

DESARROLLO DE UNA APLICACIÓN INFORMÁTICA PARA LA RESOLUCIÓN DEL PROBLEMA DE PLANIFICACIÓN DE RUTAS DE TRANSPORTE EN ENTORNOS DINÁMICOS

Eduardo Guillén Solórzano, edugs@udc.es
Susana Barbeito Roibal, sbar@cdf.udc.es
Manuel Martínez Carballo, mmc@udc.es
Universidade da Coruña

ABSTRACT

En este trabajo se presenta un nuevo software para la resolución del conocido problema de la planificación de rutas de transporte en entornos de tiempo restringido, también denominado VRPTW. Este nuevo software de resolución se fundamenta en el algoritmo desarrollado a tal efecto y que ha sido denominado ALADIN (Algoritmo de Adición en Inserción). Este algoritmo integra procedimientos de construcción de las rutas de transporte en los que se implementan reglas de adición e inserción de nodos para configurar las diferentes rutas que han de ser servidas. El software desarrollado presenta numerosas ventajas derivadas de su flexibilidad tanto para la modificación de los datos de partida, como para la introducción de nuevos parámetros de usuario, lo que permiten su aplicación incluso en entornos dinámicos en los que las visitas surgen a medida que las rutas se van cubriendo por la flota de vehículos.

Palabras Clave: VRPTW, Métodos Heurísticos, Optimización, Logística, Rutas de Transporte

1. INTRODUCCIÓN

En esta sección se presentan los datos de partida que configuran el problema conocido como Vehicle Routing Problem with Time Windows, y que hace referencia a la planificación de rutas de transporte con ventanas de tiempo. La descripción del problema consiste en que un número determinado de vehículos dispuestos en un almacén central y con unas capacidades y tiempo limitadas, deben de servir a un conjunto de clientes dispersos aleatoriamente en el espacio, ciertas cantidades de mercancía, y que han de suministrarse en momentos concretos del horizonte temporal de planificación. Esta planificación de rutas ha de realizarse recorriendo una distancia total mínima, siendo este el objetivo último del problema.

Notación

Para la resolución de este problema se ha desarrollado un algoritmo aproximativo de solución y se ha programado una aplicación informática de apoyo a la toma de decisiones por parte del usuario. La interfaz del programa se presenta en la figura 1. Esta interfaz contempla dos áreas básicas. En el lado izquierdo de la interfaz se presentan todos los datos requeridos por el programa para la aplicación de las reglas de decisión, y en el lado derecho se proporcionarán las soluciones al usuario.

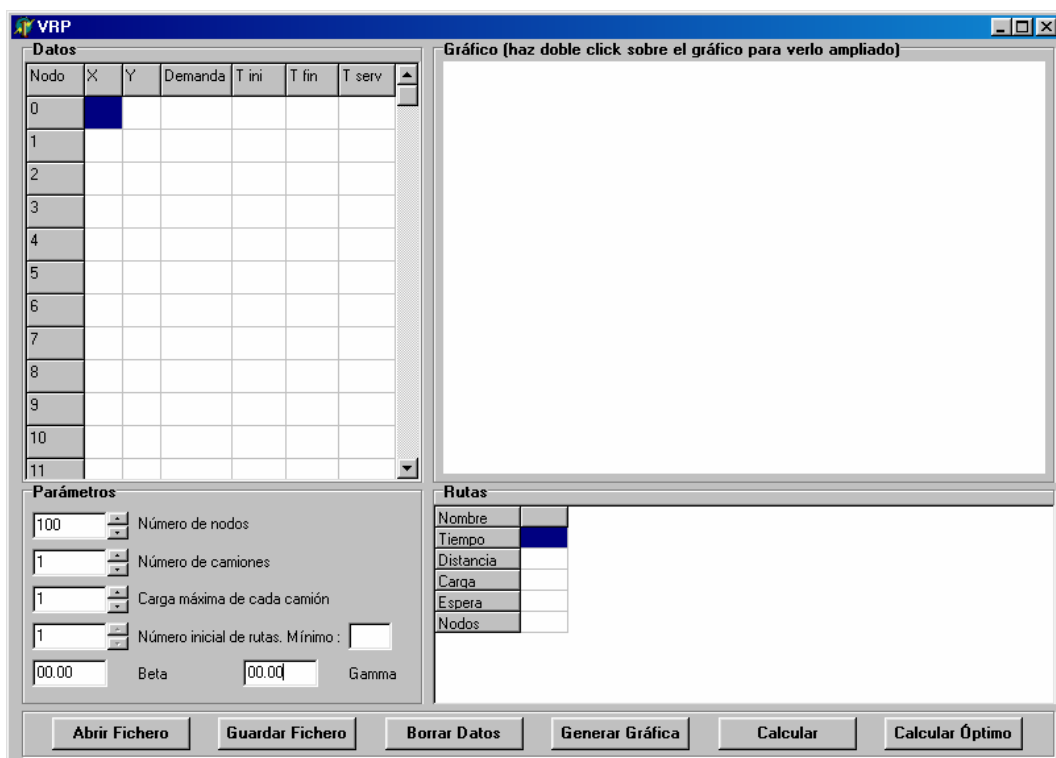


Figura 1. Imagen de la Interfaz de la aplicación

En la sección de los datos de partida se contemplan los siguientes:

Nodo: La columna nodo identifica a todos y cada uno de los nodos existentes en el problema, siendo θ el nombre del nodo que hace de depósito, y $1... N$ los nombres de los siguientes nodos. En una planificación real, los nombres de los nodos serían propiamente nombres de clientes o de ciudades.

Coordenada X: Es la coordenada relativa al eje horizontal, y que se podría identificar con las coordenadas geográficas del nodo en cuestión. En este caso se correspondería con la “*longitud*” Mide la distancia horizontal con respecto al eje de ordenadas.

Coordenada Y: Análogamente representa la coordenada relativa al eje vertical, y por ello se identificaría con la “*latitud*”. Mide la distancia vertical con respecto al eje de abscisas.

Demanda: La demanda hace referencia a la cantidad de producto demandada por el cliente en cuestión. Se podría identificar con los kilogramos, litros, o cualquier otra unidad de medida que se ha de servir, o bien recoger, si aplicamos el criterio de recogidas. Si bien en este trabajo no se ha contemplado la coexistencia de recogidas y entregas simultáneas.

Tiempo de apertura: El tiempo o momento de apertura hace referencia al momento temporal en el que el nodo en cuestión abre para la entrega de mercancías, y a partir del cual ha de entrar un vehículo para realizar el servicio.

Tiempo de cierre: Es el momento temporal en el que el nodo en cuestión cierra sus puertas y a partir del cuál no podría ser servido por ningún vehículo.

Horizonte total de planificación: El nodo central o depósito marca el horizonte temporal en el que abren y cierran todos los nodos.

Tiempo de servicio: Por último, se establece la última variable que es el tiempo de servicio requerido por el vehículo para realizar la entrega en el nodo correspondiente.

Número de nodos de los que consta el problema: Esta variable del problema hace referencia al número total de clientes que han de ser servidos desde el depósito central, y que puede oscilar según los problemas que se traten de resolver. En los problemas tipo aportados en la literatura generalmente se contemplan problemas de 100 nodos con diferentes distribuciones de los mismos. Sin embargo también se han desarrollado versiones más simplificadas con 50 y 25 nodos de estos mismos problemas tipo. Por ello en este trabajo se ha optado por establecer la posibilidad de introducir cantidades variables de clientes a ser servidos por la flota de camiones disponible, para poder afrontar cualquiera de los problemas tipo de la literatura, así como para garantizar una mayor adaptabilidad de los fundamentos del algoritmo a la situación real del problema. Estos nodos cuentan además con el nodo central o depósito desde el que parten y al que han de llegar todos y cada uno de los camiones. Este nodo se identifica con el nodo θ , por ello el número total de nodos de todo problema será el número de clientes $+1$.

Igualmente se establecen como datos el **número de vehículos** disponible para realizar la planificación de las rutas, así como la **capacidad máxima** de cada uno de los vehículos.

En el cuadrante inferior izquierda se contemplan los diferentes parámetros de las que consta el problema. Estos parámetros son definidos por el usuario para conseguir diferentes soluciones.

Número inicial de rutas R : Este parámetro obedece a la necesidad de planificar desde el comienzo del horizonte un número determinado de rutas, a partir del cuál se irá incrementando en función de las necesidades del problema. Se corresponde con el número total de nodos semilla con los que el usuario decide iniciar la planificación.

Parámetro β : Se trata del parámetro penalizador de desplazamientos elevados en las posibles inserciones simples de nodos cada vez que se considera un nodo crítico.

Parámetro γ : Consiste en el parámetro penalizador de desplazamientos en las posibles inserciones dobles de nodos entre el nodo cabeza de ruta y el nodo crítico anterior. Recordemos que se considera conjuntamente con el parámetro β .

2. MÉTODO DE CÁLCULO

A partir de la introducción de estos datos, el programa establece el conjunto de reglas y algoritmos necesarios para obtener una solución. 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. Este algoritmo constituye el alma del programa y su objetivo es resolver el problema de forma aproximativa en un tiempo muy reducido, de manera que la solución se devolverá posteriormente a la Interfaz 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 . 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.
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. En las figuras 2 y 3 se recoge el diagrama de flujo del algoritmo en el que se ha resumido de forma muy escueta su funcionamiento y reglas básicas. Estas reglas se analizan y desarrollan con mayor profundidad en las próximas líneas.

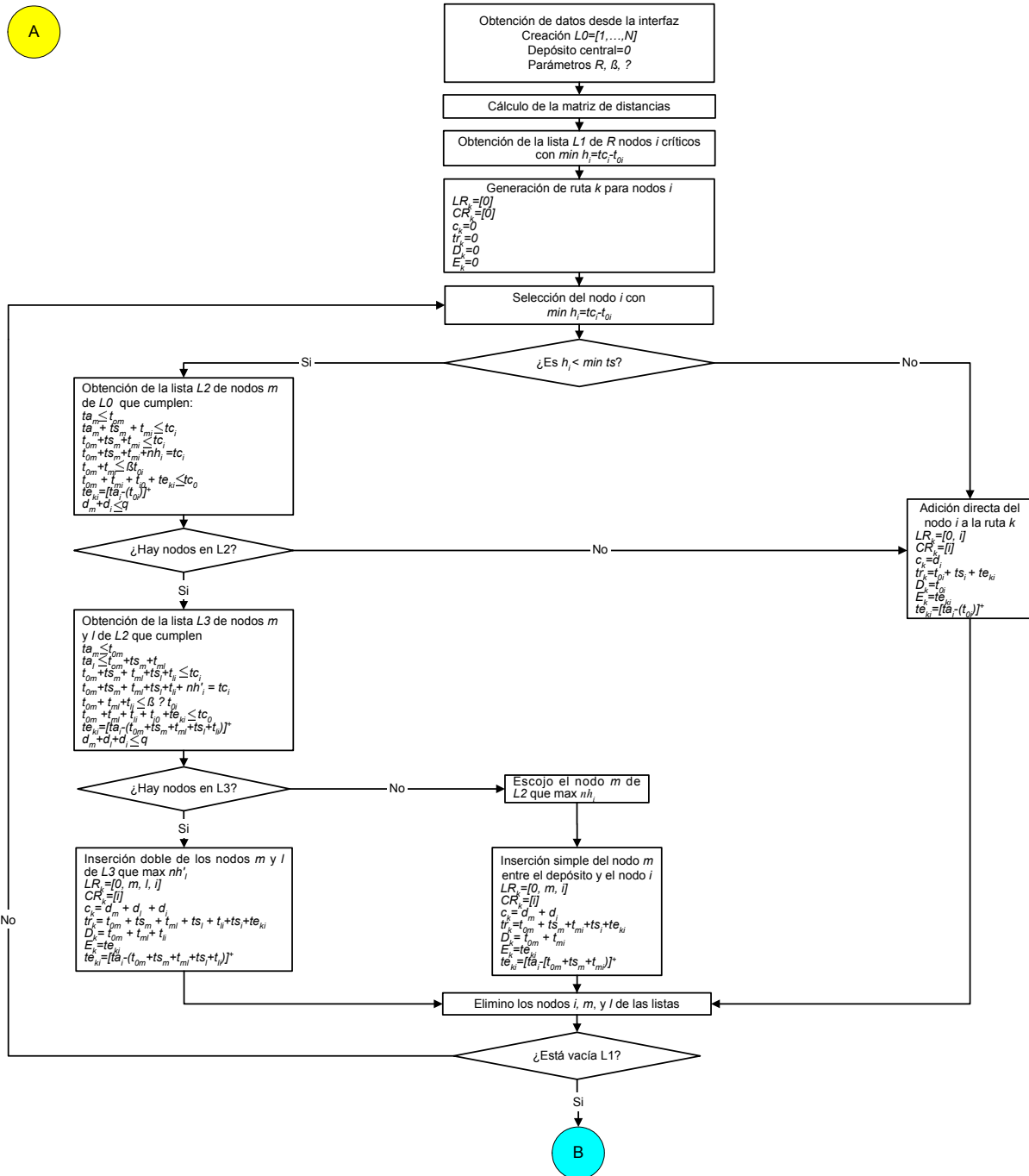


Figura 2. Primera fase del Algoritmo ALADIN

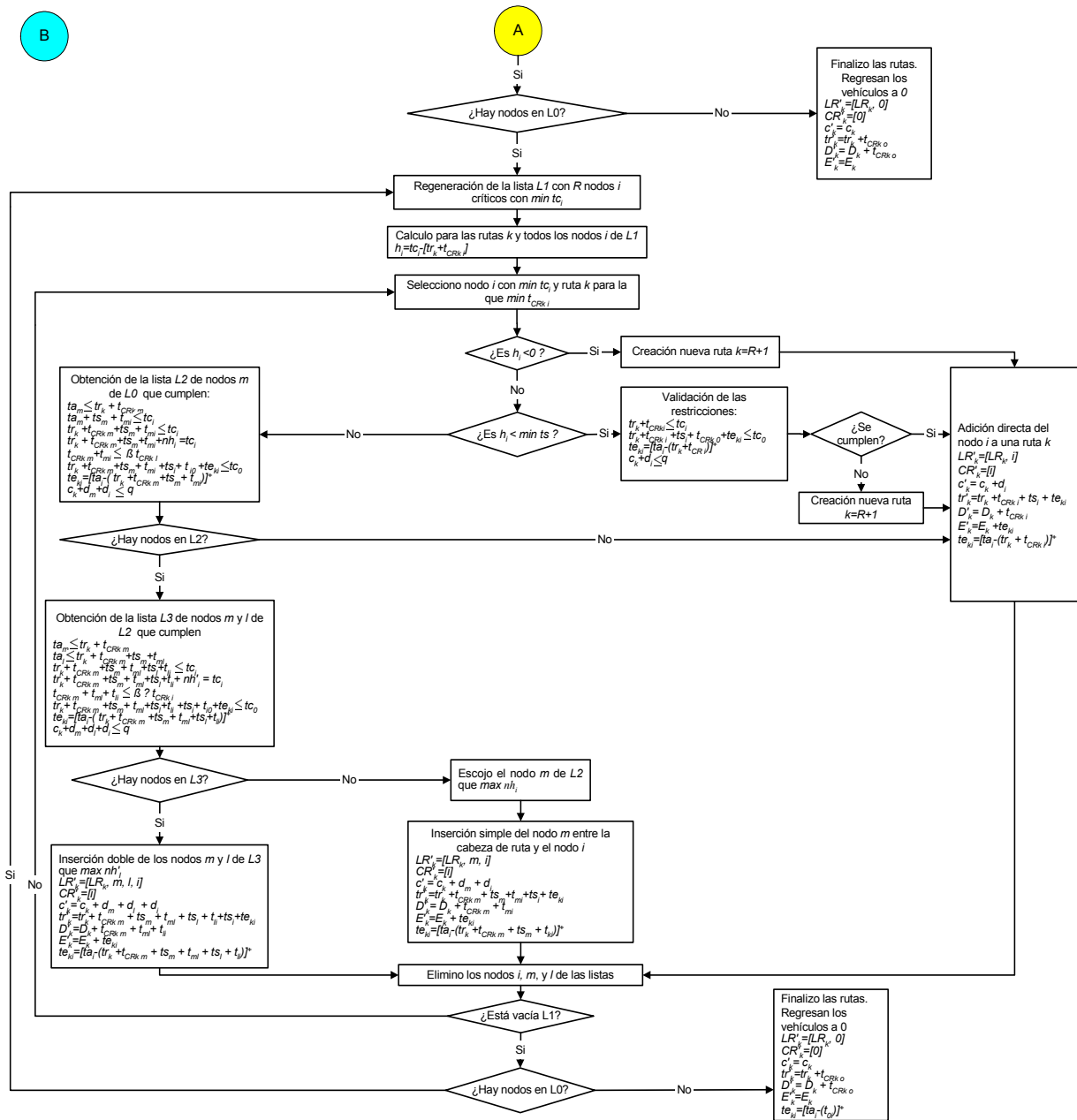


Figura 3. Segunda y posteriores fases del Algoritmo ALADIN

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 $L1$, 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.

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 inequación 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:

$$LR_k = [0, i]$$

$$CR_k = [i]$$

$$c_k = d_i$$

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

$$D_k = t_{0i}$$

$$E_k = te_{ki}$$

En este caso la espera sería $te_{ki} = [t_{si} - (t_{0i})]^+$. 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 $ta_l \leq t_{0m} + t_{sm} + t_{ml}$

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 . $t_{0m} + t_{sm} + t_{ml} + t_{sl} + t_{li} \leq tc_i$. Incluyendo la variable de holgura para deshacer la inecuación, la expresión anterior resultaría en $t_{0m} + t_{sm} + t_{ml} + t_{sl} + t_{li} + nh'_i = tc_i$ 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:

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

$$CR_k = [i]$$

$$c_k = d_m + d_i$$

$$tr_k = t_{0m} + t_{sm} + t_{mi} + t_{si} + te_{ki}$$

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

$$E_k = te_{ki}$$

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:

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

$$CR_k = [i]$$

$$c_k = d_m + d_l + d_i$$

$$tr_k = t_{0m} + t_{sm} + t_{ml} + t_{sl} + t_{li} + t_{si} + te_{ki}$$

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

$$E_k = te_{ki}$$

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 $L1$ (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 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 :

$$LR'_k = [LR_k, 0]$$

$$CR'_k = [0]$$

$$c'_k = c_k$$

$$tr'_k = tr_k + t_{CRk0}$$

$$D'_k = D_k + t_{CRk0}$$

$$E'_k = E_k$$

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

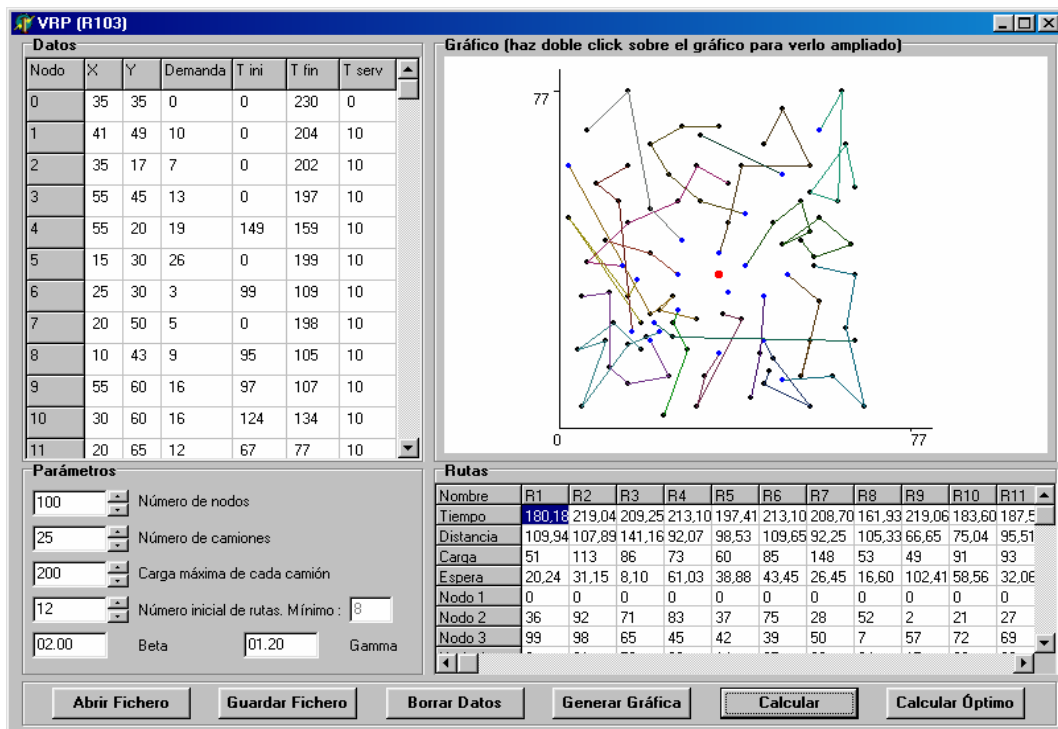


Figura 4. Representación en la interfaz de los datos integrantes de la solución

Finalmente el algoritmo genera una tabla con los datos resumen de cada una de las rutas. Esta tabla contiene la solución del problema aportada por el algoritmo, y su diseño obedece al siguiente esquema.

Número inicial de rutas = 13
Beta = 2,08
Gamma = 1,05

Número final de rutas = 20
Distancia total = 1729,15

Nombre	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13
Tiempo	159,90	218,25	157,06	180,18	207,01	215,54	208,70	161,93	219,06	183,60	194,04	197,41	212,00
Distancia	109,40	91,99	116,03	80,06	127,01	104,32	75,89	105,33	66,65	75,04	101,26	103,52	97,00
Carga	78	127	52	146	124	82	103	53	49	91	93	74	77
Espera	0,50	56,26	1,02	10,12	0,00	61,23	62,81	16,60	102,41	58,56	32,78	33,89	57,00
Nodo 1	0	0	0	0	0	0	0	0	0	0	0	0	0
Nodo 2	36	92	71	83	37	75	28	52	2	21	27	1	8
Nodo 3	47	98	65	45	42	39	50	7	57	72	69	31	1
Nodo 4	48	91	78	5	14	67	33	64	15	23	30	62	1
Nodo 5	8	38	34	96	44	55	3	49	40	22	9	88	6
Nodo 6	46	86	0	99	87	25	76	0	58	74	81	84	9
Nodo 7	0	85	0	6	97	0	79	0	0	0	35	17	1
Nodo 8		93		94	59		77				0	0	7
Nodo 9		0		95	43		0				0	0	0
Nodo 10				13	0								
Nodo 11				0									

Figura 5. Informe de la solución final a través del algoritmo desarrollado

4. RESULTADOS Y CONCLUSIONES

En este apartado se detallan los resultados más destacables de la aplicación de este tipo de software a la resolución del problema del VRPTW.

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.

Otro de las conclusiones más destacables de este Software es su **flexibilidad**. En contra de otras aplicaciones informáticas similares para la resolución del VRPTW, en este caso se permite la modificación de los parámetros del algoritmo para adaptar las reglas de cálculo a cada situación. De esta manera se consigue un mejor resultado a través de la modificación de determinados parámetros.

Igualmente es posible alterar la matriz de distancias. Habitualmente se trabaja con distancias euclídeas entre cada par de puntos, utilizando para su cálculo las posiciones de cada nodo en términos de longitud y latitud. De esta manera el algoritmo obtiene una matriz de distancias que almacena en su memoria para proceder al cálculo de distancias mínimas, tiempos de desplazamiento, etc. Sin embargo en este caso el algoritmo presenta la posibilidad de alterar esa matriz de distancias de forma artificial de manera que sería posible trabajar con distancias reales entre cada par de puntos, lo que permitiría una fácil **adaptación a situaciones reales**. De esta manera sería incluso posible que la matriz de distancias no fuera necesariamente simétrica, de manera que las distancias de ida y vuelta entre dos puntos, podrían ser diferentes, algo que en la realidad ocurre con frecuencia sobre todo en entornos urbanos.

Por último también merece la pena resaltar la facilidad de integración de este algoritmo en **procedimientos de cálculo metaheurísticos**, especialmente la búsqueda tabú o el recocido simulado. Como es sabido los procedimientos de cálculo metaheurísticos obtienen mejores resultados a costa de mayores tiempos de cálculo. Estos procedimientos se fundamentan en búsquedas intensivas guiadas, o aleatorias a través de la prueba y error. En este caso y debido a la gran velocidad de cálculo de este algoritmo ALADIN, obtiene un especial interés para su integración en este tipo de procedimientos más avanzados.

Por último también es conveniente resaltar su **adaptabilidad a entornos dinámicos**. Generalmente los métodos de construcción de rutas de transporte tienen un enfoque global por cuanto contemplan desde el primer momento la ubicación de todos los nodos, así como los posibles servicios de los mismos, según estén o no abiertos en cada momento. Esto puede llevar a decisiones en las que se rechacen buenas uniones de nodos que por ejemplo están cercanos, por otras en las que se obtiene una mejor distancia total, aún no teniendo esas buenas uniones de nodos. Es decir que se trata de buscar una buena aproximación al óptimo global. En este planteamiento, un pequeño cambio o eventualidad en la planificación puede provocar graves consecuencias en las distancias finalmente recorridas, debido a los escasos márgenes de maniobra que presentan las rutas finales obtenidas de esta manera.

Sin embargo el método desarrollado en este trabajo realiza una optimización por etapas, ya que el horizonte de planificación se divide de forma que en cada uno de los intervalos se tiende a saturar las rutas, con las mejores opciones de entre todos los nodos posibles en ese intervalo, pero que en ningún caso son todos los nodos del problema. En este sentido, simplemente no se contemplan estos nodos, hasta que no están abiertos. Por ello cualquier eventualidad próxima que pueda acontecer, no va a afectar a la planificación ya realizada, simplemente porque en esa planificación esa eventualidad no se habría contemplado, como así hubiera ocurrido en los métodos de construcción clásicos.

Por esto el método desarrollado en este trabajo se presenta como especialmente interesante para la planificación dinámica de rutas de transporte.

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.
- Potvin J.Y., y Bengio S., (1996), "The Vehicle Routing Problem with Time Windows - Part 2: Genetic Search", *INFORMS J. Comp.*, Vol. 8, No. 2, p. 165-172.
- Potvin J.Y., Dubé D. y Robillard C., (1996), "A Hybrid Approach to Vehicle Routing Using Neural
- 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.