li} \leq \gamma \beta t_{0i}$

Una vez computadas todas las nuevas holguras, se procede de la siguiente manera:

Si $L3$ está vacía, entonces escogemos el nodo m de $L2$ que maximiza nh_i , y se añaden a una de las rutas creadas al principio de la siguiente manera:

$$LR_k = [0, m, i]$$

$$CR_k = [i]$$

$$c_k = d_m + d_i$$

$$tr_k = t_{0m} + ts_m + t_{mi} + ts_i + te_{ki}$$

$$D_k = t_{0m} + t_{mi}$$

$$E_k = te_{ki}$$

En este caso $te_{ki} = [ta_i - (t_{0m} + ts_m + t_{mi})]^+$ A continuación se elimina el nodo i de la lista $L1$ y se elimina el nodo m de la lista $L0$

Si $L3$ no está vacía entonces se seleccionan los nodos m y l que maximizan el valor de la nh'_i de manera que se procede a añadir estos dos nodos a una de las rutas originalmente creadas al principio de la planificación de la siguiente manera:

$$LR_k = [0, m, l, i]$$

$$CR_k = [i]$$

$$c_k = d_m + d_l + d_i$$

$$tr_k = t_{0m} + ts_m + t_{ml} + ts_l + t_{li} + ts_i + te_{ki}$$

$$D_k = t_{0m} + t_{ml} + t_{li}$$

$$E_k = te_{ki}$$

En este caso el $te_{ki} = [ta_i - (t_{0m} + ts_m + t_{ml} + ts_l + t_{li})]^+$ A continuación se elimina el nodo i de la lista $L1$ y se eliminan los nodos m y l de la lista $L0$. El hecho de escoger aquel o aquellos nodos que minimicen el desplazamiento total, o lo que es lo mismo, que maximizan la nueva holgura resultante para el nodo i en estudio, implica lo mismo que escoger aquel o aquellos nodos que estando entre los dos nodos en consideración, minimizan la desviación sobre la pendiente original, en este sentido se minimiza la desviación generada por los nodos insertados y que serán escalas antes de llegar al nodo destino.

De esta manera se procede a la construcción de las R primeras rutas de la solución, con los primeros R nodos semilla seleccionados, hasta que se vacía la lista $L1$.

Regeneración de la lista de los R siguientes nodos críticos $L1$ (Segunda y posteriores fases)

Hasta el momento se han construido un total de R rutas iniciales y se han actualizado todos los valores para cada una de las mismas y que servirán para esta segunda etapa de planificación. Esta segunda fase se desarrolla de la siguiente manera. La idea consiste en ir seleccionando en cada una de las iteraciones del problema un total de R nodos para su asignación a las R rutas ya existentes. En este sentido se trata de un algoritmo de construcción paralela dado que las rutas se construyen simultáneamente en contra de los procedimientos de construcción de rutas secuenciales.

Una vez seleccionados los R siguientes nodos a ser asignados se buscan aquellas rutas en las que su inserción es factible en términos de temporalidad, y en segundo lugar se tiene en cuenta los criterios de cercanía de cada uno de los nuevos nodos a las cabezas de ruta existentes. Una vez asignados los nodos en consideración se vuelven a contemplar las posibilidades de inserción simple y doble.

A partir de esta segunda etapa se procede a tener en cuenta los tiempos de cierre, y no las holguras como se realizó en la primera fase de la planificación. En este sentido nos encontramos con los nodos que requieren una atención más inmediata en la planificación de las rutas, y que por ello han de ser asignados inmediatamente a las rutas existentes. Estos nodos pasan a conformar la lista LI , que en este caso se ordena de forma creciente atendiendo a los momentos de cierre de cada uno de los R nodos seleccionados.

Generación de la solución final

Una vez que se han terminado todos los nodos de las listas, entonces se procede a la inclusión del nodo θ para todas las rutas generadas, de manera que se hace regresar a todos los vehículos al nodo original. Igualmente se actualizan todos los valores de las rutas procediendo a los siguientes cálculos para toda ruta k :

Tras la aplicación del algoritmo de solución, ésta se transfiere a la interfaz para su representación.

4. RESULTADOS Y CONCLUSIONES

En primer lugar se ha testado la bondad de cálculo del algoritmo de solución en comparación con otros algoritmos similares. Para que las comparativas sean homogéneas es costumbre aplicar los diferentes métodos a un conjunto de problemas estándar que sirven como batería de prueba a los diferentes autores. El conjunto de problemas más utilizados son los de Solomon (1987), compuesta por un total de 56 problemas tipo d 100 nodos cada uno. En este caso para algunos de estos problemas se

superaban los resultados obtenidos por algunos métodos similares recogidos en la literatura, aunque en otros casos, el algoritmo matemático desarrollado no lograba superar estos resultados. La tabla 1 recoge a modo de resumen los valores medios obtenidos para los diferentes grupos de problemas.

Para ello se aplicó este modelo a los 56 problemas tipo de Solomon (1987) para contrastar la bondad de los procedimientos aquí expuestos. Las soluciones obtenidas se presentan a continuación a modo de comparativa con las obtenidas a través de otros métodos de construcción.

Método	R1	R2	RC1	RC2	C1	C2
	Dist NV	Dist NV	Dist NV	Dist NV	Dist NV	Dist NV
Clarke y Wright Method	1499 16,60-	- -	- -	- -	976 11,70-	- -
Savings, waiting time limit	1517 15,10-	- -	- -	- -	987 10,70-	- -
Solomon I1	1437 13,60	1402 3,30	1597 13,50	1682 3,90	952 10,00	692 3,13
Solomon I2	1639 14,50	1471 3,30	1874 14,20	1798 4,10	1050 10,10	921 3,40
Solomon I3	1652 14,10	1475 3,40	1850 14,00	1816 4,00	1103 10,00	1073 3,50
Nearest Neighbour	1600 14,50	1472 3,40	1800 14,20	1755 3,90	1171 10,20	963 3,50
Gillet y Miller	1500 14,60	1449 3,20	1804 14,90	1736 4,00	941 10,00	712 3,00
Potvin y Rousseau (1993)	1509 13,30	1387 3,10	1724 13,40	1651 3,60	1343 10,67	797 3,38
Ioannu et al. (2001)	1370 12,67	1310 3,09	1512 12,50	1483 3,50	865 10,00	662 3,13
Guillén et al. (2004)	1955 26,00	1239 8,00	2247 21,00	1573 11,00	1955 25,00	1485 14,00

Tabla 1 Comparación de los métodos de construcción de rutas

En la tabla anterior se presentan los resultados medios para los diferentes grupos de problemas de Solomon (1987). Destaca el mejor comportamiento de estas técnicas en los problemas de tipo R2 y RC2, que como es sabido se corresponden con aquellos casos en los que el horizonte temporal es más amplio, y por ello los vehículos han de realizar un mayor número de visitas. Por el contrario, se obtienen peores resultados para los problemas en los que el horizonte temporal es más reducido, y que se corresponden con casos en los que las rutas son más cortas.

BIBLIOGRAFÍA

- Bodin, L., Golden B.L., Assad A. A. y Ball M., (1983), “Routing and Scheduling of Vehicles and Crews, The State of the Art”, *Comp. Oper. Res.*, Vol. 10, p. 63-211.
- Clarke, G. y Wright J.W., (1964), “Scheduling of Vehicles from a Depot to a number of Delivery Points”, *Oper. Res.*, Vol. 12, p. 568-581.
- Solomon, M., (1983), “*Vehicle Routing and Scheduling With Time Window Constraints: Models and Algorithms*”, working paper, No. 83-42, College of Business Administration, Northeastern University.
- Solomon M., (1986), “On the Worst-Case Performance of Some Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints”, *Networks*, Vol. 16, p. 161-174.
- Solomon M., (1987), “Algorithms for the vehicle routing and scheduling problem with time window constraints”, *Oper. Res.* Vol. 32, p.254-265.
- Solomon M., Baker E.K. y Schaffer J.R., (1988), “Vehicle Routing and Scheduling Problems with Time Windows Constraints: Efficient Implementations of Solution Improvement Procedures”, en: *Vehicle Routing: Methods and Studies*, Ed. Golden B. G. and Assad A. A., Elsevier Science Publishers B.V. (North Holland), p. 85-105.
- Solomon M. y Desrosiers J., (1988), “Time Window Constrained Routing and Scheduling Problems”, *Transp. Sci.*, Vol. 22, No. 1, p.1-13.