

ALADIN: ALGORITMO DE ADICIÓN E INSERCIÓN PARA LA PLANIFICACIÓN DE RUTAS DE TRANSPORTE EN ENTORNOS DE TIEMPOS RESTRINGIDOS

Eduardo Guillén Solórzano
Susana Barbeito Roiba
Manuel Martínez Carballo

RESUMEN

En este artículo presentamos un Nuevo modelo para resolver el conocido problema de planificación de rutas de transporte con restricciones temporales. El método se basa en una nueva combinación de los métodos tradicionales de adición e inserción, al que hemos denominado ALADIN. Presentamos igualmente una descripción de los métodos clásicos comparados con los nuevos procedimientos de construcción recogidos en nuestro modelo. Todos ellos se han validado a través de su aplicación a los 56 problemas tipo de Solomon, demostrando que nuestro método obtiene mejores resultados en algunos de los ejemplos. El método se presenta a través de su notación matemática.

PALABRAS CLAVE: VRPTW, Logística, heurísticos

ABSTRACT

In this paper we present a new construction model for solving the well known problem of vehicle routing with time windows. The method is based on a new combination of the traditional addition and insertion procedures, which we have named ALADIN. A description of some traditional methods is presented, compared to the new constructional procedures developed in our model. All methods are tested on Solomon's 56 benchmark problems, showing that our method provides better results for some of the instances. The method is deployed by means of its mathematical notation.

KEYWORDS: VRPTW, Logistics, heuristics.

INTRODUCCIÓN

El problema de la planificación de rutas de transporte se conoce en la literatura como VRP o *Vehicle Routing Problem*, que deriva de la problemática del TSP, o *Traveling Salesman Problem*, en la que un vehículo ha de encontrar la ruta mínima entre un conjunto de puntos geográficamente dispersos. En el caso del VRP, se generaliza el problema del TSP, y se contemplan un número NV de vehículos, con unas limitaciones de tiempos, o de cargas, de manera que es necesaria la utilización de esta flota de vehículos para visitar todos los puntos del problema. Desde mediados del siglo XX, esta problemática fue abordada por la línea de Investigación Operativa, donde se enmarcan la mayor parte de los trabajos existentes, y cuya evolución ha sido notable sobre todo en los últimos 10 años. Merece la pena destacar una fecha, 1987, que es cuando Marius M. Solomon recoge toda la información relativa a los sistemas existentes para la planificación de rutas de transporte en el marco del VRP, y los adapta al problema del VRP con *Time Windows* (VRPTW), o VRP con Ventanas de Tiempo, con el ánimo de

CITIES IN COMPETITION

acercar todavía más la línea de investigación existente a la problemática empresarial. La peculiaridad del VRPTW deriva en la consideración de intervalos de apertura para cada uno de los clientes, así como los plazos de apertura del depósito central en el que se encuentran los vehículos. De esta manera se aporta un mayor realismo al problema general del VRP. Para ello Solomon (1983) recoge en su Tesis Doctoral esta problemática, publicando en 1987 uno de los artículos más recurrentes en todos los trabajos en el que resume los resultados de aplicación de diferentes métodos de construcción de rutas al problema del VRPTW. Igualmente desarrolla una batería de ejemplos sobre los que valida estos desarrollos, y que servirían de base para la inmensa mayoría de los autores con la finalidad de normalizar los resultados de los diferentes métodos aplicados.

En este trabajo se presenta un método propio desarrollado para resolver el problema del VRPTW, el cuál consta de las siguientes características:

- Depósito central único, y múltiples clientes con cualquier tipo de dispersión (uniforme o clusterizados)
- Depósito sujeto a ventana de tiempo, indicativa del horizonte temporal de las rutas.
- Clientes sujetos a ventanas de tiempo no necesariamente iguales.
- Distancias euclídeas entre nodos (Fácilmente ampliable a tiempos de desplazamiento no euclídeos)
- Tiempos de servicio en cada cliente no necesariamente iguales
- Capacidades de los vehículos limitadas e iguales.
- Flota de vehículos ilimitada
- Objetivo último de la planificación: Minimizar la distancia total recorrida por todos los vehículos.

Estas características de los problemas particulares del VRPTW que trata de resolver el algoritmo, coinciden con las características de los problemas tipo planteados por Solomon (1987), por lo que servirán en este trabajo como medida comparativa de la eficiencia y eficacia del modelo desarrollado, a través de la comparación de la distancia total conseguida por los modelos de otros autores.

MODELO DE ADICIÓN E INSERCIÓN: ALADIN

En esta sección se desarrolla la modelización matemática del algoritmo. Para ello se ha seguido un esquema lógico de trabajo que comienza con las primeras tareas realizadas en el algoritmo. Es necesario tener en cuenta que en este trabajo se ha partido de cero para la construcción del método heurístico, lo cual lo diferencia de la mayor parte de los algoritmos presentados en la literatura y que se servían los unos de los otros para establecer combinaciones entre los mismos tratando siempre de superar las mejores soluciones halladas hasta la fecha.

La notación a ser utilizada 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
c_{ij}	Coste de viajar del nodo i al nodo j .
t_{ij}	Tiempo de desplazamiento entre el cliente i y el cliente j
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
R	Número de nodos semilla a considerar inicialmente por el algoritmo.
β	Parámetro penalizador de desplazamientos en inserciones simples
γ	Parámetro multiplicador de zonas de inserción para inserciones dobles
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 .

Cálculo de la matriz de distancias

El algoritmo calcula una matriz de distancias entre todos y cada uno de los nodos, incluyendo tanto los clientes como el depósito central. Las distancias entre cada par de nodos i y j se calcula a través de la fórmula de Pitágoras que proporciona la distancia euclídea entre dos puntos.

$$d_{ij} = \left[(x_j - x_i)^2 + (y_j - y_i)^2 \right]^{1/2}$$

De esta manera se calculan todas las distancias entre cada dos nodos del problema resultando en una matriz de $N+1$ filas y $N+1$ columnas. Esta matriz de distancias, en el caso del VRPTW también es la matriz de tiempos de desplazamiento, ya que se asume que todos los camiones viajan a una velocidad de 1 unidad de espacio por unidad de tiempo.

Definición de parámetros

Número de nodos semilla R

R : Número de rutas a considerar inicialmente por el algoritmo. En este sentido el usuario puede optar por dos vertientes. La primera consiste en considerar inicialmente el número de rutas reducido, dejando al propio algoritmo la generación o creación de posteriores rutas cuando esto se considere necesario, y la segunda alternativa consiste en considerar un número elevado de rutas iniciales, de manera que el algoritmo contempla

CITIES IN COMPETITION

desde el momento inicial la existencia de nuevas rutas que proporcionan más flexibilidad a la planificación inicial.

Parámetro β

Se trata del parámetro que impide grandes desplazamientos. El algoritmo puede considerar en la programación de las rutas movimientos que aparentemente supongan grandes desplazamientos. Sin embargo, ha de ser el propio usuario quien determine qué desplazamientos son grandes y cuáles no. El parámetro β hace referencia al número de veces de la distancia original entre dos puntos que se permite como desplazamiento total máximo para realizar una visita intermedia. La opción contra el rechazo de posibles inserciones es la espera, por lo que indirectamente este parámetro β es un parámetro penalizador de las esperas.

Parámetro γ

El algoritmo presentado en este trabajo contempla la posibilidad de realizar inserciones dobles de dos nodos simultáneamente, entre el nodo origen y el nodo destino. En ese sentido la limitación de desplazamientos grandes impuesta por β , resulta insuficiente para el caso de la inserción de dos nodos simultáneamente, teniendo en cuenta que estas inserciones dobles provocan necesariamente desplazamientos superiores, y que no por ello han de ser penalizadas. En este sentido se plantea la posibilidad de incrementar esta zona de inserción impuesta por el parámetro β . Las dos opciones eran, o bien incrementarlo dentro del propio algoritmo a través de un multiplicador fijo, o bien establecer un nuevo parámetro multiplicador con el que se pudiese flexibilizar el algoritmo. En este caso se optó por la segunda opción de manera que se ha incluido un valor γ a ser fijado por el usuario para incrementar el valor de β originalmente dispuesto para el caso de las inserciones sencillas.

Inicio del problema

Una vez introducidos todos los datos necesarios para el algoritmo, entonces da comienzo la planificación de las rutas. En este tipo de problemas, generalmente las variables más críticas son las ventanas de tiempo que proporcionan los nodos y que otorgan al planificador muy breves espacios de tiempo en los que se ha de servir el nodo en cuestión. Por ello en un elevado grado, las variables a tener en cuenta en todo momento son las ventanas de tiempo de estos nodos, además de los tiempos de desplazamiento a los mismos y de los tiempos de servicio. En este sentido las distancias, quedan relegadas a un segundo puesto en importancia, dado que las ventanas de tiempo son restricciones que han de ser respetadas en el problema muchas veces a costa de recorrer mayores distancias, en contra de lo que establecen otras variantes del problema, como el VRPSTW (VRP con ventanas de tiempo no rígidas). Por otro lado es conveniente minimizar el número de vehículos a ser utilizados en la planificación, y que generalmente también tiene una relación inversa con la distancia total recorrida.

En primer lugar, se genera la Lista de Nodos no asignados $L0$. Esta lista recoge todos los nodos del problema en cuestión.

En el momento inicial de la computación todos los tiempos de las rutas son equivalentes a 0, al igual que las cargas de las mismas. Estas variables son meros contadores acumulativos de las inserciones que se van realizando a cada una de las rutas, al igual que las variables de esperas, y distancias totales recorridas. Se generan las $k=1, \dots, R$ rutas a tener en cuenta desde el momento inicial de la planificación, y a todas ellas se les añade el nodo 0 como punto de partida. Para todas las rutas $k=1, \dots, R$ se definen las siguientes variables, utilizando la notación presentada en la sección anterior:

$$LR_k = [0]$$

$$CR_k = [0]$$

$$c_k = 0$$

$$tr_k = 0$$

$$\begin{aligned} D_k &= 0 \\ E_k &= 0 \end{aligned}$$

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 $L1$, y contendrá exactamente R nodos semilla. $L1$ 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. En este modelo el valor de referencia es la holgura resultante tras la potencial visita de ese nodo, de manera que cuanto menor sea, mayor prioridad de asignación tendrá el nodo a una ruta, debido a su mayor urgencia.

$$h_i = t_{c_r} - t_{0i}$$

A continuación se ordena la lista con todos los nodos de menor a mayor holgura, y se escogen aquellos R nodos con holguras menores para formar parte de la lista de nodos semilla $L1$.

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:

$$\begin{aligned} LR_k &= [0, i] \\ CR_k &= [i] \\ c_k &= d_i \\ tr_k &= t_{0i} + ts_i + te_{ki} \\ D_k &= t_{0i} \\ E_k &= te_{ki} \end{aligned}$$

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

Para el resto de los nodos de $L1$ 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.

CITIES IN COMPETITION

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 consiguiente 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.

$$t_{0m} + ts_m + t_{mi} + nh_i = tc_i \quad \text{donde } nh_i \text{ ha de ser mayor o igual que } 0.$$

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 .

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

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:

$$\begin{aligned} LR_k &= [0, i] \\ CR_k &= [i] \\ c_k &= d_i \\ tr_k &= t_{0i} + ts_i + te_{ki} \\ D_k &= t_{0i} \\ E_k &= te_{ki} \end{aligned}$$

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:

$$\begin{aligned} ta_m &\leq t_{0m} \\ ta_l &\leq t_{0m} + ts_m + t_{ml} \end{aligned}$$

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:

$$t_{0m} + ts_m + t_{ml} + ts_l + t_{li} + nh'_i = tc_i \text{ donde } nh'_i \text{ ha de ser mayor o igual que } 0.$$

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:

CITIES IN COMPETITION

$$t_{0m} + t_{ml} + t_{li} \leq \gamma \beta t_{0i}$$

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:

$$\begin{aligned} LR_k &= [0, m, i] \\ CR_k &= [i] \\ c_k &= d_m + d_i \\ tr_k &= t_{0m} + t_{sm} + t_{mi} + t_{si} + te_{ki} \\ D_k &= t_{0m} + t_{mi} \\ E_k &= te_{ki} \end{aligned}$$

En este caso $te_{ki} = [ta_i - (t_{0m} + t_{sm} + t_{mi})]^+$

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:

$$\begin{aligned} LR_k &= [0, m, l, i] \\ CR_k &= [i] \\ c_k &= d_m + d_l + d_i \\ tr_k &= t_{0m} + t_{sm} + t_{ml} + t_{sl} + t_{li} + t_{si} + te_{ki} \\ D_k &= t_{0m} + t_{ml} + t_{li} \\ E_k &= te_{ki} \end{aligned}$$

En este caso el $te_{ki} = [ta_i - (t_{0m} + t_{sm} + t_{ml} + t_{sl} + t_{li})]^+$

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 $L1$

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 $L1$, que en este caso se ordena de forma creciente atendiendo a los momentos de cierre de cada uno de los R nodos seleccionados.

Procedimiento de asignación de un nodo i a una ruta k

Se seleccionan los R nodos con momentos de cierre más tempranos de la lista de nodos no asignados $L0$. A continuación estos nodos se ordenan de forma ascendente teniendo en cuenta sus momentos de cierre, de forma que los nodos con momentos de cierre más tempranos son aquellos que primero serán atendidos. En este caso, en contra de lo que ocurría al comienzo de la fase de planificación, la holgura ya no es una medida válida, debido a que para cada uno de los nodos en consideración de la lista $L1$, las holguras son diferentes según la ruta que se considere. Por ello se opta primero por ordenar los nodos según su requerimiento de inmediatez representado ahora por su momento de cierre, para posteriormente analizarlos en detalle.

Para estos nodos se genera una tabla en la que establecen los cálculos de los tiempos de llegada estimados desde cada una de las cabezas de ruta existentes hasta cada uno de los nodos en consideración. El cómputo del tiempo estimado de llegada del vehículo que está sirviendo la ruta k al nodo en consideración i equivale al cálculo de $tr_k + t_{CRki}$

Es decir que se computa el tiempo acumulado de la ruta k en cuestión más el tiempo de desplazamiento entre el último nodo de la ruta k , denominado *cabeza de ruta k* , hasta el nodo i en cuestión.

Este cómputo se realiza para todos y cada uno de los nodos integrados en $L1$, y para todas las rutas existentes en ese momento, de manera que obtenemos una matriz con $L1$ filas y R columnas donde se disponen todos los valores del problema.

A continuación se comparan estos tiempos estimados de llegada con los momentos de cierre de cada uno de los nodos de la lista $L1$ para verificar si la llegada de los vehículos se produce a tiempo para realizar el servicio. De esta forma es posible volver a calcular unas holguras que van a servir para decidir cuáles de los nodos serán asignados a continuación. El cálculo de las holguras para un nodo i a ser visitado por la ruta k se realiza de la siguiente manera:

$$h_i = t_{c_i} - [tr_k + t_{CRki}]$$

En este sentido solamente se pueden considerar las holguras positivas dado que en caso de resultar negativa implicaría que el vehículo de esa ruta no llegaría a tiempo de realizar el servicio de ese nodo.

CITIES IN COMPETITION

Una vez obtenidos estos valores, entonces se selecciona el nodo con cierre más temprano y se estudia en detalle. A este nodo en consideración se denota como nodo i , y es el que tiene $\min tc_i$ para los nodos de la lista LI . Entonces se procede al estudio de la holgura de este nodo i .

La primera operación que se realiza es la de asignación del nodo i a una ruta k existente, siempre y cuando la holgura resultante sea positiva. Si esto no ocurre, entonces es que no existe ninguna ruta a la que poder asignar este nodo i , por lo que se procedería a la creación de una nueva ruta, tal y como se describía al inicio del algoritmo.

Sin embargo, si existiesen holguras resultantes positivas, entonces se escoge aquella ruta a la que esté más cercano, y que no tiene por qué ser la de menor holgura. Esta consideración es muy importante. La asignación a una ruta determinada se hace en función del criterio de cercanía del nodo en cuestión a esa ruta, y no de la holgura. La holgura es un valor que se utiliza para establecer si el vehículo llega a tiempo o no, y si llegase a tiempo, entonces sirve para analizar la posible inserción de nodos intermedios, según sea la amplitud de la holgura.

Procedimiento de adición de un nodo i a una ruta k

Una vez asignado un vehículo a una ruta, entonces se procede al estudio de la holgura. Si esta es inferior al menor valor de los tiempos de servicio ($\min ts_i$), entonces no es posible realizar ningún tipo de parada intermedia, por lo que se procedería a la adición directa de este nodo a la cabeza de ruta k , de manera que la actualización de los valores de la ruta k serían los siguientes²¹ siempre y cuando se cumplan las restricciones:

Llegada a tiempo al nodo i desde la ruta k

$$tr_k + t_{CRk i} \leq tc_i$$

Eventual llegada al depósito antes de que éste cierre

$$tr_k + t_{CRk i} + ts_i + t_{i0} + te_{ki} \leq tc_0 \quad \text{donde } te_{ki} = [ta_i - (tr_k + t_{CRk i})]^+$$

Cumplimiento con la carga total del vehículo

$$c_k + d_i \leq q$$

Los valores resultantes de la ruta k a la que se añade el nodo i serían los siguientes:

$$\begin{aligned} LR'_k &= [LR_k, i] \\ CR'_k &= [i] \\ c'_k &= c_k + d_i \\ tr'_k &= tr_k + t_{CRk i} + ts_i + te_{ki} \\ D'_k &= D_k + t_{CRk i} \\ E'_k &= E_k + te_{ki} \end{aligned}$$

En este caso la espera es $te_{ki} = [ta_i - (tr_k + t_{CRk i})]^+$

²¹ Se ha utilizado la notación con apóstrofe, indicando los nuevos valores de la ruta y que son acumulativos de los valores anteriores.

Generación de la lista $L2$ de escalas intermedias entre el final de la ruta k y un nodo i

Si la holgura resultante es superior al valor del mínimo tiempo de servicio, entonces se procede al análisis de las posibles inserciones de nodos entre CR_k y el nodo i en consideración. Para ello se genera una lista $L2$ de posibles inserciones. Esta lista $L2$ va a estar compuesta por nodos m para los que se cumplen una serie de condiciones, al igual que ocurría al principio, aunque ahora el nodo de partida en vez de ser el nodo 0 , será el nodo correspondiente a la *cabeza de ruta k* . Por ello las restricciones correspondientes serán las siguientes para los nodos m a ser insertados entre CR_k y el nodo i :

Restricción de apertura del nodo m

El nodo m en consideración ha de estar abierto para cuando llegue el vehículo procedente del nodo correspondiente a la *cabeza de ruta k* .

$$ta_m \leq tr_k + t_{CRk m}$$

Restricción de llegada a tiempo al nodo i en cuestión

Los nodos m han de estar abiertos para realizar su visita antes de que cierre el nodo i .

$$ta_m + ts_m + t_{mi} \leq tc_i$$

Restricción de tiempo de desplazamientos total

En segundo lugar la visita del nodo m desde el final de la ruta k , ha de permitir la llegada a tiempo al nodo i .

$$tr_k + t_{CRk m} + ts_m + t_{mi} \leq tc_i$$

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.

$$tr_k + t_{CRk m} + ts_m + t_{mi} + nh_i = tc_i \text{ donde } nh_i \text{ ha de ser mayor o igual que } 0.$$

Restricción de desplazamiento geográfico máximo autorizado

No ha de suponer un gran desplazamiento ya que solamente se permitirán aquellos desplazamientos para realizar las visitas de los nodos intermedios k que supongan un máximo de β veces el desplazamiento original entre la cabeza de la ruta k y el nodo i .

$$t_{CRk m} + t_{mi} \leq \beta t_{CRk i}$$

Eventual llegada estimada de retorno al depósito antes que éste cierre

$$tr_k + t_{CRk m} + ts_m + t_{mi} + ts_i + t_{i0} + te_{ki} \leq tc_0 \text{ siendo } te_{ki} = [ta_i - (tr_k + t_{CRk m} + ts_m + t_{mi})]^+$$

Cumplimiento con la carga total del vehículo

$$c_k + d_m + d_i \leq q$$

CITIES IN COMPETITION

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 actualizan los valores de la ruta k en cuestión de la siguiente manera:

$$\begin{aligned} LR'_k &= [LR_k, i] \\ CR'_k &= [i] \\ c'_k &= c_k + d_i \\ tr'_k &= tr_k + t_{CRk\ i} + ts_i + te_{ki} \\ D'_k &= D_k + t_{CRk\ i} \\ E'_k &= E_k + te_{ki} \end{aligned}$$

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

Generación de la lista $L3$ de dobles escalas intermedias entre el final de la ruta k y un nodo i

Si la lista $L2$ no está vacía entonces el algoritmo procede a la generación de la lista de posibles inserciones dobles, $L3$ entre la cabeza de la ruta k y el nodo i en consideración. Para ello selecciona pares de nodos de $L2$ para generar la lista $L3$ de los nodos a ser insertados simultáneamente. Estos nodos m y l cumplen las siguientes restricciones:

Restricción de apertura de los nodos m y l

$$\begin{aligned} ta_m &\leq tr_k + t_{CRk\ m} \\ ta_l &\leq tr_k + t_{CRk\ m} + ts_m + t_{ml} \end{aligned}$$

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 .

$$tr_k + t_{CRk\ m} + ts_m + t_{ml} + ts_l + t_{li} \leq tc_i$$

Incluyendo la variable de holgura para deshacer la inecuación, la expresión anterior resultaría en:

$$tr_k + t_{CRk\ m} + ts_m + t_{ml} + ts_l + t_{li} + nh'_i = tc_i \text{ donde } nh'_i \text{ ha de ser mayor o igual que } 0.$$

Restricción de desplazamiento geográfico máximo autorizado

Además de las restricciones anteriores, es necesario comprobar que no implican un gran desplazamiento, y en este caso el desplazamiento total será menor o igual que γ veces el valor de β veces la distancia original. Matemáticamente:

$$t_{CRk\ m} + t_{ml} + t_{li} \leq \gamma \beta t_{CRk\ i}$$

Eventual llegada estimada de retorno al depósito antes que éste cierre

$$tr_k + t_{CRk\ m} + ts_m + t_{ml} + ts_l + t_{li} + ts_i + te_{ki} + t_{i0} \leq tc_0 \text{ donde } te_{ki} = [ta_i - (tr_k + t_{CRk\ m} + ts_m + t_{ml} + ts_l + t_{li})]^+$$

Cumplimiento con la carga total del vehículo

$$c_k + d_m + d_l + d_i \leq q$$

Una vez computadas todas las nuevas holguras para todas las posibles combinaciones de nodos m y l , 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 inicializadas al principio de la siguiente manera:

$$\begin{aligned} LR'_k &= [LR_k, m, i] \\ CR'_k &= [i] \\ c'_k &= c_k + d_m + d_i \\ tr'_k &= tr_k + t_{CRkm} + t_{sm} + t_{mi} + t_{si} + te_{ki} \\ D'_k &= D_k + t_{CRkm} + t_{mi} \\ E'_k &= E_k + te_{ki} \end{aligned}$$

En este caso $te_{ki} = [ta_i - (tr_k + t_{CRkm} + t_{sm} + t_{mi})]^+$

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 de $L3$ que maximizan el valor de la nh'_i de manera que se añaden estos dos nodos a la ruta k de la siguiente manera:

$$\begin{aligned} LR'_k &= [LR_k, m, l, i] \\ CR'_k &= [i] \\ c'_k &= c_k + d_m + d_l + d_i \\ tr'_k &= tr_k + t_{CRkm} + t_{sm} + t_{ml} + t_{sl} + t_{li} + t_{si} + te_{ki} \\ D'_k &= D_k + t_{CRkm} + t_{ml} + t_{li} \\ E'_k &= E_k + te_{ki} \end{aligned}$$

En este caso el $te_{ki} = [ta_i - (tr_k + t_{CRkm} + t_{sm} + t_{ml} + t_{sl} + t_{li})]^+$

A continuación se elimina el nodo i de la lista $L1$ y los nodos m y l de la lista $L0$

De esta manera se procede hasta que todos los nodos han sido asignados, y finalmente se genera la solución final con el detalle de todas las rutas.

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 0 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 :

$$\begin{aligned} LR'_k &= [LR_k, 0] \\ CR'_k &= [0] \\ c'_k &= c_k \\ tr'_k &= tr_k + t_{CRk0} \\ D'_k &= D_k + t_{CRk0} \\ E'_k &= E_k \end{aligned}$$

CITIES IN COMPETITION

Finalmente el algoritmo genera una tabla con los datos resumen de cada una de las rutas

SOLUCIONES

Para ello se desarrolló un programa informático y se aplicó 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.

CONCLUSIONES

Las técnicas de construcción presentadas en este trabajo para resolver el problema del VRPTW, permiten mejorar los métodos originales de construcción de rutas propuestos por Solomon (1987), y desarrollados por otros tantos autores. En este sentido en este trabajo se buscaba un perfeccionamiento de las reglas de decisión subyacentes en los métodos de construcción para conseguir una mejor combinación de clientes en las rutas de los vehículos. Tal y como se demuestra en los resultados obtenidos, estas soluciones son tanto mejores cuanto mayor es el horizonte temporal del problema. Igualmente se obtienen mejores soluciones cuanto mayor es la dispersión de los nodos en el espacio. En este sentido nos encontramos precisamente con los problemas menos intuitivos, ya que en los casos en los que los nodos están agrupados, y que como es sabido se corresponden con los problemas tipo C (*Clusterized*), la construcción de las rutas es mucho más intuitiva.

Lo interesante de estas reglas también estiba en la velocidad en la construcción de las rutas. Debido al proceso discriminante que se contempla, el número de opciones en las inserciones de nodos consideradas, al igual que en las adiciones de los mismos, permite una mayor agilidad al algoritmo, de manera que es capaz de arrojar soluciones muy rápidas.

Otra de las ventajas reside en la parametrización del modelo, de manera que es posible adaptar el modelo a los condicionantes del problema. Por ejemplo, si existe una mayor dispersión de nodos o clientes en el espacio, entonces será posible reducir las áreas de inserción consideradas, dado que siempre habrá nodos intermedios en el camino. Caso contrario, si nos encontramos con que los nodos está bastante agrupados, entonces será necesario ampliar enormemente las áreas de inserción para encontrar nodos que saturen los tiempos de espera que en otro caso se computarían.

Por último cabe destacar la necesidad de implementar sobre este tipo de procedimientos heurísticos de construcción de rutas, otras reglas o procedimientos de mejora a través de los cuáles se puedan mejorar las soluciones inicialmente obtenidas. Estos procedimientos generalmente hace referencia a los procedimientos de intercambios miopes de nodos o arcos entre diferentes rutas, de manera que paulatinamente se va mejorando la solución inicial. También cabe destacar la posibilidad de implementar este tipo de técnicas de construcción en métodos de búsqueda metaheurísticos, a través de los cuáles se procede a búsquedas intensivas de posibles combinaciones de arcos, para mejorar las soluciones de partida.

A pesar de ello estas técnicas suponen un avance hacia métodos más rápidos y eficaces en la construcción de rutas más cortas para el problema del VRPTW.

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.
- Dullaert W., (2000a), "Impact of Relative Route Length on the Choice of Time Insertion Criteria for Insertion Heuristics for the Vehicle Routing Problem with Time Windows", en *Proceedings of the Rome Jubilee 2000 Conference Improving Knowledge and Tools for Transportation and Logistics Development: 8th Meeting of the Euro Working Group Transportation*, Faculty of Engineering, "La Sapienza", University of Rome, Italy, B. Maurizio (ed), 153-156, Rome
- Dullaert W., (2000b), "*A Sequential Insertion Heuristic for the Vehicle Routing Problem with Time Windows with Relatively Few Customers per Route*", Research Paper, Faculty of Applied Economics UFSIA-RUCA, 2000:014, Antwerpen, Bélgica
- Gillet B. y Miller L.R., (1974), "A Heuristic Algorithm for the Vehicle Dispatch Problem", *Oper. Res.*, Vol. 22, p. 340-349.
- Ioannou G., Kritikos M., y Prastacos G., (2001), "A Greedy Look-Ahead Heuristic for the Vehicle Routing Problem with Time Windows", *Journal of the Operational Research Society* 52, 523-537
- Potvin J.Y., (1993), "The Traveling Salesman problem: A Neural Network Perspective", *INFORMS J. Comp.*, Vol. 5, No. 4, p. 328-348.
- Potvin J.Y. y Rousseau J.M., (1993), "A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows", *Eur. J. Oper. Res.*, Vol. 66, p. 331-340.
- Potvin J.Y. y Rousseau J.M., (1995), "An Exchange Heuristic for Routeing Problems with Time Windows", *J. Oper. Res. Soc.*, Vol. 46, p. 1433-1446.
- Potvin J.Y., Dufors G. Rousseau J.M., (1993), "Learning Vehicle Dispatching with Linear Programming Models", *Comp. Ops. Res.*, Vol. 20, p. 371-380.
- Potvin J.Y., Lapalme G. y Rousseau J.M., (1989), "A generalized K-Opt Exchange Procedure for the MTSP", *INFOR*, Vol. 27, no. 4, p. 474-481.
- Potvin J.Y., Shen Y. y Rousseau J.M., (1992), "Neural Networks for Automated Vehicle Dispatching", *Comp. Ops. Res.*, Vol. 19, No. 3/4, p. 267-276.
- Potvin J.Y., Kervahut T., Garcia B.L. y Rousseau J.M., (1996), "The Vehicle Routing Problem with Time Windows - Part 1: Tabu Search", *INFORMS J. Comp.*, Vol. 8, No. 2, p. 158-164.