

1. Resumen

En este proyecto se aborda un problema de un comercial que debe visitar a diferentes clientes un número determinado de veces, en un intervalo de tiempo preestablecido, por ejemplo, un mes.

Se trata de determinar los comercios que ha de visitar cada día y en qué orden debe visitarlos, de forma que cumpla unos requisitos determinados optimizando una medida de eficacia. Este problema se engloba dentro del problema del viajante de comercio, pero en su variante periódica: el problema del viajante de comercio periódico.

Para esta tipología de problemas se presentan dos planteamientos: por un lado, una variante en la que las visitas a los clientes cumplen un patrón de distancias entre visitas prefijado de antemano; y por otro lado, el caso de añadir un factor de regularidad entre visitas que penaliza una asignación de visitas a los clientes poco homogénea.

La forma de valorar una solución contempla: la distancia recorrida, la regularidad y el exceso de tiempo en cada jornada respecto a la jornada estándar.

Se han desarrollado tres algoritmos para solucionar el problema, que consisten en encontrar una solución inicial y mejorarla con un proceso de optimización local. Cada algoritmo proporciona una solución inicial diferente, ya que se aplican diferentes criterios a la hora de asignar los clientes a los días de visita.

Se ha realizado una experiencia computacional en la que se han encontrado buenas soluciones para el problema, así como para algunos ejemplares de la literatura.

Además del problema de asignación regular de clientes a rutas, se propone aplicar el concepto de regularidad a las ventas de los días de un mes, así como proponer una herramienta para gestionar el incremento de ventas que se puede producir al incrementar el número de visitas a los clientes.





2. Sumario

1. RESUMEN	1
2. SUMARIO	3
3. INTRODUCCIÓN	5
3.1. Objetivos del proyecto	6
3.2. Alcance del proyecto	7
4. ESTADO DEL ARTE	8
4.1. El problema TSP (<i>Traveling Salesman Problem</i>).....	8
4.2. Algoritmos de resolución del TSP	8
4.2.1. Algoritmos exactos.....	9
4.2.2. Heurísticas	9
4.2.3. Algoritmo de <i>Clarke and Wright</i>	10
4.3. Caso particular del problema TSP, el PTSP (<i>Periodic Traveling Salesman Problem</i>).....	12
4.4. Algoritmos de resolución del PTSP	12
4.5. Estado del concepto regularidad.....	13
5. DESCRIPCIÓN DEL PROBLEMA A RESOLVER	14
5.1. Modelización del problema.....	14
5.2. Restricciones del problema a resolver	15
5.3. Función objetivo	15
5.4. Representación de la solución	17
5.5. Cálculo de soluciones iniciales.....	18
5.5.1. Primer procedimiento	18
5.5.2. Segundo procedimiento	24
5.5.3. Tercer procedimiento	27
6. PROCESO DE OPTIMIZACIÓN	29
6.1. Proceso de optimización local.....	29
6.1.1. Reordenar clientes en un mismo día.....	29
6.1.2. Intercambio de clientes entre días	30
6.2. Proceso de perturbación	31
6.2.1. Move_opeartor.....	31
6.2.2. Cross_exchange.....	33



6.3. Algoritmo de optimización.....	34
6.4. Elementos aleatorios	35
7. EXPERIENCIA COMPUTACIONAL	37
7.1. Datos iniciales.....	37
7.2. Datos del caso real	37
7.3. Tiempo de cálculo.....	37
7.4. Resultados de la experiencia computacional	38
8. CASO DE APLICACIÓN	41
8.1. Resultados.....	43
8.1.1. Primer algoritmo. Soluciones y valores obtenidos	43
8.1.2. Segundo algoritmo. Soluciones y valores obtenidos.....	49
8.1.3. Tercer algoritmo. Soluciones y valores obtenidos.....	52
8.1.4. Resumen de resultados.....	55
9. NUEVAS POSIBILIDADES DE APLICACIÓN	62
9.1. Regularidad de ventas.....	62
9.2. Incremento de visitas	65
CONCLUSIONES	71
COSTES DEL ESTUDIO EFECTUADO	73
IMPACTO AMBIENTAL DEL ESTUDIO	75
BIBLIOGRAFÍA	76



3. Introducción

En el proyecto se aborda el caso de un comerciante que trabaja para una empresa de distribución de *dvd's* comerciables en grandes almacenes y centros especializados. Se planifican las rutas que debe realizar el comerciante durante el mes a los diferentes clientes.

La finalidad concreta consiste en determinar qué clientes debe visitar cada día y en qué orden debe hacerlo. A modo de ilustración se adjuntan dos posibles rutas (de un día de duración cada una) que debe realizar un comercial que reside en Valencia. (Fig. 1)

Día 16:

Itinerario: Valencia > Zaragoza 1 > Villanueva de Gállego > Valencia

Día 17:

Itinerario: Valencia > Castellón > Valencia 3 > Valencia



Fig.1 Ejemplo de viajes realizados por el comercial

Se resuelve el problema del viajante de comercio periódico o PTSP (*Periodic Traveling Salesman Problem*). Este es un problema común, que aparece en diferentes ámbitos, tales como: una actividad industrial en la que se deben buscar los suministros a diferentes proveedores, una empresa de recogida de residuos de diferentes localizaciones o el caso de distribución de *dvd's* aquí tratado.



3.1. Objetivos del proyecto

Los objetivos englobados dentro del presente proyecto son:

- Resolver un problema de asignación de visitas de vendedores a puntos de venta, y de optimización de las rutas a realizar.
- Analizar el impacto que tiene la introducción del concepto de regularidad en los problemas PTSP.
- Adaptar el problema de regularidad considerando la cantidad de ventas; es decir, en lugar de considerar la regularidad de los días entre las visitas del comercial, considerar la regularidad de las ventas (o, mejor, previsión de ventas) entre dichas visitas.
- Comprobar la sensibilidad de la solución al incremento de visitas a los clientes: calcular cómo afecta, al resultado final, el hecho de incrementar una visita a un cliente, considerando el beneficio económico que se obtiene con ello.



3.2. Alcance del proyecto

Este proyecto de asignación de visitas a clientes, cuenta con las siguientes hipótesis de partida:

- Los datos son conocidos (deterministas).
- La duración y las distancias de los desplazamientos, los tiempos de visita y el número de visitas, son datos.
- La jornada laboral se compone de tiempo de trabajo efectivo, ya que no se tienen en cuenta los descansos.

Se ha diseñado un procedimiento para determinar las visitas que debe realizar un comercial cada día y en qué orden se han de efectuar. El proyecto no considera conceptos de capacidad o de demanda variable, ya que se supone que el comercial no distribuye el producto (únicamente visita a las tiendas para realizar pedidos).

Los valores de las ventas diarias de cada cliente y los incrementos de ventas que suponen las visitas extra han sido estimados de forma realista. Dichos datos son necesarios para satisfacer el tercer y cuarto objetivo propuestos en el apartado 3.1.



4. Estado del arte

4.1. El problema TSP (*Traveling Salesman Problem*)

El problema del viajante de comercio está clasificado como *NP-hard*, (*nondeterministic polynomial-time hard*), es decir, la dificultad aumenta de forma exponencial con el tamaño del problema a resolver.

Hace dos siglos, los matemáticos Sir William Rowan Hamilton y Thomas Kirkman [Rowan y Kirkman, 1736], trataron problemas relacionados con el TSP. La forma general del TSP fue formulada en las universidades de Viena y Harvard. Más adelante, el problema se retomó de la mano de Hassler Whitney y Merrill M. Flood en la universidad de Princeton. En el artículo *On the history of combinatorial optimization* [Schrijver, 2005] se encuentra un resumen del desarrollo en el estudio del TSP.

El problema del viajante de comercio consiste en encontrar el camino mínimo con el que llegar a todos los vértices de un grafo, saliendo de uno de dichos vértices y retornando a dicho vértice. El problema se puede representar como un grafo completo orientado. Los nodos representan las ciudades y los valores asociados a las aristas, las distancias entre ellas. Estos valores nunca toman valores negativos. La finalidad del problema es que desde un nodo seleccionado como el de partida, se debe visitar el resto una sola vez y regresar al vértice de partida, de forma que una función de coste sea optimizada.

4.2. Algoritmos de resolución del TSP

Las formas tradicionales de tratar este problema son: encontrar una solución exacta, cuando las dimensiones del problema son relativamente pequeñas; utilizar algoritmos heurísticos, que proporcionan buenas soluciones, aunque no se puede garantizar siempre que sean las óptimas; o bien, dividir el problema en subproblemas y reducir así la dificultad de éstos.

Un desarrollo más elaborado, es la aplicación de algunas meta heurísticas como la de colonia de hormigas o de cambio genético o pseudo-genético.



4.2.1. Algoritmos exactos

La forma más directa de encontrar la solución exacta sería formular todas las permutaciones, siendo el número de soluciones posibles $n!$. Esta posibilidad se convierte en impracticable a medida que aumenta el número de variables.

Otras opciones son: utilizar un proceso de branch-and-bound, válido con ejemplos que contengan entre 40 y 60 ciudades; implementar algoritmos basados en programación lineal, con un buen funcionamiento hasta las 200 ciudades; o bien, combinar ambas opciones. [Applegate, 2006]

4.2.2. Heurísticas

Se han desarrollado varios algoritmos heurísticos, que proporcionan rápidamente buenas soluciones. Actualmente se pueden encontrar soluciones para problemas extremadamente largos en un tiempo razonable, con una desviación del óptimo de un 2 ó 3%. [Applegate, 2006]

Estos procedimientos se suelen dividir en heurísticas constructivas y procesos iterativos.

- I) **Heurísticas constructivas:** Las heurísticas constructivas construyen la solución paso a paso, de manera que se añaden elementos a la solución a medida que tienen en cuenta los diferentes criterios.

Por ejemplo, el algoritmo del vecino más cercano (NN, nearest neighbour), que consiste en considerar la ruta desde la ciudad elegida como central y escoge la ciudad más cercana no visitada todavía. Dentro de esta categoría también se encuentra el algoritmo de los ahorros, o de Clarke&Wright (el cual es utilizado en este trabajo y presentado en el apartado 4.2.3).

- II) **Procesos iterativos:** Los métodos que se basan en procedimientos iterativos parten de una solución, modificándola y obteniendo soluciones parecidas que pueden ser mejores.

Por ejemplo, en el algoritmo de Lin-Kernighan, o 2-opt, una vez encontrada una solución, elimina dos aristas y reordena las ciudades creando una posible una solución. Como ampliación a este algoritmo, existe el algoritmo k-opt, dónde se eliminan k aristas o el V-opt dónde el número de aristas a eliminar es variable.



4.2.3. Algoritmo de *Clarke and Wright*

Los procedimientos propuestos en este PFC consisten, básicamente, en asignar clientes a días y, para cada día, generar una ruta de visitas. Una vez llegados a este punto, el programa se encuentra delante del problema del viajante de comercio (TSP).

Para resolver los TSP's, en este proyecto se ha optado por el algoritmo de ahorros de Clarke and Wright, en su versión en serie, que se introduce a continuación.

El algoritmo se basa en el concepto de ahorro de una cadena de arcos respecto a la residencia del comercial(X).

En la figura 2 se puede apreciar la representación del concepto ahorro entre dos clientes. Este concepto se entiende como la distancia o el tiempo ganado al evitar tener que volver a la residencia del comercial para llegar de un cliente a otro.

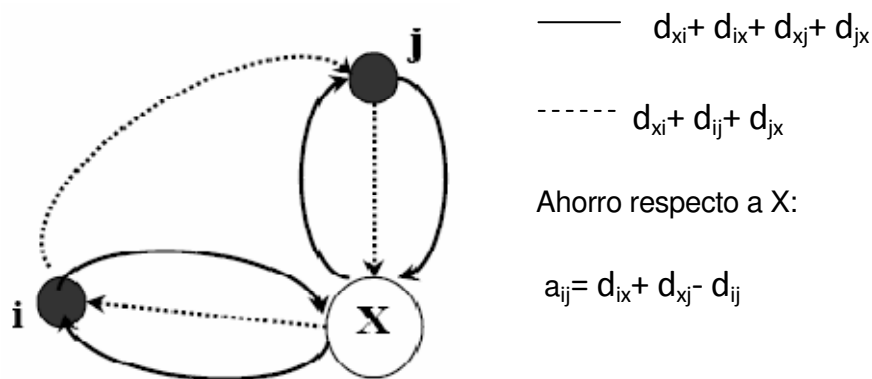


Fig. 2 Representación del concepto de ahorro en un grafo

A partir de la matriz de costes (tiempos de viaje), se calculan los ahorros para el conjunto de arcos que van de i a j respecto al nodo central, X (donde i y j son las ciudades a visitar).

A continuación, se crea una lista de parejas de clientes ordenada de forma creciente según los ahorros. Se selecciona la primera pareja de la lista (i, j) y se fija la visita de i a j, a continuación se busca el ahorro mayor entre las combinaciones posibles de los clientes extremos (i y j) y los no seleccionados. En cada asignación, se busca entre las parejas formadas por el cliente de la primera posición (i) con el resto de clientes y las formadas por el cliente de la última posición (j) con el resto.



A modo de ejemplo se considera el siguiente caso:

Se necesita crear una solución diaria con los clientes 2, 6, 24 y 27.

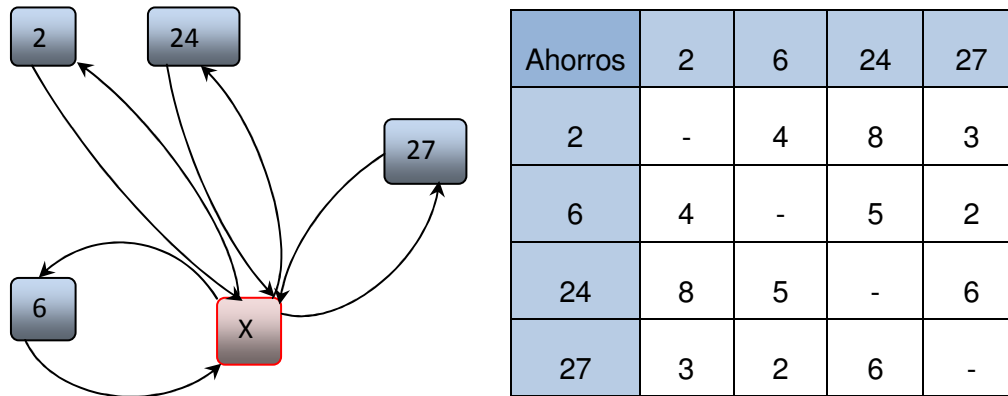


Fig. 3 Ejemplo de aplicación del algoritmo de Clarke and Wright

Aplicando el proceso se obtiene que el mayor ahorro que existe entre cualquier pareja de estos cuatro clientes es el (2,24), con un valor de 8 unidades, por lo tanto la solución parcial queda:

Día <i>i</i>	2	24	
--------------	---	----	--

A continuación se pasa a buscar cual es el mayor ahorro por la izquierda con las siguientes posibilidades: (2,6) ó (2,27); y cuál es el mejor ahorro por la derecha, cuyas posibilidades son: (24,6) ó (24,27). Se determina que el mayor ahorro es el (24,27) y como viene de un ahorro por la derecha, la solución parcial queda:

Día <i>i</i>	2	24	27	
--------------	---	----	----	--

Por último se repite el proceso, quedando pues evaluar los ahorros (2,6) y (27,6). Al ser mejor el primero, obtenemos el orden de visitas de ese día. Finalmente el orden de visitas queda de la siguiente forma:

Día <i>i</i>	6	2	24	27	
--------------	---	---	----	----	--

Colocando así todas las visitas programadas para ese día.



4.3. Caso particular del problema TSP, el PTSP (*Periodic Traveling Salesman Problem*).

A menudo se presenta una versión ampliada del problema, que consiste en que un cliente debe ser visitado diferentes veces durante un horizonte de planificación (por ejemplo, un mes).

Este problema se conoce como el problema del viajante de comercio periódico (PTSP), y se puede encontrar en [Cordeau et al. ,1997].

Resolver el PTSP consiste, por tanto, en asignar los clientes a los días (a varios días diferentes cuando un cliente debe ser visitado más de una vez) y en decidir las rutas de visita de cada día.

Hasta el momento, los algoritmos existentes en la literatura para resolver el problema utilizan patrones. Aplicar el concepto de patrón es definir los intervalos entre visitas a un cliente, de manera que la distancia entre visitas es constante. Por ejemplo, un cliente con dos visitas en cuatro días tiene asignado un patrón 2-2, y puede ser asignado el primer y tercer día o el segundo y cuarto día.

4.4. Algoritmos de resolución del PTSP

En la literatura se pueden encontrar algunos procedimientos heurísticos para solucionar el PTSP; además de técnicas de optimización local potencialmente aplicables a este tipo de problemas.

Las primeras heurísticas para el PTSP se describen en [Christofides y Beasley,1984] y en [Paletta, 1992]. Christofides y Beasley resuelve el PTSP desarrollando el siguiente procedimiento heurístico: en primer lugar, los clientes se ordenan según un coste de inserción en los diferentes días; se elige el de menor coste y se inserta en la solución recalculando el resto, de manera que, el cliente que se inserta en cada paso, minimiza el coste. El procedimiento para resolver el problema TSP que se presenta cada día es un 2-opt intercambio.

Paletta desarrolla dos heurística para resolver el PTSP. Ambas asignan primero una combinación factible de días de visita para un cliente no colocado, en función del coste de la colocación. Cuando un cliente es asignado, un algoritmo de programación dinámica se utiliza para resolver el TSP en cada día. Los costes asociados al resto de clientes se recalculan en función de la distancia de los



clientes sin asignar a cada día de viaje. Este procedimiento se repite hasta que todos los clientes son asignados.

Estas técnicas han sido superadas por enfoques meta heurísticos. Chao [Chao et al, 1995b] parte de una solución inicial e intercambia combinaciones de días de visita utilizando la optimización local record-to-record de Dueck, [Dueck, 1993]. Dicho autor varía parámetros de intercambio para buscar otras soluciones, siendo éstas cercanas a la mejor obtenida, pero permitiendo un margen de error. Cordeau, [Cordeau et al.,1997], desarrolla un método de búsqueda tabú basado en el operador GENI explicado en [Gendreau et al., 1992]. Paletta, [Paletta, 2002], propone un procedimiento mejorado, en el que, durante la generación de una ruta, se produce optimización local.

Hemmelmayr, [Hemmelmayr et al, 2007], desarrolla otro enfoque meta heurístico para la solución del PTSP que se basa en búsqueda en vecindarios variables (VNS). VNS es una búsqueda local basada en meta heurísticas propuesta por primera vez en [Mladenovic, 1995], [Mladenovic and Hansen, 1997] y [Hansen y Mladenovic, 1997].

4.5. Estado del concepto regularidad

Hasta el momento, el problema PTSP no contemplaba la posibilidad de introducir el concepto de regularidad, como la variación del intervalo entre dos visitas fijadas por un patrón y el intervalo obtenido. Debido a que se trabajaba con asignación de patrones de visitas a un cliente, dicha regularidad estaba garantizada.

Este concepto se trata en el artículo "*Response time variability*" [Corominas et al.2007], como la variabilidad del tiempo de respuesta (RTV), y se define como la diferencia entre el tiempo entre dos visitas consecutivas y el tiempo ideal, elevada al cuadrado.

Lo que se plantea en este proyecto es introducir el término de regularidad en la función de evaluación, de manera que se pueda mejorar la calidad de la solución en el aspecto de la distancia recorrida.

Además, parece adecuado proponer una parametrización de la función objetivo, de manera que sea el usuario final quien pueda dar mayor o menor importancia al término regularidad o a la distancia recorrida, dependiendo de sus intereses.



5. Descripción del problema a resolver

En este apartado se caracteriza la tipología de problemas PTSP abordada en los apartados posteriores.

5.1. Modelización del problema

Sea un vendedor que ha de visitar N clientes $(i, j=1, \dots, N)$ en D días. Se conoce una matriz T , con el tiempo de viaje entre cualquier pareja de clientes, y entre el domicilio del vendedor y cada uno de los clientes: t_{ij} es el tiempo de viaje entre el cliente i y el cliente j , y t_{0j} es el tiempo de viaje entre el lugar de residencia del vendedor y el cliente j .

También se conoce la matriz K , que manteniendo el símil con la matriz T , proporciona la distancia existente en kilómetros entre cualquier pareja de clientes o entre un cliente y la residencia del vendedor.

Además se conoce el tiempo de visita del vendedor a cada uno de los N clientes (tv_i es el tiempo que el vendedor está en el cliente i).

Existen clientes que han de ser visitados más de una vez, de forma que v_i es el número de visitas que hay que realizar al cliente i en un mes.

La finalidad del proyecto consiste en asignar los N clientes a visitar, considerando que hay clientes que se visitan varias veces, a los D días laborables, de forma que se minimice una función objetivo parametrizada por el usuario.

Las rutas a obtener deben tener una duración no mayor a 7 horas ya que, aunque la jornada diaria es de 8 horas de duración, se desea que el comercial dedique una hora cada día a realizar gestiones.



5.2. Restricciones del problema a resolver

Las restricciones a aplicar son:

1. Asegurar el número de visitas a cada cliente.
2. No pasar de 7 horas en la realización de las visitas, para dedicar una hora a gestionar la cartera de clientes.
3. No pasar de 8 horas en el total de la jornada diaria.

En el caso real se clasifican estas restricciones en restricciones *soft* y restricciones *hard*. Las restricciones *soft*, a diferencia de las restricciones *hard*, permiten soluciones que no cumplen un requisito, pero son penalizadas. Mientras que la primera restricción es *hard*, la segunda y la tercera son *soft*.

La forma de trabajar con las restricciones *soft*, es penalizar las soluciones que contengan jornadas que excedan las 7 horas, en dos niveles, de forma que las jornadas que tienen una duración entre 7 y 8 horas penalizan con un valor de orden inferior a si tienen una duración superior a 8 horas.

El motivo de estas penalizaciones es que se supone que el comercial debe dedicar 7 horas a viajes y visitas y una hora a gestiones. Si de esa hora dedica un tiempo a acabar un viaje, se considera permisible (si no fuera permisible sería una restricción *hard*), pero se penaliza. En cambio, no se considera permisible a priori que debe realizar una jornada superior a 8 horas.

La forma de valorar estas soluciones es introducir una penalización en la función objetivo.

5.3. Función objetivo

La función objetivo diseñada cuenta con cuatro elementos para valorar la calidad de las soluciones obtenidas.

- I) La **cantidad de kilómetros** recorridos por el viajante durante todo el mes.



- II) La **regularidad** entre visitas obtenidas, para aquellos clientes que se visitan varias veces al mes.

Se entiende por regularidad entre visitas a un mismo cliente, la suma de los cuadrados de las diferencias entre los intervalos de dos visitas y el intervalo ideal entre éstas. Dicho intervalo deseable se obtiene dividiendo el número total de días que tiene el periodo entre el número de visitas que se han de realizar.

Por ejemplo, sea un cliente que hay que visitar 3 veces en 22 días y se ha decidido visitarlo los días 1, 8 y 18; entre dos visitas consecutivas hay 7, 10 y 5 días, que se obtienen de las siguientes diferencias (8-1), (18-8) y (1+22-18). Así el valor de la regularidad obtenida es:

$$\left(\frac{22}{3} - 7\right)^2 + \left(\frac{22}{3} - 10\right)^2 + \left(\frac{22}{3} - 5\right)^2 = \frac{38}{3}$$

Se comprueba que un menor valor de este parámetro de regularidad indica más regularidad, por este motivo este valor debe ser minimizado, ya que realmente, indica la falta de regularidad.

- III) el **exceso de tiempo de ruta** cuando las jornadas superan las 7 horas y no superan las 8 horas
- IV) el **exceso de tiempo de ruta** cuando las jornadas superan las 8 horas.

Se penaliza la cantidad de tiempo en exceso que tiene cada jornada, de manera que a más horas, peor valor de la función objetivo. Además estos dos términos se elevan al cuadrado y a la cuarta, respetivamente, para evitar que, por ejemplo, en una solución de diez días de duración, penalice de la misma manera si cada jornada excede una hora o si una única jornada tiene un exceso de diez horas.



La función objetivo propuesta, a minimizar, es:

$$F. Obj. = \alpha \cdot (Cantidad_km) + (1-\alpha) \cdot \gamma_1 \cdot Regularidad + \gamma_2 \cdot Ruta_7^2 + \gamma_3 \cdot Ruta_8^4 \quad (Ec. 1)$$

Donde *Cantidad_km* es la cantidad total de kilómetros, *Regularidad* es el valor referido en el segundo lugar de la función objetivo, *Ruta₇* incluye el exceso de tiempo diario por encima de 7 horas y *Ruta₈* el exceso de tiempo diario de más de 8 horas.

5.4. Representación de la solución

Una solución se representaría mediante una matriz. El número de filas es igual al número de días que comprende horizonte a planificar y las columnas, a las visitas a realizar cada día.

Un ejemplo posible de solución con 44 clientes y 22 días, dónde el cliente 9, 10, 12 se han de visitar dos veces sería la siguiente:

	Visita 1	Visita 2	Visita 3	Visita 4	Visita 5	Visita 6	...
Día 1	9	44					
Día 2	34	36	37	10			
Día 3	12	14					
Día 4	5						
Día 5	40	35					
Día 6	1						
Día 7	2	6	11				
Día 8	3	31					
Día 9	4	42					
Día 10	15	29					
Día 11	16						
Día 12	43	7	9				
Día 13	10	39	8				
Día 14	12						
Día 15	17						
Día 16	21	20					
Día 17	23	19					
Día 18	18	24	22				
Día 19	25	13					
Día 20	26	30					
Día 21	28	27					
Día 22	38	32	33	41			



De forma que la primera jornada está compuesta por los siguientes desplazamientos: de casa del viajante al cliente número 9, del cliente número 9 al 44 y del cliente 44 a casa del viajante.

Y así sucesivamente para el resto de días.

5.5. Cálculo de soluciones iniciales

Como se ha introducido anteriormente, los procedimientos de resolución se basan en obtener una solución inicial y mejorarla con un procedimiento de optimización local.

La solución inicial se obtiene mediante tres procedimientos heurísticos. El primero se basa en utilizar patrones de visitas, en el tercero la asignación de un cliente es posible en cualquier día, y el segundo, que es un procedimiento intermedio entre los dos anteriores, permite variar el día que se asignan los clientes con el patrón en más menos un día.

5.5.1. Primer procedimiento

Como se ha comentado, el primer algoritmo trabaja con patrones de visitas (que han de ser determinados). La forma de calcular los patrones se describe a continuación:

Se tenga un número total de días (D) igual a 20, y un número de visitas o frecuencia (v) igual a 4; en este caso el tiempo ideal entre dos visitas consecutivas es de 5. En otro caso con la misma frecuencia pero con un número de días que no sea múltiplo de 4, por ejemplo 19, se crea un conflicto a resolver: ¿cómo repartir con números enteros el día que falta?

Para solucionar este problema se ha optado por utilizar el vector de descomposición, propuesto en [Corominas *et al*, 2007]. Para cada cliente se crea un vector de v_i componentes. El vector se puede definir como $\lambda = (\lambda_1, \dots, \lambda_{v_i})$, donde λ_i son v_i enteros positivos tal que $\lambda_1 \leq \dots \leq \lambda_{v_i}$. Los componentes del vector λ son los días transcurridos entre las v_i visitas al cliente i . Así pues, estos componentes se pueden calcular como " $D \bmod v_i$ " y " $v_i - D \bmod v_i$ " componentes



de λ , siendo mod el operador división entera. Las componentes son iguales a “Parte entera de $\left(\frac{D}{v_i}\right)$ ” y “Parte entera de $\left(\frac{D}{v_i}\right)+1$ ” respectivamente.

Siguiendo con el ejemplo citado, $D=19$ y $v_i=4$ el vector de descomposición es $\lambda = (4,5,5,5)$, que proporciona el mejor valor posible de regularidad.

El primer procedimiento para el cálculo de una solución inicial está formado por los siguientes pasos:

Paso 1: Buscar la frecuencia de visita mayor (es decir, el número máximo de visitas a un mismo cliente).

Paso 2: Para la frecuencia de visita mayor, seleccionar el cliente más lejano a la residencia del vendedor y asignarlo a los días que correspondan según el patrón de ese cliente.

La forma de asignar es la siguiente. Una vez que se ha seleccionado el cliente que se quiere asignar, éste se coloca en las diferentes opciones, variando la colocación de la primera visita, y respetando las distancias indicadas por el patrón. Cada vez que se decide colocar la visita al cliente en un día, se resuelve el problema del viajante de comercio, reordenando adecuadamente las visitas. El algoritmo utilizado para resolver este subproblema es el algoritmo de los ahorros o de **Clarke & Wright**. Finalmente se selecciona la asignación que presente un menor valor de la función objetivo.

Paso 3: Una vez colocado el cliente más lejano se coloca el segundo cliente más lejano hasta que no queden cliente con esa frecuencia

Paso 4: Volver al Paso 1 si quedan clientes por asignar. Es decir, repetir el proceso de asignación para los clientes de frecuencia inferior hasta los de frecuencia unidad.

Cabe destacar que, en este algoritmo inicial, si en el momento de asignar una visita a un día este día no tiene ninguna otra visita asignada, la cantidad de kilómetros de este día no se suma en la función objetivo. La finalidad de esta excepción es facilitar que haya un mayor reparto de visitas entre todos los días disponibles.



A continuación se explica un ejemplo en detalle para facilitar la comprensión del procedimiento presentado.

Sea un período de 15 días en los que se tiene que visitar a 25 clientes varias veces.

La información referente al número de visitas de cada cliente es:

Cliente	Número de visitas
1	1
2	4
3	1
4	1
5	4
6	3
7	2
8	1
9	2
10	2
11	1
12	2
13	1
14	1
15	4
16	1
17	1
18	5
19	1
20	1
21	6
22	1
23	1
24	2
25	1

En primer lugar se busca la frecuencia más alta, en este caso es 6. De los clientes con frecuencia 6 se busca el más alejado (cliente 21) y se coloca siguiendo el vector de descomposición.

El vector de descomposición resultante es (2, 2, 2, 3, 3, 3).



Las tres opciones posibles son:

Día1	21	
Día2		
Día3	21	
Día4		
Día5	21	
Día6		
Día7	21	
Día8		
Día9		
Día10	21	
Día11		
Día12		
Día13	21	
Día14		
Día15		
F.Obj	0	

Día1		
Día2	21	
Día3		
Día4	21	
Día5		
Día6	21	
Día7		
Día8	21	
Día9		
Día10		
Día11	21	
Día12		
Día13		
Día14	21	
Día15		
F.Obj	0	

Día1		
Día2		
Día3	21	
Día4		
Día5	21	
Día6		
Día7	21	
Día8		
Día9	21	
Día10		
Día11		
Día12	21	
Día13		
Día14		
Día15	21	
F.Obj	0	

Teniendo todas el mismo valor de la función objetivo, se selecciona la primera.

La solución queda:

Día1	21	
Día2		
Día3	21	
Día4		
Día5	21	
Día6		
Día7	21	
Día8		
Día9		
Día10	21	
Día11		
Día12		
Día13	21	
Día14		
Día15		

Se busca ahora el siguiente cliente con frecuencia 6.



Al no haber ninguno, se busca el de frecuencia 5 más alejado (cliente 18) y se repite el procedimiento anterior, siendo las opciones:

Día1	18	21		
Día2				
Día3	21			
Día4	18			
Día5	21			
Día6				
Día7	18	21		
Día8				
Día9				
Día10	18	21		
Día11				
Día12				
Día13	18	21		
Día14				
Día15				
F.Obj	634			

Día1	21			
Día2	18			
Día3	21			
Día4				
Día5	18	21		
Día6				
Día7	21			
Día8	18			
Día9				
Día10	21			
Día11	18			
Día12				
Día13	21			
Día14	18			
Día15				
F.Obj	25			

Día1	21			
Día2				
Día3	18	21		
Día4				
Día5	21			
Día6	18			
Día7	21			
Día8				
Día9	18			
Día10	21			
Día11				
Día12	18			
Día13	21			
Día14				
Día15	18			
F.Obj	25			

Se selecciona la segunda opción, que es la que tiene una función objetivo menor.

Este procedimiento se va repitiendo para cada cliente. Por ejemplo, al llegar al momento de colocar al cliente 12, se tienen las siguientes opciones:

Día1	5	12	21		
Día2	18	6	9		
Día3	15	21			
Día4	5	2	10		
Día5	18	21			
Día6	15				
Día7	2	6	21		
Día8	5	12	18		
Día9	9				
Día10	15	21			
Día11	18	2	10		
Día12	5	6			
Día13	21				
Día14	15	18			
Día15	2				
F.Obj	12.702				

Día1	21	5			
Día2	6	12	18	9	
Día3	15	21			
Día4	5	2	10		
Día5	18	21			
Día6	15				
Día7	2	6	21		
Día8	18	5			
Día9	9	12			
Día10	15	21			
Día11	18	2	10		
Día12	5	6			
Día13	21				
Día14	15	18			
Día15	2				
F.Obj	12.213				

Día1	21	5			
Día2	18	6	9		
Día3	12	15	21		
Día4	5	2	10		
Día5	18	21			
Día6	15				
Día7	2	6	21		
Día8	18	5			
Día9	9				
Día10	12	15	21		
Día11	18	2	10		
Día12	5	6			
Día13	21				
Día14	15	18			
Día15	2				
F.Obj	12.634				



Día1	21	5			
Día2	18	6	9		
Día3	15	21			
Día4	12	5	2	10	
Día5	18	21			
Día6	15				
Día7	2	6	21		
Día8	18	5			
Día9	9				
Día10	15	21			
Día11	2	12	18	10	
Día12	5	6			
Día13	21				
Día14	15	18			
Día15	2				
F.Obj	12.823				

Día1	21	5			
Día2	18	6	9		
Día3	15	21			
Día4	5	2	10		
Día5	12	18	21		
Día6	15				
Día7	2	6	21		
Día8	18	5			
Día9	9				
Día10	15	21			
Día11	18	2	10		
Día12	12	5	6		
Día13	21				
Día14	15	18			
Día15	2				
F.Obj	12.744				

Día1	21	5			
Día2	18	6	9		
Día3	15	21			
Día4	5	2	10		
Día5	18	21			
Día6	12	15			
Día7	2	6	21		
Día8	18	5			
Día9	9				
Día10	15	21			
Día11	18	2	10		
Día12	5	6			
Día13	21	12			
Día14	15	18			
Día15	2				
F.Obj	12.063				

Día1	21	5			
Día2	18	6	9		
Día3	15	21			
Día4	5	2	10		
Día5	18	21			
Día6	15				
Día7	21	12	2	6	
Día8	18	5			
Día9	9				
Día10	15	21			
Día11	18	2	10		
Día12	5	6			
Día13	21				
Día14	12	15	18		
Día15	2				
F.Obj	12.799				

Día1	21	5			
Día2	18	6	9		
Día3	15	21			
Día4	5	2	10		
Día5	18	21			
Día6	15				
Día7	2	6	21		
Día8	5	12	18		
Día9	9				
Día10	15	21			
Día11	18	2	10		
Día12	5	6			
Día13	21				
Día14	15	18			
Día15	2	12			
F.Obj	12.215				



Finalmente se llega a la solución siguiente:

Día1	21	5							
Día2	18	6	9	7	8				
Día3	15	21	11						
Día4	5	2	10						
Día5	21	23	18						
Día6	12	17	15						
Día7	6	2	24	25	19	21	16	13	
Día8	18	20	5						
Día9	9	7							
Día10	15	21							
Día11	18	2	10						
Día12	6	5	1						
Día13	21	12	22						
Día14	18	24	15						
Día15	4	2	3						
F.Obj	21.418								

Con una inspección rápida se aprecia que hay un día con una gran cantidad de visitas. Este día excede las horas permitidas. El problema se soluciona al aplicar el proceso de optimización local. La función objetivo utilizada en este ejemplo no contempla el exceso de trabajo diario.

5.5.2. Segundo procedimiento

El segundo procedimiento difiere con el primero en la paso número dos.

La diferencia radica en que la asignación del cliente no se evalúa una vez que se han colocado todas las visitas del cliente; si un cliente tiene frecuencia 4, no se evalúan las 4 visitas a la vez, sino que se permite variar la regularidad en el momento de asignar cada nueva visita.



El segundo paso queda de la siguiente manera:

Paso 2: Para la frecuencia de visita mayor, seleccionar el cliente más lejano a la residencia del vendedor y asignar las visitas.

La forma de asignar es la siguiente. Una vez que se ha seleccionado el cliente a asignar, se coloca la primera visita. A continuación se coloca la segunda en tres opciones diferentes: la primera a la distancia que marca el patrón menos un día, la segunda a la distancia del patrón y la tercera a la distancia del patrón más un día. Se selecciona la mejor de las tres según el valor de la función objetivo y se pasa a asignar la siguiente visita. Cuando se completan las visitas se obtiene un valor de la función objetivo para ese cliente. Este valor se compara con el que adquiere en las diferentes posibilidades de colocar la primera visita, seleccionando finalmente la de mejor valor.

Los siguientes pasos son iguales a los del primer proceso.

En el siguiente ejemplo se explica el proceso de asignación de las visitas de un cliente. Partiendo de una solución intermedia tal como esta:

Día1	5	21		
Día2	18	6	9	
Día3	15	21		
Día4	5	2	10	
Día5	18	21		
Día6	15			
Día7	2	6	21	
Día8	5	18		
Día9	9			
Día10	15	21		
Día11	18	2	10	
Día12	5	6		
Día13	21			
Día14	15	18		
Día15	2			

Imagínese que se ha de asignar el cliente número 12 que tiene frecuencia 2.



El proceso de asignación es el siguiente:

Se coloca el cliente 12 en el primer día (primera posibilidad).

Día1	5	12	21	
Día2	18	6	9	
Día3	15	21		
Día4	5	2	10	
Día5	18	21		
Día6	15			
Día7	2	6	21	
Día8	5	18		
Día9	9			
Día10	15	21		
Día11	18	2	10	
Día12	5	6		
Día13	21			
Día14	15	18		
Día15	2			

Y a continuación se coloca la siguiente visita a este cliente en las tres posiciones comentadas.

Día1	5	12	21	
Día2	18	6	9	
Día3	15	21		
Día4	5	2	10	
Día5	18	21		
Día6	15			
Día7	2	6	12	21
Día8	5	18		
Día9	9			
Día10	15	21		
Día11	18	2	10	
Día12	5	6		
Día13	21			
Día14	15	18		
Día15	2			
F.Obj	6.346			

Día1	5	12	21	
Día2	18	6	9	
Día3	15	21		
Día4	5	2	10	
Día5	18	21		
Día6	15			
Día7	2	6	21	
Día8	5	12	18	
Día9	9			
Día10	15	21		
Día11	18	2	10	
Día12	5	6		
Día13	21			
Día14	15	18		
Día15	2			
F.Obj	6.290			

Día1	5	12	21	
Día2	18	6	9	
Día3	15	21		
Día4	5	2	10	
Día5	18	21		
Día6	15			
Día7	2	6	21	
Día8	5	18		
Día9	9	12		
Día10	15	21		
Día11	18	2	10	
Día12	5	6		
Día13	21			
Día14	15	18		
Día15	2			
F.Obj	6.444			



Seleccionando finalmente la que tenga mejor valor de la función objetivo. Una vez colocadas las dos visitas, la primera asignación posible se ha completado, y se prueba con la siguiente posibilidad; poner la primera visita en el segundo día.

El procedimiento acaba cuando se prueba en todas las posiciones y se encuentra la mejor.

5.5.3. Tercer procedimiento

El tercer procedimiento propuesto es el que permite una mayor versatilidad a la hora de asignar las visitas a los clientes.

La diferencia con los procedimientos anteriores se presenta en el paso número dos. Este paso queda de la siguiente forma.

Paso 2: Para la frecuencia de visita mayor, seleccionar el cliente más lejano de la residencia del vendedor y asignarlo a los días que correspondan.

La forma de asignar es la siguiente. Se coloca la primera visita del cliente seleccionado en el primer día, a continuación, se prueba a colocar la segunda visita en todos los días libres disponibles. Se selecciona la mejor según la función objetivo y se pasa a asignar la siguiente visita. Cuando se completan todas las visitas se obtiene un valor de la función objetivo para ese cliente. Este valor se compara con el que adquiere durante las diferentes posibilidades, colocando la primera visita en el resto de días, y se selecciona finalmente la de mejor valor.

Cabe comentar cómo se valora la regularidad de una solución cuando todas las visitas a un cliente no están asignadas: se considera la máxima regularidad que se podría alcanzar con el resto de visitas no asignadas.

Por ejemplo, supóngase que el número de días totales es 19 y se debe asignar un cliente con 4 visitas. Una vez asignadas las dos primeras visitas al cliente i , que distan entre sí una cantidad de 9 días, se puede calcular el siguiente vector de descomposición, $\lambda=(9,\lambda')$. En este caso el vector λ' , vector de descomposición



que resuelve el siguiente subproblema: número días totales igual a 10 (19-9), y 3 (4-1) intervalos entre visitas. ($\lambda' = (3,3,4)$)

Finalmente se obtiene un vector $\lambda = (9,3,3,4)$. Este vector indica la mejor regularidad posible que se puede obtener al colocar los dos primeros clientes a una distancia de 9 días. La parte de la función objetivo relativa a la regularidad para dicho cliente quedaría tal como se indica en la ecuación 2.

$$\left(9 - \frac{19}{4}\right)^2 + \left(3 - \frac{19}{4}\right)^2 + \left(3 - \frac{19}{4}\right)^2 + \left(4 - \frac{19}{4}\right)^2 \quad (\text{Ec. 2})$$



6. Proceso de optimización

Una vez obtenida la solución inicial, se ejecuta un proceso de mejora mediante cambios en la posición de las visitas a los clientes, ya sea entre rutas de diferentes días o en la ruta de un día. Este proceso sigue un ciclo iterativo de optimización local que incluye fases de perturbación de las soluciones (en el apartado 6.3 se expone con detalle dentro).

6.1. Proceso de optimización local

El proceso de optimización local está basado en dos procedimientos. Por un lado reordenar los clientes de un día y, por otro, mover clientes entre días. Estos procedimientos se combinan y se realizan a la vez, ya que mientras se reordenan los clientes entre días, también se reordenan los clientes de un mismo día.

6.1.1. Reordenar clientes en un mismo día

Este procedimiento busca una posición de un cliente, en la ruta de un día, que reduzca el valor de la función objetivo.

Sea, por ejemplo, la ruta 2, 3, 12, 23 y 45. El algoritmo consiste en mover a un cliente de la posición actual, al resto de posiciones entre otros dos clientes, y evaluar la función objetivo.

Es este caso se movería los clientes (por ejemplo, el 2 y el 3) de la siguiente forma:

Cliente 2

Posibilidades:

(0-3-**2**-12-23-45-0)

(0-3-12-**2**-23-45-0)

(0-3-12-23-**2**-45-0)

(0-3-12-23-45-**2**-0)

Cliente 3

Posibilidades:

(0-**3**-2-12-23-45-0)

(0-2-12-**3**-23-45-0)

(0-2-12-23-**3**-45-0)

(0-2-12-23-45-**3**-0)

Si en alguno de estos casos la función objetivo fuera mejor, dicho movimiento de inserción sería fijado y se volvería a ejecutar el proceso.

Este algoritmo mejora en algunos casos las soluciones, pero solamente en la parte referente a la cantidad de kilómetros o duración de la jornada, ya que la regularidad no se ve afectada.



6.1.2. Intercambio de clientes entre días

Este procedimiento consiste en realizar cambios en la asignación de los días de las visitas de un cliente: toda visita de cada cliente se cambia a los otros días posibles, es decir, a los días que no haya otra visita a ese mismo cliente, y se evalúa dicha solución parcial. Si la función objetivo es mejor se fija dicho cambio y se pasa a la siguiente visita.

Sea la solución representada por la figura 4, consistiría en mover el cliente 2 entre los días 4 (solución inicial) y 6 (solución obtenida).

Solución inicial								Solución obtenida								
Día1	21	5						Día1	21	5						
Día2	18	6	9	7	8			Día2	18	6	9	7	8			
Día3	15	21	11					Día3	15	21	11					
Día4	5	2	10					Día4	5	10						
Día5	21	23	18					Día5	21	23	18					
Día6	12	17	15					Día6	12	17	2	15				
Día7	6	2	24	25	19	21		Día7	6	2	24	25	19	21		
Día8	18	20	5					Día8	18	20	5					
Día9	9	7						Día9	9	7						
Día10	15	21						Día10	15	21						
Día11	18	2	10					Día11	18	2	10					
Día12	6	5	1					Día12	6	5	1					
Día13	21	12	22					Día13	21	12	22					
Día14	18	24	15					Día14	18	24	15					
Día15	4	2	3					Día15	4	2	3					

Fig. 4: Representación del movimiento Cross_exchange

Es un proceso de barrido, en el que se empieza con el primer cliente y se van eligiendo las visitas a medida que aparecen en la solución, por lo que el proceso de optimización depende del orden de la solución. Si se encuentra una solución mejor se fija y se pasa a buscar la siguiente visita o el siguiente cliente en el caso que no haya más visitas. El barrido se realiza intercambiando días de forma creciente y de forma decreciente (del primer al último y del último al primer día).



6.2. Proceso de perturbación

Esta parte del proceso de optimización ha sido diseñada para salir de óptimos locales. Se puede dar el caso que moviendo cualquier cliente de un día a otro o en el propio día, no se mejore la función objetivo.

Para solucionar este problema se perturba la solución, de manera que la nueva solución está lejos de la solución inicial, para, a continuación, repetir el proceso de optimización local.

Se han implementado dos formas de perturbar la solución “estancada”. Estas dos formas se conocen como el *Move_operator* y el *Cross_exchange*, que se pueden encontrar en el artículo “A variable neighborhood search heuristic for periodic routing problems” [Hammelmayr, V. C et al, 2007].

Ambas formas se aplican aleatoriamente, seleccionando una u otra en el momento de introducir la perturbación. Únicamente se realizan perturbaciones que proporcionan soluciones factibles.

6.2.1. Move_opeartor

En este tipo de perturbación se eliminan dos visitas consecutivas de un día y se incluyen en otro día.

Se selecciona aleatoriamente los días y las visitas a cambiar.

El proceso se puede representar gráficamente de la siguiente forma:

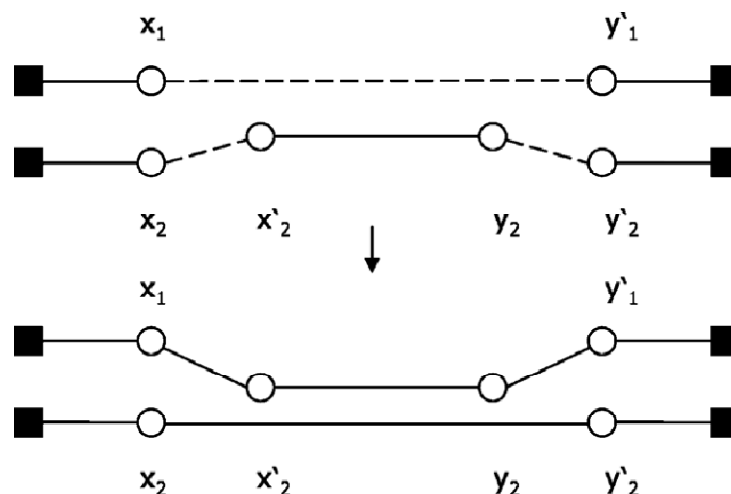


Fig. 5: Representación del movimiento Move_operator



Dónde existen dos rutas, entre las que se intercambia el arco (x'_2-y_2) desde la segunda a la primera.

Sea por ejemplo, la solución inicial representada en la figura 6. En el caso que se quiera cambiar el segundo arco del séptimo día y pasarlo al noveno, la solución que se obtiene es:

Solución inicial								Solución obtenida							
Día1	21	5						Día1	21	5					
Día2	18	6	9	7	8			Día2	18	6	9	7	8		
Día3	15	21	11					Día3	15	21	11				
Día4	5	2	10					Día4	5	2	10				
Día5	21	23	18					Día5	21	23	18				
Día6	12	17	15					Día6	12	17	15				
Día7	6	2	24	25	19	21		Día7	6	25	19	21			
Día8	18	20	5					Día8	18	20	5				
Día9	9	7						Día9	9	2	24	7			
Día10	15	21						Día10	15	21					
Día11	18	2	10					Día11	18	2	10				
Día12	6	5	1					Día12	6	5	1				
Día13	21	12	22					Día13	21	12	22				
Día14	18	24	15					Día14	18	24	15				
Día15	4	2	3					Día15	4	2	3				

Fig. 6: Aplicación del movimiento Move_operator



6.2.2. Cross_exchange

El siguiente sistema de perturbación es el resultado de realizar un doble *Move_operator* entre dos días. Se trata, por tanto, de intercambiar dos arcos de dos días diferentes, siendo seleccionados estos de forma aleatoria.

El proceso se puede representar gráficamente de la siguiente forma:

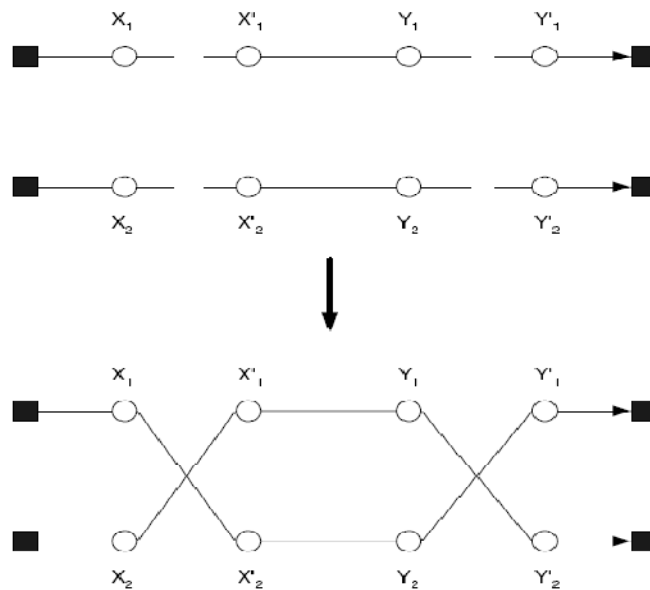


Fig. 7: Representación del movimiento *Cross_exchange*

Donde existen dos rutas, entre las que se intercambian dos arcos, uno de cada ruta entre ellas, pasando de las rutas $(X_1-X'_1-Y_1-Y'_1)$ y $(X_2-X'_2-Y_2-Y'_2)$ a las rutas $(X_1-X'_2-Y_2-Y'_1)$ y $(X_2-X'_1-Y_1-Y'_2)$.



Sea por ejemplo, la solución inicial representada en la figura 8. En el caso que se quiera cambiar el primer y el tercer arco de los días 5 y 6, la solución obtenida es:

Solución inicial								Solución obtenida								
Día1	21	5						Día1	21	5						
Día2	18	6	9	7	8			Día2	18	6	9	7	8			
Día3	15	21	11					Día3	15	21	11					
Día4	5	2	10					Día4	5	2	10					
Día5	21	23	18					Día5	21	23	18					
Día6	12	17	15					Día6	15	24	25					
Día7	6	2	24	25	19	21		Día7	6	12	17	2	19	21		
Día8	18	20	5					Día8	18	20	5					
Día9	9	7						Día9	9	7						
Día10	15	21						Día10	15	21						
Día11	18	2	10					Día11	18	2	10					
Día12	6	5	1					Día12	6	5	1					
Día13	21	12	22					Día13	21	12	22					
Día14	18	24	15					Día14	18	24	15					
Día15	4	2	3					Día15	4	2	3					

Fig. 8: Aplicación del movimiento Cross_exchange

6.3. Algoritmo de optimización

La estrategia de optimización se basa en realizar diversas veces el intercambio entre días combinándolo con el intercambio diario y añadiendo etapas de perturbación.

Una vez que se llega al punto en que cualquier movimiento perjudica el valor de la función objetivo, se procede a crear una perturbación. Se selecciona uno de los dos sistemas de perturbación aleatoriamente.

Este proceso se repite hasta que finaliza un tiempo de proceso limitado por el usuario.



Una posible representación del valor de la función objetivo a lo largo del tiempo sería:

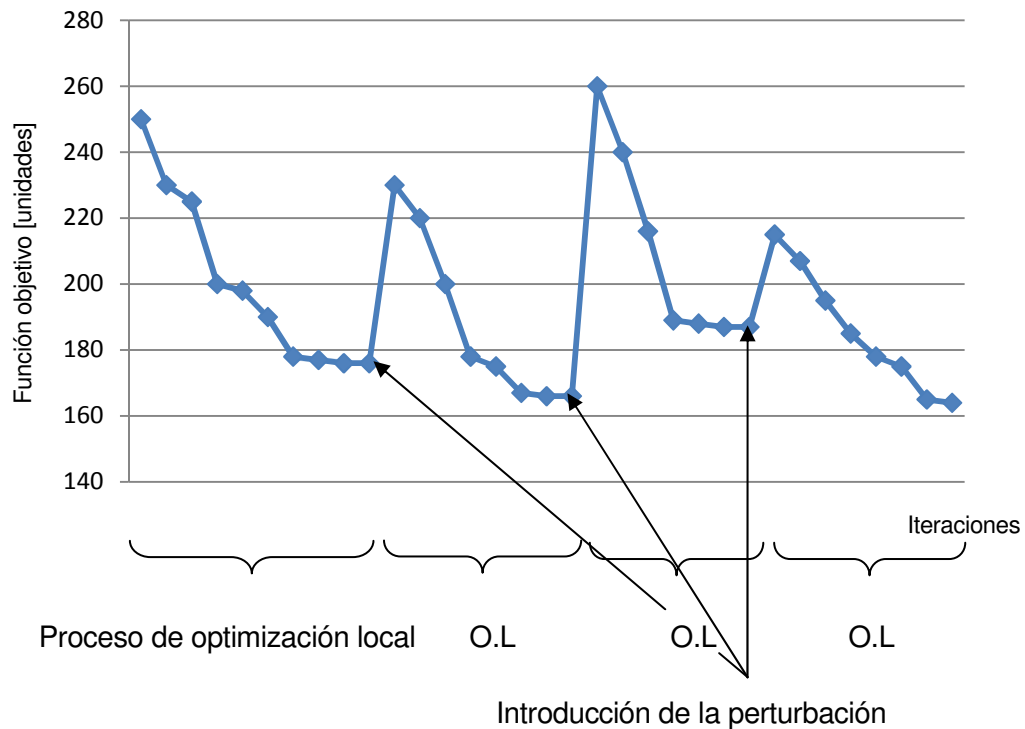


Fig. 9: Ejemplo de evolución de la solución

Se puede observar que a medida que se aplica la optimización local se reduce el valor de la función objetivo, hasta que llega a un punto en el que no existe ningún intercambio que haga decrecerla.

En ese momento se realiza una perturbación. El resultado es un empeoramiento de la función objetivo que posteriormente se vuelve a optimizar localmente para, en algunos casos, conseguir una solución mejor.

6.4. Elementos aleatorios

El concepto de aleatoriedad cobra gran importancia dentro de este proyecto, ya que la solución final depende de la asignación aleatoria en determinadas fases.



Ejemplos de esta aleatoriedad los tenemos en:

- I) **Introducción de la perturbación.** La perturbación se basa en el movimiento de la/s visita/s a un cliente, de una jornada de trabajo cargada a otra jornada, seleccionada aleatoriamente.

- II) **Asignación de clientes.** En el cálculo de la solución inicial, el asignar una segunda o tercera visita depende de las asignaciones anteriores, sin poder modificarlas.

Para intentar reducir los efectos de la aleatoriedad se repiten varias perturbaciones, aunque con este procedimiento, no se asegura que en todos los procesos en los que se introduzca un cambio que, en un principio, pueda parecer que empeore la solución, disponga de las circunstancias necesarias para encontrar un camino a una buena solución, y así pues, la mejore.



7. Experiencia computacional

Para probar y validar los procedimientos propuestos se ha trabajado sobre dos conjuntos de ejemplares. Un primer conjunto que se encuentra en la literatura y sirve para comparar algoritmos para la resolución del PTSP y otras variantes, y un segundo conjunto de datos que pertenecen a un caso real.

7.1. Datos iniciales

El primer conjunto de ejemplares se ha conseguido como réplica a una consulta realizada a Jean-François Cordeau del “*Service de l’enseignement de la gestion des opérations et de la logistique*” en el centro “*Chaire de recherche du Canada en logistique et en transport HEC Montréal*”. Dichos ejemplares se pueden encontrar en internet. [<http://neumann.hec.ca/chairedistributique/data/ptsp>]

Son problemas que presentan una cantidad de clientes de entre 50 y 300, en un horizonte de planificación de 4 a 6 días y entre 1 y 6 visitas a realizar a dichos clientes.

El problema que existe con estos datos es que el número de días del horizonte de planificación es muy corto respecto a los casos reales (en los que es más común trabajar con un mes de unos 20-22 días laborables).

7.2. Datos del caso real

Los datos reales corresponden a un conjunto de centros comerciales de la zona de levante principalmente, que deben ser visitados por el comercial de una empresa de reparto de *dvd*'s.

Las diferentes zonas a visitar por dicho comercial están localizadas en las regiones de Zaragoza, Tarragona y Valencia.

Se conocen las distancias en kilómetros y en minutos además del tiempo y el número de visitas de cada cliente.

7.3. Tiempo de cálculo

La experiencia computacional se ha realizado con un ordenador con procesador Inter Core2 Duo T5450 a 1,66Ghz y una memoria RAM de 2046 MB.



El tiempo de cálculo se divide entre el necesario para la obtención de la solución inicial y utilizado en el proceso de optimización. El tiempo transcurrido oscila entre décimas de segundo y tres segundos para calcular la solución inicial y un tiempo variable al realizar la optimización local, debido a que esta parte depende de la cantidad de iteraciones que el usuario desea realizar.

A modo de ejemplo, se opta por ejecutar seis veces el proceso de optimización local combinado con diez perturbaciones, se necesita un intervalo de tiempo de unos cuatro minutos, pudiendo variar en función de la aleatoriedad de las perturbaciones.

7.4. Resultados de la experiencia computacional

En un primer análisis se han comparado los resultados de las heurísticas diseñadas con los obtenidos con los procedimientos de la literatura que se pueden encontrar en el artículo anteriormente comentado [Hemmelmayr et al, 2007]. Cabe destacar que en este caso no se introduce el concepto de regularidad, puesto que este concepto es nuevo y no se ha aplicado en los procedimientos existentes.

En la comparación de las soluciones iniciales con las mejores encontradas por el resto de algoritmos de la literatura, se puede observar que el resultado es, en promedio un 9% peor.

La diferencia es lógica ya que los procedimientos de la literatura son muy especializados y elaborados e incorporan potentes procedimientos de optimización local.



Ejemplar	Elementos (clientes)	Mejor valor de la literatura [Km]	Obtenido [Km]	Diferencia [Km]	%
ej1	49	2064	2146	82	3,64%
ej2	97	3205	3382	177	5,52%
ej3	145	4027	4405	378	9,39%
ej4	193	4538	5022	484	10,67%
ej5	241	4613	5024	411	8,91%
ej6	289	5521	6181	660	11,95%
ej7	73	4435	4706	271	6,11%
ej8	145	5366	5965	599	11,16%
ej9	217	7234	8115	881	12,18%
ej10	289	8199	8964	765	9,33%
Promedio					8,86%

Tabla1. Comparativa de los resultados de los problemas de la literatura (sin regularidad)

A continuación se introduce el concepto de regularidad entre visitas para intentar determinar cuál de los tres algoritmos es el más eficiente.

Ejemplar	Mejor valor de la literatura [km]	Obtenido sin Regularidad [km]	Obtenido con Regularidad [km+reg]	Diferencia con mejor [km+reg]	Regularidad	Algoritmo mejor solución	%
ej1	2064	2146	2146	82	0	1	3,64%
ej2	3205	3382	3022	-183	28	3	-5,71%
ej3	4027	4405	3965	-62	41	2	-1,54%
ej4	4538	5022	4521	-17	55	2,3	-0,37%
ej5	4613	5024	4566	-47	30	2	-1,02%
ej6	5521	6181	5553	32	85	3	0,58%
ej7	4435	4706	4213	-222	43	3	-5,01%
ej8	5366	5965	5965	599	0	1	11,16%
ej9	7234	8115	7679	445	133	3	6,15%
ej10	8199	8964	8102	-97	168	2	-1,18%
Promedio							0,64%

Tabla2. Comparativa de los resultados de los problemas de la literatura (con regularidad)

La conclusión a la que se llega trabajando con este conjunto de ejemplares es que no se puede calificar a un único procedimiento como el mejor de todos ellos. Hay que considerar además que el periodo de días es pequeño para determinar el efecto de introducir la regularidad.



Además, en todos los ejemplares, la frecuencia de visitas a los clientes es múltiplo del número de días. Por ejemplo, el primer ejemplar tiene una duración de 4 días, y las frecuencias son de 4, 2 y 1 días. De esta forma, solo alteran la regularidad los clientes con dos visitas, ya que, por un lado, hay cuatro opciones de colocar una única visita, (todas de igual valor de regularidad) y, por otro, solo hay una opción de colocar los clientes con cuatro visitas.

Tampoco se dispone del tiempo de desplazamiento ni de la duración de la jornada laboral, con lo que no se pueden introducir las penalizaciones del caso real.



8. Caso de aplicación

En este trabajo, se presenta el caso real de una compañía que quiere generar rutas para las visitas que debe hacer cada comercial de plantilla. La compañía se dedica al negocio de venta de películas en grandes superficies. Cada vendedor tiene una zona de trabajo independiente y cada cliente siempre es asignado a un mismo vendedor particular.

El área comercial está dividida en diferentes zonas, una por cada comercial, siendo la zona a estudiar principalmente la de levante.

La compañía quiere generar las rutas comerciales para un ciclo, que dura un mes. Como ya se ha comentado, las rutas del comercial se caracterizan porque cada cliente tiene que ser visitado un número determinado de veces. El objetivo es minimizar una función objetivo que considera la distancia recorrida, la regularidad entre visitas o el tiempo excedido de siete y ocho horas.

El periodo de este caso particular es de 26 días (4 semanas trabajando de lunes a sábado más dos días de la siguiente semana).

Por motivos de confidencialidad no se ha especificado el nombre de cada cliente, asignándole a cada uno un código.

El número de visitas, código del cliente y población correspondientes a cada cliente se muestra en la tabla 3.

Código cliente	Nº Visitas	Población
2	1	Castellón
3	1	Villareal
4	1	Vinaroz
5	1	Castellón
7	1	Huesca
9	1	Tarragona
15	1	Sagunto
16	1	Gandía
17	1	Paterna
18	1	Valencia
20	1	Xirivella
22	1	Burjassot
23	1	Valencia



24	1	Alboraya
25	1	Alfarfar
26	1	La Eliana
28	1	Alzira
32	1	Zaragoza
34	1	Zaragoza
35	1	Zaragoza
36	1	Utebo
37	1	Zaragoza
39	1	Zaragoza
40	1	Zaragoza
41	1	Zaragoza
44	1	Zaragoza
6	2	Castellón
12	2	Aldaya
19	2	Valencia
33	2	Zaragoza
1	3	Castellón
11	3	Tarragona
10	4	Reus
38	4	Zaragoza
43	5	Villanueva de Gallego

Tabla3. Código cliente, número de visitas y población

Los parámetros de la función objetivo, parametrizada en la ecuación 1, dan la misma importancia a la regularidad y a la cantidad de kilómetros, siendo α igual a 0,5, con unas penalizaciones γ_1 y γ_2 de 100 unidades y γ_3 de 10E10 unidades. La función objetivo queda representada en la ecuación 3.

$$F.Obj. = 0,5 \cdot (Cantidad_km) + 0,5 \cdot 100 \cdot Regularidad + 100 \cdot Ruta_7^2 + 10E10 \cdot Ruta_8^4 \quad (Ec. 3)$$



8.1. Resultados

En este apartado se describen los resultados obtenidos por los diferentes algoritmos diseñados. Para cada resultado se muestra la solución inicial y final, la distribución de la jornada laboral a lo largo del mes para ambas soluciones y los valores que toma la función objetivo.

Además para el primer algoritmo se ha representado la evolución del valor de la función objetivo, de la regularidad y de la distancia recorrida, en las diferentes iteraciones y procesos de optimización local. Para obtener las diferentes soluciones se ha ejecutado el proceso de optimización 6 veces introduciendo 6 perturbaciones.

8.1.1. Primer algoritmo. Soluciones y valores obtenidos

Solución inicial:

Día 1	43	38			
Día 2	1	18			
Día 3	10	26			
Día 4	24	19			
Día 5	12	23			
Día 6	43	39			
Día 7	38	32			
Día 8	20	5			
Día 9	4	10	9	11	
Día 10	6	1	3		
Día 11	43	44			
Día 12	33	34			
Día 13	38	35			
Día 14	40				
Día 15	10				
Día 16	43	7			
Día 17	11	19			
Día 18	12	25			
Día 19	1	22			
Día 20	38	36			
Día 21	43	37			
Día 22	10				
Día 23	6	15			
Día 24	2	17			
Día 25	33	41			
Día 26	28	16	11		



La distribución del tiempo de trabajo de cada ruta en la solución inicial se aprecia en la figura 10.

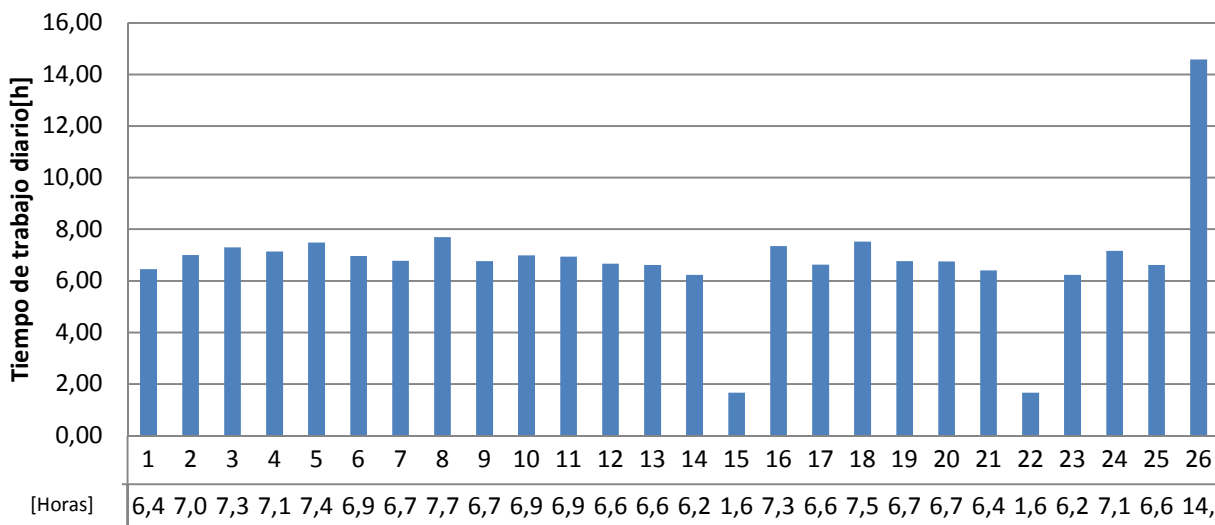


Fig.10: Tiempo de trabajo al ejecutar el primer algoritmo

Valores de los resultados iniciales:

Función objetivo 8,75e12
 Cantidad de kilómetros 8062,5 km
 Regularidad 4,13

La evolución de la función objetivo para el primer algoritmo se muestra en la fig. 11.

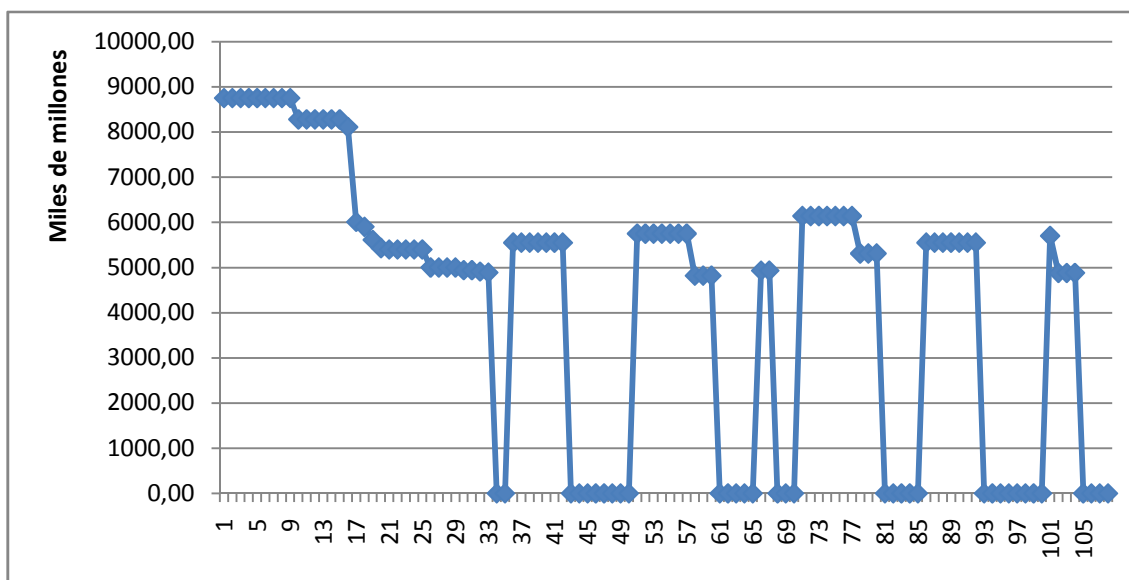
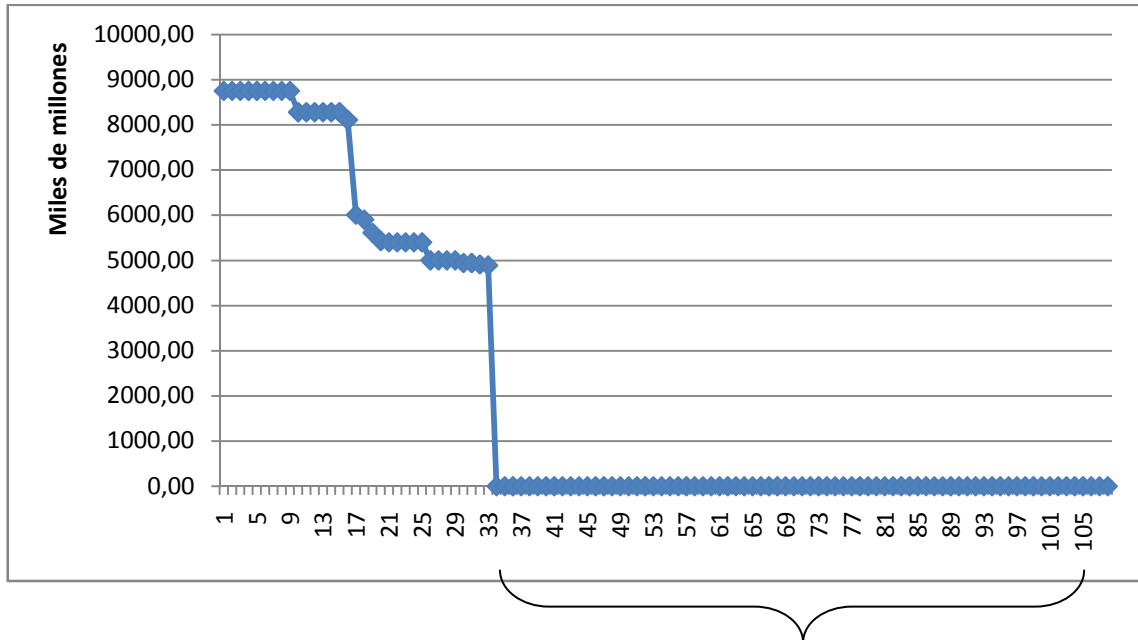


Fig. 11: Evolución de la función objetivo al ejecutar el primer algoritmo



A continuación se representan la evolución del valor de la mejor solución obtenido por dicho procedimiento, así como el detalle a partir del paso 34 (figuras 12 y 13). Se puede observar que la función objetivo se mejora en la última iteración.



Detalle de la mejor solución a partir del paso 34

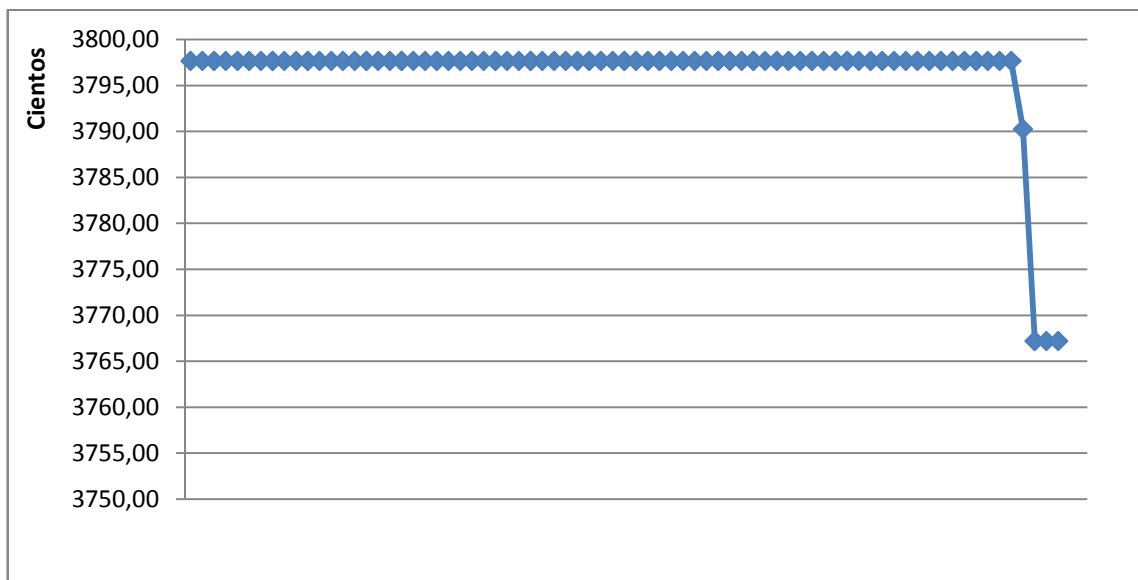


Fig. 12 y 13: Evolución de la mejor solución al ejecutar el primer algoritmo



La evolución de la regularidad para el primer algoritmo se muestra en la fig. 14.

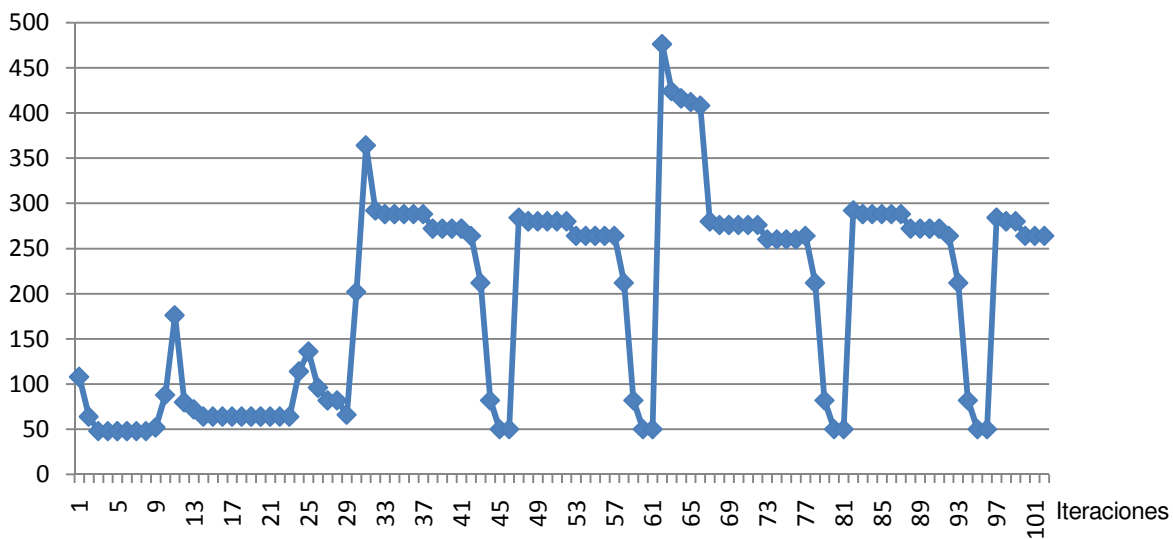


Fig. 14: Evolución de la regularidad al ejecutar el primer algoritmo

La figura 15 muestra la evolución de la distancia recorrida al aplicar el primer algoritmo.

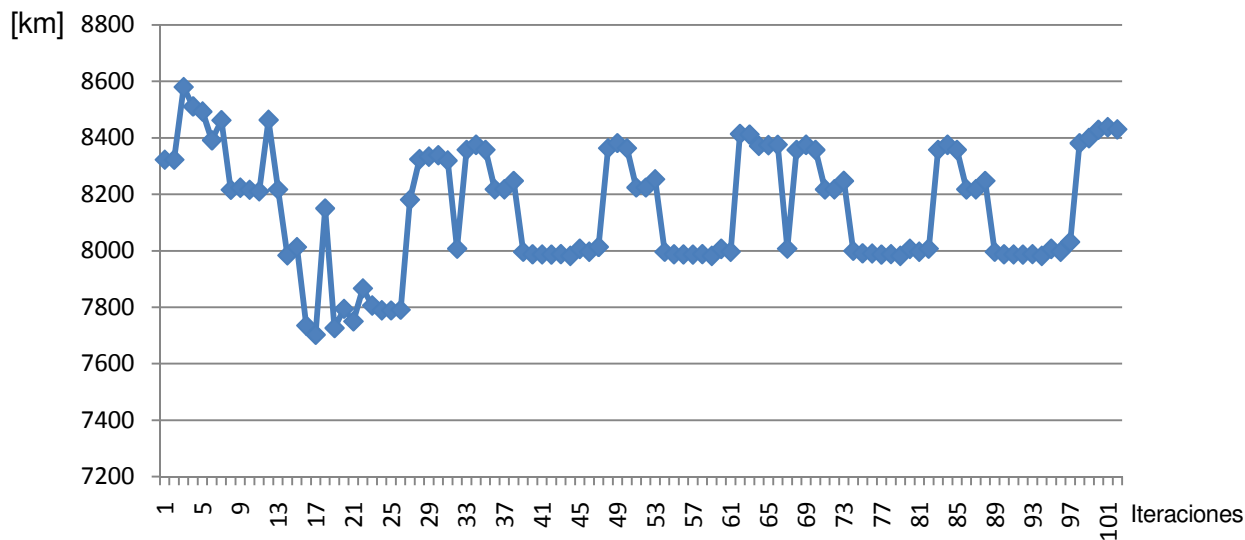


Fig. 15: Evolución de la cantidad de km's al ejecutar el primer algoritmo



En las dos figuras anteriores se puede apreciar como varían los diferentes parámetros de la función objetivo (regularidad y distancia recorrida) en el momento en que se introduce la perturbación, reduciéndose en el proceso de optimización local.

Solución final:

Día 1	43	38		
Día 2	18	2		
Día 3	26	10		
Día 4	24	19		
Día 5	12	3		
Día 6	43	39		
Día 7	38	32		
Día 8	20	11		
Día 9	10	9	6	
Día 10	4	15		
Día 11	43	44		
Día 12	33	34		
Día 13	38	35		
Día 14	40			
Día 15	10	11	1	
Día 16	43	7		
Día 17	22	1		
Día 18	12	25		
Día 19	23	6		
Día 20	38	36		
Día 21	43	37		
Día 22	10	11	1	
Día 23	16			
Día 24	17	5		
Día 25	33	41		
Día 26	19	28		



La distribución del tiempo de trabajo de cada ruta final se representa en la figura 16.

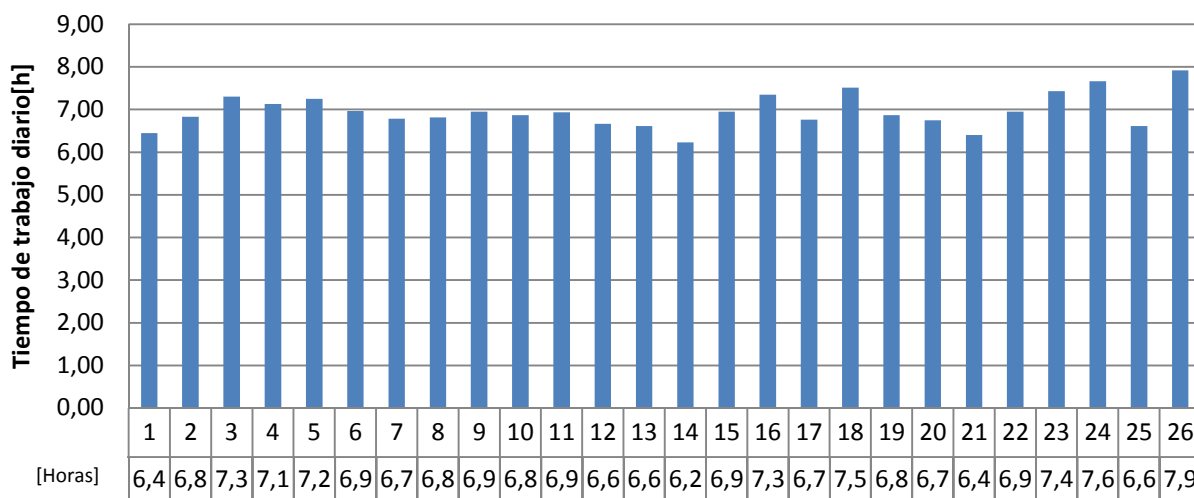


Fig. 16: Tiempo de trabajo final al ejecutar el primer algoritmo

Valores de los resultados finales:

Función objetivo 379766,17
 Cantidad de kilómetros 8319 km
 Regularidad 364,13



8.1.2. Segundo algoritmo. Soluciones y valores obtenidos

Solución inicial:

Día 1	43	38		
Día 2	10	11		
Día 3	5			
Día 4	6	15		
Día 5	12	23		
Día 6	43	36	41	
Día 7	38	32		
Día 8	10	26		
Día 9	1	25		
Día 10	11	19		
Día 11	43	39		
Día 12	40			
Día 13	38	33		
Día 14	10	22		
Día 15	28	16	2	
Día 16	43	44		
Día 17	6	1	3	
Día 18	11	12		
Día 19	38	35		
Día 20	10	37		
Día 21	43	7		
Día 22	4	17		
Día 23	24	19		
Día 24	20	9		
Día 25	1	18		
Día 26	33	34		



La distribución del tiempo de trabajo en cada ruta en la solución inicial se aprecia en la figura 17.

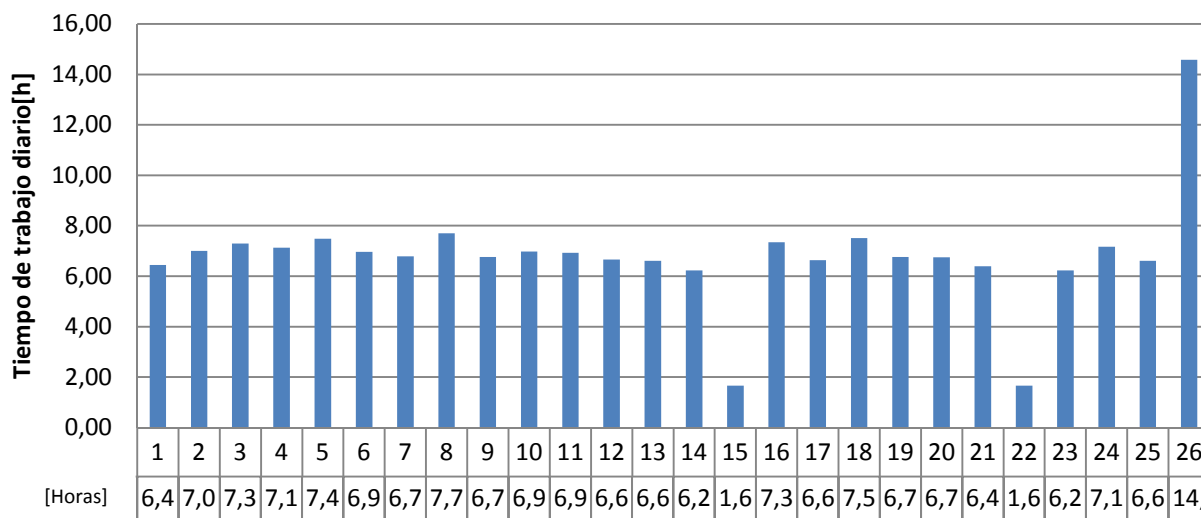


Fig. 17: Tiempo de trabajo al ejecutar el segundo algoritmo

Valores de los resultados iniciales:

Función objetivo 8,98e12
 Cantidad de kilómetros 9093,9 km
 Regularidad 12,13

La distribución del tiempo de trabajo final se representa en la figura 18.

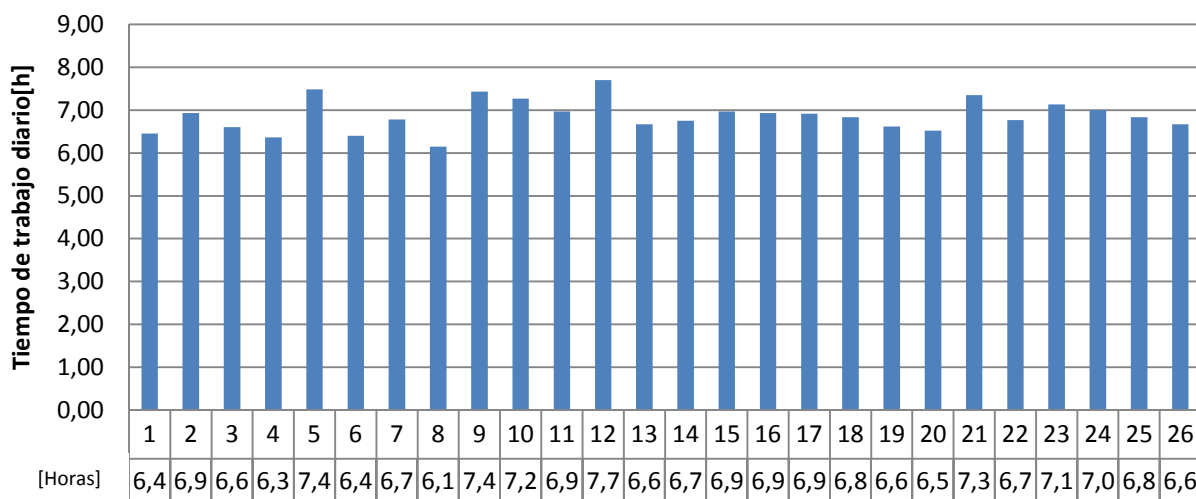


Fig. 18: Tiempo de trabajo final al ejecutar el segundo algoritmo



Valores de los resultados finales:

Función objetivo 315408,57
 Cantidad de kilómetros 8203,8 km
 Regularidad 62,13

Solución final:

Día 1	43	38		
Día 2	15	1		
Día 3	2	5		
Día 4	11	10	6	
Día 5	23	12		
Día 6	43	41		
Día 7	38	32		
Día 8	1	10		
Día 9	16			
Día 10	19	25		
Día 11	43	39		
Día 12	40	36		
Día 13	38	33		
Día 14	26	11		
Día 15	28			
Día 16	43	44		
Día 17	6	1	4	
Día 18	12	22		
Día 19	38	35		
Día 20	37	10		
Día 21	43	7		
Día 22	17	11		
Día 23	24	19		
Día 24	10	9	20	
Día 25	18	3		
Día 26	33	34		



8.1.3. Tercer algoritmo. Soluciones y valores obtenidos

Solución inicial:

Día 1	6	1	12	19	33	38	43	10	11	
Día 2	5	3								
Día 3	41	40	37							
Día 4	15									
Día 5	16									
Día 6	43	35								
Día 7	17									
Día 8	10	38								
Día 9	1	4	11							
Día 10	20									
Día 11	43	39								
Día 12	22	19								
Día 13	12									
Día 14	6	2	10							
Día 15	38	33								
Día 16	23									
Día 17	43	44								
Día 18	1	11								
Día 19	18	24								
Día 20	10	9								
Día 21	38	32								
Día 22	43	7								
Día 23	25									
Día 24	26									
Día 25	28									
Día 26	36	34								



La distribución del tiempo de trabajo de cada ruta en la solución inicial se aprecia en la figura 19.

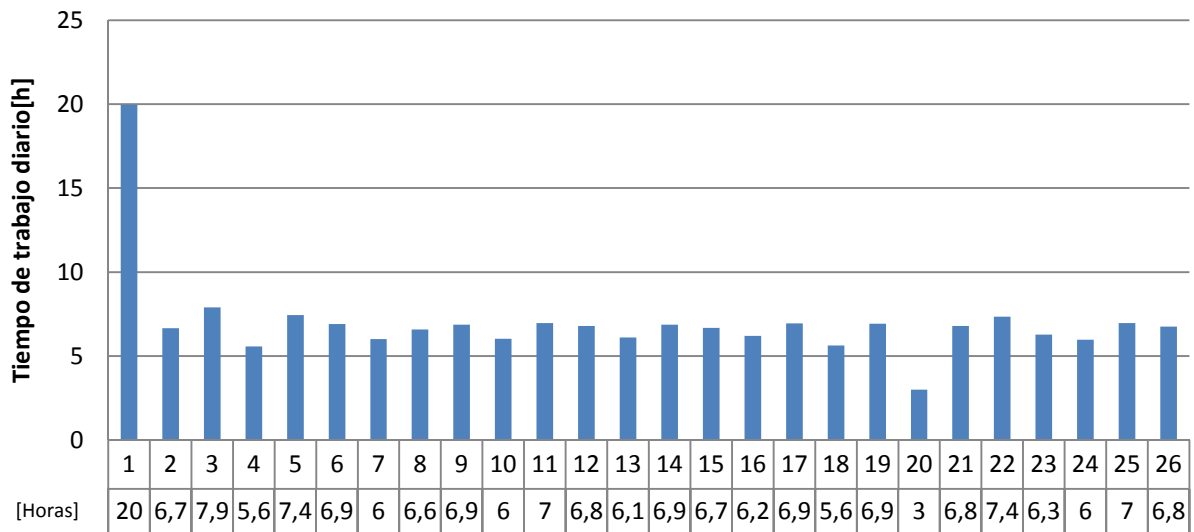


Fig. 19: Tiempo de trabajo al ejecutar el tercer algoritmo

Valores de los resultados iniciales:

Función objetivo 1,20e13
 Cantidad de kilómetros 7909,6 km
 Regularidad 16,13

La distribución del tiempo de trabajo final se representa en la figura 20.

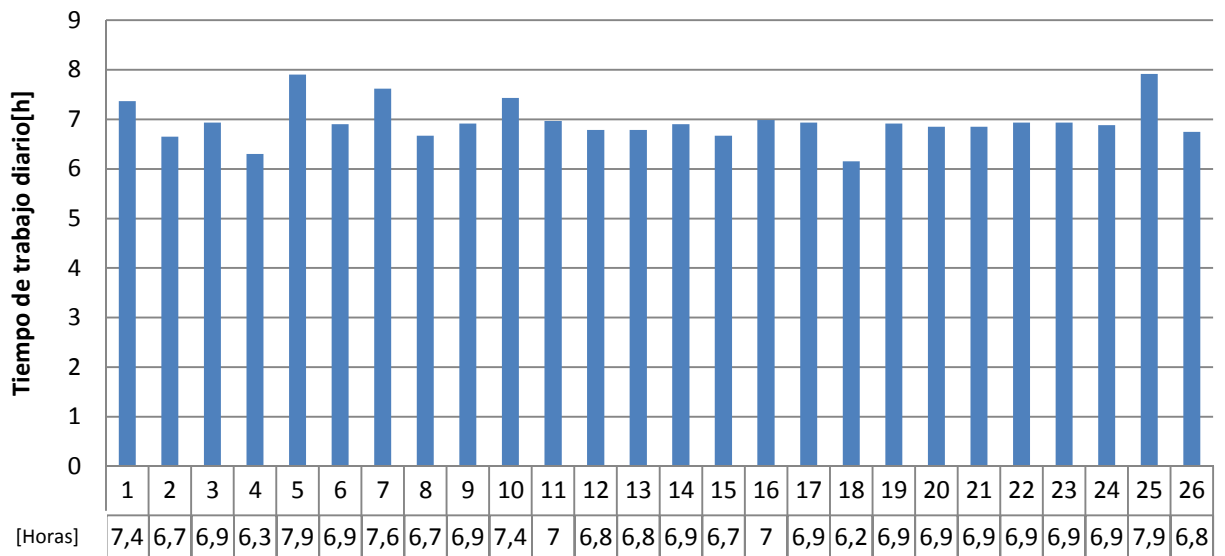


Fig. 20 Tiempo de trabajo final al ejecutar el tercer algoritmo



Valores de los resultados finales:

Función objetivo 262141,82
 Cantidad de kilómetros 8011,9 km
 Regularidad 568,13

Solución final:

Día 1	12	26		
Día 2	3	5		
Día 3	43	10		
Día 4	7			
Día 5	41	40	37	
Día 6	43	35		
Día 7	17	20		
Día 8	38	33		
Día 9	4	1	6	
Día 10	16			
Día 11	43	39		
Día 12	22	19		
Día 13	38	32		
Día 14	15	10		
Día 15	38	33		
Día 16	23	11		
Día 17	43	44		
Día 18	10	1		
Día 19	18	24		
Día 20	1	2	11	
Día 21	34	9		
Día 22	43	10		
Día 23	25	6		
Día 24	12	11		
Día 25	19	28		
Día 26	38	36		



A modo de ejemplo y para esta solución, se expone en qué días se realizan las visitas a un par clientes en la solución final.

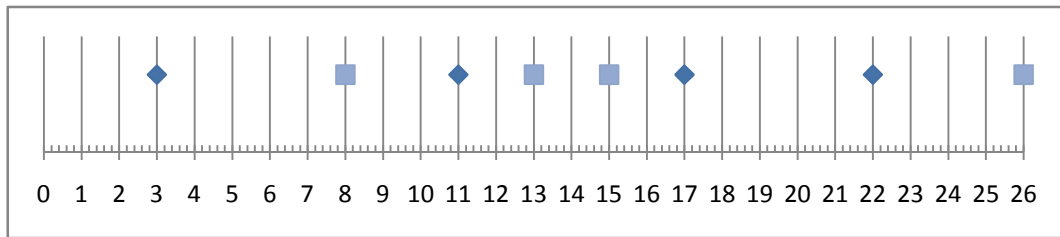


Fig. 21: Disposición de las visitas a los clientes 38 y 43

Los intervalos de visita del cliente 43, representado con un rombo en la figura 21, son de 8-6-5-7, mientras que los del cliente 38, representado con un cuadrado, son de 5-2-11-8, claramente alejados del patrón para propiciar una mejor solución.

8.1.4. Resumen de resultados

La tabla 4 muestra los resultados obtenidos por los tres algoritmos.

	Inicial		Final	
Algoritmo 1	Cantidad km's	8062,5	Cantidad km's	8319
	Regularidad	4,13	Regularidad	364,13
	F.Obj:	8,75e12	F.Obj:	379766,17
Algoritmo 2	Cantidad km's	9093,9	Cantidad km's	8432
	Regularidad	12,13	Regularidad	62,13
	F.Obj:	8,98e12	F.Obj:	315408,57
Algoritmo 3	Cantidad km's	7909,6	Cantidad km's	8203,8
	Regularidad	16,13	Regularidad	568,13
	F.Obj:	1,20e13	F.Obj:	262141,82

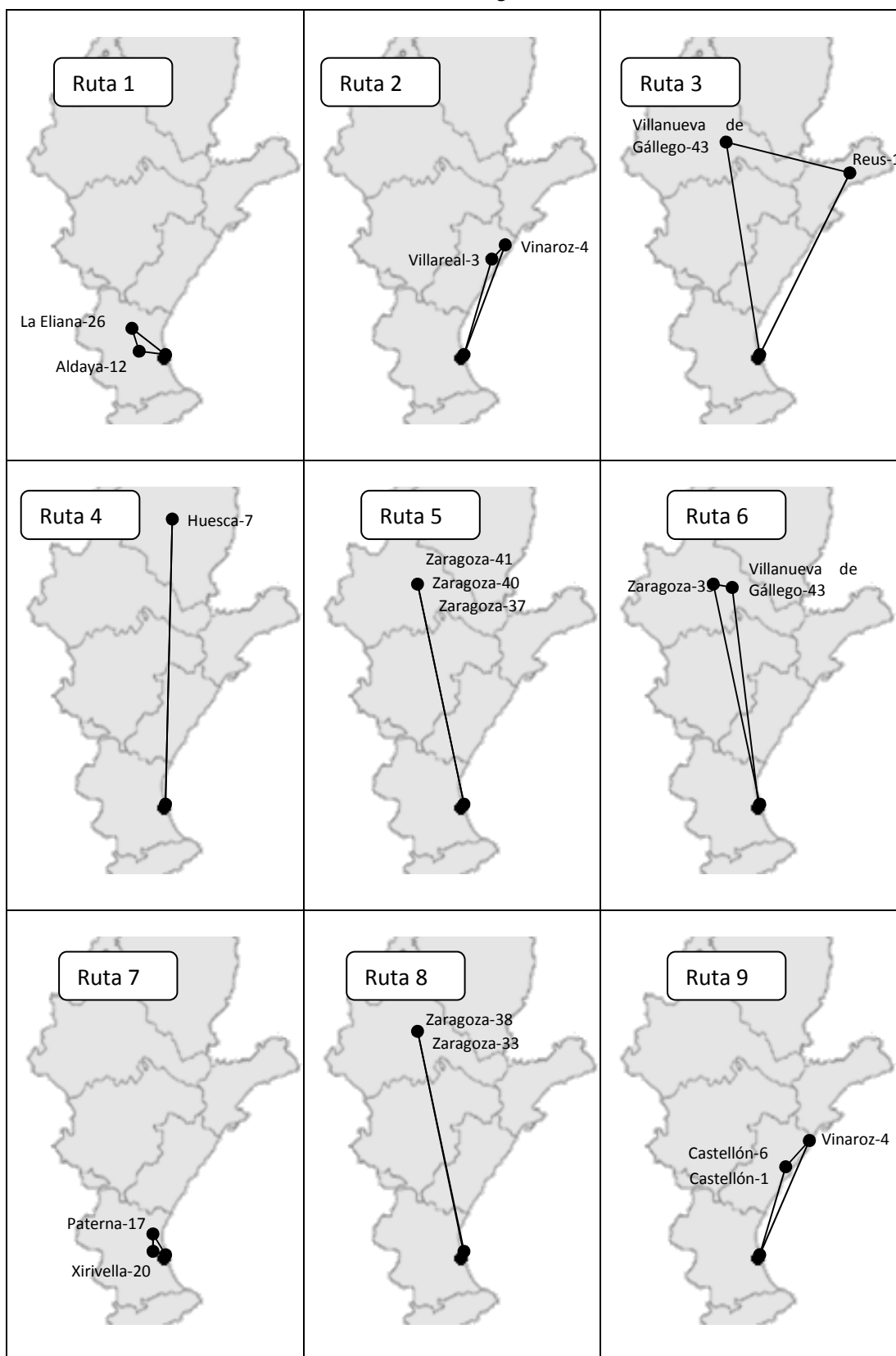
Tabla4. Resumen de los resultados

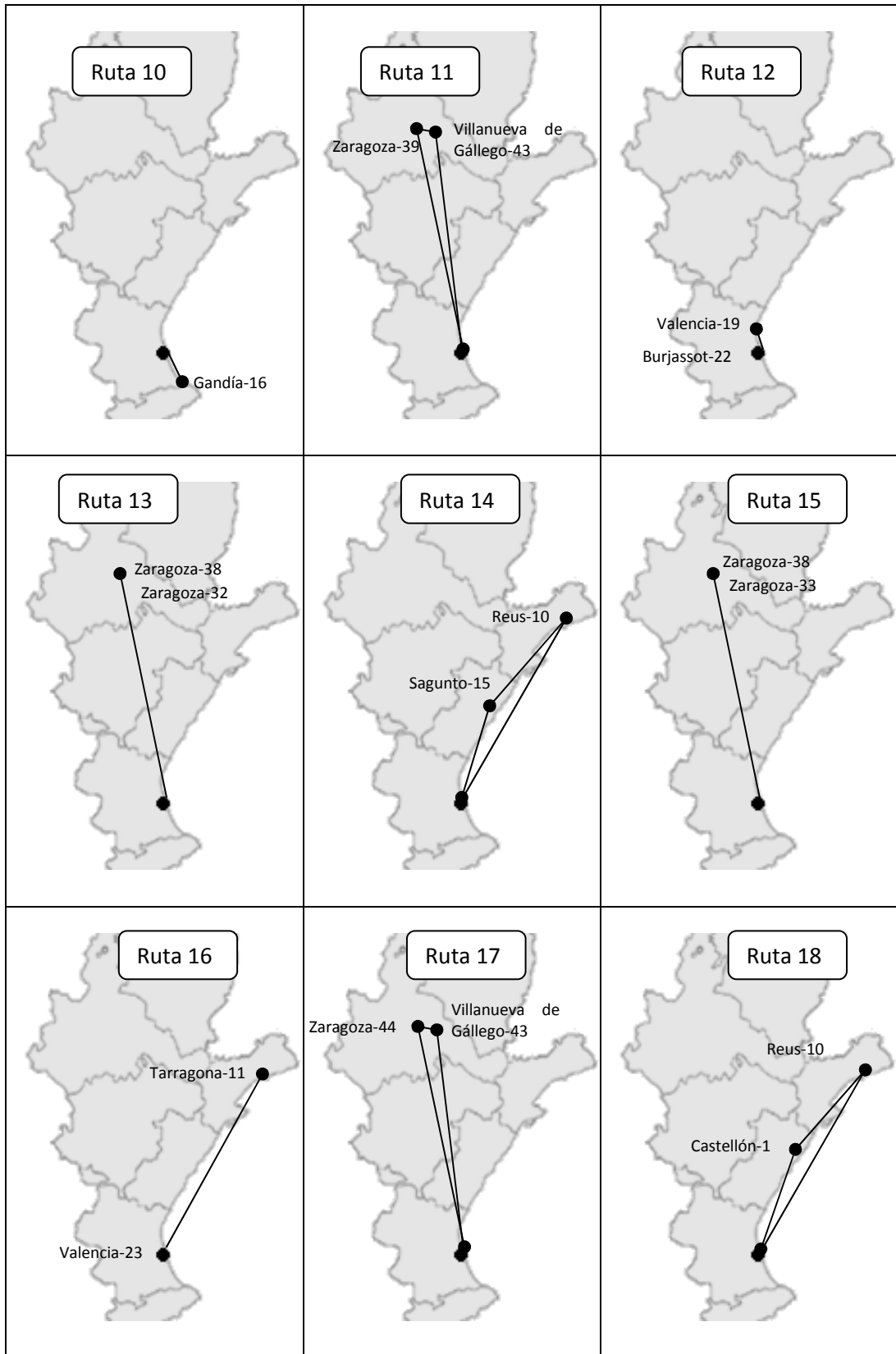
En este caso, atendiendo sólo al valor de la función objetivo, es razonable indicar que la mejor solución se obtiene con el tercer algoritmo (a pesar de la regularidad).

La cantidad de kilómetros a efectuar o la regularidad aumentan en algún caso, lo que es debido a que se obtienen soluciones con una mejor distribución del tiempo de las rutas, entendiendo rutas que no sobrepasan las 8 horas. Esta mejora se puede apreciar en el valor de la función objetivo.



Las rutas obtenidas finalmente con el tercer algoritmo han sido:





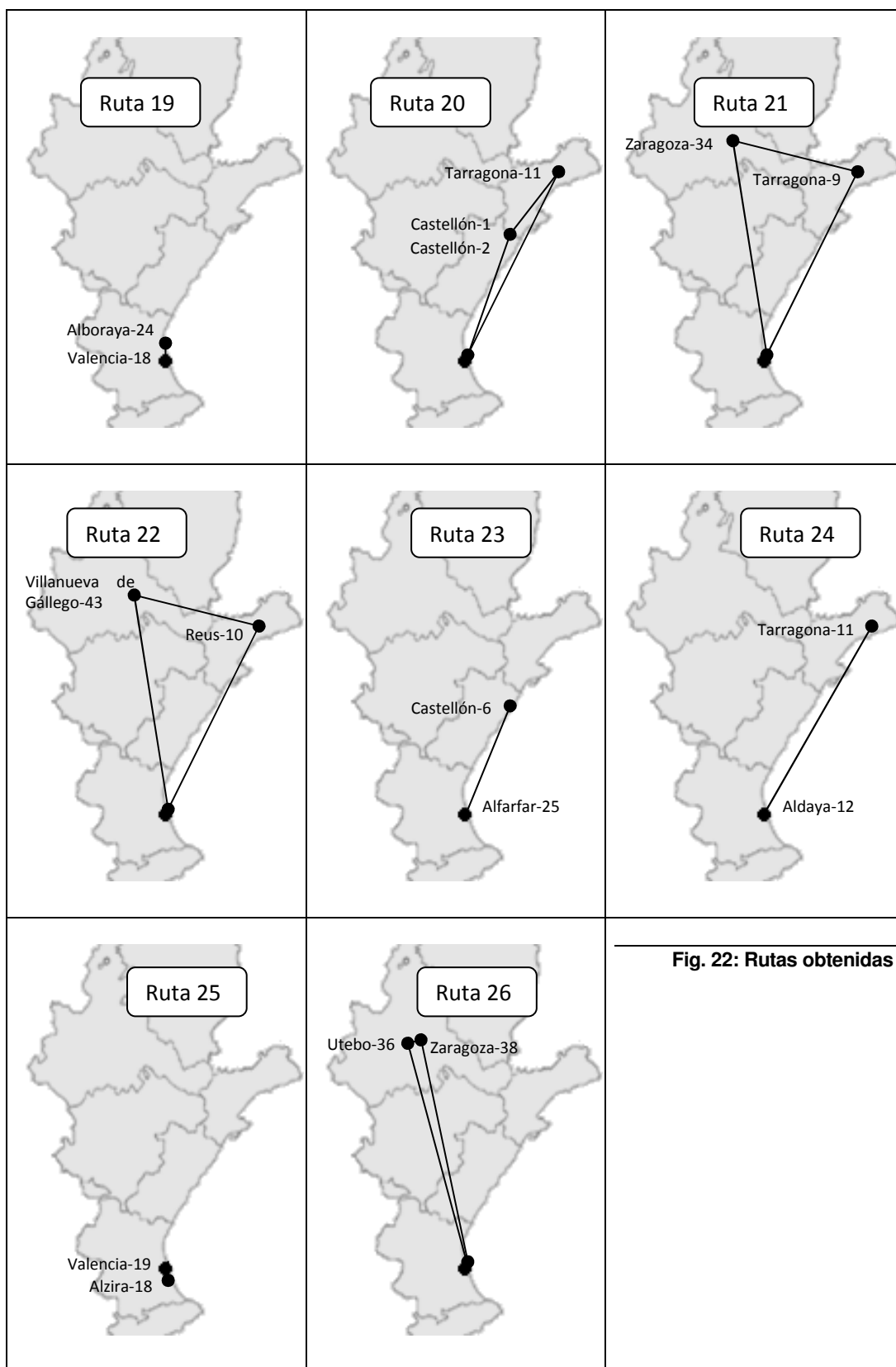


Fig. 22: Rutas obtenidas



Para calcular las rutas del resto de zonas de la empresa, se recomienda efectuar los tres procedimientos propuestos y proporcionar las tres soluciones, y que sea la propia empresa quien decida cual es mejor, ya que existe una diferencia en la cantidad de horas trabajadas en los diferentes días, y se puede dar el caso, que una solución a priori peor, se pueda adaptar mejor a las necesidades del vendedor.

Además el decisor puede contemplar cuestiones cualitativas que no se han cuantificado en este proyecto.

A continuación se hace un análisis para mostrar cómo puede afectar la parametrización de la función objetivo a la solución. Para ello se varían los factores de la función objetivo que priorizan la parte de la regularidad o la de cantidad de kilómetros recorridos. Para este procedimiento se ha repetido el proceso de optimización local tres veces y se han añadido cuatro perturbaciones.

En la tabla número 5 se muestran los valores obtenidos al variar la importancia de cada elemento de la función objetivo.

Importancia Cantidad kilómetros	Importancia Regularidad	Regularidad	Cantidad de kilómetros	Función objetivo
0%	100%	170,13	8.282,70	292.113,33
10%	90%	176,13	7.883,70	336.840,37
20%	80%	288,13	8.273,90	296.605,45
30%	70%	112,13	7.928,40	331.427,85
40%	60%	544,13	7.812,90	306.073,16
50%	50%	280,13	7.939,70	289.176,52
60%	40%	292,13	7.868,20	290.506,25
70%	30%	254,13	8.046,40	281.256,48
80%	20%	176,13	7.883,70	330.029,63
90%	10%	220,13	7.658,10	324.993,62
100%	0%	508,13	7.601,90	270.801,90

Tabla5. Valores de los términos de la F.Obj. con diferentes parametrizaciones

La forma de variar la importancia de cada uno de los parámetros es variar el parámetro α de la ecuación 1. Dándole valores entre 0 y 1. Se puede apreciar que a medida que aumenta la importancia que se le da a cada parámetro, éste disminuye, habiendo algunos resultados que no siguen la tendencia y que son provocados por la aleatoriedad del proceso.



A continuación se representan la cantidad de kilómetros, figura 23, la regularidad, figura 24, y la función objetivo, figura 25, para diferentes parametrizaciones.

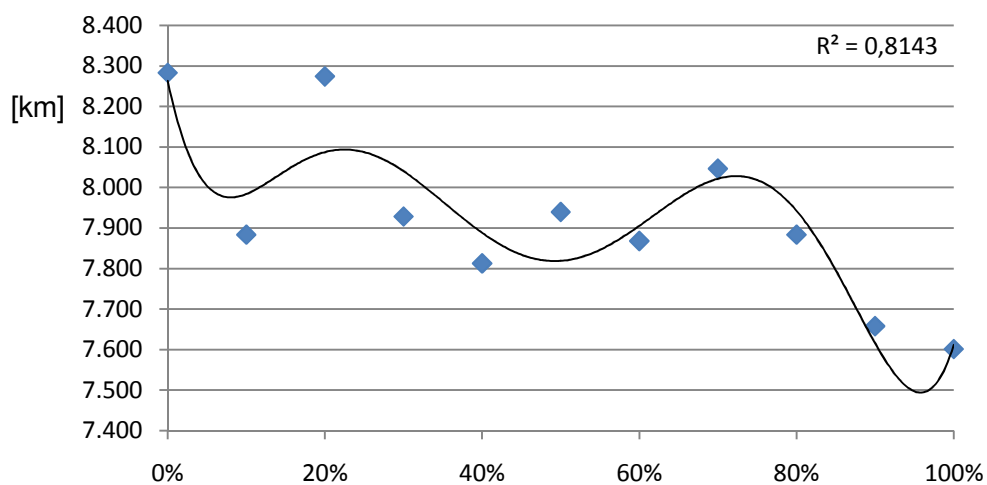


Fig. 23: Cantidad de kilómetros para diferentes parametrizaciones

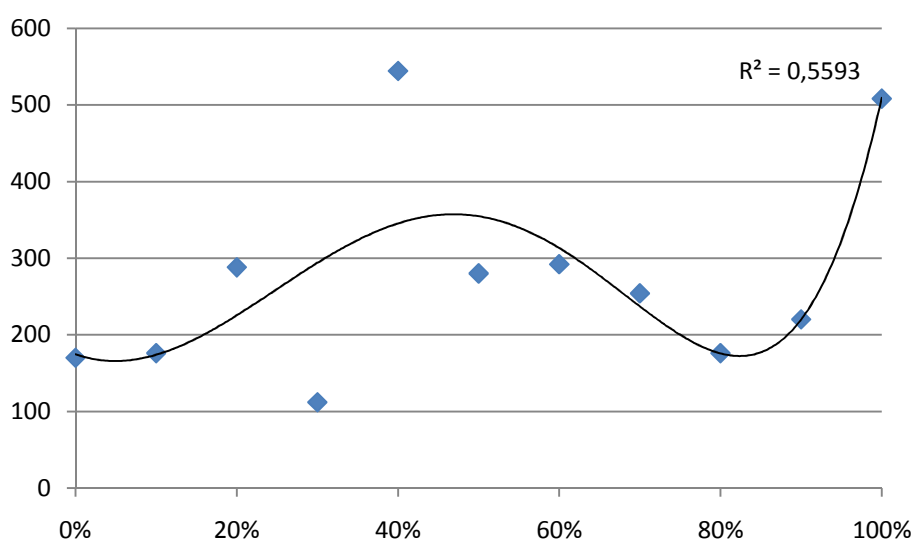


Fig. 24 Regularidad para diferentes parametrizaciones



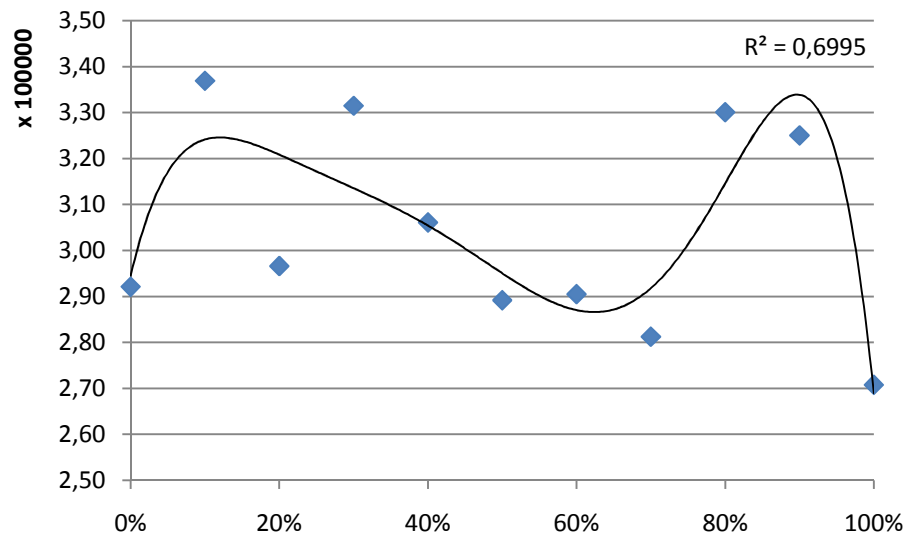


Fig. 25 Ucción objetivo para diferentes parametrizaciones

Se puede observar que al variar los parámetros de la función objetivo se obtienen diferentes valores de regularidad y de distancia recorrida.

En función de la importancia que se dé a cada factor, se puede lograr una mayor adaptación a las necesidades reales. También se pueden introducir modelos (líneas de tendencia) en cada gráfico. Estos modelos se introducen desde la aplicación Excel y se pueden utilizar para poder parametrizar la función objetivo correctamente, siempre y cuando, la calidad del coeficiente de determinación de la regresión obtenida (R^2) alcance valores que puedan garantizar el uso de la función.



9. Nuevas posibilidades de aplicación

En este apartado se desarrollan dos nuevas herramientas de gestión. La primera aplica el concepto de regularidad a las ventas y la segunda evalúa qué cliente se debe priorizar a la hora de buscar beneficios por visitas extra.

En este caso, se ha estimado un conjunto de datos necesarios para la aplicación: la demanda diaria de un comercio, el coste de las horas extra o el incremento de ventas para una visita extra. Estos datos intentan ajustarse a los que se tendrían en un caso real, aunque la verdadera finalidad de este apartado es crear un conjunto de herramientas que permitan gestionar estos problemas, más que obtener los valores exactos de los datos de éstos.

9.1. Regularidad de ventas

Una vez completado el estudio del caso con los datos reales, se plantea la posibilidad de ampliar el significado de la regularidad.

Intentar que el tiempo entre visitas sea el mismo, o lo más parecido posible, tiene sentido cuando la demanda diaria durante el mes es homogénea, es decir, cuando dejar de ir un sábado e ir un viernes es lo mismo que dejar de ir un miércoles e ir un martes. En cambio, en la mayoría de productos la demanda no es homogénea, por lo que resulta interesante no ir cada ciertos días, si no ir, cuando se ha consumido una cantidad de producto (si es posible, la misma). Por este motivo se introduce el concepto de regularidad de ventas.

Para plantear esta hipótesis se ha establecido una cantidad de ventas para cada día. Teniendo un mayor volumen de ventas las jornadas de principio del mes y las de fin de semana.

Se entiende como regularidad entre ventas óptima, aquella en la que la visita se produce en el momento de venta parcial según el número de visitas. Por ejemplo, si en un mes se venden 100 unidades y se visita al cliente 4 veces, se debe visitar de forma que entre visita y visita se venda 25.



El valor que se ha dado a la cantidad de ventas durante el mes para todos los clientes más un cierto ruido es el siguiente:

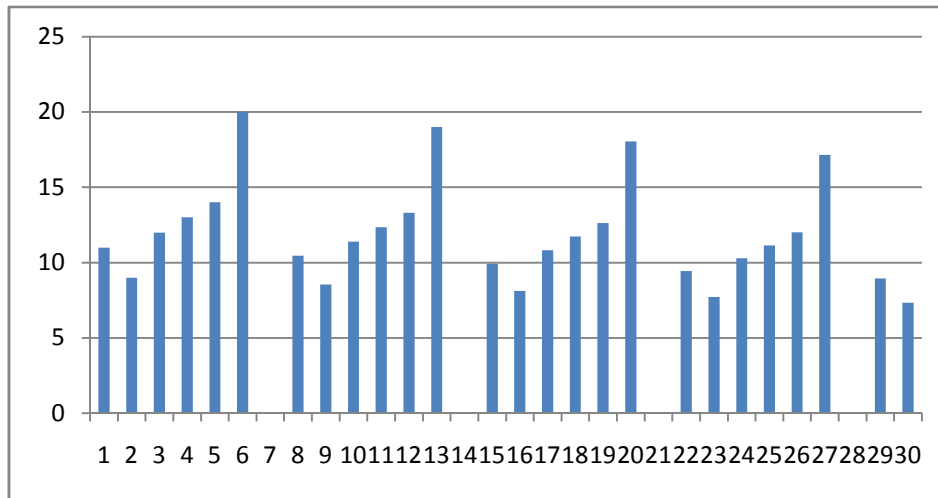


Fig. 26: Evolución de ventas durante el mes

Se puede apreciar que hay días en los que la cantidad de ventas es igual a cero. Estos días representan a los domingos, ya que se ha supuesto que el establecimiento comercial no trabaja. Como se ha comentado hay un incremento progresivo de ventas durante la semana con una venta mayor el sábado, y un descenso durante el mes, siendo menores las ventas al final de éste.

Para analizar el funcionamiento de la herramienta, se ha aplicado el tercer algoritmo al ejemplar (considerando la regularidad de ventas) y se ha comparado con la solución anterior que consideraba la regularidad entre visitas. En primer lugar se muestra el reparto de las jornadas de trabajo considerando la regularidad entre visitas, figura 27.

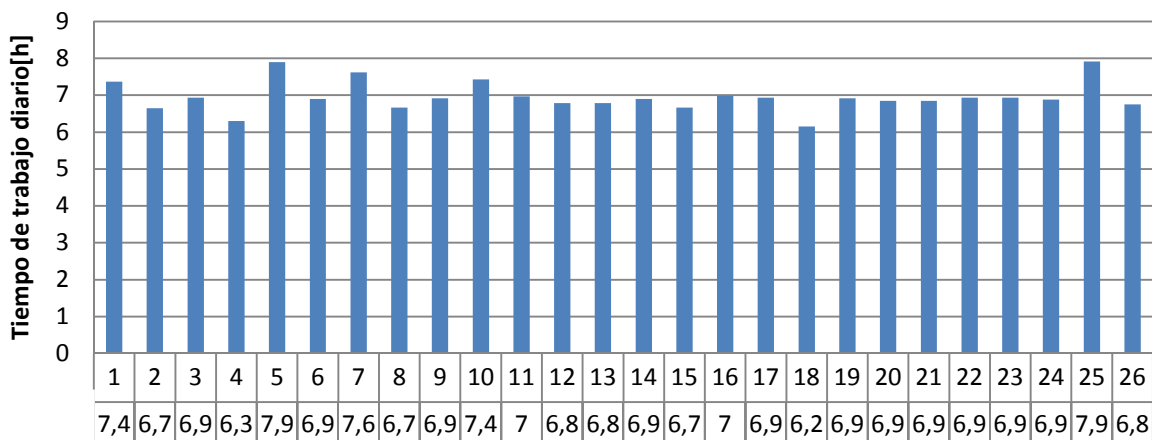


Fig. 27: Evolución del tiempo de trabajo anterior



Se puede apreciar que no se adapta a lo que en realidad se necesita. Por ejemplo con el cliente 1, tiene como intervalos de cantidad de producto vendida entre visita y visita, 143-106-60, siendo los óptimos de 103-103-103.

Por lo tanto se procede a lanzar el algoritmo con el concepto de regularidad aplicado a ventas.

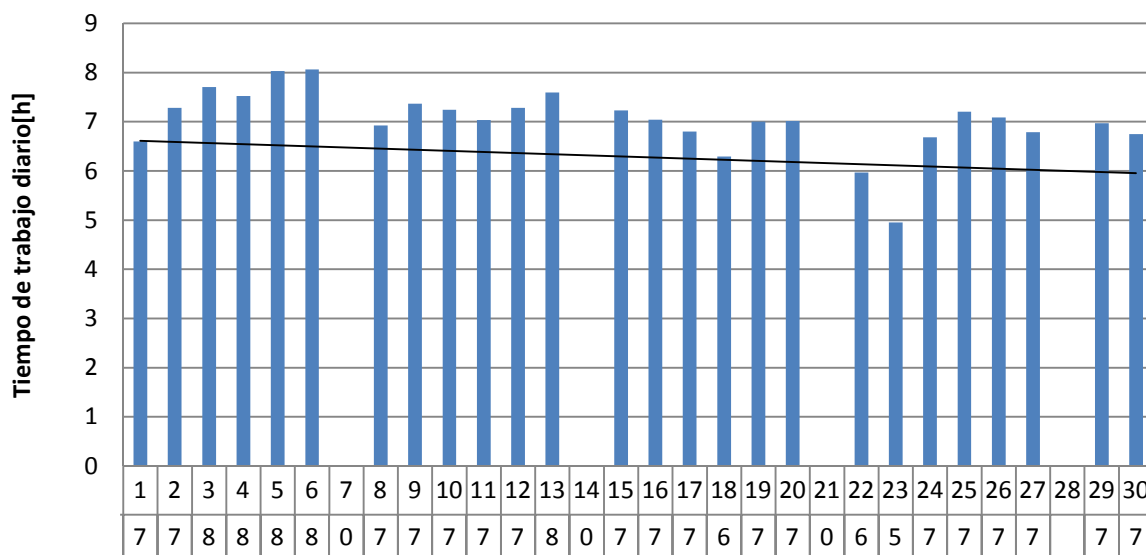


Fig. 28: Evolución del tiempo de trabajo con la ampliación

Para poder comparar mejor la solución con la demanda supuesta, se han insertado los huecos correspondientes a los domingos.

Al insertar la línea de tendencia, se puede ver que el tiempo de trabajo del mes, se adapta a la demanda de los centros comerciales; es decir, se visita menos a final del mes ya que hay menos venta. Siguiendo con el ejemplo del cliente1, las ventas entre visitas son 102-106-100.

Evaluando la función objetivo adaptada a la regularidad de las ventas, se llega a la conclusión que la solución se mejora alrededor de un 30%, pasando de un valor de 385.651 a 270.617 unidades.



9.2. Incremento de visitas

Otra aplicación posible es conocer cómo afecta al beneficio total, el incremento del número de visitas a un cliente. Esto es interesante, ya que se puede dar el caso de que al aumentar el número de visitas a un cliente, también se aumente la calidad del servicio y esto conlleve un aumento de las ventas.

De esta forma, se debe determinar si aporta un beneficio mayor que el coste de aumentar el número de kilómetros o la cantidad de horas necesarias de trabajo del comercial.

En este ejemplo, se ha estimado que el incremento de una visita supone un aumento de ingresos inversamente proporcional al número total de visitas, aumentar de una a dos visitas supone un 10% más, de dos a tres visitas un 7%, de tres a cuatro un 5%...

Se ha determinado que el coste por kilómetro extra es 0,3 um. También se ha valorado el coste del incremento de tiempo de trabajo en 15 um por hora, si se excede la jornada de trabajo y el incremento del valor regularidad (a más valor, menos regular es) en 0,5 um por unidad.

Los resultados obtenidos para el caso del apartado 8, con el tercer algoritmo sin visitas extra, son:

Regularidad	Kilómetros	Función objetivo	Horas totales
112,13	7.988,30	329.400,82	176,51



Aumentando una visita por cliente se obtiene la tabla 6.

Visita extra al cliente	Regularidad	Kilómetros	Función objetivo	Horas totales
-	112,13	7988,30	329400,82	176,51
1	134,46	8438,50	332999,18	177,03
2	298,13	8531,50	286525,92	180,13
3	370,13	7832,00	295022,67	180,03
4	246,13	7552,90	292883,12	179,85
5	88,13	8152,80	280883,07	177,53
6	236,80	7616,80	244648,40	179,57
7	392,13	7846,40	342729,87	179,72
9	536,13	8430,90	305622,12	181,76
10	121,93	7410,90	377002,12	179,08
11	76,46	8083,60	317565,13	179,23
12	120,80	7912,90	324296,45	180,13
15	214,13	8025,00	327419,17	179,80
16	112,13	7732,50	411672,91	183,03
17	194,13	7393,50	331603,41	177,90
18	300,13	7766,00	378989,66	177,68
19	104,80	7851,90	282465,95	179,68
20	262,13	7882,00	377447,66	179,51
22	134,13	8259,40	324736,36	180,43
23	114,80	8165,30	378122,65	181,92
24	426,13	8211,00	291212,16	179,95
25	540,13	8259,70	296136,51	179,90
26	506,13	7741,90	300977,62	180,12
28	216,13	8241,50	335227,41	182,15
32	412,13	7912,00	381062,66	180,55
33	124,80	8049,50	330664,75	179,18
34	310,13	7766,90	294890,12	178,60
35	338,13	7827,20	296320,27	179,05
36	334,13	7807,70	296110,52	179,18
37	310,13	7769,50	294891,42	178,55
38	441,93	8133,30	342863,32	176,86
39	338,13	7825,20	296319,27	179,10
40	278,13	7595,90	335304,62	180,50
41	322,13	7778,50	295495,91	178,55
43	110,66	8138,70	279602,68	181,02
44	338,13	7765,90	294289,62	179,12

Tabla6. Valores obtenidos con una visita extra



En la tabla 7 se puede ver la diferencia con la solución previa.

Código cliente	Nº Visitas inicial	Nº Visitas final	Incremento regularidad	Incremento km's	Increment. tiempo	Increment. ingresos	Diferencia con F.Obj
1	3	4	22,33	450,2	0,52	5%	3.598,37
2	1	2	186	543,2	3,62	10%	-42.874,90
3	1	2	258	-156,3	3,52	10%	-34.378,15
4	1	2	134	-435,4	3,34	10%	-36.517,70
5	1	2	-24	164,5	1,02	10%	-48.517,75
6	2	3	124,67	-371,5	3,06	7%	-84.752,42
7	1	2	280	-141,9	3,21	10%	13.329,05
9	1	2	424	442,6	5,25	10%	-23.778,70
10	4	5	9,8	-577,4	2,57	4%	47.601,30
11	3	4	-35,67	95,3	2,72	5%	-11.835,69
12	2	3	8,67	-75,4	3,62	7%	-5.104,37
15	1	2	102	36,7	3,29	10%	-1.981,65
16	1	2	0	-255,8	6,52	10%	82.272,09
17	1	2	82	-594,8	1,39	10%	2.202,59
18	1	2	188	-222,3	1,17	10%	49.588,84
19	2	3	-7,33	-136,4	3,17	7%	-46.934,87
20	1	2	150	-106,3	3	10%	48.046,84
22	1	2	22	271,1	3,92	10%	-4.664,46
23	1	2	2,67	177	5,41	10%	48.721,83
24	1	2	314	222,7	3,44	10%	-38.188,66
25	1	2	428	271,4	3,39	10%	-33.264,31
26	1	2	394	-246,4	3,61	10%	-28.423,20
28	1	2	104	253,2	5,64	10%	5.826,59
32	1	2	300	-76,3	4,04	10%	51.661,84
33	2	3	12,67	61,2	2,67	7%	1.263,93
34	1	2	198	-221,4	2,09	10%	-34.510,70
35	1	2	226	-161,1	2,54	10%	-33.080,55
36	1	2	222	-180,6	2,67	10%	-33.290,30
37	1	2	198	-218,8	2,04	10%	-34.509,40
38	4	5	329,8	145	0,35	4%	13.462,50
39	1	2	226	-163,1	2,59	10%	-33.081,55
40	1	2	166	-392,4	3,99	10%	5.903,80
41	1	2	210	-209,8	2,04	10%	-33.904,91
43	5	6	-1,47	150,4	4,51	3%	-49.798,13
44	1	2	226	-222,4	2,61	10%	-35.111,20

Tabla7. Diferencias con los valores de referencia



Comparando los datos obtenidos con la solución inicial se obtiene la tabla 8, con los costes e ingresos de las visitas extras.

Código cliente	Gastos (um)	Ingresos (um)	Beneficio (um)
1	190,01	50	-140,01
2	-118,49	100	218,49
3	-208,87	100	308,87
4	-378,7	100	478,7
5	-432,48	100	532,48
6	-850,8	66,67	917,47
7	278,81	100	-178,81
9	185,74	100	-85,74
10	346,29	40	-306,29
11	-66,8	50	116,8
12	-15,03	66,67	81,7
15	91,54	100	8,46
16	843,78	100	-743,78
17	-94,56	100	194,56
18	540,75	100	-440,75
19	-466,38	66,67	533,05
20	568,58	100	-468,58
22	104,49	100	-4,49
23	622,74	100	-522,74
24	-106,48	100	206,48
25	13,63	100	86,37
26	-107,06	100	207,06
28	270,83	100	-170,83
32	704,33	100	-604,33
33	77,43	66,67	-10,76
34	-281,18	100	381,18
35	-228,03	100	328,03
36	-236,03	100	336,03
37	-281,13	100	381,13
38	348,28	40	-308,28
39	-227,9	100	327,9
40	84,17	100	15,83
41	-266,39	100	366,39
43	-386,01	33,33	419,34
44	-265,74	100	365,74

Tabla8. Beneficio por visita extra a los diferentes clientes



Ordenando los clientes en función del beneficio en um, que aporta el hecho de crear una visita extra, se obtiene la figura 29.

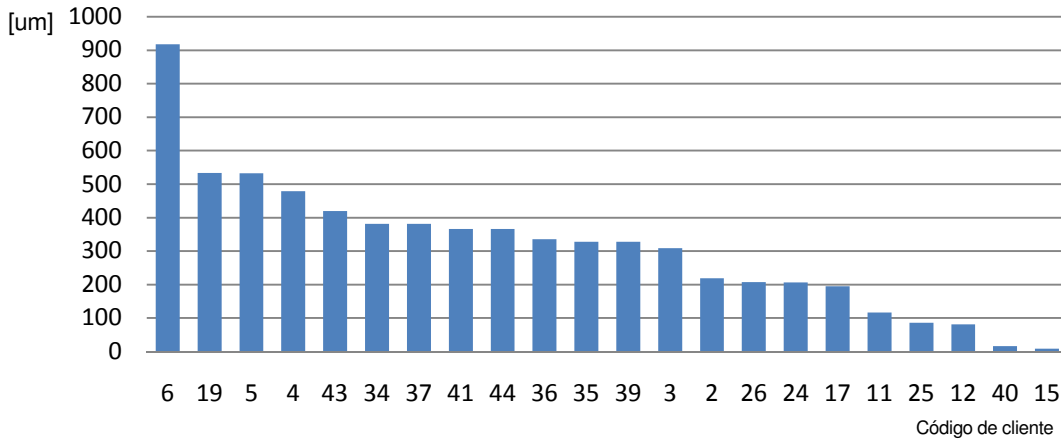


Fig. 29: Beneficio por visita extra (primera iteración)

Al ser el cliente 6 el que más beneficio aporta con una visita extra, se le asigna la tercera visita y se vuelve a repetir el proceso, siendo ahora los datos iniciales:

Regularidad	Kilómetros	Función objetivo	Horas totales
236,80	7616,80	244648,40	179,57

Una vez ejecutados todos los cálculos se obtienen los beneficios por visita extra, que se resumen en la figura 30.

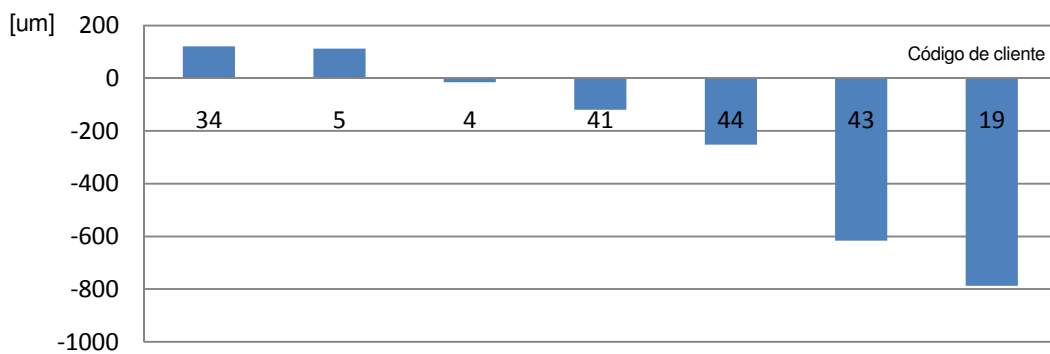


Fig. 30: Beneficio por visita extra (segunda iteración)



De esta forma se procede a introducir una visita extra al cliente 34.

Se vuelve a repetir el proceso pero, en este caso, sin obtener ningún cliente que con una visita extra obtenga un beneficio.

En resumen, se insertan las visitas extras a los cliente 6 y 34, obteniendo un beneficio de 1038,06 um; 917 um de la visita al cliente 6 y 120 de la visita al cliente 34.



Conclusiones

Las conclusiones que se extraen del presente proyecto son las siguientes:

Aplicar el proceso de optimización local y perturbación, para mejorar una solución inicial, es el método más adecuado para mejorar soluciones ya que permite salir de óptimos locales.

Al establecer restricciones *soft*, que añaden penalizaciones en la función objetivo cuando la solución contiene días que exceden la cantidad de horas de trabajo permitidas, se pueden encontrar soluciones iniciales que tengan valores de la función objetivo muy altos, teniendo jornadas de trabajo de más de ocho horas. Estas soluciones se corrigen en el proceso de optimización local, dando buenas soluciones en un plazo corto de tiempo.

Para solucionar problemas de la misma empresa en otras zonas es conveniente solucionar el problema con los tres algoritmos, para tener diferentes puntos de partida con los que elaborar el proceso de optimización local, y disponer así, de más opciones de llegar a un resultado mejor.

Debido a la aleatoriedad del proceso de optimización y si el tiempo de cálculo dispensable lo permite, se debería repetir un número elevado de veces la perturbación de la solución.

Se ha conseguido que los días tengan una carga de trabajo ajustada a los requisitos iniciales. En la representación gráfica de desplazamientos puede parecer que hay días con un desplazamiento mucho más corto que otros, pero es debido a que estos días cuentan con varias visitas a centros comerciales de una misma ciudad, o a tiempos de visita más largos.

La introducción del concepto de regularidad en esta clase de problemas amplía la calidad de éstos, ya que se adaptan más a la realidad.



Una vez solucionado el caso de estudio, se proponen dos ampliaciones. Con la regularidad aplicada a las ventas, se sitúa el problema dentro de otro marco de actuación, en el que se capacita al cliente a adaptarse a criterios como la demanda variable o la previsión de algún lanzamiento que modifique el volumen de ventas. Se ha comprobado que si se aplica bien este concepto, la solución se amolda a las necesidades reales.

Por otro lado, también se trabaja con posibles visitas extras. Se desarrolla una herramienta con la que se puede incrementar el beneficio obtenido si se conoce el incremento de ventas por una visita extra.



Costes del estudio efectuado

El proyecto se ha desarrollado individualmente en las instalaciones públicas de la escuela (ETSEIB). La evaluación económica del proyecto se basa en partidas tales como: el coste de material, la amortización, el valor residual y el tiempo de desarrollo.

Material:

Parámetro	Coste (€)	Coste sobre proyecto (€)
Ordenador: Acer 5920G	800	100
Impresora y material de impresión: HP 5940	90	35

Tabla9. Coste del material

Coste horario:

El proyecto se ha realizado en el periodo de tiempo comprendido entre febrero y junio, con una media de trabajo de 6 horas al día. Durante este tiempo se han realizado las actividades de recogida de información, diseño y estructuración del algoritmo, implementación del algoritmo en Visual Basic, experiencia computacional, análisis de resultados y escritura de la memoria del proyecto.

El trabajo se divide entre el tiempo dedicado por un ingeniero junior, que desarrolla el proyecto y el dedicado por un ingeniero sénior, que se dedica a dirigir y supervisar el trabajo. Partiendo de que un ingeniero junior cobra 35€ la hora y uno sénior 90€ por hora, se estima el siguiente coste relativo al tiempo de desarrollo.

$$6 \frac{\text{horas}}{\text{día}} \cdot 5 \frac{\text{días}}{\text{semana}} \cdot 4,2 \frac{\text{semanas}}{\text{mes}} \cdot 5 \text{ meses} = 630 \text{ horas}$$

$$630 \text{ horas Ing. junior} \cdot 35 \frac{\text{€}}{\text{hora}} = 22.050\text{€}$$

$$40 \text{ horas Ing. senior} \cdot 90 \frac{\text{€}}{\text{hora}} = 3.600\text{€}$$



En la tabla 10 se observan en detalle los costes del proyecto, aplicando los costes indirectos, el beneficio industrial, el IVA y el beneficio propio.

Total previo de costes	25.785 €
Costes indirectos (5%)	1.289 €
Subtotal 1	27.074 €
Impuestos Valor Añadido (IVA 16%)	4.332 €
Subtotal 2	31.406 €
Beneficio Industrial (13%)	4.083 €
Subtotal 3	35.489 €
Beneficio propio (3000€)	3.000 €
TOTAL	38.489 €

Tabla 10. Resumen de costes del proyecto

El coste total del proyecto es de 38.489 €.



Impacto ambiental del estudio

El uso de las herramientas desarrolladas para la empresa que comercializa dvd's, conlleva un ahorro de tiempo y de distancia recorrida por cada comercial. Estos ahorros se pueden extrapolar al resto de zonas en las que la empresa divide el territorio.

Se consigue una mejora de la eficiencia en los desplazamientos, reduciendo las emisiones que se producirían al hacer más kilómetros. Esta mejora se ve reflejada en diferentes aspectos. El primero de ellos es una reducción del combustible empleado por cada comercial, que supone el ahorro energético necesario para producirlo y transportarlo.

También se reducen las emisiones contaminantes, que afectan al problema del cambio climático, o a los problemas respiratorios.

Por último, también se ve reducida la contaminación ambiental sonora que producen los vehículos a su paso por las diferentes ciudades.

A modo de ejemplo, se procede a cuantificar una parte de este ahorro.

La empresa logra reducir la cantidad de kilómetros recorridos alrededor del 15%. Como todas las zonas se dividen en sectores más o menos iguales, la cantidad de kilómetros que debe realizar el comercial de cada zona es parecida. Por lo tanto, los resultados se pueden aplicar de la siguiente manera en las 9 zonas.

$$9 \text{ zonas optimizadas} \cdot 8500 \frac{\text{km de recorrido}}{\text{zona optimizada}} \cdot \frac{15 \text{ km ahorrados}}{100 \text{ km recorridos}} \cdot 0,152 \frac{\text{kg. CO}_2 \text{ emitido coche medio}}{\text{km recorrido}} = 1.744 \text{ kg CO}_2 \text{ no emitidos}$$

Una forma de dar la magnitud del ahorro que supone este proyecto es convertir los 1.744 kg de CO_2 a la cantidad de árboles necesarios para poder transformar esta contaminación en oxígeno.

La conclusión es que se necesitaría un bosque con 80 árboles adultos, con una relación de transformación de 0,022 toneladas por año y por árbol, que realizara el proceso de fotosíntesis durante el tiempo que los comerciales estuvieran trabajando.



Bibliografía

- APPLGATE, D. BIXBY, R. CHVÁTAL, V. COOK. *The Traveling Salesman Problem*, (2006).
- ANGELELLI, E. SPERANZA. MG., *The periodic vehicle routing problem with intermediate facilities. European Journal of Operational Research* Vol 137, 2002, p. 233-247.
- AZI, N. GENDREAU, M. POTVIN J., *An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. European Journal of Operational Research*. Vol 178, 2007 p.755–766.
- BERTAZZI, L., PALETTA, G., SPERANZA, M.G.,. *An improved heuristic for the period traveling salesman problem. Computers & Operations Research*. Vol 31, 2004, p.1215–1222.
- BRAYSY, O.,. *A reactive variable neighborhood search for the vehiclerouting problem with time windows. Journal on Computing* .Vol 15, 2003, p.347–368.
- CALVETE, H. GALÉ, C. OLIVEROS, MJ. SÁNCHEZ-VALVERDE, B. *A goal programming approach to vehicle routing problems with soft time Windows. European Journal of Operational Research*. Vol. 177, 2007, p.1720–1733.
- CAMPBELL AND SAVELSBERGH: *Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems, Transportation Science*. Vol. 38(3), p. 369–378.
- CHAO, I.M., GOLDEN, B.L., WASIL, E.A., *A new heuristic for the period traveling salesman problem. Networks*. Vol.26, 1995a, p.25–44.
- CHAO, I.M., GOLDEN, B.L., WASIL, E.A.,. *A new heuristic for the period traveling salesman problem. Computers & Operations Research*. Vol.22, 1995b, p.553–565
- CHRISTOFIDES, N., BEASLEY, J.E.,. *The period routing problem. Networks* 14, 1984, p.237–256.
- CORDEAU, J.F., GENDREAU, M., LAPORTE, G., *A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks*. Vol.30, 1997, p.105–119.



- COROMINAS, A., KUBIAK, W. AND MORENO, N., *Response time variability*, *Journal of Scheduling*, Vol. 10, 2007, p. 97-110.
- DUECK, G., *NEW OPTIMIZATION HEURISTICS: The great deluge algorithm and the record-to-record travel*. *Journal of Computational Physics*. Vol.104, 1993, p.86–92.
- FERRER, L. PASTOR, R. GARCÍA, A. *Designing salespeople's multiple routes with periodic constraints: a case study*. *IOC Research Institute*. UPC
- FRANCIS, P. SMILOWITZ, K. TZUR, M. *The Period Vehicle Routing Problem with Service Choice*, *Transportation Science*. Vol.40, 2006, p. 439–454.
- GARCÍA, A. PASTOR, R. *Solving the Response Time Variability Problem by means of a psychoclonal approach*. *Institute of Industrial and Control Engineering*.
- GARCÍA, A. COROMINAS, A. PASTOR, R. *Solving the Response Time Variability Problem by means of the Cross-Entropy Method*. *IJMTM, Production Line Systems: Concepts, Methods and Applications*, 2007.
- GENDREAU, M., HERTZ, A., LAPORTE, G.,. *New insertion and postoptimization procedures for the traveling salesman problem*. *Operations Research*. Vol.40, 1992. p.1086–1094.
- HANSEN, P., MLADENOVIC´, N., *Variable Neighborhood Search for the p-median*. *Location Science*. Vol.5, 1997. p.207–226.
- HEMMELMAYR, V.C. ET AL., *A variable neighborhood search heuristic for periodic routing problems*, *European Journal of Operational Research*. Vol.182, 2007, p.305-320.
- MLADENOVIC´, N. *A variable neighbourhood algorithm – a new metaheuristic for combinatorial optimization*. *Abstract of Papers presented at Optimization Days*. Vol.112, 1995.
- MLADENOVIC´, N., HANSEN, P., *Variable Neighborhood Search*. *Computers & Operations Research*. Vol.24, 1997, p.1097–1100.
- LAPORTE, G. GENDREAU, M. POTVIN, J. SERNET, F. *Classical and modern heuristics for the vehicle routing problem*. *European Journal of Operational Research*. Vol. 178, 2007, p.755–766.



- PALETTA, G., *A multiperiod traveling salesman problem: heuristic algorithms. Computers & Operations Research*. Vol.19, 1992, p.789–795.
- PALETTA, G., *The period traveling salesman problem: a new heuristic algorithms. Computers & Operations Research*. Vol.29, 2002, p.1343–1352.
- ROWAN, W., KIRKMAN, T., *Graph Theory 1736-1936*.
- SCRIVVER, A., *On the history of combinatorial optimization*, 2005.

