Elocons: Un algoritmo de construcción de rutas eficiente para la Pequeña y Mediana Empresa de distribución

Elocons: An efficient construction algorithm for vehicle routing problems with time windows in Small and Medium Enterprises

Arturo Nieto de Almeida, Alejandro García del Valle, Diego Crespo Pereira

Grupo Integrado de Ingeniería, Universidad de A Coruña, España

Arturo Nieto de Almeida
EUAT – Universidad de A Coruña
Campus A Zapateira,
La Coruña

e-mail: anieto@udc.es
Telef. 981 167000 ext. 2734

RESUMEN

Este artículo presenta un algoritmo de construcción eficiente y de bajo coste de implementación para la resolución de problemas de rutas de transporte con ventanas de tiempo (VRPTW). En el desarrollo del mismo se ha tenido en cuenta la problemática real de las pequeñas y medianas empresas de transporte, caracterizada por un número reducido de clientes atendidos por una pequeña flota de vehículos. La capacidad organizativa de estas empresas es baja, están poco informatizadas y disponen de poco personal administrativo. El objetivo perseguido fue, por lo tanto, diseñar un heurístico eficiente basado en reglas de decisión lógicas, con un bajo coste de puesta en marcha, que pudiera proporcionar soluciones efectivas a problemas reales con un tiempo de computación bajo utilizando ordenadores de gama media baja. El algoritmo obtenido se testeó con los problemas tipo de Solomon de 100 clientes mostrando un alto grado de desempeño como método de construcción, lo que lleva a considerarlo como altamente eficiente y competitivo.

ABSTRACT

In this paper an efficient low cost construction algorithm is proposed for vehicle routing problems with time windows (VRPTW). The development of *ELOCONS* is focused on Small and Medium Enterprises (SMEs) which can be characterized as having few customers to be attended by a small number of vehicles, and limited IT and organizational capabilities. The objective is to design an efficient heuristic with logical rules of decision making, with a low cost of implementation in SMEs which will bring effective solutions to real problems without the need for excessive computational time. Experimental results on Solomon's 100 customers instances show that the suggested construction method is highly efficient and competitive.

Palabras Clave: Rutas de Transporte; Ventanas de Tiempo; Heurísticos, Logística
Keywords: Routing; Time windows; Heuristics, Logistics

INTRODUCCION

La búsqueda de la eficiencia en todos los procesos de la industria, unida a la deslocalización fragmentada de los procesos productivos hace que la importancia de los sistemas logísticos sea cada vez mayor. La planificación de rutas de transporte, la racionalización de las actividades de distribución y la optimización de rutas de reparto son fundamentales dentro de dichos sistemas.

En los últimos años, en base a los desarrollos de los Sistemas de Información Geográfica y a la disponibilidad de hardware cada vez más potente, se han desarrollado aplicaciones de gestión logística y Sistemas Integrados de Información Logística, dirigidos a grandes empresas dejando desasistidas las pequeñas empresas que cubren la distribución puerta a puerta.

El Problema de rutas de vehículos con ventanas de tiempo (VRPTW) consiste en diseñar un conjunto de rutas con el menor coste posible desde un único depósito hacia un conjunto de clientes geográficamente dispersos, cada uno con su demanda, debe ser atendido por una flota de vehículos, respetando las restricciones de carga máxima, los momentos de apertura y cierre y tiempos de servicio de cada uno de los clientes, el número total de vehículos, y los momentos de apertura y cierre del depósito central del que todos los vehículos parten y al que todos deben volver.

Las aplicaciones de este tipo de problemas abarcan el aprovisionamiento de cadenas de supermercados, distribución de prensa, servicios de vigilancia privados y un largo etc.

En el entorno de las Pymes las rutas tienden a ser cortas y las capacidades financieras, organizativas y de gestión tienden a ser muy reducidas. Por ello y siguiendo los trabajos previos de Guillén y del Valle, circunscribimos este trabajo a las técnicas de construcción de rutas desarrollando un método eficiente que no necesite usar grandes recursos de ningún tipo, ya que más del 70% del total de empresas de transporte españolas son pequeñas compañías con menos de 10 empleados (Estudio Socioeconómico del sector de Transporte por Carretera, Consultrans para Mº de Fomento, 2005). Esta situación refuerza la necesidad de obtener heurísticos rápidos y de fácil implementación, que no requieran un gran desarrollo informático ni un alto nivel organizacional.

La aplicabilidad de soluciones informáticas en el entorno de dichas pymes se basa en cuatro pilares:

Velocidad. Obtención de buenas soluciones en un corto espacio de tiempo, como política de calidad de servicio diferenciadora de la competencia.

Versatilidad. Resolución de problemas muy diversos en cuanto a tipología de clientes, rutas e incluso incidencias durante la ejecución de las mismas.

Bajo costo tecnológico. Mínima inversión en hardware y software.

Generación de buenas soluciones prácticas: La aleatoriedad de incidentes en la práctica puede hacer que soluciones óptimas o cercanas al óptimo se transformen en malas soluciones en la realidad. Por ello las pymes buscan una combinación práctica que les proporcione buenas soluciones junto a la capacidad de cambiar las rutas en caso de que sucedan imprevistos.

El algoritmo diseñado cumple los requerimientos anteriores. La velocidad en la obtención de la mejor solución, realizando todas las combinaciones de parámetros es, en promedio, de 3,1 minutos. Al operar

mediante listas de distancias es muy versátil y no requiere una gran inversión en hardware ni software, de hecho se ha experimentado utilizando un ordenador tipo ofimática con 2 Gb de memoria RAM y Sistema Operativo Windows XP SP3 32 bits. Los resultados obtenidos son muy homogéneos en los diferentes tipos de problemas de Solomon, siendo su desviación media respecto a los mejores resultados conocidos (incluidos óptimos) de 17,2%, y respecto a los mejores resultados obtenidos por métodos heurísticos su desviación promedio es solamente del 14,29%.

La contribución de los autores es el desarrollo de un conjunto de reglas de decisión lógicas para obtener rutas factibles que minimicen los costes totales de transporte en el marco de las restricciones que atenazan a las pymes en la actualidad; implementar ese conjunto de reglas decisionales en un algoritmo efectivo para la toma de decisiones en la planificación de rutas de transporte en un corto espacio de tiempo e integrar el algoritmo en un interface de usuario que permita al planificador incluir datos y recibir información lógica y gráfica para facilitar la toma de decisiones.

Para medir la eficiencia y eficacia del algoritmo que hemos denominado "ELOCONS" lo hemos enfrentado a los problemas tipo de Solomon (1987), que son la referencia en la que este tipo de algoritmos se comparan entre sí.

Descripción del Algoritmo "ELOCONS"

Utilizamos como datos iniciales, el conjunto de datos utilizados en los problemas de Solomon (1987):
Identificación del cliente;
Coordenadas *x,y* de posición del cliente;
Demanda de cada cliente;
Ventana de tiempo de cada cliente, y
Tiempo de servicio de cada cliente;
La notación utilizada en la descripción es la siguiente:

N	*Número total de clientes*
NR	*Número de rutas generadas hasta el momento considerado*
i	*Índice de los clientes: i = 1, 2, …, N*
x_i	*Abscisa de la posición del cliente i*
y_i	*Ordenada de la posición del cliente i*
c_{ij}	*Coste de viajar desde el cliente i al cliente j*
t_{ij}	*Tiempo de desplazamiento entre el cliente i y el cliente j*
ts_i	*Tiempo de servicio del cliente i*
d_{ij}	*Distancia entre el cliente i y el cliente j*
$CR1$	*Cabecera de Ruta más cercana al cliente*
$CR2$	*Segunda cabecera de ruta más cercana al cliente*
d_{iCR1}	*Distancia entre el cliente i y la cabecera de ruta más cercana*
d_{iCR2}	*Distancia entre el cliente i y la segunda cabecera de ruta más cercana*
V	*Número total de vehículos disponibles*
k	*Índice del vehículo k=1,...,V*
Q	*Capacidad de los vehículos en consideración.*
q_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 clientes a asignar por iteración por el algoritmo*

R_0 *Número de clientes 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 del vehículo k en el cliente i*

LR_k *Lista de clientes pertenecientes a la ruta k. Se trata de un vector de clientes 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 clientes que visita el vehículo k, más los tiempos de servicio de los clientes 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 entre dos clientes será la suma de* t_{pq} *cuando* $q > p$

c_k *Carga total del vehículo k. Es la suma de las demandas de los clientes ya visitados.*

CR_k *La cabeza de la ruta k, se corresponde con el cliente 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 clientes que visita este vehículo.*

E_k *Espera total en la que incurre el vehículo k.*

El usuario puede definir la capacidad de carga del vehículo, los vehículos disponibles, y algunos parámetros para ajustar el desarrollo del algoritmo ELOCONS, tales como el número R_0 clientes/nodos semilla, que son el número de clientes a considerar en la iniciación del algoritmo, el número de clientes *R* que se asignan en cada iteración, el parámetro de penalización *β* que permite acotar la región de búsqueda de clientes para inserciones sencillas, y el parámetro multiplicador γ que permite ampliar la región de búsqueda de clientes a considerar en inserciones dobles.

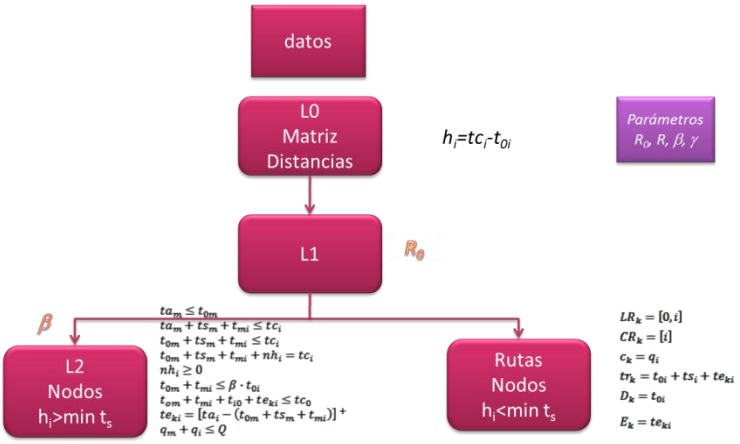

Figura 1: Proceso de Inicialización de Rutas

En primer lugar ELOCONS genera la lista *L0* con el total de clientes del problema, en dicha lista se distingue si los clientes están asignados o no asignados a rutas, y calcula la matriz de distancias entre clientes y de éstos últimos al depósito central.

A partir de la matriz de distancias y las ventanas de tiempo de los clientes ELOCONS genera la lista *L1* que incluye los R_0 clientes críticos, es decir con menor holgura, considerando ésta como la diferencia entre el momento de cierre del cliente y el tiempo de desplazamiento desde el depósito central hasta el cliente *i*. La lista se ordena de menor a mayor holgura y se inicializan las R_0 rutas a tener en cuenta en el momento inicial de la planificación, añadiéndoles el nodo 0 (depósito central) como punto de partida.
Se seleccionan los clientes cuyas holguras sean menores que los tiempos de servicio y se asignan a las *k* rutas, actualizando los valores de dicha ruta y eliminando los clientes de la lista *L0*.
Los clientes cuya holgura es superior al menor tiempo de servicio, se asignan a una nueva lista *L2* de clientes a ser insertados. Para ello deben cumplir que el vehículo llegará desde el depósito central al cliente con posterioridad a la apertura del mismo, y una vez atendido éste, se desplazará al cliente *i,* llegando antes de que este cierre. La holgura Pasa a ser la diferencia entre la suma de los tiempos de desplazamiento desde el depósito central al cliente *m* y después al cliente *i* y el momento de cierre del cliente *i*.
Se aplica el parámetro *β* de tal forma que el tiempo de desplazamiento desde el depósito central al cliente *m* y desde este al cliente *i* sean menores que el tiempo de desplazamiento directo desde el depósito central al cliente *i* multiplicado por el parámetro *β*, esto significa en la práctica que el algoritmo solo tendrá en consideración los clientes que se encuentren en una elíptica entre el depósito central y el nodo *i*.

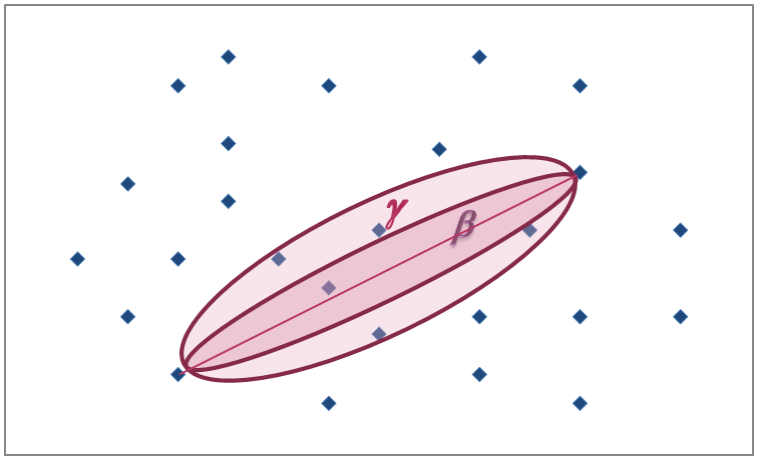

Figura 2 Representación visual de los parámetros β y γ

El vehículo después de desplazarse a los clientes, servirlos, esperar en su caso en el cliente *i* servirlo y volver al depósito central llegará antes de que éste haya cerrado. Además la carga del vehículo no se verá excedida por las demandas atendidas.
Si todo lo anterior no se cumple la lista *L2* estará vacía el vehículo no tendrá otra opción que ir directamente al cliente *i* y esperar que éste abra para ser servido. Se elimina el cliente de las listas *L0* y *L1*.

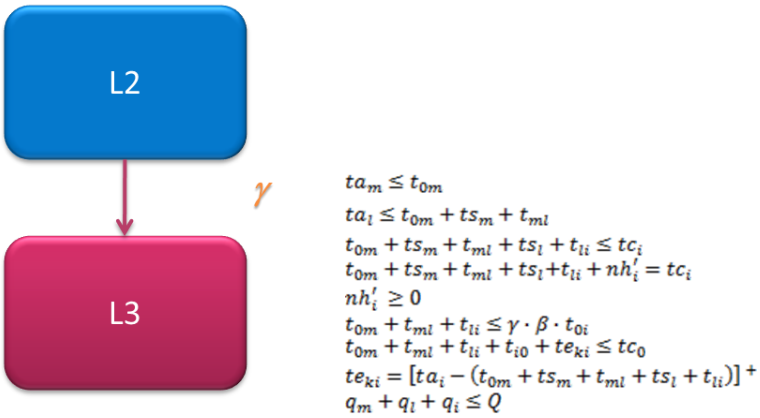

Figura 3 Utilización del parámetro γ en inserciones dobles

Si, por el contrario, la lista *L2* tiene clientes pendientes de ser visitados, se puede generar una nueva lista *L3* de clientes *m* y *l* de la lista *L2* para realizar inserciones dobles, si se cumplen que los clientes hayan abierto sus ventanas de tiempo cuando llegue el vehículo, al primero directamente desde el depósito central y al segundo después de haber servido al primero y desplazarse al segundo, servirlo y darle tiempo a llegar al tercer cliente *i* antes de que cierre, la diferencia entre el momento en que llega el vehículo *k* al cliente *i* y el momento de cierre es la nueva holgura.

Estos clientes, que serán insertados en pareja, estarán comprendidos en un espacio elíptico comprendido entre el depósito central y el cliente i, calculado mediante la utilización de los parámetros *γ* y *β*.

Por supuesto debe cumplirse que el vehículo después de atender a los clientes, podrá volver antes del cierre al depósito central y su carga no se verá excedida por las demandas atendidas.

ELOCONS procede a realizar inserciones dobles de los clientes de la lista *L3* que tienen máxima nueva holgura en el cliente *i* en la ruta originalmente creada quedando la ruta actualizada en el cliente *i*:

Una vez realizada la doble inserción ELOCONS procede a borrar al cliente *i* de las listas *L1* y *L0* y a los clientes *m* y *l* de la lista *L0* y *L3*.

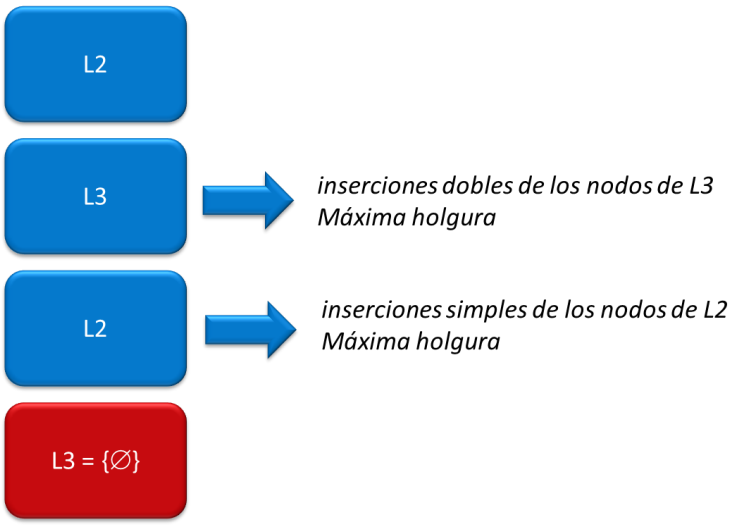

Figura 4 Inserciones dobles de clientes de L3 e inserciones simples de clientes de L2 hasta que L3 se vacíe

Si la lista *L3* no contiene clientes se elige el cliente *m* de la lista *L2* que maximice la holgura en el cliente *i*, adicionándolo a una de las rutas creadas quedando ésta actualizada en el cliente *i* después de haber visitado el cliente *m*:
ELOCONS procede a borrar el cliente *i* de las listas *L0* y *L1* y al cliente *m* de la lista *L0* y *L2*.
El algoritmo realiza iterativamente estos pasos hasta que la lista *L1* esté vacía, lo que significa que los R_0 clientes críticos están ya asignados a rutas. En ese momento ELOCONS comprueba si la lista *L0* también está vacía.
SI así fuera, significaría que todos los clientes, críticos y no críticos estarían asignados a ruta y por lo tanto la construcción de rutas habría finalizado por lo que el algoritmo ELOCONS procedería a finalizarlas todas haciendo volver a todos los vehículos al depósito central.

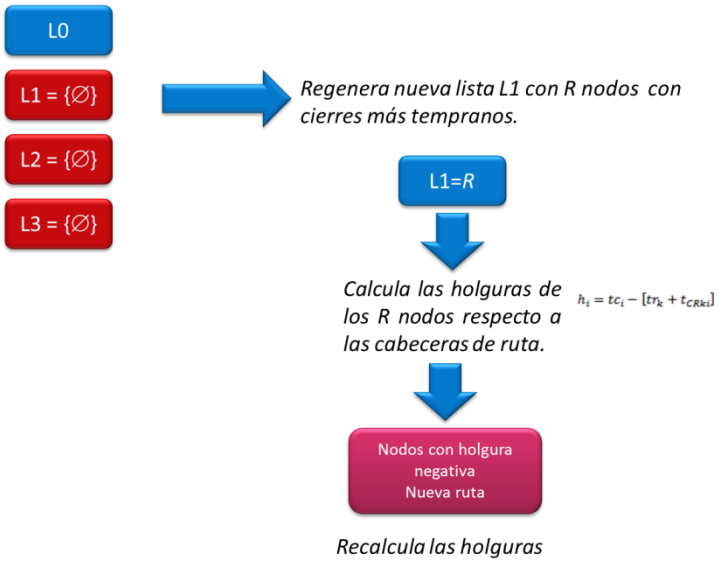

Figura 5 Regeneración de la lista L1con R nuevos clientes

Mientras haya clientes en la lista *L0*, se vuelven a generar nuevas listas *L1* con *R* nuevos clientes en un proceso iterativo. *R* es distinto de R_0 y es un parámetro definido por el usuario. Estos *R* nuevos clientes son seleccionados por orden de momento de cierre más temprano.
ELOCONS calcula si los nuevos *R* clientes pueden ser accedidos desde las rutas actuales mediante el cálculo de las holguras de cada una de las *k* rutas a cada uno de los *i* clientes de la lista *L1*. Si la holgura es negativa en cada ruta significa que el cliente no puede ser accedido desde ninguna de las rutas existentes en la actualidad, por lo que se incluye en una lista denominada *LSR*.

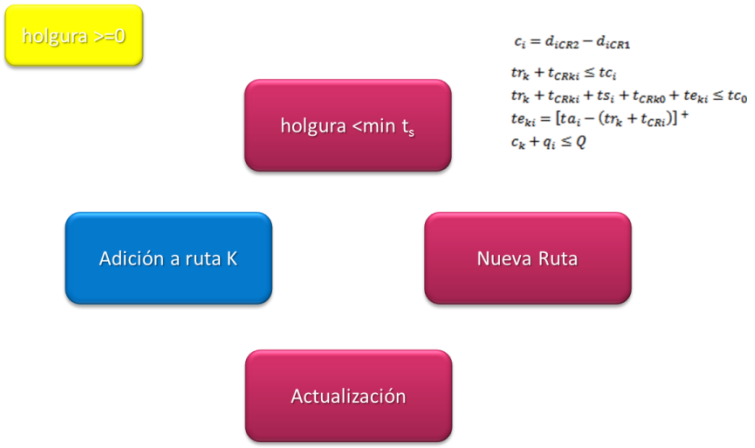

Figura 6 Cálculo de la holgura en cada ruta y opciones si la holgura es menor que el menor tiempo de servicio

SI la holgura es positiva, se calcula el coste en distancia c_i de no asignar el cliente *i* a su ruta más cercana. Este coste lo definimos como la diferencia entre las distancias desde las cabeceras de ruta y la cabecera de ruta más cercana.

ELOCONS elige el cliente con el mayor coste en términos de distancia, esto es la mayor diferencia en distancia desde el cliente hasta la cabecera de ruta más cercana comparado con cualquier otra cabecera de ruta, eligiendo la ruta con menor distancia desde cabecera de ruta al cliente *i* con holgura no negativa. Si la holgura fuese negativa entonces se crearía una nueva ruta hasta el cliente *i*.

Se comprueba para esta ruta si la holgura es mayor o menor que el tiempo de servicio.

Si la holgura es menor que el menor tiempo de servicio, *ELOCONS* comprueba que el vehículo *k* puede acceder al cliente *i* antes de que cierre y que llegará al depósito central antes de su cierre después de haber servido al cliente *i* e incluso haber esperado la apertura del mismo en su caso, y su capacidad no se excede con la adición de la demanda del cliente *i*. Si se cumplen todas las restricciones indicadas el cliente *i* se adiciona directamente a la ruta *k* y se actualiza la ruta, si no, se genera una nueva ruta, inicializándola. En ambos casos el cliente *i* es eliminado de las listas *L0* y *L1*.

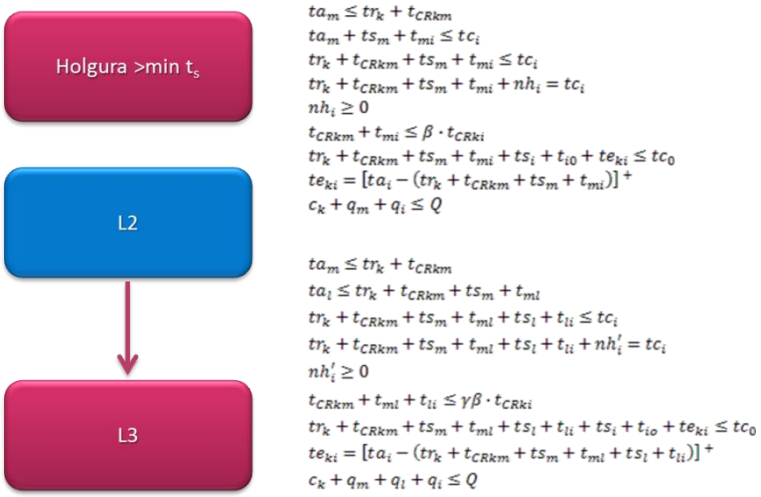

Figura 7 Rutina de ELOCONS si la holgura es mayor que el menor tiempo de servicio

Si la holgura es mayor que el menor tiempo de servicio, ELOCONS genera una nueva lista *L2* de clientes *m* de *L0* que se encuentran en el camino para ser insertados y que cumplan todas las restricciones explicitadas anteriormente para la inserción simple.

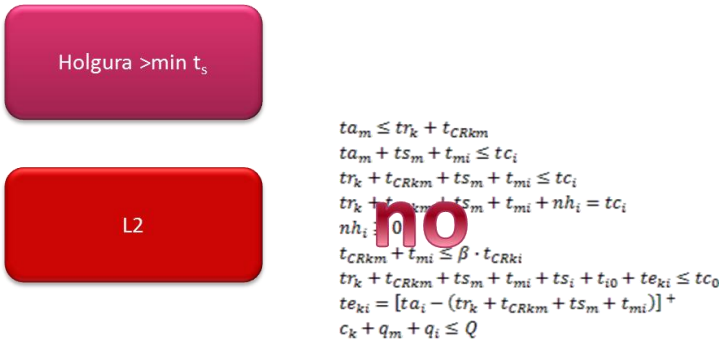

Figura 8 Rutina del algoritmo Elocons si la holgura es mayor al menor tiempo de servicio y la lista L2 está vacía

Si la lista *L2* no contiene clientes, el algoritmo no puede proceder a realizar inserciones en la ruta *k* que visita al cliente *i*, por lo tanto la única opción que queda es ir al cliente *i* y esperar que este abra. ELOCONS procede entonces a actualizar la ruta *K* y eliminará el cliente *i* de las listas *L0* y *L1*.
El proceso se repetirá mientras haya clientes en la lista *L1* calculando holguras en las demás rutas. Cuando *L1* se vacía, ELOCONS recalcula *L1*.

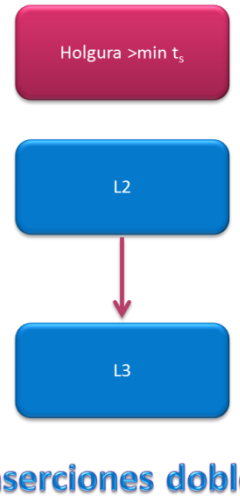

Figura 9 Rutina seguida por Elocons si la holgura es mayor que el menor servicio y L2 tiene clientes

Si la lista *L2* tiene clientes, ELOCONS genera una nueva lista *L3* de posibles inserciones dobles de clientes intermedios entre la cabecera de ruta *k* y el cliente *i* en estudio. Se seleccionan los pares de clientes *m* y *l* que cumplan las condiciones explicitadas anteriormente para la realización de inserciones dobles.
El algoritmo realiza inserciones dobles en la ruta *k* mientras haya clientes en *L3*, seleccionando pares de clientes que maximicen el valor de la holgura en el cliente *i*.
Hecha la inserción, ELOCONS procede a eliminar el nodo *i* de las listas *L0* y *L1* y los nodos *m* y *l* de la lista *L0*.

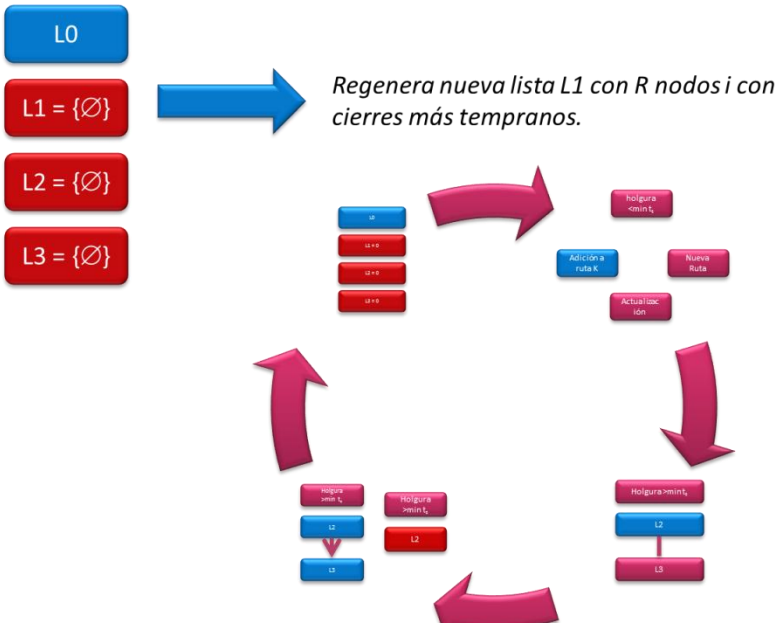

Figura 10 Proceso Iterativo

ELOCONS repite el proceso anterior, recalculando las holguras mientras haya clientes en *L1*. Cuando *L1* esté vacía *ELOCONS* procede a recalcular *L1* con nuevos clientes.
Cuando la lista *L3* queda vacía, ELOCONS procede a elegir el cliente *m* de la lista *L2* que maximice la holgura en el cliente *i*, y se añade a una de las rutas inicializadas al principio y se elimina el cliente *i* de las listas *L0* y *L1*.
Este proceso se repite hasta que no queden clientes en la lista *L0*. En ese momento ELOCONS procede a retornar los vehículos al depósito central y cerrar todas las rutas.
Finalizadas todas las operaciones, ELOCONS genera una tabla con los datos resumen de cada una de las rutas y los datos globales de la solución obtenida.

Figura 11 Finalización de todo el proceso de ELOCONS

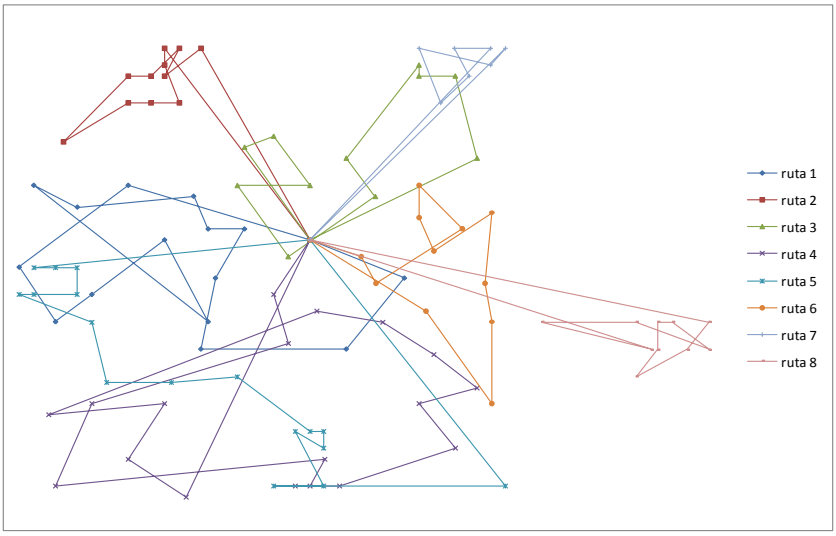

Figura 12 Ejemplo de ruta calculada por el algoritmo

RESULTADOS

En todo momento se ha mantenido como objetivo el diseño de un método que, aplicando una lógica práctica, pudiera resolver eficientemente, y en un tiempo reducido, el problema diario al que se enfrentan multitud de micro y pequeñas empresas de transporte, con capacidades de hardware y software relativamente bajas.

Para establecer comparaciones con la mayoría de los algoritmos existentes, se ha sometido el algoritmo a los problemas tipo de 100 clientes de Solomon (1987) en todas sus variantes.

El hardware utilizado ha sido un ordenador AMD Athlon 64 Processor 3000+ a 2Ghz. La RAM del equipo era de 2Gb y el sistema operativo del equipo era el Windows XP SP3 32 bits.

El algoritmo de cálculo de rutas se programó en Java 6 y el IDE (Entorno de Desarrollo Integrado) empleado fue el Netbeans 6.0 que es un entorno de desarrollo multiplataforma abierto y gratuito, de disponibilidad inmediata en la red.

Para cada problema se buscó la mejor combinación de parámetros, variando R_0 y R entre los valores 1 y 20, β entre 1 y 2,5 con incrementos de 0,1 y γ entre 1 y 2,5 con incrementos de 0,5. Ello supone, para la resolución de cada uno de los problemas tipo, realizar 25.600 combinaciones diferentes de dichos parámetros. Los tiempos de CPU obtenidos incluyen, por lo tanto, la realización de las 25.600 combinaciones. Dado que los tiempos de resolución eran muy cortos, no se estimó la realización de un análisis sobre las mejores combinaciones de parámetros.

Los problemas tipo se clasifican según la distribución geográfica de los clientes y el tipo de restricciones temporales, como se puede ver en la tabla siguiente:

Tabla 1 Variedades y características de los problemas tipo

Problemas tipo	Distribución de Nodos	Restricciones temporales
R1	Aleatoria uniforme	Muy Restrictivas
R2	Aleatoria uniforme	Menos restrictivas
C1	Clusterizados	Muy restrictivas
C2	Clusterizados	Menos Restrictivas
RC1	Clusterizados y aleatorios	Muy restrictivas
RC2	Clusterizados y aleatorios	Menos restrictivas

La referencia alfabética (**R** ó **C**) nos indica el tipo de dispersión geográfica de los clientes; en los problemas del tipo **R** nos enfrentamos a clientes dispersos uniformemente en el plano, y en los problemas de tipo **C** nos enfrentamos a clientes agrupados. Obviamente en los problemas tipo **RC** los clientes están distribuidos en el plano de forma mixta.

La referencia numérica nos indica el nivel de "estrés" provocado por las ventanas de tiempo, así, los problemas tipo **1**, tienen un horizonte de planificación muy breve, por lo que cada ruta puede realizar pocas visitas a clientes, y los problemas tipo **2** tienen horizontes de planificación más amplios, de manera que no son necesarios tantos vehículos como en el caso anterior. Generalmente se trata de problemas cuya solución integra pocas rutas, y éstas son muy largas, incluyendo muchas visitas.

Variando, en cada una de las posibles combinaciones de dispersión y ventanas de tiempo, las demandas de cada uno de los clientes y sus ventanas de tiempo, obtenemos entre 8 y 12 problemas diferentes por cada una de las combinaciones.

La siguiente tabla muestra los resultados obtenidos por ELOCONS y la comparación con los mejores resultados conocidos, obtenidos por métodos heurísticos, publicados en Enero de 2.010 en la página web de SINTEF:

http://www.sintef.no/Projectweb/TOP/Problems/VRPTW/Solomon-benchmark/100-customers/

Tabla 2 Comparativa Resultados ELOCONS - Mejores resultados obtenidos por heurísticos

	Distancia			
	Mejor Heurístico	**ELOCONS**	**Diferencia**	**Ratio**
r101	1.645,79	1.782,44	136,65	8,30%
r102	1.486,12	1.621,47	135,35	9,11%
r103	1.292,68	1.358,04	65,36	5,06%
r104	1.007,24	1.230,89	223,65	22,20%
r105	1.377,11	1.518,08	140,97	10,24%
r106	1.251,98	1.384,62	132,64	10,59%
r107	1.104,66	1.237,53	132,87	12,03%
r108	960,88	1.167,71	206,83	21,53%
r109	1.194,73	1.432,72	237,99	19,92%
r110	1.118,59	1.328,80	210,21	18,79%
r111	1.096,72	1.205,48	108,76	9,92%
r112	982,14	1.357,02	374,88	38,17%
r201	1.252,37	1.325,65	73,28	5,85%
r202	1.191,70	1.259,15	67,44	5,66%
r203	939,50	1.057,75	118,25	12,59%
r204	825,52	880,20	54,68	6,62%
r205	994,42	1.146,14	151,72	15,26%
r206	906,14	1.080,75	174,61	19,27%
r207	890,61	984,58	93,97	10,55%
r208	726,75	851,32	124,57	17,14%
r209	909,16	1.037,47	128,31	14,11%
r210	939,34	1.105,35	166,01	17,67%
r211	885,71	1.057,48	171,77	19,39%
c101	**828,94**	**828,94**	**0,00**	**0,00%**
c102	828,94	866,21	37,27	4,50%
c103	828,06	1.021,55	193,49	23,37%
c104	824,78	1.098,51	273,73	33,19%
c105	828,94	834,79	5,85	0,71%
c106	828,94	872,93	43,99	5,31%
c107	**828,94**	**828,94**	**0,00**	**0,00%**
c108	828,94	1.001,28	172,34	20,79%
c109	828,94	1.030,68	201,74	24,34%
c201	**591,56**	**591,56**	**0,00**	**0,00%**
c202	591,56	706,63	115,07	19,45%
c203	591,17	840,08	248,91	42,11%
c204	590,60	992,84	402,24	68,11%
c205	588,88	591,56	2,68	0,45%
c206	588,49	625,50	37,01	6,29%
c207	588,29	623,20	34,91	5,93%
c208	588,32	620,18	31,86	5,41%
rc101	1.696,94	1.862,62	165,68	9,76%
rc102	1.554,75	1.638,16	83,41	5,36%
rc103	1.261,67	1.504,29	242,62	19,23%
rc104	1.135,48	1.395,82	260,34	22,93%
rc105	1.629,44	1.714,14	84,70	5,20%
rc106	1.424,73	1.694,37	269,64	18,93%
rc107	1.230,48	1.520,02	289,54	23,53%
rc108	1.139,82	1.653,24	513,42	45,04%

	Distancia			
	Mejor Heurístico	**ELOCONS**	**Diferencia**	**Ratio**
rc201	1.406,91	1.515,09	108,18	7,69%
rc202	**1.365,65**	**1.302,56**	**- 63,09**	**-4,62%**
rc203	1.049,62	1.164,90	115,28	10,98%
rc204	798,41	990,47	192,06	24,05%
rc205	1.297,19	1.382,80	85,61	6,60%
rc206	1.146,32	1.274,85	128,53	11,21%
rc207	1.061,14	1.213,50	152,36	14,36%
rc208	828,14	1.149,83	321,69	38,85%

Tal como se comprueba en las tablas anteriores el desempeño de ELOCONS es realmente bueno, sobre todo si se tiene en consideración que es un heurístico de construcción de rutas que no incluye ninguna rutina de postoptimización. De hecho ELOCONS obtiene un resultado que mejora en 4,72% el mejor resultado conocido en la resolución del problema *RC202*, e iguala los mejores resultados conocidos en los problemas *C201* y *C101*.
También se puede concluir que ELOCONS alcanza un alto grado de homogeneidad, toda vez que la desviación media a los mejores resultados conocidos de todos los problemas tipo es solamente del 17,2% y respecto a la media de solamente 2,99%.

Tabla 3 Tiempo medio de resolución de problemas

	Total	**Promedio**
C1	27,5248039	3,05831154
C2	18,8392003	2,35490004
R1	42,1248170	3,51040142
R2	35,9919990	3,27199991
RC1	25,8490317	3,23112896
RC2	23,4851064	2,93563830
TOTAL	**173,8149580**	**3,10383854**

Tal como se comprueba en la gráfica anterior el tiempo usado por el algoritmo para resolver los problemas tipo de Solomon es, en media, 3,1 minutos por problema, lo que demuestra con creces que la rapidez en la resolución de problemas requerida en las condiciones iniciales se ha alcanzado. Teniendo en cuenta que realiza todas las combinaciones de valores de parámetros para obtener el mejor resultado, su desempeño es muy rápido y permite en poco tiempo resolver rutas complejas.

CONCLUSIONES

Desde el año 1.967, la comunidad científica ha ido planteando distintos algoritmos para la resolución del problema de planificación de rutas con restricciones temporales en la entrega de la mercancía.
Aprovechando el incremento en las capacidades computacionales existentes, los algoritmos han evolucionado hacia formulaciones más evolucionadas y sofisticadas, y por ende, más complicadas,

tendiendo más hacía la obtención de los resultados óptimos que hacia la obtención de buenos resultados, combinando eficiencia en la resolución con eficacia en la solución obtenida.
Así, se han planteado la utilización de diferentes métodos de resolución, brillantes desde un punto de vista conceptual, pero limitativos en su aplicación en el entorno empresarial de la pyme y la micro empresa de distribución.
Como se ha referido a lo largo del presente trabajo, el problema al que nos enfrentamos incide fundamentalmente en pequeñas o muy pequeñas empresas, con recursos y capacidades bajas. Su problemática no supera, en general, los 100 clientes, por lo que sus necesidades están más enfocadas a la obtención de una buena solución para sus rutas de reparto en el más corto espacio de tiempo mediante una herramienta que no les exija inversiones fuertes en tecnología.
Se ha tratado en este trabajo de dar respuesta a estas dos condiciones, por lo que se ciñó la investigación a la obtención de un algoritmo de construcción rutas. Estas condiciones básicas, rapidez en la obtención de soluciones efectivas y bajo coste de infraestructura informática, se han alcanzado.
Desde un punto de vista de la investigación académica de meta heurísticos, ELOCONS puede proporcionar soluciones iniciales para su utilización en meta heurísticos más complejos de post optimización.
Una de las ventajas principales del algoritmo presentado es su forma de trabajar mediante listas de clientes. Esta característica le proporciona una gran versatilidad para trabajar en condiciones más cercanas a la realidad, como por ejemplo la utilización de distancias no simétricas o la utilización de tiempos diferentes según la dirección hacia la que se dirija el vehículo. También permite intercambiar matrices de distancias por matrices de tiempos y, por lo tanto, incluir diferentes matrices de tiempo dependiendo del horario de reparto, de la época del año, o de cualquier otro tipo de condiciones que el programador considere que pueda afectar al cálculo de las rutas. Consideramos estas características como muy importantes para su futura aplicabilidad en la resolución de problemas prácticos reales de distribución.

En un futuro y en aras a su aplicabilidad en la vida real, se podrían realizar los siguientes desarrollos:

Incluir un módulo de post optimización, que se utilizaría fundamentalmente en rutas que se repiten continuadamente en el tiempo.
Desarrollar una interfaz de usuario que permitiera al decisor modificar las rutas de forma local en algunos nodos, lo que proporcionaría una mayor versatilidad en la práctica.
Un módulo previo que permitiera al usuario la utilización de diferentes matrices de tiempos, según la conveniencia horaria o estacional.
La integración con sistemas GIS, para facilitar vía GPS las informaciones necesarias a los conductores, y a su vez, combinado con herramientas de geolocalización, para poder controlar el estado del reparto, prestar una información relevante en tiempo real al cliente e igualmente la modificación de una ruta por contingencias acaecidas sobre el recorrido de la ruta con posterioridad a la planificación, dotando a la herramienta y por ende al planificador, de una mayor capacidad de gestión y control.
La integración con sistemas de gestión y control de costes, proporcionando los datos que el planificador considere relevantes.

BIBLIOGRAFÍA

1. Bräysy O, Gendreau M. "Metaheuristics for the Vehicle Routing Problem with Time Windows" *(Internal Report STF42 A01025)*. Norway, Oslo: SINTEF Applied Mathematics, Department of Optimization. (2001).
2. Bräysy O, Gendreau M. "Route Construction and Local Search Algorithms for the Vehicle Routing Problem with Time Windows" *(Report Nº STF42 A01024)*. Norway, Oslo: SINTEF Applied Mathematics, Department of Optimization. (2001).
3. Clarke G, Wright J W. "Scheduling of Vehicles from a Depot to a number of Delivery Points". *Operations Research*. Vol.12 p.568-581. (1964).
4. Cordeau JF et al. *Vehicle Routing*. Barnhart C, Laporte G. Editores. *Transportation* (p. 367- 428). North Holland. (2007).ISBN: 9780444513465
5. Dullaert W, Bräysy O. "New insertion criteria for Solomon's (1987) insertion heuristic for the vehicle routing problem with time windows" *(Report Nº STF42 A03002)*. Norway, Oslo: SINTEF Applied Mathematics, Department of Optimization. (2003).
6. Dullaert W. "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, Belgium. (2000).
7. Faulín, J, García del Valle A. "Solving the capacitated vehicle routing problem using the ALGELECT electrostatic algorithm". *Journal of the Operational Research Society*. Vol.59 p.1685–1695. (2008).
8. Fischer M.L, Jaikumar R. "A Generalized Assignment Heuristic for Vehicle Routing". *Networks*. Vol. 11 p. 109-124. (1981).
9. García del Valle A. "Método de ATRACCIÓN para la optimización de rutas de transporte". *Estudios de Construcción y Transporte*. Vol.94 p. 77-95. (2002).
10. Gendreau M, Hertz A, Laporte G. et al. "A Generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows". *Operations Research*. Vol. 46 p.330-335. (1998).
11. Gendreau M, Laporte G, Potvin J.Y. "Local Search Algorithms for the Vehicle Routing Problem". Montreal, Canada: Université de Montreal, Centre de Recherche sur les Transport, *(CRT-963)*. (1994).
12. Golden B L, Raghavan S, Wasil E. *The Vehicle Routing Problem. Latest Advances and New Challenges*. USA: ORCS Interfaces, Ed. Springer. (2008). ISBN: 9780387777771
13. Golden B L, Assad A.A. *Vehicle Routing: Methods and Studies*. Amsterdam, Netherlands: North-Holland Publ. (1988).ISBN: 0444704078 (U.S.)
14. Hall R.W. "Vehicle Routing Software Survey". *OR/MS Today, INFORMS*, Vol. August, 2002.
15. Paessens, H. "The Savings Algorithm for the Vehicle Routing Problem". *European Journal Operations Research*. Vol.34 p. 336-344. (1988).
16. Solomon M, Desrosiers J. "Time Window Constrained Routing and Scheduling Problems". *Transportation Science*. Vol.22 (1) p. 1-13. (1988).
17. Solomon M. "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. (1986).
18. Solomon M. "Algorithms for the vehicle routing and scheduling problem with time window constraints". *Operations Research*. Vol.32 p.254-265. (1987).

19. Solomon, M. “Vehicle Routing and Scheduling With Time Window Constraints: Models and Algorithms” (*working paper, No. 83-42*). Boston, USA: Northeastern University, College of Business Administration. (1983).
20. Toth P, Vigo D. *The Vehicle Routing Problem*. Philadelphia, USA: SIAM Monographs. (2002). ISBN: 0898714982