

Logística y Servicios

Sesión de práctica de laboratorio 1, TSP

En esta sesión, haremos un modelo para el problema del viajante de comercio y lo resolveremos con Excel y con el solver en la nube NEOS.

Problema:

La empresa QUIROTEC vende equipamiento médico de alta tecnología a hospitales, principalmente equipamiento electrónico para quirófanos. Los beneficios de la empresa dependen de una política eficiente de ventas, en la cual sus comerciales visitan los hospitales más importantes para conseguir contratos de suministro con las distintas administraciones locales. El problema es que las ciudades a visitar son muchas y están geográficamente dispersas. El comercial se pasa mucho tiempo viajando, e incurre en altos costes de dietas y kilometrajes. La empresa desea diseñar rutas para visitar los hospitales de manera que se minimice la distancia total de viaje, lo que se traduce en una reducción de costes. En particular, se planea visitar 20 ciudades. Las ciudades a visitar y las distancias, en Km. entre ellas, se muestran en la siguiente tabla:

	A Coruña	Albacete	Alicante	Badajoz	Barcelona	Burgos	Castellón	Córdoba	León	Lleida	Madrid	Málaga	Ourense	Pamplona	Salamanca	Santander	Sevilla	Valencia	Vigo	Zaragoza
A Coruña	0	806	969	615	1029	461	896	823	286	876	564	971	144	657	426	435	825	908	141	737
Albacete	806	0	163	504	497	461	228	324	562	451	240	416	722	519	438	610	460	167	796	379
Alicante	969	163	0	651	494	624	225	442	725	455	403	452	885	601	601	773	569	158	959	456
Badajoz	615	504	651	0	974	528	710	256	490	830	381	369	488	727	295	646	207	649	478	687
Barcelona	1029	497	494	974	0	577	269	818	755	155	590	894	964	422	757	651	954	336	1039	296
Burgos	461	461	624	528	577	0	470	589	180	424	231	713	403	202	234	147	697	489	477	285
Castellón	896	228	225	710	269	470	0	549	629	230	395	625	831	422	598	620	686	67	906	252
Córdoba	823	324	442	256	818	589	549	0	630	747	358	155	678	718	436	738	136	488	723	633
León	286	562	725	490	755	180	629	630	0	602	320	778	235	381	194	229	659	642	323	463
Lleida	876	451	455	830	155	424	230	747	602	0	446	845	811	269	604	498	884	297	886	142
Madrid	564	240	403	381	590	231	395	358	320	446	0	482	480	371	200	380	488	341	555	303
Málaga	971	416	452	369	894	713	625	155	778	845	482	0	827	827	584	863	191	558	850	732
Ourense	144	722	885	488	964	403	831	678	235	811	480	827	0	604	305	445	680	823	87	672
Pamplona	657	519	601	727	422	202	422	718	381	269	371	827	604	0	433	229	854	452	678	171
Salamanca	426	438	601	295	757	234	598	436	194	604	200	584	305	433	0	352	465	544	380	465
Santander	435	610	773	646	651	147	620	738	229	498	380	863	445	229	352	0	815	639	528	371
Sevilla	825	460	569	207	954	697	686	136	659	884	488	191	680	854	465	815	0	625	689	770
Valencia	908	167	158	649	336	489	67	488	642	297	341	558	823	452	544	639	625	0	898	306
Vigo	141	796	959	478	1039	477	906	723	323	886	555	850	87	678	380	528	689	898	0	747
Zaragoza	737	379	456	687	296	285	252	633	463	142	303	732	672	171	465	371	770	306	747	0

Sigue esta guía para contestar a las preguntas de la 1 a la 5 siguientes:

1. A partir de los modelos para el problema del viajante de comercio (TSP) de la unidad 4, ¿podrías decir cuántas variables binarias y continuas tendrá el modelo?
2. ¿Cuántas soluciones diferentes tendrá este problema de TSP? ¿Cuánto tiempo necesitaría un ordenador para procesar todas estas soluciones, si cada solución requiere solamente 1 milisegundo para procesarse?
3. Dado que no es posible obtener la solución óptima de forma sencilla, abre el fichero de Excel que acompaña esta guía e intenta obtener una solución “a mano” tal y como se hace en muchas empresas. Intenta conseguir una solución donde la distancia total viajada sea inferior a 7.000 Km. Escribe el orden en el que se visitan las ciudades y compara el resultado obtenido con el de tus compañeros.
4. Resuelve el modelo usando el solver Evolutivo de Excel. Escribe el orden en el que se visitan las ciudades y compara el resultado obtenido con el del punto anterior.
5. Usando el fichero MPS proporcionado junto con la guía, intenta obtener la solución óptima usando el solver NEOS en la nube. ¿Cómo se compara esta solución con las obtenidas previamente?

1. Tal y como se vio en la unidad 4, el modelo del TSP es el siguiente:

Variables

- $X_{ij} \in \{0, 1\}$, toma el valor 1 si el arco (i, j) está en la ruta
- $U_i \geq 0$ define la posición del punto i en la ruta (número de puntos visitados antes de i)

$$\min \sum_{(i,j) \in A} c_{ij} X_{ij} \quad (1)$$

$$\text{s.a.: } \sum_{i \neq j} X_{ij} = 1, \quad j = 0, \dots, n \quad (2)$$

$$\sum_{j \neq i} X_{ij} = 1, \quad i = 0, \dots, n \quad (3)$$

$$U_i - U_j + nX_{ij} \leq n - 1, \quad 1 \leq i \neq j \leq n \quad (4)$$

Tenemos tantas variables X como arcos, pero no tenemos un arco desde cada ciudad consigo mismo. Como tenemos 20 ciudades, tenemos 20 ciudades, tendremos $20 \cdot 19 = 380$ variables binarias.

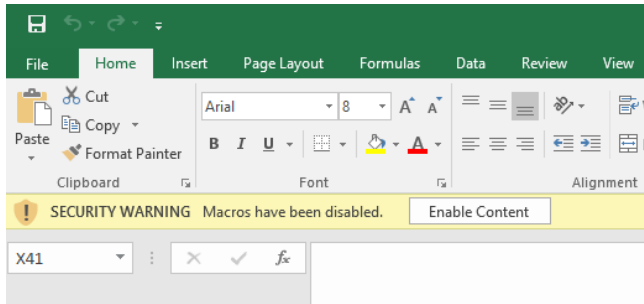
También tenemos 20 variables U (una por ciudad). Por lo tanto, hay 20 variables continuas.

En total hay 400 variables.

2. Tenemos 20 ciudades. Hay tantas rutas posibles como posibles ordenaciones o permutaciones de estas ciudades, esto es, $20! = 2.432.902.008.176.640.000$. Si dividimos este número por 1000 (una solución, lleva 1 milisegundos, 1000 soluciones, 1 segundo) obtenemos el número de segundos que necesitamos para calcular todas las soluciones. Si dividimos el resultado por 3600 obtenemos horas, si volvemos a dividir por 24 días y si

volvemos a dividir por 365,25 obtenemos el número de años. Aun así ¡necesitamos 77.094.012 años para calcular todas las soluciones!

3. Abramos el fichero Excel. Lo primero es asegurarnos de que activamos los macros contenidos en el fichero. En la siguiente figura se resalta la banda amarilla con la advertencia de seguridad. Es necesario darle al botón “Enable content”:



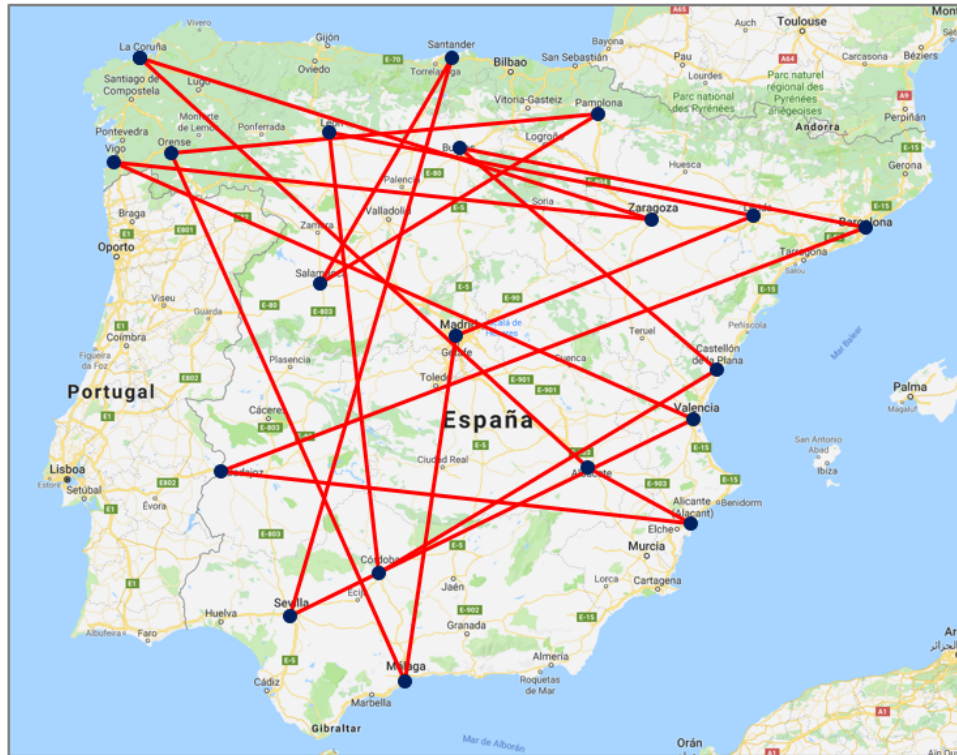
El fichero Excel tiene una columna de variables “Orden visita” con el nombre “Orden” en el rango B4:B23. Hay otros nombres en la hoja, pero no son necesarios para seguir (Los puedes verificar en Formulas-Name Manager). Aparece un mapa de España donde están marcadas las ciudades a visitar como puntos y la ruta se representa con líneas rojas. La solución inicial del fichero es visitar las ciudades en orden alfabético de su nombre. La distancia total de esta ruta (celda C40) es de 12.388 Km.

Máster Universitario en Ingeniería Informática

El problema del viajante de comercio "The Traveling Salesman Problem" (TSP)

Orden visita

- 1 A Coruña
- 2 Albacete
- 3 Alicante
- 4 Badajoz
- 5 Barcelona
- 6 Burgos
- 7 Castellón
- 8 Córdoba
- 9 León
- 10 Lleida
- 11 Madrid
- 12 Málaga
- 13 Ourense
- 14 Pamplona
- 15 Salamanca
- 16 Santander
- 17 Sevilla
- 18 Valencia
- 19 Vigo
- 20 Zaragoza



Distancia Total

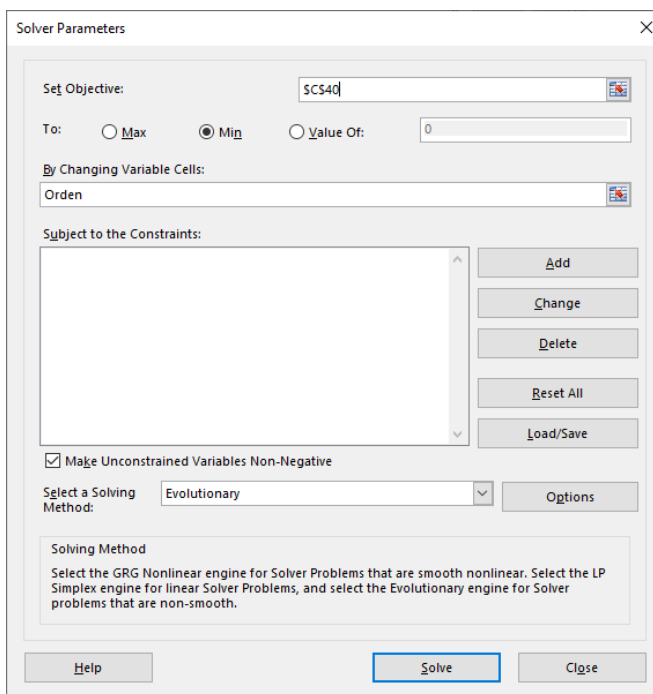
12388

	A Coruña	Albacete	Alicante	Badajoz	Barcelona	Burgos	Castellón	Córdoba	León	Lleida	Madrid	Málaga	Ourense	Pamplona	Salamanca	Santander	Sevilla	Valencia	Vigo	Zaragoza
Distancias	0	806	969	615	1029	461	896	823	286	876	564	971	144	657	426	435	825	908	141	737
2 Albacete	806	0	163	504	497	461	228	324	562	451	240	416	722	519	438	610	460	167	796	379
3 Alicante	969	163	0	651	494	624	225	442	725	455	403	452	885	601	601	773	569	158	959	456
4 Badajoz	615	504	651	0	974	528	710	256	490	830	381	369	488	727	295	646	207	649	478	687
5 Barcelona	1029	497	494	974	0	577	269	818	755	155	590	894	964	422	757	651	954	336	1039	296
6 Burgos	461	461	624	528	577	0	470	589	180	424	231	713	403	202	234	147	697	489	477	285
7 Castellón	896	228	225	710	269	470	0	549	629	230	395	625	831	422	598	620	686	67	906	252
8 Córdoba	823	324	442	256	818	589	549	0	630	747	358	155	678	718	436	738	136	488	723	633
9 León	286	562	725	490	755	180	629	630	0	602	320	778	235	381	194	229	659	642	323	463
10 Lleida	876	451	455	830	155	424	230	747	602	0	446	845	811	269	604	498	884	297	886	142
11 Madrid	564	240	403	381	590	231	395	358	320	446	0	482	480	371	200	380	488	341	555	303
12 Málaga	971	416	452	369	894	713	625	155	778	845	482	0	827	827	584	863	191	558	850	732
13 Ourense	144	722	885	488	964	403	831	678	235	811	480	827	0	604	305	445	680	823	87	672
14 Pamplona	657	519	601	727	422	202	422	718	381	269	371	827	604	0	433	229	854	452	678	171
15 Salamanca	426	438	601	295	757	234	598	436	194	604	200	584	305	433	0	352	465	544	380	465
16 Santander	435	610	773	646	651	147	620	738	229	498	380	863	445	229	352	0	815	639	528	371
17 Sevilla	825	460	569	207	954	697	686	136	659	884	488	191	680	854	465	815	0	625	689	770
18 Valencia	908	167	158	649	336	489	67	488	642	297	341	558	823	452	544	639	625	0	898	306
19 Vigo	141	796	959	478	1039	477	906	723	323	886	555	850	87	678	380	528	689	898	0	747
20 Zaragoza	737	379	456	687	296	285	252	633	463	142	303	732	672	171	465	371	770	306	747	0

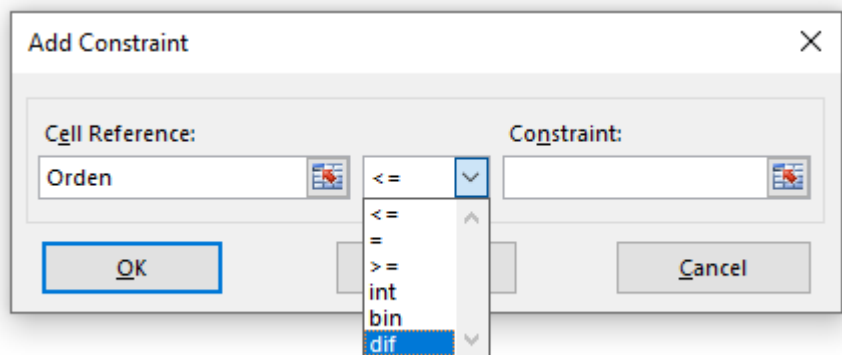
Para calcular una heurística a mano, podemos, por ejemplo, empezar de la Coruña, ir a la siguiente ciudad más cercana, y desde esa a la más cercana, y así sucesivamente. Realiza este cálculo a mano. ¿La ruta que sale tiene una distancia menor que la calculada alfabéticamente?

Una vez has obtenido este orden, vamos a meterlo en el Excel. Se indica el número de la ciudad, en la columna correspondiente, y el nombre aparece solo. Por ejemplo, si la primera ciudad a visitar es A Coruña, y la segunda es Vigo, escribe “1” en B4 y “A Coruña” aparecerá automáticamente en C5. Escribe “19” en B5 (y en C5 aparecerá “Vigo” de forma automática). Sigue con todas las ciudades asegurándote de no repetir ninguna. Cuando hayas terminado, obtienes en C40 la distancia total de la ruta.

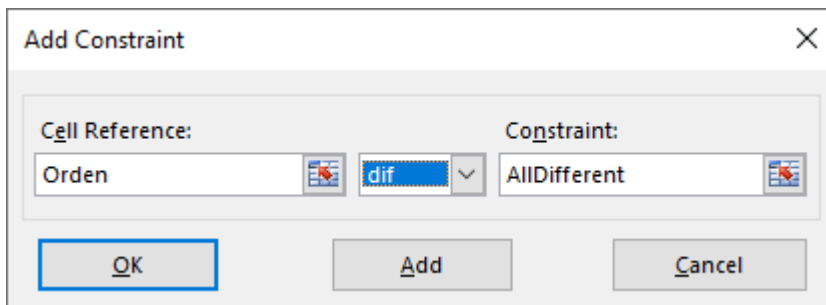
4. Vamos a resolver ahora el problema con el solver evolutivo de Excel. Ves al menú Data y luego Solver. Hay que minimizar la distancia total, es decir, la celda C40 y las variables son “Orden”. Asegúrate de que seleccionas el método de resolución “Evolutionary”:



Aún no hemos terminado. Necesitamos especificar al solver evolutivo la representación de la solución. Estamos usando una permutación para representar las soluciones. Para indicar esto al solver añadimos una restricción, indicando que nuestras variables son “all different”:



Nota que hay que seleccionar “dif” en el tipo de restricción:



La ventana del solver debe quedar así (notar que estamos minimizando):

Solver Parameters

Set Objective:

SCS40

To:

☐ Max
 ☒ Min
 ☐ Value Of:

0

By Changing Variable Cells:

Orden

Subject to the Constraints:

Orden = AllDifferent

Add

Change

Delete

Reset All

Load/Save

☒ Make Unconstrained Variables Non-Negative

Select a Solving Method:

Evolutionary

Options

Solving Method

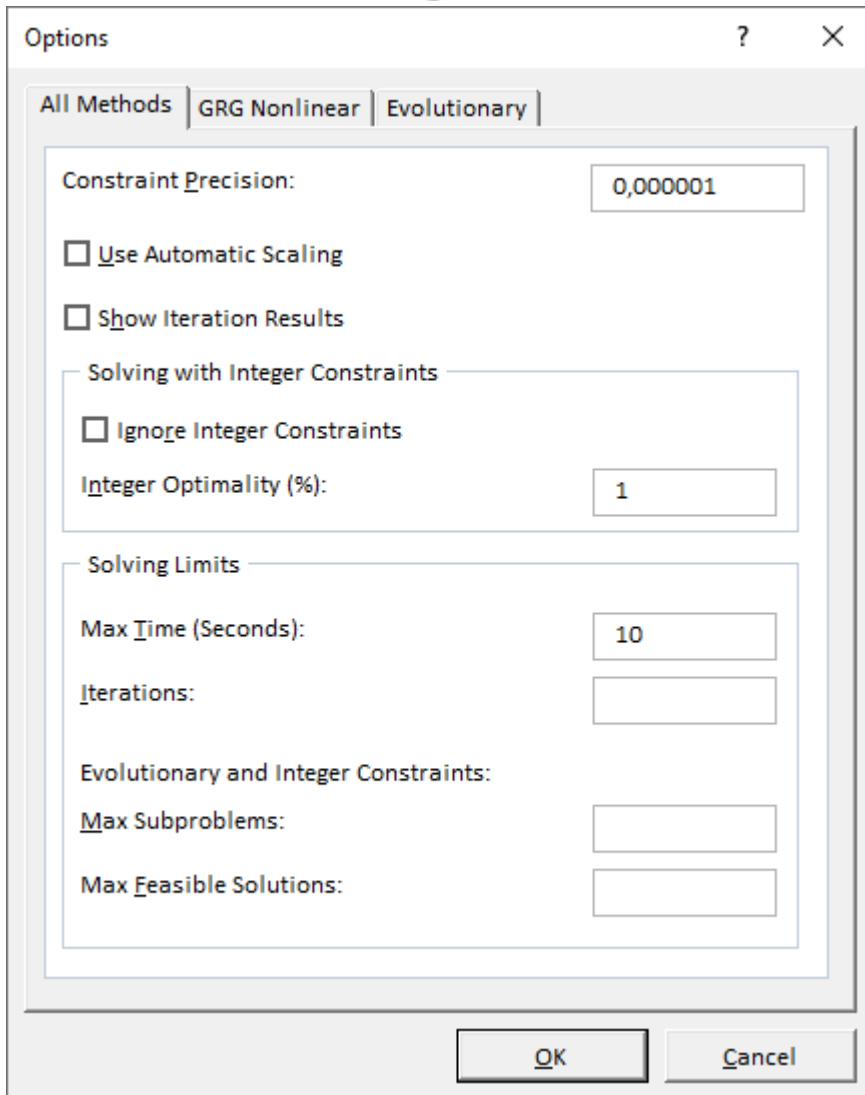
Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.

Help

Solve

Close

Ahora vamos a las opciones con el botón “Options”. En la pestaña “All Methods”, indicamos el tiempo máximo que le queremos dar al algoritmo genético, dentro de Solving Limits, y el campo Max Time (Seconds). Vamos a darle 10 segundos de tiempo, tal y como se muestra a continuación:

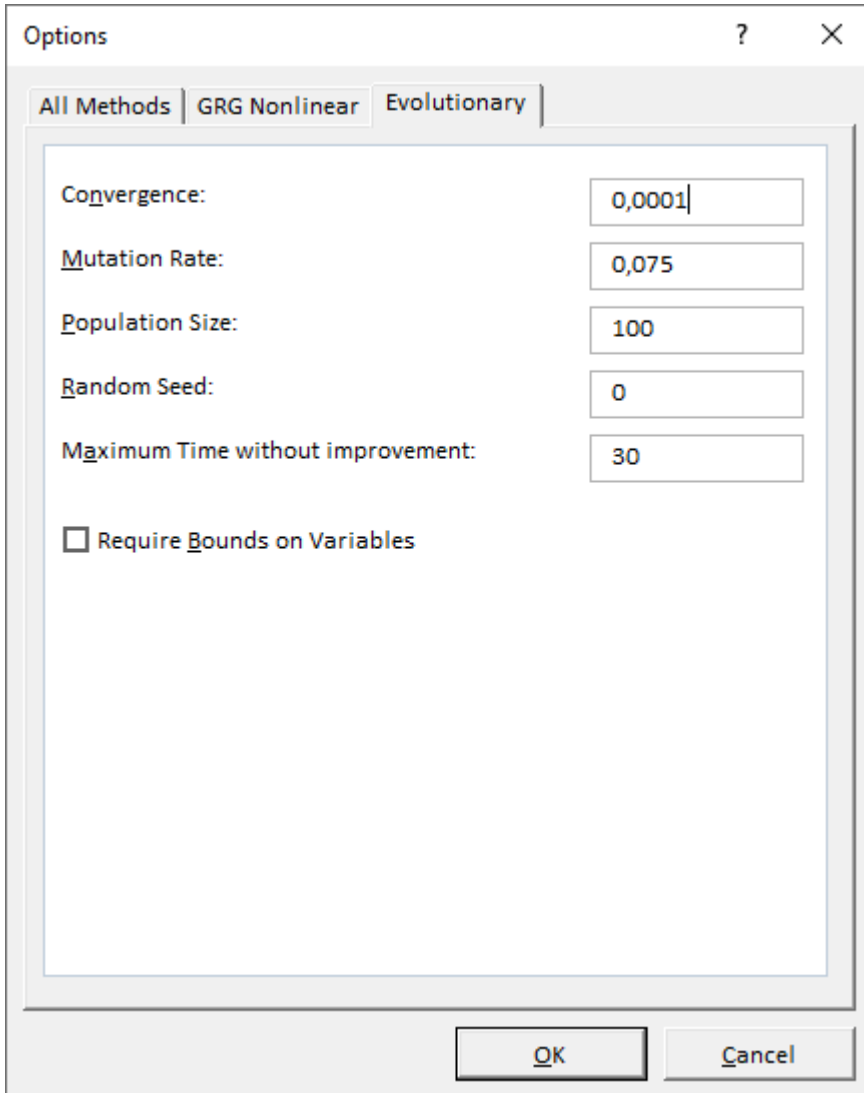


The image shows a screenshot of a software application's 'Options' dialog box. The dialog has a title bar with a question mark and a close button. It features three tabs: 'All Methods', 'GRG Nonlinear', and 'Evolutionary'. The 'Evolutionary' tab is currently selected. Inside this tab, there are several settings:

- Constraint Precision:** A text box containing the value '0,000001'.
- Use Automatic Scaling:** An unchecked checkbox.
- Show Iteration Results:** An unchecked checkbox.
- Solving with Integer Constraints:** A section header.
- Ignore Integer Constraints:** An unchecked checkbox.
- Integer Optimality (%):** A text box containing the value '1'.
- Solving Limits:** A section header.
- Max Time (Seconds):** A text box containing the value '10'.
- Iterations:** An empty text box.
- Evolutionary and Integer Constraints:** A section header.
- Max Subproblems:** An empty text box.
- Max Feasible Solutions:** An empty text box.

At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

En la pestaña Evolutionary podemos introducir otros campos, como el tamaño de la población. Por ejemplo:



Options

All Methods | GRG Nonlinear | Evolutionary

Convergence: 0,0001

Mutation Rate: 0,075

Population Size: 100

Random Seed: 0

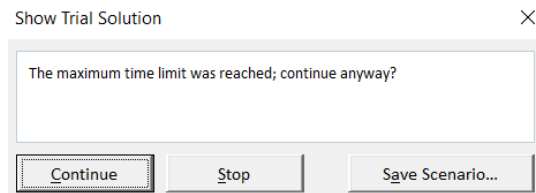
Maximum Time without improvement: 30

☐ Require Bounds on Variables

OK Cancel

Para este caso, no vamos a tocar nada. Le damos a OK y luego a Solve.

Tras unos segundos, obtendrás este mensaje.

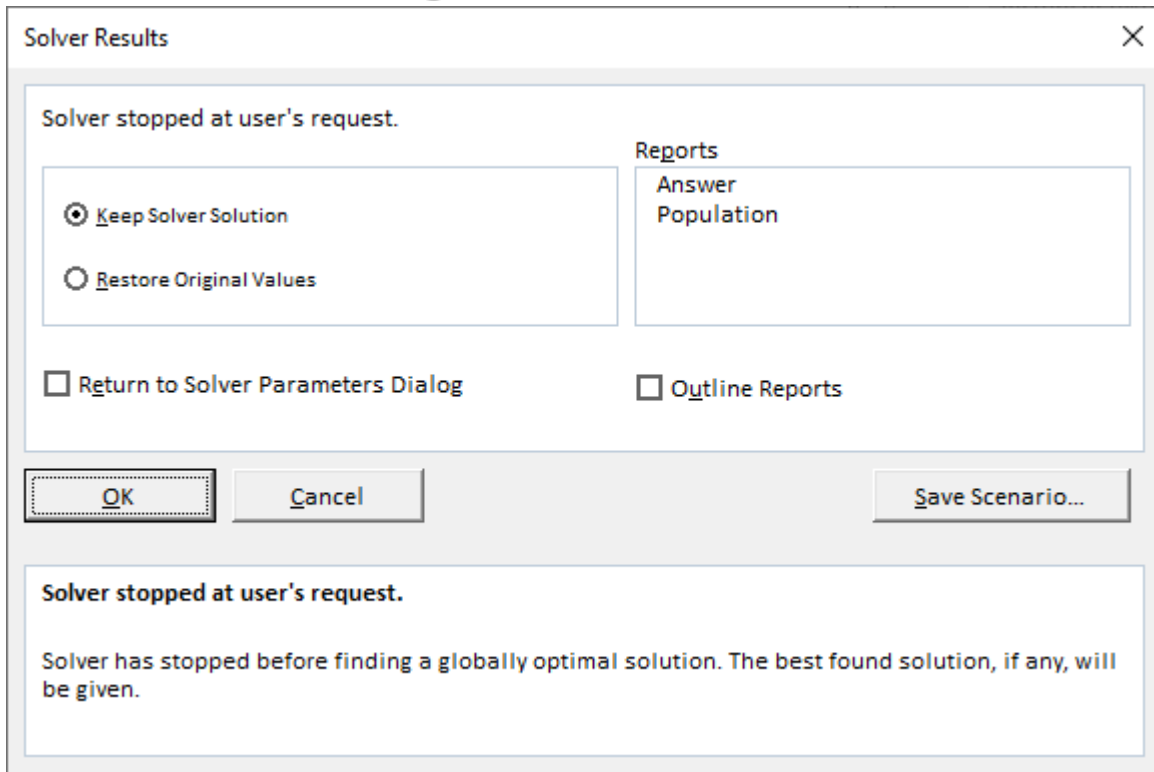


Show Trial Solution

The maximum time limit was reached; continue anyway?

Continue Stop Save Scenario...

Le damos a Stop y nos lleva a la ventana de la solución:



Seleccionamos Answer y Population dentro de Reports y le damos a OK.

El Answer report nos da los valores de las variables. Los algoritmos genéticos son estocásticos (no dan siempre la misma solución) así que tu solución es probable que no coincida con la de esta guía:

Microsoft Excel 16.0 Answer Report

Result: Solver stopped at user's request.

Solver Engine

Engine: Evolutionary

Solution Time: 7,39 Seconds.

Iterations: 0 Subproblems: 3959

Solver Options

Max Time 5 sec, Iterations Unlimited, Precision 0,000001

Convergence 0,0001, Population Size 100, Random Seed 0, Mutation Rate 0,075, Time w/o Improve 30 sec

Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume NonNegative

Objective Cell (Min)

Cell	Name	Original Value	Final Value
\$C\$40	Total Distance	12388	4151

Variable Cells

Cell	Name	Original Value	Final Value	Integer
\$B\$4	Visit Order	1	14	AllDiff
\$B\$5	Visit Order	2	20	AllDiff
\$B\$6	Visit Order	3	10	AllDiff
\$B\$7	Visit Order	4	5	AllDiff
\$B\$8	Visit Order	5	18	AllDiff
\$B\$9	Visit Order	6	7	AllDiff
\$B\$10	Visit Order	7	3	AllDiff
\$B\$11	Visit Order	8	2	AllDiff
\$B\$12	Visit Order	9	12	AllDiff
\$B\$13	Visit Order	10	8	AllDiff
\$B\$14	Visit Order	11	17	AllDiff
\$B\$15	Visit Order	12	4	AllDiff
\$B\$16	Visit Order	13	11	AllDiff
\$B\$17	Visit Order	14	15	AllDiff
\$B\$18	Visit Order	15	13	AllDiff
\$B\$19	Visit Order	16	19	AllDiff
\$B\$20	Visit Order	17	1	AllDiff
\$B\$21	Visit Order	18	9	AllDiff
\$B\$22	Visit Order	19	16	AllDiff
\$B\$23	Visit Order	20	6	AllDiff

Constraints

NONE
\$B\$4:\$B\$23=AllDiff

Hemos conseguido bajar de una solución alfabética con una distancia total viajada de 12.388 Km. a una solución optimizada con una distancia de 4.151 Km. Es una mejora notable.

El informe de Population tiene este aspecto:

Microsoft Excel 16.0 Population Report
Worksheet: [P05_1G_en.xlsm]TSP

Variable Cells

Cell	Name	Best Value	Mean Value	Standard Deviation	Maximum Value	Minimum Value
\$B\$4	Visit Order	14	14,21	1,759792228	20	3

\$B\$5	Visit Order	20	18,69	3,600490988	20	2
\$B\$6	Visit Order	10	9,85	2,75745088	20	3
\$B\$7	Visit Order	5	5,73	1,91144876	15	5
\$B\$8	Visit Order	18	17,29	2,362565803	18	5
\$B\$9	Visit Order	7	7,05	1,659834481	19	1
\$B\$10	Visit Order	3	3,65	2,854643568	20	1
\$B\$11	Visit Order	2	3,76	4,235778344	20	2
\$B\$12	Visit Order	12	10,7	3,471383737	18	2
\$B\$13	Visit Order	8	10,53	5,742724866	17	4
\$B\$14	Visit Order	17	11,38	5,014685504	19	2
\$B\$15	Visit Order	4	7,56	4,699709856	17	2
\$B\$16	Visit Order	11	11,95	2,75745088	15	1
\$B\$17	Visit Order	15	12,63	3,347409672	19	1
\$B\$18	Visit Order	13	10,66	7,321229681	19	1
\$B\$19	Visit Order	19	10,8	8,6608372	20	1
\$B\$20	Visit Order	1	10,94	6,023221058	20	1
\$B\$21	Visit Order	9	10,05	2,668086743	17	3
\$B\$22	Visit Order	16	8,61	4,315569651	20	6
\$B\$23	Visit Order	6	13,96	4,341298042	20	1

Constraints

NONE

Este informe nos indica la media del valor de cada variable, junto con la desviación típica y los valores mínimos y máximos para todas las soluciones encontradas por el algoritmo. Si la media fuera igual al mejor y la desviación típica fuera de 0, implicaría que todos los miembros de la población (individuos) serían iguales. Esto requeriría, por ejemplo, una mayor tasa de mutación.

La solución obtenida se muestra en el fichero Excel:

Máster Universitario en Ingeniería Informática

El problema del viajante de comercio "The Traveling Salesman Problem" (TSP)

Orden visita

- 14 Pamplona
- 20 Zaragoza
- 10 Lleida
- 5 Barcelona
- 18 Valencia
- 7 Castellón
- 3 Alicante
- 2 Albacete
- 12 Málaga
- 8 Córdoba
- 17 Sevilla
- 4 Badajoz
- 11 Madrid
- 15 Salamanca
- 13 Ourense
- 19 Vigo
- 1 A Coruña
- 9 León
- 16 Santander
- 6 Burgos

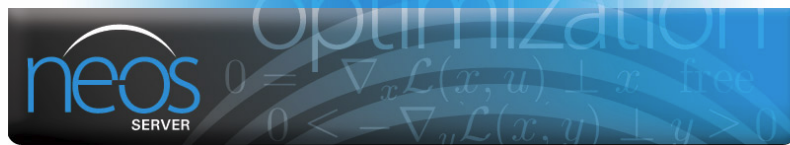


Distancia Total

4151

	A Coruña	Albacete	Alicante	Badajoz	Barcelona	Burgos	Castellón	Córdoba	León	Lleida	Madrid	Málaga	Ourense	Pamplona	Salamanca	Santander	Sevilla	Valencia	Vigo	Zaragoza
Distancias																				
1 A Coruña	0	806	969	615	1029	461	896	823	286	876	564	971	144	657	426	435	825	908	141	737
2 Albacete	806	0	163	504	497	461	228	324	562	451	240	416	722	519	438	610	460	167	796	379
3 Alicante	969	163	0	651	494	624	225	442	725	455	403	452	885	601	601	773	569	158	959	456
4 Badajoz	615	504	651	0	974	528	710	256	490	830	381	369	488	727	295	646	207	649	478	687
5 Barcelona	1029	497	494	974	0	577	269	818	755	155	590	894	964	422	757	651	954	336	1039	296
6 Burgos	461	461	624	528	577	0	470	589	180	424	231	713	403	202	234	147	697	489	477	285
7 Castellón	896	228	225	710	269	470	0	549	629	230	395	625	831	422	598	620	686	67	906	252
8 Córdoba	823	324	442	256	818	589	549	0	630	747	358	155	678	718	436	738	136	488	723	633
9 León	286	562	725	490	755	180	629	630	0	602	320	778	235	381	194	229	659	642	323	463
10 Lleida	876	451	455	830	155	424	230	747	602	0	446	845	811	269	604	498	884	297	886	142
11 Madrid	564	240	403	381	590	231	395	358	320	446	0	482	480	371	200	380	488	341	555	303
12 Málaga	971	416	452	369	894	713	625	155	778	845	482	0	827	827	584	863	191	558	850	732
13 Ourense	144	722	885	488	964	403	831	678	235	811	480	827	0	604	305	445	680	823	87	672
14 Pamplona	657	519	601	727	422	202	422	718	381	269	371	827	604	0	433	229	854	452	678	171
15 Salamanca	426	438	601	295	757	234	598	436	194	604	200	584	305	433	0	352	465	544	380	465
16 Santander	435	610	773	646	651	147	620	738	229	498	380	863	445	229	352	0	815	639	528	371
17 Sevilla	825	460	569	207	954	697	686	136	659	884	488	191	680	854	465	815	0	625	689	770
18 Valencia	908	167	158	649	336	489	67	488	642	297	341	558	823	452	544	639	625	0	898	306
19 Vigo	141	796	959	478	1039	477	906	723	323	886	555	850	87	678	380	528	689	898	0	747
20 Zaragoza	737	379	456	687	296	285	252	633	463	142	303	732	672	171	465	371	770	306	747	0

5. Te hemos proporcionado un fichero con extensión MPS, que contiene el modelo matemático para ejemplo (escribir a mano el modelo llevaría demasiado tiempo). No es posible además resolver este modelo con el solver estándar de Excel. Vamos a resolver este modelo de forma exacta en la nube, usando el solver NEOS. Ves a <https://neos-server.org/neos/>:



NEOS Server: State-of-the-Art Solvers for Numerical Optimization

The **NEOS Server** is a free internet-based service for solving numerical optimization problems. Hosted by the [Wisconsin Institute for Discovery at the University of Wisconsin in Madison](#), the NEOS Server provides access to more than 60 state-of-the-art solvers in more than a dozen optimization categories. Solvers hosted by the University of Wisconsin in Madison run on distributed high-performance machines enabled by the [HTCondor](#) software; remote solvers run on machines at [Arizona State University](#), the [University of Klagenfurt](#) in Austria, and the [University of Minho](#) in Portugal.

The **NEOS Guide** website complements the NEOS Server, showcasing optimization case studies, presenting optimization information and resources, and providing background information on the NEOS Server.

NEOS Server

- [Submit a job to NEOS](#)
- [View Job Queue and Job Results](#)
- [User's Guide to the NEOS Server](#)
- [NEOS Server FAQ](#)
- [NEOS Support](#)


NEOS Guide

- [NEOS Case Studies](#)
- [NEOS Optimization Guide](#)
- [NEOS Server Information](#)
- [Optimization Resources, LP FAQ and NLP FAQ](#)

Advanced Tools


- [Statistics: solvers, web sites, cluster](#)
- [Job Archives](#) (password required)
- [Downloads: Client Tools \(GitHub\) and Kestrel](#)

Latest NEOS News

- 


NEOS
@NeosOpt

Global optimization solver RAPOSa now available on the @NeosOpt server. Check it out now!
neos-server.org/neos/solvers/g...

Jul 12, 2019
- 

NEOS
@NeosOpt

Attention NLP-Lovers. KNITRO has been updated to the most recent version on @NeosOpt

Jul 2, 2019
- 

NEOS
@NeosOpt

All of the @coin_or solvers have been updated to latest versions on @NeosOpt.

Jul 2, 2019

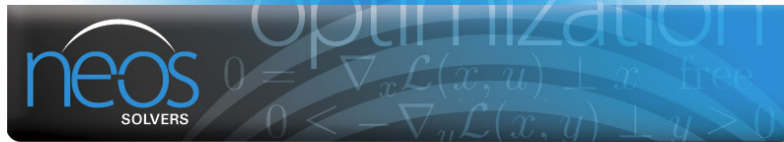
We want to keep our services as available and free as possible. Please consider making a [contribution](#) to help us keep the optimizations flowing.



[Terms of Use](#) · [Acknowledgements](#) · [Questions and Comments](#)

Copyright © 2019, Wisconsin Institutes for Discovery at the University of Wisconsin, Madison

Dale a "Submit a job to NEOS":



Listed below are the available solvers organized by Problem Type. An additional list is available for searching by Solver if you prefer.

If you need help in selecting a solver, consult the [Optimization Tree](#) of the NEOS Guide. The choice of solver then determines the available input options for defining the optimization problem.

Each solver has sample problems and background information on the solver. Be sure to submit a sample problem to get a feel for how to submit optimization problems to NEOS. If you encounter problems, consult the [NEOS Server FAQ](#), or contact us by clicking on the **Comments and Questions** link at the bottom of the page.

Problem Type

Solver

Expand All

Collapse All

Job Queue Tools

- View Job Queue
- View Job Results / Kill a Job

Application

- Domino [jpeg Input]
- ECM [csv Input][single_text Input][zip Input]
- Fishwerks [csv Input]

Bound Constrained Optimization

- L-BFGS-B [AMPL Input]

Combinatorial Optimization and Integer Programming

- BiqMac [SPARSE Input]
- concorde [TSP Input]

Complementarity Problems

- Knitro [AMPL Input]
- MILES [GAMS Input]
- NLPEC [GAMS Input]
- PATH [AMPL Input][GAMS Input]

Extended Mathematical Programming

- DE [GAMS Input]
- JAMS [GAMS Input]

Global Optimization

- ANTIGONE [GAMS Input]
- ASA [AMPL Input]
- BARON [AMPL Input][GAMS Input]
- Couenne [AMPL Input][GAMS Input]
- icos [AMPL Input]
- LGO [AMPL Input]

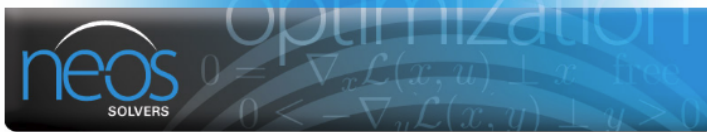
Baja en la pantalla y selecciona la opción [MPS Input] para el solver GUROBI en la sección “Mixed Integer Linear Programming”:

Mixed Integer Linear Programming

- Cbc [AMPL Input][GAMS Input][MPS Input]
- CPLEX [AMPL Input][GAMS Input][LP Input][MPS Input][NL Input]
- feaspump [AMPL Input][CPLEX Input][MPS Input]
- FICO-Xpress [AMPL Input][GAMS Input][MOSEL Input][MPS Input][NL Input]
- Gurobi [AMPL Input][GAMS Input][LP Input][MPS Input][NL Input]
- MINTO [AMPL Input]
- MOSEK [AMPL Input][GAMS Input][LP Input][MPS Input][NL Input]
- proxy [CPLEX Input][MPS Input]
- qsopt_ex [AMPL Input][LP Input][MPS Input]
- scip [AMPL Input][CPLEX Input][GAMS Input][MPS Input][OSIL Input][Python Input][ZIMPL Input]
- SYMPHONY [MPS Input]

(también podríamos seleccionar, CPLEX, Cbc o scip, entre muchos otros, siempre que tengan la opción MPS Input)

Sube el fichero MPS usando el botón indicado en la siguiente pantalla, seleccionando la opción “check the box to include the solution file as part of the results” y después selecciona el botón “Submit to NEOS” tal y como se indica:



NEOS Interfaces to Gurobi

[Sample Submissions](#)
[WWW Form](#) - [Email](#) - [XML-RPC](#) - [Disabled](#)

Gurobi

The NEOS Server offers the [Gurobi Optimizer](#) for the solution of linear programming (LP) problems, mixed-integer linear programming (MILP) problems, and second-order conic programming (SOCP) problems. LPs and MILPs can be submitted to Gurobi on the NEOS server in AMPL, GAMS, LP, or MPS format. SOCPs can be submitted in AMPL, GAMS or NL format.

The Gurobi Optimizer is a state-of-the-art solver for mathematical programming. The solvers in the Gurobi Optimizer were designed from the ground up to exploit modern architectures and multi-core processors. For more information on Gurobi products and services, see the [Gurobi website](#).

Using the NEOS Server for Gurobi/MPS

To solve a linear programming problem using MPS as input, you need only submit the MPS file. The MPS file may be submitted in gzipped or zipped format, and NEOS will automatically uncompress it.

An optional parameter file may also be specified. This file should consist of lines in the format **Keyword Value** where **Keyword** is one of the [Gurobi parameter keywords](#) and **Value** is the value of the parameter.

Web Submission Form

MPS file

Enter the path to the MPS file

1

Parameter file

Enter the path to the parameter file

Return .sol file

☒ Check the box to include the solution file as part of the results

2

Comments

Additional Settings

☐ Dry run: generate job XML instead of submitting it to NEOS

☐ Short Priority: submit to higher priority queue with maximum CPU time of 5 minutes

E-Mail address:

3

Please do not click the "Submit to NEOS" button more than once.

By submitting a job, you have accepted the [Terms of Use](#)

Aparecerá una pantalla con el progreso de la solución. No le des a ningún botón y espera hasta que se carguen los resultados. Recuerda, este es un problema muy difícil. Obtendrás un resultado en texto como este:

```
NEOS Server Version 5.0
Job#       : 7340211
Password   : pWtAkzSO
User       : None
Solver     : milp:Gurobi:MPS
Start      : 2019-07-22 18:08:51
End        : 2019-07-22 18:09:11
Host       : NEOS HTCondor Pool
```

Disclaimer:

This information is provided without any express or implied warranty. In particular, there is no warranty of any kind concerning the fitness of this information for any particular purpose.

```
*****
/opt/gurobi/latest/linux64/lib:/opt/gurobi/latest/linux64/lib:/opt/rh/python27/root/usr/lib64
```

```
4
Load Avg: ( 5.98 , 5.88 , 5.87 )
Read MPS format model from file model.mps
Reading time = 0.00 seconds
NO_TITLE: 401 rows, 400 columns, 1843 nonzeros
Changed value of parameter Threads to 1
  Prev: 0 Min: 0 Max: 1024 Default: 0
Optimize a model with 401 rows, 400 columns and 1843 nonzeros
Variable types: 20 continuous, 380 integer (380 binary)
Coefficient statistics:
  Matrix range      [1e+00, 2e+01]
  Objective range   [7e+01, 1e+03]
  Bounds range      [1e+00, 1e+00]
  RHS range         [1e+00, 2e+01]
Presolve removed 0 rows and 1 columns
Presolve time: 0.00s
Presolved: 401 rows, 399 columns, 1824 nonzeros
Variable types: 19 continuous, 380 integer (380 binary)
```

Root relaxation: objective 3.197200e+03, 59 iterations, 0.00 seconds

Nodes		Current Node		Objective Bounds		Work	
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd	Gap It/Node Time
	0	0	3197.20000	0	21	- 3197.20000	- - 0s
H	0	0			5883.0000000	3197.20000	45.7% - 0s
	0	0	3496.00000	0	34	5883.00000 3496.00000	40.6% - 0s
	0	0	3496.00000	0	34	5883.00000 3496.00000	40.6% - 0s
	0	0	3496.00000	0	34	5883.00000 3496.00000	40.6% - 0s
	0	0	3496.00000	0	34	5883.00000 3496.00000	40.6% - 0s
	0	2	3496.00000	0	34	5883.00000 3496.00000	40.6% - 0s
*	156	152		154	5226.0000000	3500.16000	33.0% 3.5 0s
H	297	265			4572.0000000	3502.22917	23.4% 3.8 0s
H	494	319			4093.0000000	3526.73437	13.8% 3.6 0s
H	1387	479			4035.0000000	3611.26250	10.5% 6.3 1s
*	1610	538		56	4012.0000000	3633.00000	9.45% 6.2 1s
*	1613	494		55	3968.0000000	3633.00000	8.44% 6.2 1s
*	2415	643		41	3946.0000000	3693.00000	6.41% 6.7 1s

Cutting planes:

```
Learned: 15
Gomory: 9
Cover: 22
MIR: 5
Flow cover: 17
Inf proof: 8
```

Explored 4992 nodes (33600 simplex iterations) in 2.53 seconds
Thread count was 1 (of 24 available processors)

Solution count 8: 3946 3968 4012 ... 5883

```
Optimal solution found (tolerance 1.00e-04)
Best objective 3.946000000000e+03, best bound 3.946000000000e+03, gap 0.0000%
Optimal objective: 3946.0
***** Begin .sol file *****
# Solution for model NO_TITLE
# Objective value = 3946
U(1 0
U(2 9
U(3 10
U(4 4
U(5 13
U(6 19
U(7 12
U(8 7
U(9 20
U(10 15
U(11 8
U(12 6
U(13 2
U(14 17
U(15 3
...
```

(lo hemos recortado, pues es muy largo). Vemos que la solución óptima es 3.946 (en la línea # Objective value = 3946) que es mejor que la solución que obtuvimos con el algoritmo genético en Excel. Hay una larga lista de variables. $X(i,j)$ 0 indica que la variable $X_{ij} = 0$, mientras que $X(i,j)$ 1 indica que la variable $X_{ij} = 1$. De esta forma, puedes construir en el Excel la solución, para poderla comparar. Por ejemplo, $X(20,14)$ 1 indica que después de la ciudad 20, vamos a la 14.

Aún más fácil es ver las variables. $U(i, k)$ indica que la ciudad i se visita en la posición k de la ruta. Por ejemplo, “U(1 0” indica que la ciudad 1 (A Coruña) se visita en la posición 0 (primer lugar), y “U(2 9” indica que la ciudad 2 (Albacete) se visita en la posición 9 (décimo lugar).

Si cargas esta solución óptima en Excel obtienes la siguiente ruta:

Máster Universitario en Ingeniería Informática

El problema del viajante de comercio "The Traveling Salesman Problem" (TSP)

Orden visita

- 1 A Coruña
- 19 Vigo
- 13 Ourense
- 15 Salamanca
- 4 Badajoz
- 17 Sevilla
- 12 Málaga
- 8 Córdoba
- 11 Madrid
- 2 Albacete
- 3 Alicante
- 18 Valencia
- 7 Castellón
- 5 Barcelona
- 10 Lleida
- 20 Zaragoza
- 14 Pamplona
- 16 Santander
- 6 Burgos
- 9 León



Distancia Total

3946

	A Coruña	Albacete	Alicante	Badajoz	Barcelona	Burgos	Castellón	Córdoba	León	Lleida	Madrid	Málaga	Ourense	Pamplona	Salamanca	Santander	Sevilla	Valencia	Vigo	Zaragoza
1 A Coruña	0	806	969	615	1029	461	896	823	286	876	564	971	144	657	426	435	825	908	141	737
2 Albacete	806	0	163	504	497	461	228	324	562	451	240	416	722	519	438	610	460	167	796	379
3 Alicante	969	163	0	651	494	624	225	442	725	455	403	452	885	601	601	773	569	158	959	456
4 Badajoz	615	504	651	0	974	528	710	256	490	830	381	369	488	727	295	646	207	649	478	687
5 Barcelona	1029	497	494	974	0	577	269	818	755	155	590	894	964	422	757	651	954	336	1039	296
6 Burgos	461	461	624	528	577	0	470	589	180	424	231	713	403	202	234	147	697	489	477	285
7 Castellón	896	228	225	710	269	470	0	549	629	230	395	625	831	422	598	620	686	67	906	252
8 Córdoba	823	324	442	256	818	589	549	0	630	747	358	155	678	718	436	738	136	488	723	633
9 León	286	562	725	490	755	180	629	630	0	602	320	778	235	381	194	229	659	642	323	463
10 Lleida	876	451	455	830	155	424	230	747	602	0	446	845	811	269	604	498	884	297	886	142
11 Madrid	564	240	403	381	590	231	395	358	320	446	0	482	480	371	200	380	488	341	555	303
12 Málaga	971	416	452	369	894	713	625	155	778	845	482	0	827	827	584	863	191	558	850	732
13 Ourense	144	722	885	488	964	403	831	678	235	811	480	827	0	604	305	445	680	823	87	672
14 Pamplona	657	519	601	727	422	202	422	718	381	269	371	827	604	0	433	229	854	452	678	171
15 Salamanca	426	438	601	295	757	234	598	436	194	604	200	584	305	433	0	352	465	544	380	465
16 Santander	435	610	773	646	651	147	620	738	229	498	380	863	445	229	352	0	815	639	528	371
17 Sevilla	825	460	569	207	954	697	686	136	659	884	488	191	680	854	465	815	0	625	689	770
18 Valencia	908	167	158	649	336	489	67	488	642	297	341	558	823	452	544	639	625	0	898	306
19 Vigo	141	796	959	478	1039	477	906	723	323	886	555	850	87	678	380	528	689	898	0	747
20 Zaragoza	737	379	456	687	296	285	252	633	463	142	303	732	672	171	465	371	770	306	747	0