Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA. **Rect@**

Volumen 16 (2015), páginas 117 a 133.

EL PROBLEMA DEL VIAJANTE DE COMERCIO: BÚSQUEDA DE SOLUCIONES Y HERRAMIENTAS ASEQUIBLES

BERNAL GARCÍA, JUAN JESÚS
juanjesus.bernal@upct.es
Universidad Politécnica de Cartagena (UPCT) / Métodos Cuantitativos e Informáticos
C/Real 3. 30201 Cartagena

HONTORIA HERNÁNDEZ, ELOY
eloy.hontoria@upct.es
Universidad Politécnica de Cartagena / Organización de Empresas
Escuela Técnica Superior de Ingenieros Industriales. 30202 Cartagena

ALEKSOVSKI, DARKO
darko.aleksovski@ijs.si
Jožef Stefan Institute / Knowledge Technologies Department
1000 Ljubljana (Eslovenia)

Recibido (30/09/2015)
Revisado (22/11/2015)
Aceptado (03/12/2015)

RESUMEN: La elevada competitividad a nivel global ha reducido los márgenes de beneficios de la PYMEs y obligado a éstas a buscar nuevas herramientas de gestión. Por otra parte, sus reducidas infraestructuras en cuanto a recursos humanos y equipos en el área informática hacen que las aplicaciones de carácter sencillo y gratuito sean muy bien acogidas. Este trabajo de investigación explorará el carácter científico del Problema del Viajante de Comercio para su aplicación práctica con distancias y tiempos reales al entorno de las PYMEs. El desarrollo informático en programación abierta, se realizará mediante la herramienta Solver de *Excel* para hacerlo asequible a las citadas pequeñas y medianas empresas; investigándose su robustez respecto al dimensionamiento del problema.

Palabras claves: Problema del Viajante de Comercio, Problemas NP-Hard, Optimización Rutas, Solver de Excel

ABSTRACT: High levels of global competitiveness have reduced Small and Medium Size Enterprises´ (SMEs) profits´ margins and have forced them to search for new management tools. On the other hand, due to their reduced both human resources and computing structures make for them easy and free applications very wellcome. This research work will search the scientific side of the Travelling Salesman Problem (TSP) for its practical application with both real distances and times to the SMEs environment. The programming development through Solver by *Excel* will be shown in open source and its robustness to deal with the size of the problem dimension will be analysed.

Keywords: Travelling Salesman Problem, NP-Hard Problems, Routes Optimisation, Solver by Excel

1. Introducción y antecedentes

Según datos de la Dirección General de Industria y de la Pequeña y Mediana Empresa[1], a finales de 2013, el 99,9% del tejido productivo español son PYMEs (entre 0 y 249 trabajadores) y de ese porcentaje el 42,2% son micro-PYMEs (entre 1 y 9 trabajadores). Estas micro-PYMEs de manera generalizada, no cuentan con unas infraestructuras de tipo informático que les permita abordar el desarrollo de herramientas para sus necesidades, por ejemplo, para la búsqueda de una reducción del coste del transporte. Son estos costes de tipo logístico los que según un estudio del *Establish United Logistics Group* representaban en el año 2007 el 9,74% de las ventas en empresas americanas y el 8,39 % de las europeas y los que de alguna manera se deben intentar reducir todo lo posible.

Con el propósito de encontrar herramientas asequibles para su uso en Pymes, los autores han continuado con la investigación iniciada en la búsqueda de resolución de problemas de optimización de rutas de transporte con herramientas asequibles. En un primer artículo se analizó el tema del camino crítico mediante un modelo elaborado con la hoja de cálculo *Excel*, el complemento *Solver* y la programación de macros *VBA* [1][2].

En el presente trabajo, se continúa con un objetivo análogo, abordando el problema del viajante de comercio (*Travelling Salesman Problem*, TSP).

El problema del Viajante de Comercio

El TSP es uno de los problemas de optimización combinatoria más populares en la literatura científica. Propuesta inicialmente por G. Dantzig, R. Fulkerson and S. Johnson [2], consistente en determinar el circuito óptimo que tiene que hacer un Viajante que partiendo desde un origen debe visitar n ciudades una y solo una vez, volviendo al punto de partida.

El problema es fácil de formular, pero difícil de resolver. Aquí lo enunciaremos en dos fases:

Sea $C=(c_{ij})$ una matriz de costes, donde c_{ij}, $i, j = 1, \ldots, n$, representa el coste de ir de la ciudad i a la ciudad j. Se trata de resolver:

$$\text{Min } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}.x_{ij} \tag{1}$$

$$\text{sujeto a: } \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, 2, ..., n. \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, 2, ..., n. \tag{3}$$

$$x_{ij} \in \{0,1\}, \qquad i, j = 1, 2, ..., n. \tag{4}$$

donde $x_{ij} = 1$ significa que el viajero va desde ciudad i a ciudad j (x_{ij} = entero $i, j = 1,..., n$)

Las ecuaciones (1)-(4) definen el problema de asignación (de cada ciudad de salida se llega a una sola ciudad o que exista una salida desde la ciudad 0), pero no evita los subcircuitos. Para ello, se añaden la siguiente restricción que obligan a que solo un camino sea el que recorre todas las ciudades y donde u_i son variables ficticias que representan el paso de una ciudad a otra.

$$u_i - u_j + nx_{ij} = 1, \quad 2 \le i \ne j \le n, \tag{5}$$

$$u_i \ge 0, \quad 1 \le i \le n. \tag{6}$$

Las distancias d_{ij} entre las ciudades i y j se utilizan para clasificar el problema del Viajante:

---

[1] Dirección General de Industria y de la Pequeña y Mediana Empresa (http://www.ipyme.org). Fuente: INE, DIRCE 2014 (datos a 1 de enero de 2014).

[2] J.J. Bernal, E. Hontoria y D. Aleksovski. El problema del enrutamiento de vehículos. Propuestas para la búsqueda del camino más corto. Aplicaciones al entorno docente y Pymes. Rect@Monográfico nº4. (2013)

- Si $d_{ij} = d_{ji}$ entonces el problema es simétrico y en caso contrario es asimétrico.
- Si se cumple la desigualdad triangular o desigualdad de Minkowski ($d_{ik} \leq d_{ij} + d_{jk}$ para todo i , j y k) se dice que el problema es métrico.
- Si d_{ij} son distancias euclídeas en el plano, el problema es euclídeo (simétrico + métrico).

El TSP es considerado un problema del tipo *NP-Hard*, lo cual significa que el tiempo necesario para encontrar soluciones tiene una complejidad exponencial, es decir, no se conoce un algoritmo de tiempo polinomial que lo pueda resolver. Para el caso de n ciudades, y en un problema de tipo simétrico, existen (n-1)!/2 posibles circuitos, es decir, que para solo 20 ciudades serian 10^{18} circuitos.

El factor tiempo computacional es uno de los elementos claves en la resolución del TSP y son muchos los trabajos desarrollados para proporcionar soluciones eficientes respecto al mismo. Estas soluciones pueden obtenerse tanto por algoritmos de tipo exacto como por algoritmos aproximados (heurísticos) [3].

Para finalizar, comentar que la importancia del TSP radica en que no es un problema únicamente de tipo teórico, encontrándose aplicaciones de tipo práctico en empresas.

En este sentido, se han aplicado a:

- Perforación de placas de circuitos impresos en [4] para conectar un conductor en una capa con otro conductor en otra capa.
- Optimizar la ruta para la preparación de pedidos en un almacén logístico [5]
- Para conectar componentes en una placa base de ordenador donde se debe encontrar el camino más corto partiendo y terminando en un punto [6].
- Compañías de tipo logístico con problemas de reparto, rutas de autobuses, etc.

2. Diseño de una herramienta informática para la resolución del TSP

A la hora de buscar una herramienta al alcance las PYMEs, tanto por asequibilidad como por facilidad de uso, se sigue apostando por el programa *Excel* incluido en *Office* de *Microsoft*. Se trata por tanto, de programar mediante las fórmulas de Excel y su complemento *Solver* el anterior algoritmo de optimización del TSP; también elaborar una herramienta que mejore las prestaciones de herramientas análogas existentes, como las desarrolladas por C. Jiang [7] o M.C. Patterson and B. Harmel [8].

Para ello y en una primera fase, se establecieron una serie de requisitos al modelo a diseñar en aras del cumplimiento de las metas marcadas, debía cumplir los siguientes puntos:

- Tener una operativa sencilla e intuitiva.
- Ofrecer la posibilidad de redimensionado del tamaño del problema mediante la variación del número de ciudades a visitar.
- Contar con una programación transparente, mostrando todas las fórmulas. El objetivo de este punto es que el usuario pueda editar y modificar la programación del *Solver* y los macros *VBA* realizados, para su adaptación al problema concreto planteado en cada caso.
- Facilitar la toma de datos reales de la matriz de distancias entre ciudades, mostrando gráficamente en un mapa la ubicación de las mismas. En este aspecto, la aplicación debía incluir la posibilidad de impedir la conexión entre ciudades, bien por las condiciones reales de las carreteras que las unen, o bien por interés de la propia empresa. Finalmente se deberá permitir elegir la ciudad de partida del recorrido óptimo a efectuar.

Como colofón a los puntos anteriores, la herramienta debía presentar un entorno gráfico atractivo, mostrando la solución encontrada de una forma más visual mediante grafos.

Para cumplir con los anteriores condicionantes, se estableció el siguiente *pliego de condiciones*:

- Programar en VBA la herramienta, de forma que se iniciara sencillamente el proceso de resolución mediante el “clic” del icono correspondiente
- El modelo debía ser matricial y con rangos nombrados que no variasen con el redimensionado, lo que además haría más sencilla y comprensiva su formulación.

- La toma de datos debía realizarse de manera automática de una tabla obtenida de *Google Maps* mediante la programación necesaria. Este condicionante, además de facilitar la entrada de la información, permitiría trabajar con valores reales, incluso no simétricos.
- Para posibilitar la obtención de valores reales, mostrar la programación que permita la obtención de la matriz de tiempos y distancias. Se debe complementar dicha con otra de conexiones, que posibilite evitar el enlace directo entre dos ciudades. También como innovación, la herramienta creada debe permitir elegir la ciudad de inicio del recorrido.

Como se ha comentado con anterioridad, el TSP es un problema de optimización combinatoria de tipo *NP-hard*, lo cual significa que a medida que el número de ciudades aumenta, el tiempo computacional crece de manera exponencial.

Es por ello que, en la resolución de los objetivos anteriores, los autores quieren demostrar al mismo tiempo y de manera empírica la variación del tiempo de procesamiento en relación al número de ciudades. Los resultados nos mostrarán las limitaciones de la herramienta elaborada, dando opciones a otros desarrollos futuros de tipo heurístico que puedan hacer frente a problemas de elevada magnitud.

Por otro lado, existe la limitación de la herramienta Solver estándar incluida en Excel a un máximo de 200 variables y hasta 100 restricciones. Dichas limitaciones añaden a los objetivos anteriores dos temas nuevos a investigar:

- Determinar el tamaño matricial que implica dicho límite, o lo que es lo mismo, dimensionar el problema calculando el máximo número de ciudades del modelo construido.
- Las posibilidades de resolución de problemas más complejos mediante otras versiones mejoradas de Solver, o de otras herramientas existentes, tanto de pago como de software libre.

3. Herramienta programada

Como se ha comentado anteriormente, uno de los objetivos de este diseño es ofrecer una imagen visual sencilla que combine la simplicidad en su uso con una mejorada estética. El modelo-herramienta diseñado muestra el aspecto mostrado en la Figura 1, donde se pueden apreciar:

- Las matrices input del problema: Matriz de distancias (o tiempos) entre ciudades (***mdistan***).
- Matriz de conexiones existente entre ciudades (***mconex***).
- Matrices con las conexiones a realizar por la solución óptima: En forma de 0 y 1, donde 1 indica si la solución implica conexión entre dos ciudades. Los elementos de estas matrices forman parte de las celdas cambiantes del Solver (***mrutas***).
- La matriz necesaria para introducir la condición (5) del algoritmo (***mcondi***).
- Matriz que traduce las anteriores conexiones (una por fila y columna) de la solución en distancias (o tiempos), y cuya suma de filas o columnas nos proporciona la función objetivo (***fobje***) con la distancia (o tiempo) total de la solución óptima (***mrutas x mdistan***).
- Vector con las ciudades objeto del problema ***(vciu).***
- Vector con la ruta o recorrido (***vreco***) de la solución óptima encontrada. Este vector es otro elemento cambiante del Solver y expresa en forma de orden numérico la secuencia de ciudades (desde 1 a ***nciu***). A la derecha de este vector, figura su correspondiente traducción a los nombres de las ciudades a que pertenece dicha secuencia y los valores en distancias (o tiempos), así como el valor total del recorrido.

Finalmente, se incluye la posibilidad de elegir la ciudad inicial de la ruta, lo que reordena la solución encontrada, mostrando de forma más visual el grafo con la secuencia anterior y con los valores parciales y totales en kilómetros o en horas.

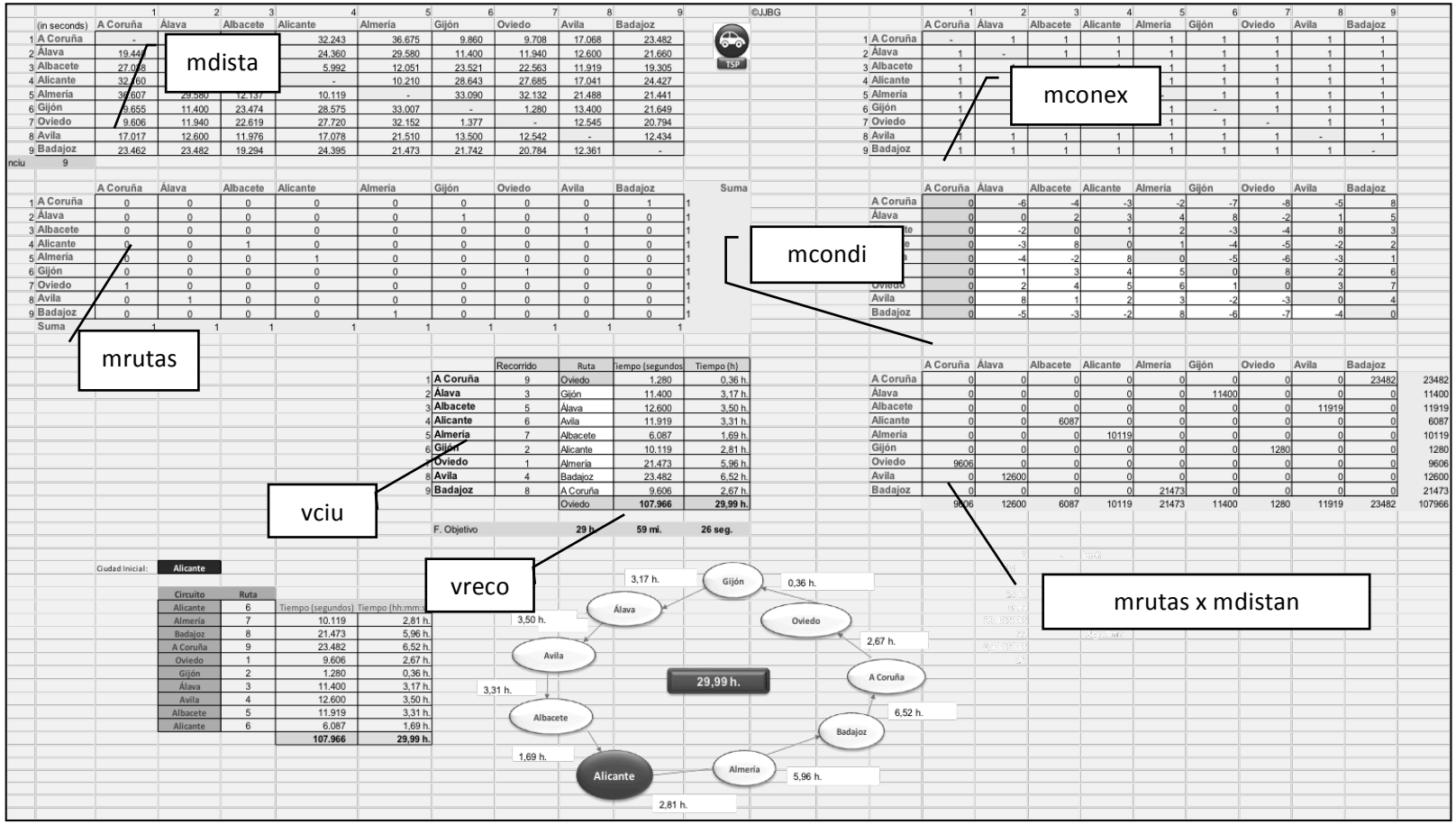


Figura 1: Aspecto general de la herramienta elaborada.

4. Testeo de la herramienta. Casos prácticos

Con la doble finalidad de testear la herramienta elaborada, y al mismo tiempo mostrar mediante casos prácticos su funcionamiento y utilidad, se presentan a continuación algunos problemas chequeados.

Comenzamos analizando un caso con 9 ciudades españolas. La Figura 2 muestra sus nombres y situaciones en el mapa (Véase anexo mapas).

A Coruña
Álava
Albacete
Alicante
Almería
Gijón
Oviedo
Avila
Badajoz

Figura 2: Las 9 ciudades del caso práctico y su ubicación en el mapa.

Como primer paso para comprobar la viabilidad de la herramienta, se debe obtener distancias y tiempos reales de recorrido entre las ciudades.

4.1. Obtención de las Matrices de Tiempos y Distancias. Búsqueda de soluciones con *Solver*.

Son varios los procedimientos para obtener las matrices de tiempos y distancias de esta API[3], pero en base a la filosofía general del trabajo consistente en desarrollar una herramienta sencilla de entender, los autores han optado por este camino para evitar el diseño de scripts que conllevan una mayor dificultad.

Es por ello, y a diferencia de otros artículos nuestros, por lo que se ha optado por la herramienta online proporcionada por la *Web Hurl* (https://www.hurl.it/). En esta web se pueden crear y enviar consultas a la API de Google Maps cuya URL es https://maps.googleapis.com/maps/api/distancematrix/xml.

Posteriormente, se añaden los parámetros que definen las ciudades que se precisa figuren en ambas matrices. Como ejemplo, y para una matriz de dos ciudades (2*2) se debe rellenar dos parámetros: *origins* y *destination* como muestra la Figura 3 de esta Web.

origins	Barcelona,Spain|Tarragona,Spain
destinations	Barcelona,Spain|Tarragona,Spain

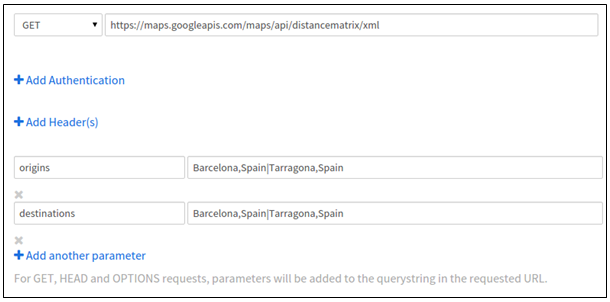


Figura 3: Obtención de las matrices tiempos y distancias.

[3] https://developers.google.com/maps/?hl=es.

En la aplicación se debe introducir el nombre de las ciudades o las coordenadas que definen las ciudades (pares de longitudes / latitudes), todas ellas separadas por una barra vertical "|" y como resultado se obtienen la información de tiempos y distancias en formato XML. Este archivo lo guardamos en el ordenador y lo importamos posteriormente a Excel para posteriormente refinar sus filas y columnas. Estas matrices de tiempos y distancias que se obtienen por este procedimiento son suministradas al algoritmo de optimización.

A partir de la matriz de distancias obtenida en el apartado anterior, se elabora la matriz de conexiones (en principio no se ha restringido ninguna), como nos presenta la Figura 4.

		1	2	3	4	5	6	7	8	9
	(en metros)	A Coruña	Alava	Albacete	Alicante	Almería	Gijón	Oviedo	Avila	Badajoz
1	A Coruña	-	603.423	851.784	1.018.468	1.136.430	281.659	286.157	519.048	655.570
2	Alava	603.423	-	604.726	761.621	906.875	322.264	339.087	379.387	654.123
3	Albacete	850.604	604.726	-	168.906	359.863	725.621	707.486	367.231	519.869
4	Alicante	1.017.856	761.621	168.218	-	295.709	892.873	874.739	534.483	687.121
5	Almería	1.132.671	906.875	355.428	290.611	-	1.007.689	989.554	649.298	612.076
6	Gijón	280.697	322.264	726.784	893.468	1.011.431	-	28.783	394.048	625.589
7	Oviedo	285.440	339.087	708.720	875.404	993.366	28.866	-	375.984	607.525
8	Avila	519.139	379.387	367.519	534.203	652.165	394.156	376.021	-	327.495
9	Badajoz	661.461	654.123	520.473	687.158	615.099	625.488	607.353	327.198	-
Nciu	9									

	A Coruña	Álava	Albacete	Alicante	Almería	Gijón	Oviedo	Avila	Badajoz
A Coruña	-	1	1	1	1	1	1	1	1
Álava	1	-	1	1	1	1	1	1	1
Albacete	1	1	-	1	1	1	1	1	1
Alicante	1	1	1	-	1	1	1	1	1
Almería	1	1	1	1	-	1	1	1	1
Gijón	1	1	1	1	1	-	1	1	1
Oviedo	1	1	1	1	1	1	-	1	1
Avila	1	1	1	1	1	1	1	-	1
Badajoz	1	1	1	1	1	1	1	1	-

Figura 4: Matriz de distancias y matriz de conexiones.

Las siguientes matrices a elaborar sirven para cumplir con las restricciones impuestas: Ecuación (5) del algoritmo y la que se produce al multiplicar la de distancias por la de conexiones de la solución, y que proporciona la función objetivo del problema, Figura 5. También es preciso preparar el vector con el recorrido óptimo pues de acuerdo a la solución óptima encontrada, este vector debe recoger el orden de visita de las ciudades.

	A Coruña	Álava	Albacete	Alicante	Almería	Gijón	Oviedo	Avila	Badajoz	Suma
A Coruña	0	0	0	0	0	0	0	0	1	1
Álava	0	0	0	0	0	1	0	0	0	1
Albacete	0	0	0	0	0	0	0	1	0	1
Alicante	0	0	1	0	0	0	0	0	0	1
Almería	0	0	0	1	0	0	0	0	0	1
Gijón	0	0	0	0	0	0	1	0	0	1
Oviedo	1	0	0	0	0	0	0	0	0	1
Avila	0	1	0	0	0	0	0	0	0	1
Badajoz	0	0	0	0	1	0	0	0	0	1
Suma	1	1	1	1	1	1	1	1	1	

Oviedo	0	2	4	5	6	1	0	3	7
Avila	0	8	1	2	3	-2	-3	0	4
Badajoz	0	-5	-3	-2	8	-6	-7	-4	0

	A Coruña	Álava	Albacete	Alicante	Almería	Gijón	Oviedo	Avila	Badajoz	
A Coruña	0	0	0	0	0	0	0	0	655570	655570
Álava	0	0	0	0	0	322264	0	0	0	322264
Albacete	0	0	0	0	0	0	0	367231	0	367231
Alicante	0	0	168218	0	0	0	0	0	0	168218
Almería	0	0	0	290611	0	0	0	0	0	290611
Gijón	0	0	0	0	0	0	28783	0	0	28783
Oviedo	285440	0	0	0	0	0	0	0	0	285440
Avila	0	379387	0	0	0	0	0	0	0	379387
Badajoz	0	0	0	0	615099	0	0	0	0	615099
	285440	379387	168218	290611	615099	322264	28783	367231	655570	3112603

	Recorrido
A Coruña	
Álava	
Albacete	
Alicante	
Almería	
Gijón	
Oviedo	
Avila	
Badajoz	

Figura 5: Matriz de las conexiones del recorrido (inicial=0).
Matriz para la restricción de los u_i, Matriz producto distancias x conexiones y Vector Recorrido.

A continuación, se rellenan las distintas casillas del cuadro de dialogo, y en el caso de que esté todo correcto, Solver informa que ha encontrado una solución. Véase Figura 6 y Anexo con la macro realizada.

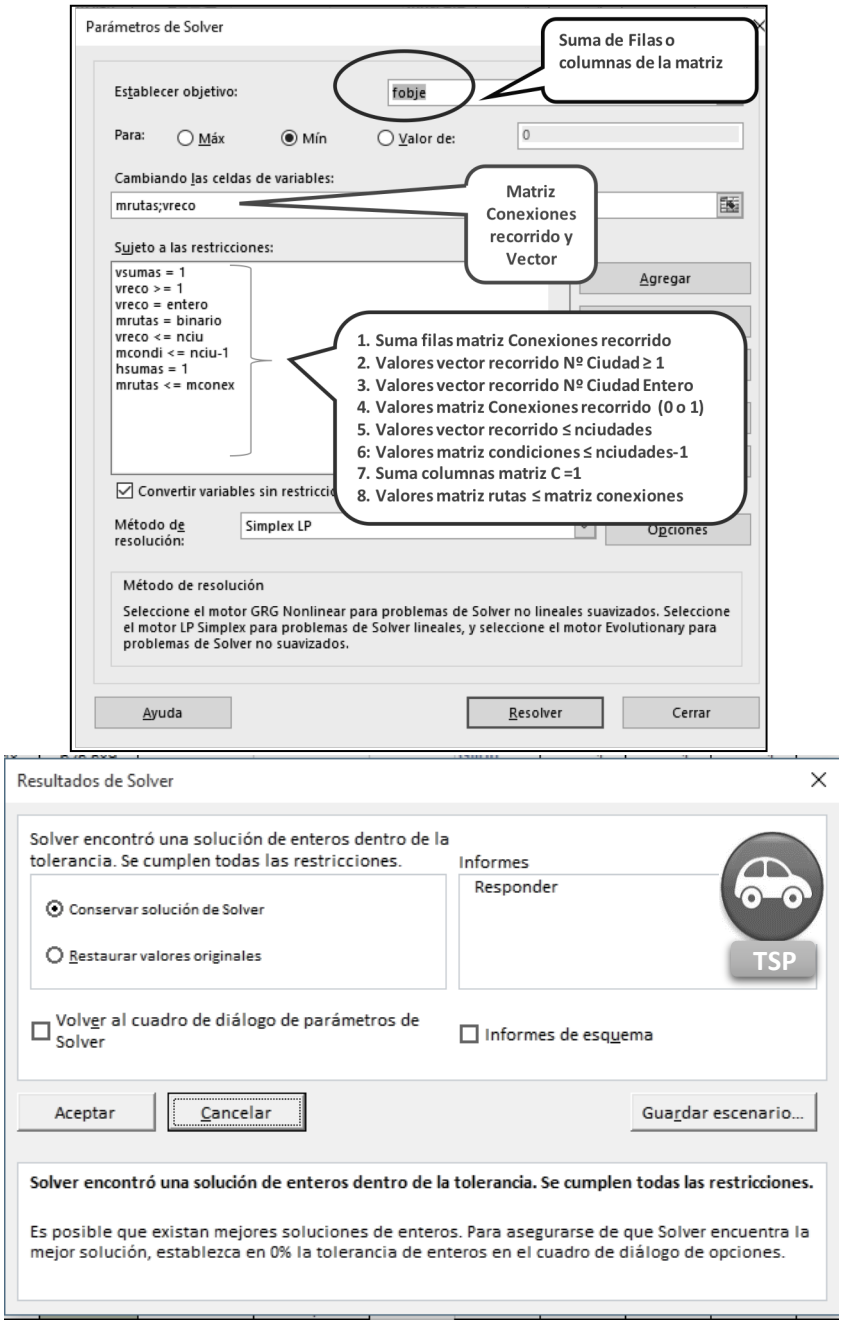


Figura 6: Explicación de los parámetros del Solver y Mensaje de solución encontrada.

El modelo recoge el recorrido óptimo, y mediante formulación, nos indica además cuales son las distancias entre ellas (en metros y Kms.) y el cómputo total del TSP. De acuerdo con nuestro pliego de condiciones, también se ha programado la solución para poder elegir, mediante un desplegable, la ciudad origen del recorrido, Figura 7.

	Recorrido	Ruta	Distancias (m)	Distancias (Km)
A Coruña	9	Oviedo	28.783	28,78 Km.
Álava	3	Gijón	322.264	322,26 Km.
Albacete	5	Alava	379.387	379,39 Km.
Alicante	6	Avila	367.231	367,23 Km.
Almería	7	Albacete	168.218	168,22 Km.
Gijón	2	Alicante	290.611	290,61 Km.
Oviedo	1	Almería	615.099	615,10 Km.
Avila	4	Badajoz	655.570	655,57 Km.
Badajoz	8	A Coruña	285.440	285,44 Km.
F. Objetivo		Oviedo	3.112.603	**3.112,60 Km.**

Ciudad Inicial:	Almería		
Circuito	Ruta	Distancias	
Almería	7		
Badajoz	8	615.099	615,10 Km.
A Coruña	9	655.570	655,57 Km.
Oviedo	1	285.440	285,44 Km.
Gijón	2	28.783	28,78 Km.
Alava	3	322.264	322,26 Km.
Avila	4	379.387	379,39 Km.
Albacete	5	367.231	367,23 Km.
Alicante	6	168.218	168,22 Km.
Almería	7	290.611	290,61 Km.
		3.112.603	**3.112,60 Km.**

Figura 7: Solución TSP de 9 ciudades y solución con elección de ciudad inicial.

Para que la solución encontrada sea lo más visual posible, se ha construido un circuito con grafos y distancias, así como su correspondiente representación en un mapa, Figura 8.

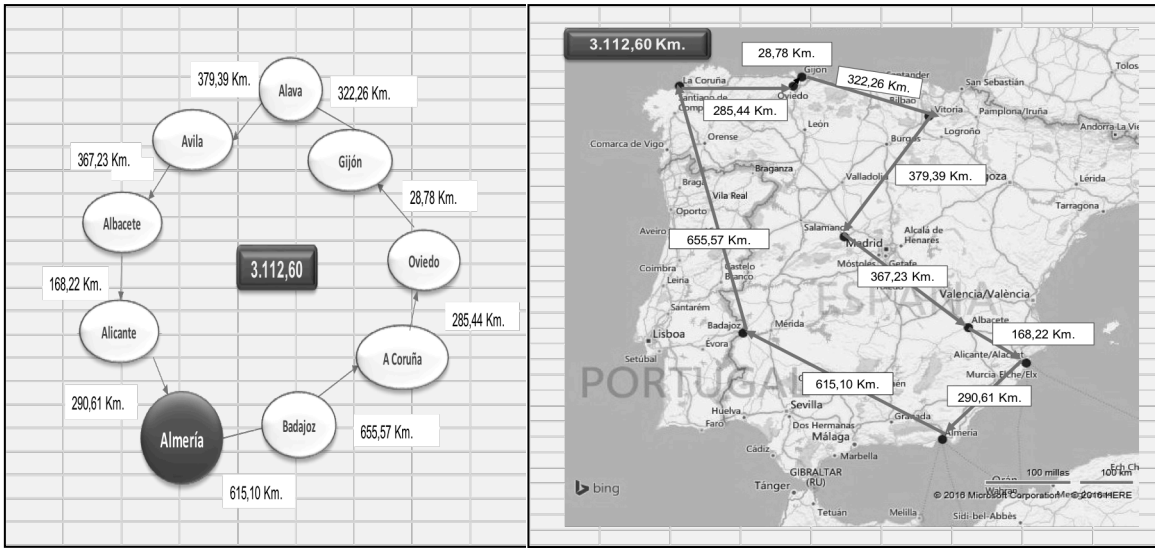


Figura 8: Solución en forma de grafo y representado sobre mapa.

A continuación, se realiza el mismo estudio para una matriz de tiempos, con la finalidad de comprobar si las soluciones encontradas eran coherentes al comparar distancias y tiempos, Figura 9.

(in seconds)	A Coruña	Álava	Albacete	Alicante	Almería	Gijón	Oviedo	Avila	Badajoz
A Coruña	-	19.440	27.141	32.243	36.675	9.860	9.708	17.068	23.482
Álava	19.440	-	19.320	24.360	29.580	11.400	11.940	12.600	21.660
Albacete	27.038	19.320	-	5.992	12.051	23.521	22.563	11.919	19.305
Alicante	32.160	24.360	6.087	-	10.210	28.643	27.685	17.041	24.427
Almería	36.607	29.580	12.137	10.119	-	33.090	32.132	21.488	21.441
Gijón	9.655	11.400	23.474	28.575	33.007	-	1.280	13.400	21.649
Oviedo	9.606	11.940	22.619	27.720	32.152	1.377	-	12.545	20.794
Avila	17.017	12.600	11.976	17.078	21.510	13.500	12.542	-	12.434
Badajoz	23.462	23.482	19.294	24.395	21.473	21.742	20.784	12.361	-

	Recorrido	Ruta	Tiempo (segundos	Tiempo (h)
A Coruña	9	Oviedo	1.280	0,36 h.
Álava	3	Gijón	11.400	3,17 h.
Albacete	5	Álava	12.600	3,50 h.
Alicante	6	Avila	11.919	3,31 h.
Almería	7	Albacete	6.087	1,69 h.
Gijón	2	Alicante	10.119	2,81 h.
Oviedo	1	Almería	21.473	5,96 h.
Avila	4	Badajoz	23.482	6,52 h.
Badajoz	8	A Coruña	9.606	2,67 h.
		Oviedo	**107.966**	**29,99 h.**
F. Objetivo		**29 h.**	**59 mi.**	**26 seg.**

Figura 9: Caso práctico con tiempos: matriz en segundos y solución TSP.

Para continuar con el proceso de testeo de la herramienta, se quiso comprobar el funcionamiento de la matriz de conexiones anulándose la unión entre dos ciudades comunicadas por la solución anterior (Álava y Albacete). En este punto, el modelo respondió eligiendo otra ruta que evitase la conexión citada a costa de incrementar el tiempo óptimo de recorrido que pasó de 29,99 a 30,22 horas, Figura 10.

(in seconds)	A Coruña	Álava	Albacete	Alicante	Almería	Gijón	Oviedo	Avila	Badajoz
A Coruña	-	19.440	27.141	32.243	36.675	9.860	9.708	17.068	23.482
Álava	19.440	-	19.320	24.360	29.580	11.400	11.940	12.600	21.660
Albacete	27.038	19.320	-	5.992	12.051	23.521	22.563	-	19.305
Alicante	32.160	24.360	6.087	-	10.210	28.643	27.685	17.041	24.427
Almería	36.607	29.580	12.137	10.119	-	33.090	32.132	21.488	21.441
Gijón	9.655	11.400	23.474	28.575	33.007	-	1.280	13.400	21.649
Oviedo	9.606	11.940	22.619	27.720	32.152	1.377	-	12.545	20.794
Avila	17.017	12.600	-	17.078	21.510	13.500	12.542	-	12.434
Badajoz	23.462	23.482	19.294	24.395	21.473	21.742	20.784	12.361	-

	1	2	3	4	5	6	7	8	9
	A Coruña	Álava	Albacete	Alicante	Almería	Gijón	Oviedo	Avila	Badajoz
A Coruña	-	1	1	1	1	1	1	1	1
Álava	1	-	1	1	1	1	1	1	1
Albacete	1	1	-	1	1	1	1	-	1
Alicante	1	1	1	-	1	1	1	1	1
Almería	1	1	1	1	-	1	1	1	1
Gijón	1	1	1	1	1	-	1	1	1
Oviedo	1	1	1	1	1	1	-	1	1
Avila	1	1	-	1	1	1	1	-	1
Badajoz	1	1	1	1	1	1	1	1	-

	Recorrido	Ruta	Tiempo (segundos	Tiempo (h)
A Coruña	9	Oviedo	1.280	0,36 h.
Álava	3	Gijón	11.400	3,17 h.
Albacete	4	Álava	19.320	5,37 h.
Alicante	5	Albacete	6.087	1,69 h.
Almería	6	Alicante	10.119	2,81 h.
Gijón	2	Almería	21.473	5,96 h.
Oviedo	1	Badajoz	12.434	3,45 h.
Avila	8	Avila	17.068	4,74 h.
Badajoz	7	A Coruña	9.606	2,67 h.
		Oviedo	**108.787**	**30,22 h.**
F. Objetivo		**30 h.**	**13 mi.**	**7 seg.**

Figura 10: Nueva solución del TSP al anular conexiones entre 2 ciudades

5. Dimensionamiento máximo del problema. Alternativas al *Solver* incluido en *Excel*

Para conocer el máximo número de ciudades a las que *Solver* es capaz de hacer frente, se establecieron varias pruebas, variando el número de ciudades a visitar y midiendo el tiempo computacional necesario para encontrar la solución.

El resultado del proceso de investigación concluyó que la limitación del *Solver* estándar estaba en un máximo de nueve ciudades para la herramienta diseñada. Al aumentar la dimensión del problema a tratar, nos encontrábamos con el mensaje de error que muestra la Figura 11.

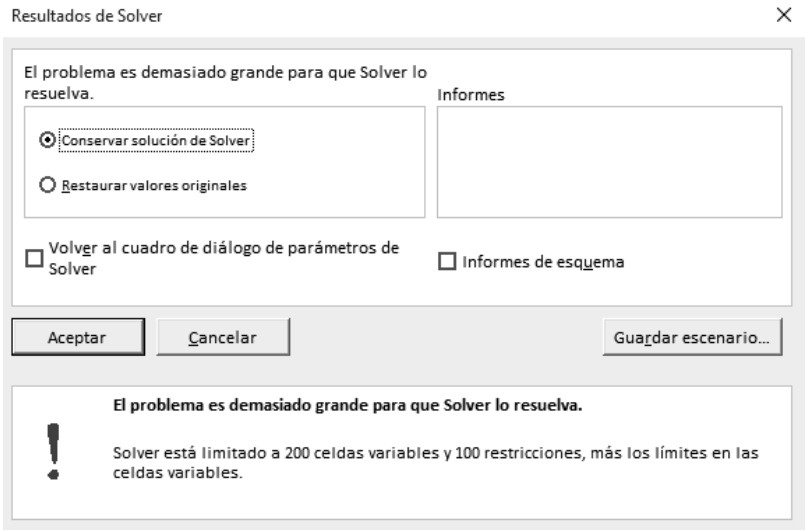


Figura 11: Mensaje de Solver para indicar que el problema excede su capacidad

Para afrontar problemas TSP de mayor dimensión, decidimos optar por una herramienta de software libre, el *Open Solver*[4] y otra de pago denominada *Premium Solver* [5] Tras descargar la versión gratuita de *Open Solver*, y una versión trial (por un mes) de *Premium Solver* (ver anexo), procedimos a analizar problemas de mayor dimensión, concretamente desde 12 hasta 50 ciudades.

Para el primer caso el test fue totalmente satisfactorio, con unos tiempos de computación razonables, pero no así para la matriz de 50x50, incluso para una menor de 21 ciudades. Se procedió entonces a ir incrementando la de 12 en más ciudades y el resultado fue que hasta 16, los tiempos, aunque aumentaban, eran asumibles. No sucedía lo mismo para problemas de mayor dimensión, y esto suponía un inconveniente si la herramienta tal, y como se estipuló, debería servir para una Pyme.

Ciudad	Recorrido	Ruta	Tiempo (s)	Tiempo (h)
A Coruña	16	Avila	8838	2,46 h.
Álava	13	Cáceres	4293	1,19 h.
Albacete	7	Badajoz	12189	3,39 h.
Alicante	8	Cadiz	17070	4,74 h.
Almería	5	Almería	6644	1,85 h.
Gijón	14	Cartagena	6643	1,85 h.
Oviedo	15	Albacete	6087	1,69 h.
Avila	1	Alicante	1959	0,54 h.
Badajoz	3	Benidorm	16207	4,50 h.
Barcelona	10	Barcelona	18921	5,26 h.
Benidorm	9	Burgos	5328	1,48 h.
Bilbao	12	Bilbao	2880	0,80 h.
Burgos	11	Álava	11400	3,17 h.
Cáceres	2	Gijón	1377	0,38 h.
Cadiz	4	Oviedo	9708	2,70 h.
Cartagena	6	A Coruña	17017	4,73 h.
		Avila	146561	40,71 h.

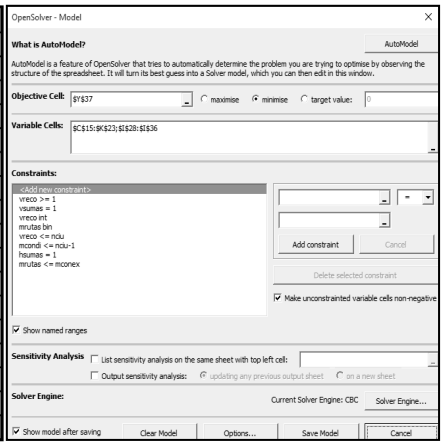


Figura 12: Solución de Open Solver para 16 ciudades. Model con constraints

[4] http://opensolver.org/

[5] http://www.solver.com/premium-solver-pro

Siguiendo con la investigación de los efectos de aumento del número de ciudades en las prestaciones del *Open Solver* y *Premium Solver*, se estudió un caso práctico entre 16 ciudades españolas. Ambos programas encontraron soluciones parecidas, en torno a las 40 horas, Figura 12.

La Figura 13 muestra la evolución de los tiempos de computo en ambos casos (cálculos realizados con un procesador Core i3 a 3,40 Ghz).

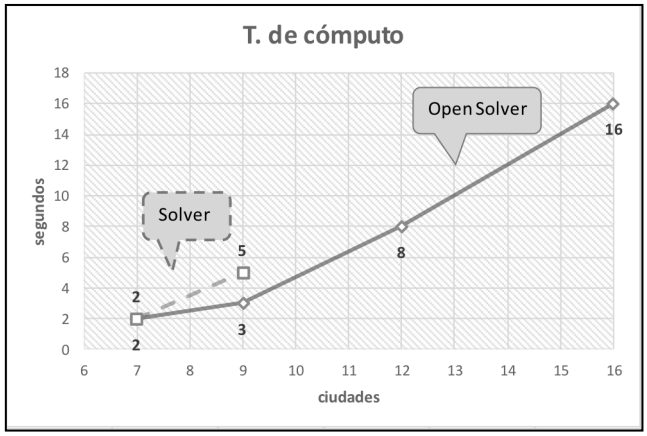


Figura 13: Tiempos de cómputo Solver vs. Open Solver

Para chequear la robustez de las soluciones anteriores, los resultados se contrastaron con los de la herramienta *OR tools*, que para 9 ciudades mostró exactamente la misma ruta y los mismos kilómetros que los calculados anteriormente por el Solver estándar incluido en *Excel* (3.112 Kms, véase Figura 7). Sin embargo, para 16 ciudades*, OR Tools* obtuvo diferentes soluciones a las calculadas por *Open Solver* y por *Premium Solver*. El tiempo necesario calculado por esta herramienta para recorrer la ruta fue de 143.915 segundos con una diferencia de 2.646 segundos (44 minutos), así como una secuencia de ciudades distinta:

AVILA- CACERES-BADAJOZ- CADIZ- ALMERIA- CARTAGENA- ALBACETE- ALICANTE- BENIDORM- BARCELONA- ALAVA- BILBAO- BURGOS- OVIEDO- GIJON- A CORUÑA- AVILA

Así pues, la ruta propuesta por esta herramienta varía respecto a la mejor solución encontrada por *Open Solver,* en la siguiente parte del recorrido, Figura 14.

Figura 14: Diferencias solución OR Tools y Open Solver

No obstante, a pesar de la mejora de la solución de *OR tools*, debido a que es un lenguaje de programación muy potente, proyectado mediante "Programación por restricciones" (constraint programming, CP), que es un paradigma de la programación en informática donde las relaciones entre variables son expresadas en términos de restricciones, al considerarlo demasiado complejo para una PYME, seguimos optando por la segunda herramienta para este tipo de empresas.

6. Conclusiones

El resultado de la investigación realizada, ha reflejado la viabilidad de la herramienta diseñada para la planificación de rutas en una pequeña empresa. Las soluciones encontradas prácticamente en tiempo real, son soluciones realistas no solo en distancia sino también en tiempo total de recorrido. Para problemas de

una mayor dimensión, se puede recurrir a la versión gratuita del *Open Solver* que es suficientemente potente para abordar hasta un máximo de 16 ciudades. La versión comercial de *Premium Solver* tiene como inconveniente además de ser de pago, el no tener una potencia computacional muy superior a la versión *Open Solver*. *OR Tools*, como ya hemos comentado, tiene el problema de una programación demasiado compleja para un entorno sencillo de una PYME.

Por tanto, aconsejamos la utilización de Excel hasta 9 ciudades y *Open Solver* hasta 16 ciudades. No obstante, aunque los resultados encontrados en cualquiera de las versiones de Solver han mostrado la limitada potencia para encontrar soluciones exactas en un tiempo computacional razonable en un ordenador estándar, debido a las características de la empresa y de su problemática a tratar, es posible que necesiten buscar soluciones para un mayor número de ciudades en un menor intervalo de tiempo. Es en estos casos en los que se precisan soluciones que sin ser las óptimas, son suficientemente buenas para ser consideradas y que además cumplan con la condición de que se obtengan rápidamente. Por ello, el equipo de investigación está desarrollando una nueva herramienta basada en algoritmos heurísticos y que permitan encontrar soluciones en un tiempo máximo fijado por las necesidades empresariales.

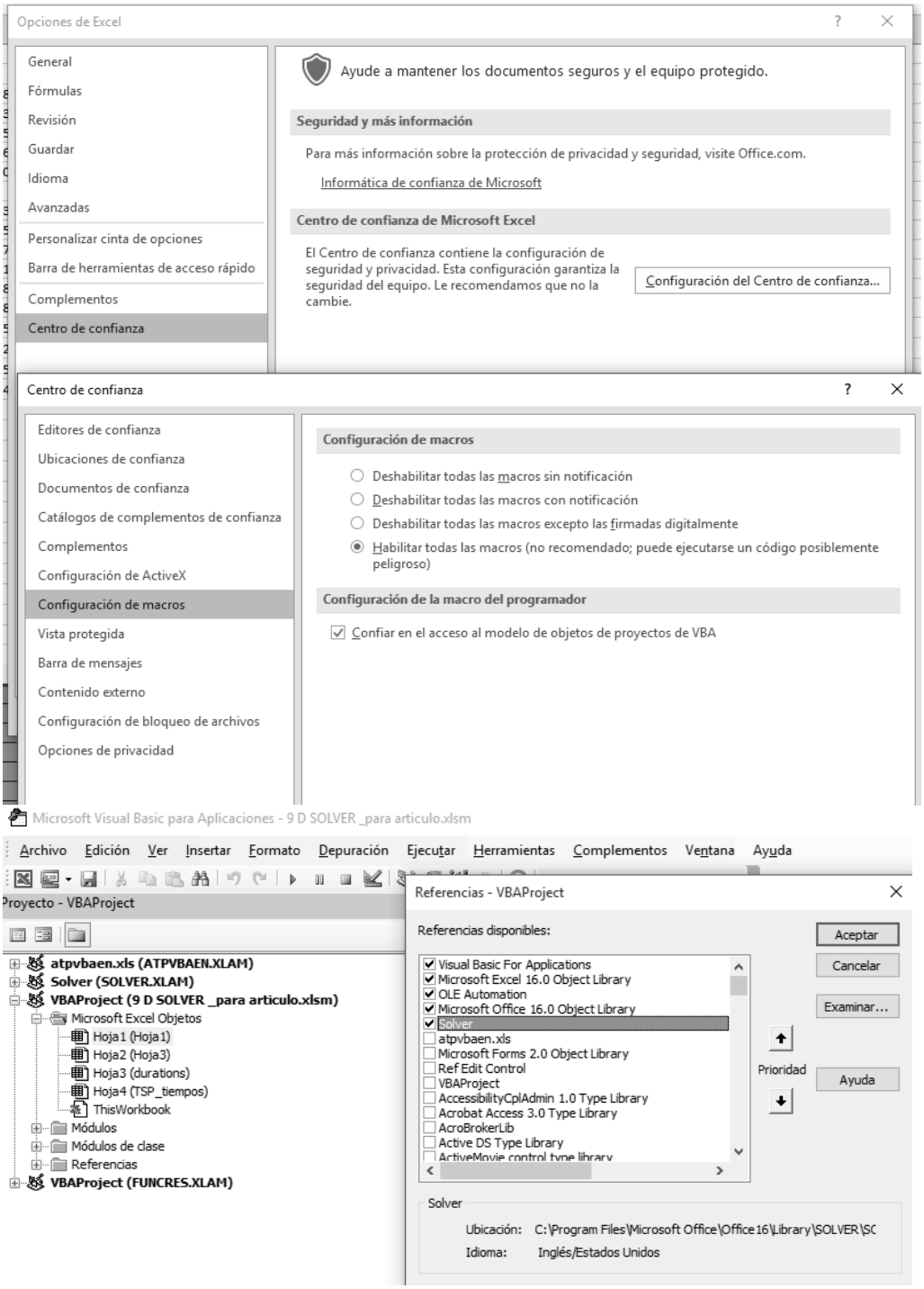
Referencias bibliográficas

1. J. J. Bernal, E. Hontoria, D. Aleksovski. El problema del enrutamiento de vehículos. Propuestas para la búsqueda del camino más corto. Aplicaciones al entorno docente y Pymes, *Rect@Monográfico nº4.* (2013) 25-38.
2. G. Dantzig, R. Fulkerson, S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, **2(4)** (1954) 393–410.
3. G. Laporte, The Travelling Salesman Problem: An overview of exact and approximate algorithms, *European Journal of Operational Research* **59** (1992) 231–247.
4. H. D. Ratliff, A.S. Rosenthal. Order-picking in a rectangular warehouse: a solvable case for the travelling salesman problem. *Operations Research,* **31** (1983) 507–552
5. M. Grötschel, M. Jünger & G. Reinelt. Optimal control of plotting and drilling machines: A case study. *Mathematical Methods of Operations Research*, **35, No. 1**, (1991) 61-84.
6. J. K. Lenstra, A.H.G. Rinnooy. Some simple applications of the traveling salesman problem. *Operational Research Quarterly*, **26** (1975) 717–33.
7. C. Jiang. A Reliable solver of euclidean traveling salesman problems with Microsoft Excel Add-in tools for small-size systems. *Journal Of Software*, **5 (7)** (2010) 761-768.
8. M .C. Patterson, B. Harmel. An algorithm for using Excel Solver for the Traveling Salesman Problems, *Journal of Education for Business*, **78 (6)** (2003) 341-346.

ANEXOS

A. Macros

Para poder trabajar con macros de *VBA* en *Excel* es preciso en primer lugar activarlos una primera vez: *Archivo/Opciones/Centro de confianza/Configuración del centro de confianza/Configuración de macros/Habilitar…*, se almacena el libro, se cierra el *Excel*, lo vuelves a abrir, y ya funcionan los macros. Seguidamente debemos activar en el programa *VBA* (con F11 de *Excel*) la *Referencia VBA Proyect.* Entonces ya podemos programas nuestra macro que ejecute el *Solver,* Figura 15.



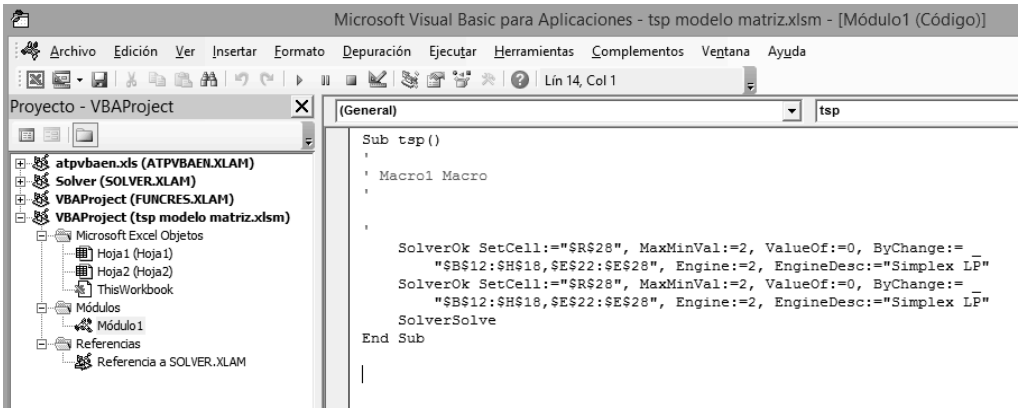


Figura 15: Activación de macros. Referencia VBA Proyect y macro de ejecución del Solver

B. Mapas

A la hora de presentar las ciudades sobre un mapa, disponemos de diversas opciones de las que hemos probado dos, la del complemento *Excel Map*, incluido en la versión profesional de *Excel*, en 3D en la de 2016, Figura 16.

Figura 16: Excel Map 3D

Otra alternativa, más accesible, es insertar en *Excel*, de la Tienda de Complementos de *Microsoft* la App gratuita, *Mapas de Bing*, que permite la presentación de datos en vista carretera o área, Figura 17. Por su condición de facilidad de uso y gratuidad, nos decantamos por esta opción, a la hora de complementar las herramientas recomendados anteriormente de *Excel* y *Open Solver*.

Figura 17: APP Mapas Bing

C. Open Solver y Premium Solver

Se muestra el aspecto del menú de *Open Solver* y como remarca las matrices del problema (Figura 18), y el modelo y cálculo de iteraciones en *Premium Solver,* Figura 18.

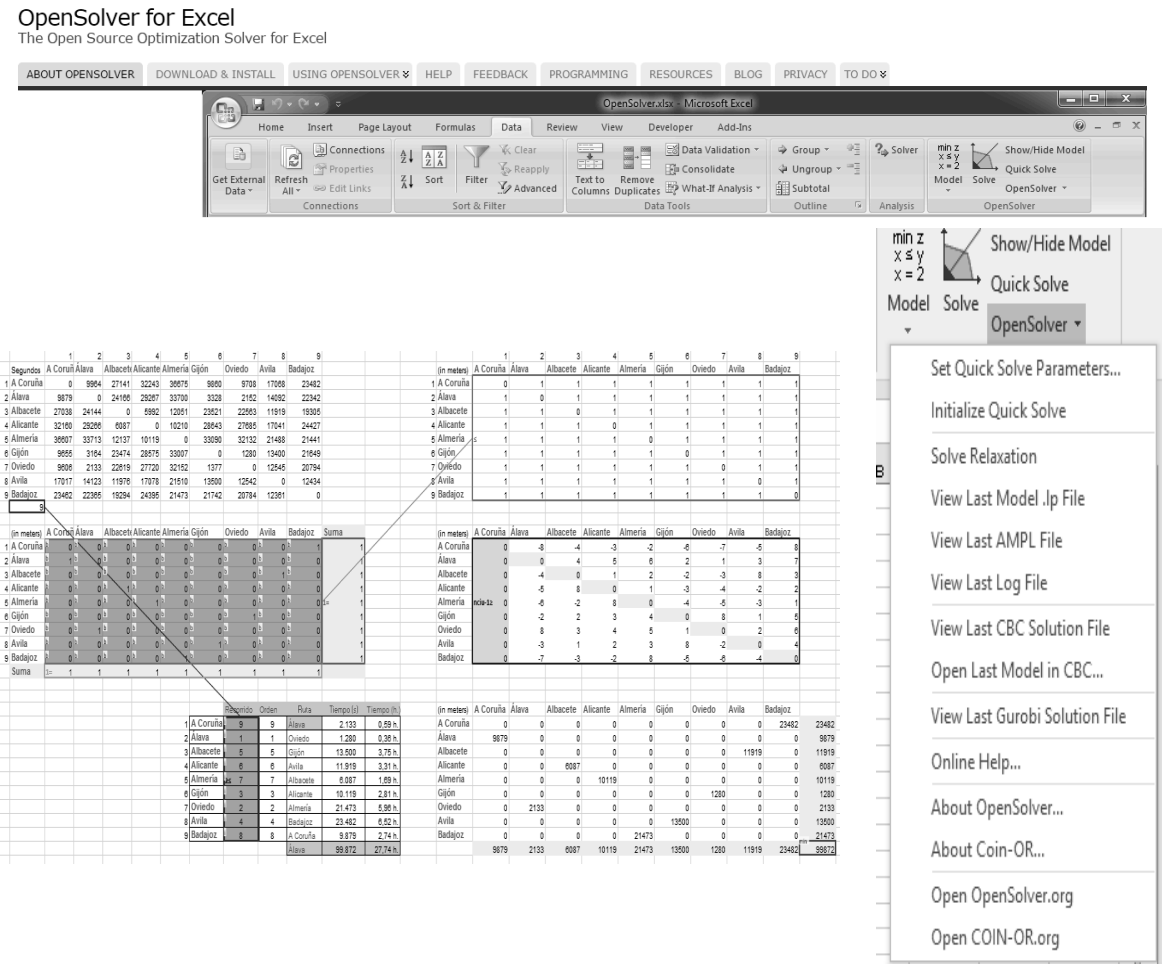
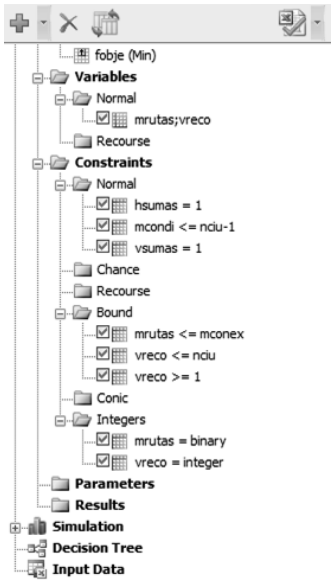


Figura 18: Open Solver



Best Integer Objective	
Current Objective	42
Nodes	
Iterations	1

Figura 19: Premium Solver. Modelo e iteraciones