
Investigaciones Turísticas

ISSN: 2174-5609



Un nuevo algoritmo heurístico para la creación de rutas turísticas personalizadas

Francisco David de la Peña Esteban

Departamento de Ingeniería Industrial
Universidad a Distancia de Madrid (UDIMA)

franciscodavid.delapena@udima.es

RESUMEN

La adaptación de las actividades turísticas a sus preferencias es un elemento crucial para los turistas del siglo XXI. El presente artículo tiene por objetivo el proponer un nuevo algoritmo heurístico para la realización de rutas turísticas personalizadas en los centros históricos de las ciudades que pueda ser aplicado a plataformas web turísticas o aplicaciones para móviles. Para su consecución se ha hecho en primera instancia un análisis del estado de la cuestión que sirviera de marco de referencia. A continuación se explica el nuevo algoritmo, que consta de dos partes. En la primera se aborda el problema de determinar los puntos de interés que debe visitar el turista en base a sus predilecciones. Para ello se van a seleccionar los parámetros que se tendrán en cuenta y se propone un nuevo método para cuantificarlos. La segunda parte del algoritmo se centra en determinar el orden en que deben ser recorridos, buscando que la distancia sea lo menor posible. Se van a realizar agrupaciones de los distintos puntos de interés, y se van a unir entre sí. El resultado final es un algoritmo que se adapta a las preferencias dadas y que logra unos desplazamientos muy ajustados.

Palabras clave: puntos de interés turísticos, centros históricos, ruta a pie, personalización.

Investigaciones Turísticas

ISSN: 2174-5609



A new heuristic algorithm to create customized tourist routes

Francisco David de la Peña Esteban

Industrial Engineering Department

Madrid Open University

franciscodavid.delapena@udima.es

ABSTRACT

The customization of tourism activities to the tourist preferences is crucial in the XXI century. This study aims to propose a new heuristic algorithm to create customized routes in historical city centers to be uploaded in web sites or mobile applications. First, a state of the art has been done followed by the description of the new algorithm proposed which consists of two different parts. In the first part, the attractions that the tourist should visit according to their preferences are selected together with the parameters and a new methodology to quantify them. In the second part, the study tries to determine the position of the attractions along the itinerary with the minimum distance between them. Finally, the attractions are grouped and linked together. The result is a customized algorithm to the tourist preferences which provides small distance walks.

Keywords: tourist attractions, historical city center, walking itinerary, customized tourist routes.

I. INTRODUCCIÓN

Cada vez está más extendido que los turistas preparen las rutas que van a realizar por una determinada ciudad con ayuda de plataformas turísticas web que les proponen distintas alternativas. En la mayoría de los casos no se tiene en cuenta ningún criterio diferenciador de los diversos turistas, y se ofrecen los mismos recorridos a todos.

Los turistas, por su lado, lo que les gustaría sería tener un recorrido personalizado que vaya por los distintos puntos de interés turísticos (a partir de ahora Pols: *Points of Interest*), en el que se tengan en cuenta sus preferencias. Además luego querrá poder realizarlo con el soporte de un *smartphone*, *tablet* o *ipad* que le guíe en el mismo. Para abordar esta nueva problemática se va a proponer un algoritmo heurístico que permite la personalización de los recorridos turísticos a pie en los centros históricos de las ciudades y que sirva de base para su implementación en plataformas web turísticas.

II. OBJETIVOS Y METODOLOGÍA

De cara a afrontar estas nuevas necesidades que tienen los turistas, este artículo tiene el objetivo de desarrollar un algoritmo heurístico para la realización de rutas a pie personalizadas en centros históricos de ciudades, en función de las preferencias de los turistas. Este algoritmo constará de dos partes bien diferenciadas. Por un lado está el determinar cuáles son los Pols que se van a elegir para ese turista, y en segundo lugar estará determinar el orden en qué deben recorrerse. Los Pols seleccionados deberán concordar lo máximo posible con las preferencias mostradas por el usuario, y la distancia a recorrer deberá ser la menor posible. Se buscará que tenga una baja complejidad algorítmica, de manera que cuando en un futuro se implemente en una plataforma web de información turística el tiempo de ejecución no sea elevado.

Los pasos a seguir para lograrlo son los siguientes. Lo primero será realizar el estado de la cuestión. Se hará un análisis de los diversos métodos existentes para seleccionar los Pols para un caso concreto y los factores que tienen en cuenta para determinarlos. También se verán las diversas aproximaciones que se han realizado para establecer el orden en qué hay que recorrer los diversos Pols. En este punto tendrá especial relevancia el análisis del clásico problema del cartero viajante (TSP- *Travelling Salesman Problem*) (Goyal, 1971). Este problema es NP-completo y por ello se necesitan las heurísticas propuestas. Por último dentro del estado de la cuestión se verán aquellos estudios que tienen en cuenta las dos partes del algoritmo.

A continuación se realizará el desarrollo del algoritmo. El algoritmo va a ser de dos pasos:

Paso 1: Determinar los Puntos de Interés (Pols) a visitar: se va a seleccionar los factores que se van a tener en cuenta. Se va a proponer una fórmula para cuantificar una puntuación asociada a cada Pol para cada caso concreto. A partir de esta puntuación se elegirán los Pols a recorrer. Las entradas en este paso serán:

- Las tipologías de patrimonio deseadas por el visitante
- El tiempo máximo disponible para realizar la ruta

La salida será el listado de los Pols que componen la ruta

Paso 2: Generación de la ruta personalizada. Determinar el orden de visita: en función de las distancias existentes entre los distintos Pols se va a realizar una agrupación de los mismos. Luego se buscará minimizar la distancia total recorrida entre las distintas agrupaciones. Como entrada a este paso se tendrá el listado de los Pols que componen la ruta, así como su punto de inicio. La salida será el orden en qué deben visitarse los Pols.

Para una mejor comprensión del algoritmo, así como para ver su funcionamiento se va a realizar un ejemplo práctico. Por último se verán las conclusiones que se pueden extraer del algoritmo propuesto y las futuras líneas de investigación.

III. ALGORITMO HEURÍSTICO PARA DISEÑAR RUTAS TURÍSTICAS

3.1 Estado de la cuestión

Existen diversas aproximaciones en los algoritmos existentes para realizar rutas turísticas, pero en todos ellos hay dos partes diferenciadas, aunque muchos de los investigadores en este tema se centran únicamente en una de ellas. Por un lado están los parámetros que va a considerar el algoritmo de cara a seleccionar los puntos de interés (Pols) que se van a visitar en la ruta, y por otro está la parte del algoritmo que se encarga de minimizar los desplazamientos entre los mismos.

De cara a la selección de los Pols y los factores a tener en cuenta se encuentran:

- la búsqueda de minimizar los costes asociados a las visitas en conjunción con satisfacer las preferencias del turista, en función del tiempo disponible para realizar el recorrido. (Caballero et al., 2005)
- la toma en consideración para cada Pol de factores como el estilo artístico, su calidad, si es adecuado para grupos, para discapacitados o para niños, o si es especialmente recomendado para algún grupo de población (Ocegueda, 2012).

En la parte del algoritmo que se basa en minimizar los desplazamientos ha habido diferentes aproximaciones, aunque la mayoría de ellas parten de la teoría de grafos:

- el problema del cartero viajante (TSP- *Travelling Salesman Problem*) y sus diferentes aproximaciones (Goyal, 1971), (Ascheuer et al., 2001), (Walshaw 2002), (Hincapié et al., 2004), (Kuo et al., 2010).
- basada en la teoría de grafos, y utilizando una combinación de diversos algoritmos (Prim, Sollin, Dijkstra) para determinar la ruta sobre unos puntos ya prefijados (Yang et al, 2012)
- también basada en grafos, y utilizando los árboles de expansión mínima para determinar el mejor camino para unos Pols ya dados (Yang y Ho, 2006).
- un algoritmo para rutas de vehículos con ventanas de tiempo aplicado para la distribución de productos. Se basa en determinar la holgura para cada uno de los destinos, y en función de esto propone la ruta a realizar (Almeida et al, 2012).

Como mejores ejemplos de autores que abordan las dos partes expuestas del algoritmo se tiene:

- la propuesta de un método TTDP - *Tourist Trip Design Models*, basado en el uso de vectores para determinar los Pol a seguir en la ruta, y usan técnicas metaheurísticas para que el desplazamiento efectuado sea mínimo (Souffriau et al., 2008).
- la personalización de rutas, buscando la más atractiva para el usuario, teniendo en cuenta ventanas de tiempo (Descamps-Vila et al., 2012a y 2012b). También consideran factores como dónde comer, si se viaja en familia, en pareja, con amigo o solo. Le da una importancia general a cada Pol, así como se les asigna una o varias tipologías de patrimonio. También permite definir al usuario su punto de partida. En la parte de establecer la ruta se basa en un algoritmo que va construyendo incrementalmente candidatos para las soluciones, y abandonando aquellos candidatos que no producen una solución válida. Va creando parejas de de Pol a visitar y busca encontrar el mejor camino entre ellas.
- la realización de rutas personalizadas para viajes que duran varios días, seleccionando previamente los Pols que se van a visitar cada día en función de parámetros tales como el tipo de Pol, su ubicación, su importancia objetiva, y los medios multimedia disponibles (Gavalas et al, 2012). Tiene también en cuenta la predicción del tiempo, haciendo una distinción entre visitas al aire libre y visitas dentro de los Pols. Se basa en teoría de grafos, y le asocia unos beneficios y unos costes a cada Pol, buscando maximizar los beneficios sin exceder el tiempo disponible al día.
- un sistema inteligente de rutas para la creación de una guía turística electrónica personalizada (PET, *Personalised Tourist Guide*) (García et al., 2009). Tiene en cuenta los Pols a visitar, así como actividades a realizar, tomando en consideración el transporte público existente así como el tiempo previsto para el caso de las actividades a realizar. También realizan una validación del sistema propuesto (García et al, 2012) en un

escenario real, como es la ciudad de San Sebastián, con 50 Pols, siendo las conclusiones obtenidas muy favorables.

- la creación de rutas personalizadas a partir de la información geo-temporal dejada por innumerables turistas en sistemas de compartición en la nube de fotografías como Flickr. Las rutas se crean basándose en el comportamiento colectivo de los turistas (Tai, et al. 2008), (De Choudhury et al. 2010).

El algoritmo que se propone quiere aportar un nuevo acercamiento, en el que primeramente se va a dar una puntuación personalizada de cada Pol para cada turista. Esta puntuación estará asociada a las tipologías de patrimonio deseadas, el estado de conservación, si es Patrimonio Nacional o si es considerado Highlight, factores que conjuntamente no se han considerado en otros algoritmos. En función de estas puntuaciones, y del tiempo disponible se seleccionarán los Pols a visitar. La creación de la ruta también se realiza de manera novedosa mediante la creación de agrupaciones de Pols y su posterior unión en la ruta.

3.2 Desarrollo del nuevo algoritmo

Existen gran cantidad de factores que pueden considerarse de cara a personalizar una ruta turística. Cuantos más parámetros se tengan en cuenta, mejor será la ruta personalizada obtenida, siempre que esos parámetros sean realmente los determinantes y no existan otros que no aporten mucho a la solución pero que compliquen el algoritmo de manera innecesaria. También habrá que considerar que pueden existir factores contrapuestos, y será necesario determinar un compromiso entre ellos.

Por el contrario, la complejidad del algoritmo será mayor, y los tiempos de respuesta obtenidos empeorarán. Un factor decisivo es la disponibilidad de muchos de estos datos, ya que en muchos Pols puede ser que no se dispongan de ellos.

De cara al diseño del nuevo algoritmo se han tenido los siguientes factores:

- Pol: Listado de todos los Puntos de Interés turísticos.
- TP: Tipologías de patrimonio asociadas a cada Pol. Un Pol puede tener asociadas varias tipologías. Las tipologías que se han determinado son las siguientes:

1-Arquitectónico; 2-Arqueológico; 3-Etnológico; 4-Inmaterial; 5-Industrial, científico, técnico; 6-Documental y Bibliográfico; 7- Museos; 8- Monumento ornamental y 9-Otros

-EC: Estado de conservación de cada Pol. Se pondera de 1 a 3, siendo 3 el mejor estado de conservación, y 1 el peor.

-PN: Si el Pol es Patrimonio nacional. Es 1 si es considerado Patrimonio nacional, y en caso contrario 0.

-HL: Si el Pol es considerado Highlight. Siendo la ponderación de 1 a 5, siendo 1 lo peor, y 5 lo más alto dentro de la escala.

-Distancias entre cada uno de los Pol.

-Tiempo estimado de visita de cada Pol.

El usuario determinará los siguientes aspectos de la ruta:

-El tiempo estimado que tiene para la ruta.

-El punto de inicio de la misma, que debe ser un Pol.

-Las tipologías de patrimonio en qué está interesado.

El algoritmo va a ser de dos pasos:

-Paso 1: Determinar los Puntos de Interés (Pol) a visitar.

-Paso 2: Generación de la ruta personalizada. Determinar el orden de visita.

Se expondrá en detalle cada uno de ellos

3.2.1. Paso 1: Determinar los puntos de interés (Pols) a visitar

Las entradas para este paso serán:

-Las tipologías de patrimonio deseadas por el visitante

-El tiempo máximo disponible para realizar la ruta.

Cada Pol debe tener una ponderación asociada, que viene dada en función de los siguientes factores:

-NTP: Nº de tipologías de patrimonio deseadas por el turista (1-..)

-EC: Estado de conservación (1-3)

-PN: Patrimonio nacional (0-1)

-HL: Highlight (1-5)

Se ha creado la siguiente fórmula que dará la ponderación a cada Pol, con valores de 0 a 100:

$$\text{Ponderación} = \text{NTP} * \text{EC} * 10 \text{ (hasta un máximo de 50 puntos)} + 30 * \text{PN} + 4 * \text{HL}$$

Esta fórmula se ha ideado de manera que se tenga una puntuación de 0 a 100, y en la que las preferencias del cliente tengan un peso del 50%, pero que también se prime a los Pols con mayor relevancia desde un punto de vista tanto institucional, como técnico. La ponderación es la siguiente:

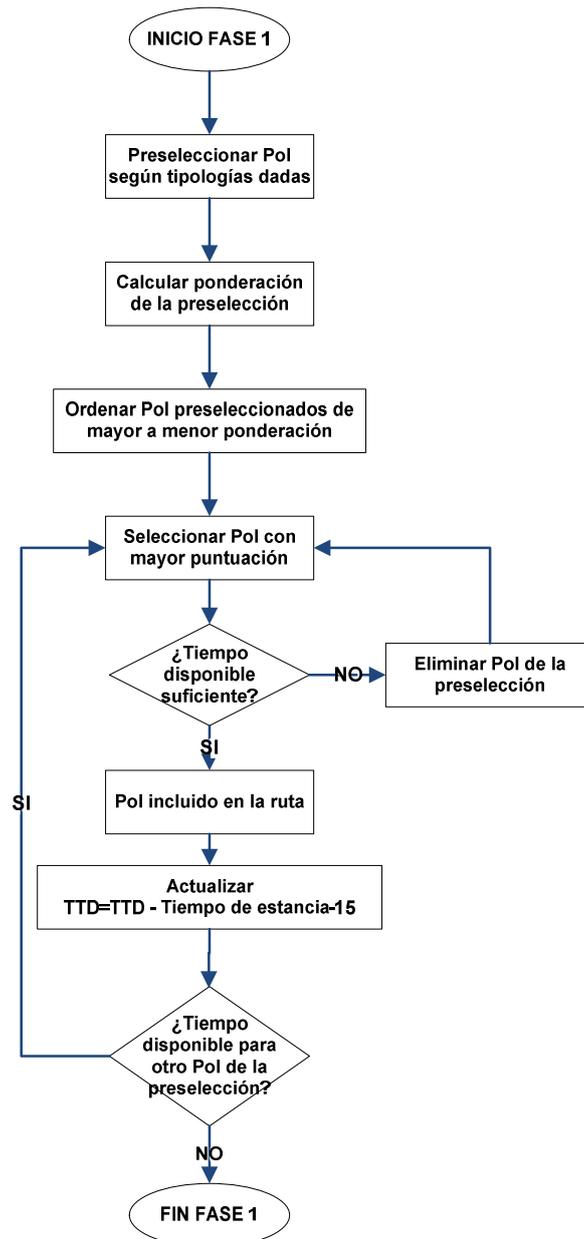
-50% para las tipologías de patrimonio deseadas por el visitante, teniendo en cuenta su estado de conservación.

-30% para los Pols que sean Patrimonio Nacional, de manera que se prime esta denominación dada su importancia.

-20% para los Pol que se consideran Highlights, beneficiando aquellos Pols que son realmente significativos y que desde un punto de vista técnico se considera que sería deseable su visita, aún cuando no se ha especificado esa tipología de patrimonio.

La metodología a seguir en este paso será la que recoge la Figura 1:

Figura 1. Paso 1 del algoritmo



1. Se preseleccionan todas aquellos Pols que cumplan con las tipologías de patrimonio dadas.
2. Se obtiene la ponderación asociada a cada uno de ellos.
3. Se ordenan de mayor a menor puntuación.
4. Se conocen los tiempos medios de estancia en cada Pol. El tiempo medio estimado de desplazamiento se considera de 15 minutos, al tratarse de un centro histórico, en el que la ruta se va a realizar a pie, y las distancias medias no son grandes. Partiendo del tiempo total disponible (TTD) en minutos, se selecciona el Pol con mayor puntuación dentro de los Pols preseleccionados que entre en ese tiempo. En caso de empate se selecciona aquel que tenga mayor HL, luego que sea PN, con mejor EC. Si persiste el empate se hace aleatoriamente.
5. Se calcula de nuevo el tiempo total disponible como $TTD = TTD - \text{Tiempo de estancia} - 15$
6. Se elimina el Pol como preseleccionado.
7. Volver al paso 4 hasta que se haya agotado el tiempo.

La salida de este paso será el listado de los Pols que componen la ruta.

3.2.2. Paso 2: Determinar el orden en qué se van a visitar los puntos de interés

Las entradas para este paso del algoritmo serán:

- Listado de los Pol que componen la ruta.
- Distancias entre ellos.
- Punto de inicio de la ruta dentro del casco histórico, definido por el visitante, a elegir de entre el listado de Pol de la ruta. La ruta al ser un centro histórico, se va a realizar a pie, sin la utilización de transporte público.

La metodología a seguir en este paso será:

1. Elaborar la tabla de distancias entre los Pols. Quitando mínimo por filas y por columnas se obtendrá la tabla de distancias con penalización.
2. Agrupar los Pols, en función de los ceros de la tabla anterior. Se van a agrupar juntos todos aquellos con los que tenga un cero entre sí, y aquellos que tengan un cero con éstos.
3. Realizar el grafo asociado a cada agrupación. Se va a realizar el árbol de expansión mínima de cada grupo.
4. Determinar nodo de inicio y fin del nodo del grupo que tiene el punto de inicio. Se elimina como posibilidad de desplazamiento el máximo por cada fila, excepto si es cero, de cara a no permitir los trayectos más largos. Se va a hacer un listado de todas las rutas permitidas que empiezan por el nodo de inicio, y se seleccionará aquella con menor recorrido. En caso de empate se escoge aleatoriamente.

En caso de no existir ninguna ruta posible, no se eliminarán los máximos desplazamientos por cada fila.

5. Determinar el siguiente grupo a unir con el nodo de fin del grupo que se acaba de determinar su orden. Se seleccionará aquel Pol que tenga menor distancia con el nodo de fin del anterior grupo. En caso de empate se elegirá aquel Pol que provenga de un grupo con mayor nº de Pol. Si persiste el empate se elige aleatoriamente.

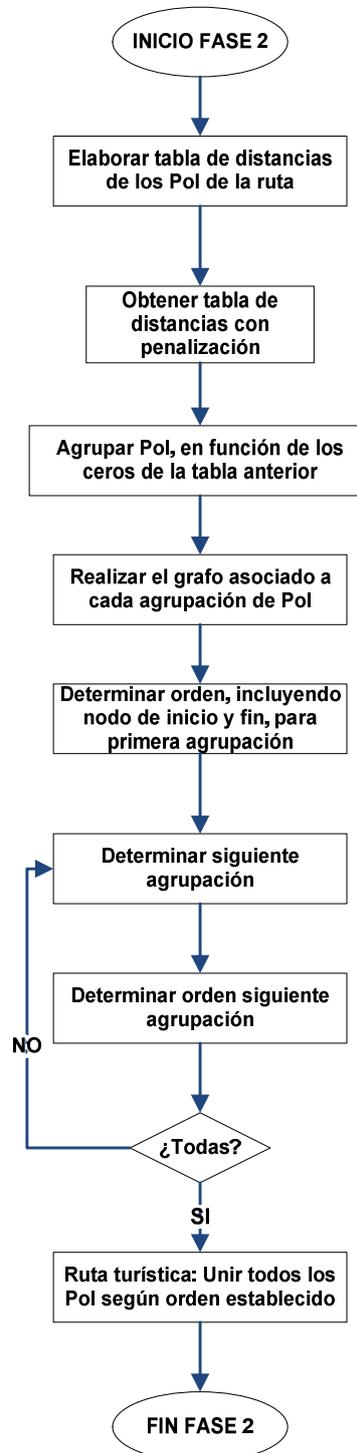
Realizar tantas veces como sea necesario hasta que todos los grupos estén unidos.

6. Unir todos los Pol para determinar la ruta turística.

Se considera que únicamente se visitarán los Pols seleccionados. No se contempla la visita de un Pol no seleccionado aunque se encuentre en la ruta turística obtenida, debido a que el tiempo de la ruta es fijo y en función de él se han determinado los Pols a visitar.

La salida de este paso será la ruta turística, indicando el orden en qué debe ser realizada.

Figura 2. Paso 2 del algoritmo



3.3 Aplicación del algoritmo a un caso práctico

Se va a realizar una aplicación práctica para una localidad hipotética que tiene 20 Pols. Se ha solicitado una ruta personalizada en la que se han seleccionado 3 tipos de patrimonio dentro de los disponibles. En la siguiente tabla se detallan para cada Pol:

- NTP: número de tipos de Patrimonio en los que coincide con la ruta solicitada
- EC: su estado de conservación.
- PN: si es patrimonio nacional (1) o no (0).
- HL: la puntuación asociada a la condición de highlight de ese Pol.

Tabla 1. Características de los Pols del ejemplo en función de las preferencias del turista

Pol	NTP	EC	PN	HL
1	2	1	1	2
2	1	1	0	4
3	1	3	1	4
4	0	2	0	2
5	1	3	0	4
6	2	2	0	1
7	1	3	0	2
8	1	3	1	3
9	0	1	0	4
10	2	3	0	2
11	2	1	0	3
12	3	3	0	3
13	3	3	1	1
14	1	1	0	5
15	1	1	0	2
16	3	2	1	2
17	1	2	0	2
18	1	1	0	4
19	1	2	1	1
20	3	2	0	1

PASO 1:

- 1-Se preseleccionan todas aquellos Pol que cumplan con las tipologías de patrimonio dadas.
- 2-Se obtiene la ponderación asociada a cada uno de ellos.

En base a esto se van a obtener tres ponderaciones:

- Ponderación asociada al número de tipos de patrimonio coincidentes = $NTP * EC * 10$ (hasta un máximo de 50 puntos)
- Ponderación asociada a si es Patrimonio Nacional = $30 * PN$
- Ponderación asociada a ser la condición de *highlight* = $4 * HL$

Siendo la ponderación total la suma de las tres.

Tabla 2. Ponderación de cada Poi del ejemplo

Poi	NTP	EC	PN	HL	Pond. NTP	Pond. PN	Pond. HL	Pond Total
1	2	1	1	2	20	30	8	58
2	1	1	0	4	10	0	16	26
3	1	3	1	4	30	30	16	76
5	1	3	0	4	30	0	16	46
6	2	2	0	1	40	0	4	44
7	1	3	0	2	30	0	8	38
8	1	3	1	3	30	30	12	72
10	2	3	0	2	50	0	8	58
11	2	1	0	3	20	0	12	32
12	3	3	0	3	50	0	12	62
13	3	3	1	1	50	30	4	84
14	1	1	0	5	10	0	20	30
15	1	1	0	2	10	0	8	18
16	3	2	1	2	50	30	8	88
17	1	2	0	2	20	0	8	28
18	1	1	0	4	10	0	16	26
19	1	2	1	1	20	30	4	54
20	3	2	0	1	50	0	4	54

3-Se ordenan de mayor a menor puntuación.

El tiempo medio estimado de desplazamiento se considera de 15 minutos, al tratarse de un centro histórico.

Tabla 3. Ponderación de cada Poi del ejemplo ordenada de mayor a menor

Poi	NTP	EC	PN	HL	Pond. NTP	Pond. PN	Pond. HL	Pond Total
16	3	2	1	2	50	30	8	88
13	3	3	1	1	50	30	4	84
3	1	3	1	4	30	30	16	76
8	1	3	1	3	30	30	12	72
12	3	3	0	3	50	0	12	62
1	2	1	1	2	20	30	8	58
10	2	3	0	2	50	0	8	58
19	1	2	1	1	20	30	4	54
20	3	2	0	1	50	0	4	54
5	1	3	0	4	30	0	16	46
6	2	2	0	1	40	0	4	44
7	1	3	0	2	30	0	8	38
11	2	1	0	3	20	0	12	32

14	1	1	0	5	10	0	20	30
17	1	2	0	2	20	0	8	28
2	1	1	0	4	10	0	16	26
18	1	1	0	4	10	0	16	26
15	1	1	0	2	10	0	8	18

4-Se conocen los tiempos medios de estancia en cada Pol. Partiendo del tiempo total disponible (TTD) en minutos, se selecciona el Pol con mayor puntuación dentro de los Pol preseleccionados que entre en ese tiempo. En caso de empate se selecciona aquel que tenga mayor HL, luego que sea PN, con mejor EC. Si persiste el empate se hace aleatoriamente.

5-Se calcula de nuevo el tiempo total disponible como $TTD = TTD - \text{Tiempo de estancia} - 15$

6-Se elimina el Pol como preseleccionado.

7-Volver al paso 4 hasta que se haya agotado el tiempo.

Partiendo de 465 minutos disponibles (8 horas menos el primer desplazamiento desde el punto de inicio seleccionado por el usuario), y conociendo el tiempo de estancia medio de cada Pol:

Tabla 4. Selección de los Pols del ejemplo

Pol	Pond Total	Tiempo estancia	TTD
16	88	30	420
13	84	60	345
3	76	20	310
8	72	35	260
12	62	15	230
1	58	80	135
10	58	30	90
19	54	10	65
20	54	20	30
5	46	15	0
6	44	70	
7	38	20	
11	32	15	
14	30	10	
17	28	35	
2	26	25	
18	26	40	
15	18	30	

Aplicando los pasos descritos se han seleccionado 10 Pols para la ruta. Ahora habrá que pasar a la fase 2 del algoritmo para determinar el orden en que deben visitarse.

El Pol seleccionado como primera parada por el visitante va a ser el 13.

PASO 2:

1-Elaborar la tabla de distancias entre los Pols. Quitando mínimo por filas y por columnas se obtendrá la tabla de distancias con penalización. Para obtener la tabla con penalización, primero se va a obtener el mínimo de cada fila.

La tabla de distancias para este ejemplo es la siguiente, estando expresada en centenas de metros:

Tabla 5. Distancias mínimas entre los Pols del ejemplo

	16	13	3	8	12	1	10	19	20	5	min
16		8	15	13	21	18	24	28	30	30	8
13	8		7	5	13	10	16	18	20	22	5
3	15	7		12	15	17	11	27	29	29	7
8	13	5	12		8	15	21	25	27	27	5
12	21	13	15	8		23	26	33	35	35	8
1	18	10	17	15	23		6	10	12	12	6
10	24	16	11	21	26	6		16	18	18	6
19	28	18	27	25	33	10	16		2	14	2
20	30	20	29	27	35	12	18	2		13	2
5	30	22	29	27	35	12	18	14	13		12

A continuación se va a restar a cada elemento de la tabla, el mínimo de su fila correspondiente. Así mismo se va a obtener el mínimo por columna de la tabla resultante:

Tabla 6. Distancias mínimas entre los Pols del ejemplo quitando mínimo por fila

	16	13	3	8	12	1	10	19	20	5
16		0	7	5	13	10	16	20	22	22
13	3		2	0	8	5	11	13	15	17
3	8	0		5	8	10	4	20	22	22
8	8	0	7		3	10	16	20	22	22
12	13	5	7	0		15	18	25	27	27
1	12	4	11	9	17		0	4	6	6
10	18	10	5	15	20	0		10	12	12
19	26	16	25	23	31	8	14		0	12
20	28	18	27	25	33	10	16	0		11
5	18	10	17	15	23	0	6	2	1	
min	3	0	2	0	3	0	0	0	0	6

A continuación se va a restar a cada elemento de la tabla, el mínimo de su columna correspondiente, y ya se tendrá la tabla con penalización que se usará en el siguiente paso del algoritmo:

Tabla 7. Distancias con penalización entre los Pols del ejemplo

	16	13	3	8	12	1	10	19	20	5
16		0	5	5	10	10	16	20	22	16
13	0		0	0	5	5	11	13	15	11
3	5	0		5	5	10	4	20	22	16
8	5	0	5		0	10	16	20	22	16
12	10	5	5	0		15	18	25	27	21
1	9	4	9	9	14		0	4	6	0
10	15	10	3	15	17	0		10	12	6
19	23	16	23	23	28	8	14		0	6
20	25	18	25	25	30	10	16	0		5
5	15	10	15	15	20	0	6	2	1	

2-Agrupar los Pol, en función de los ceros de la tabla anterior empezando por la primera fila. Se van a agrupar juntos todos aquellos con los que tenga un cero, y aquellos que tengan un cero con éstos.

Primer grupo:

- El 16 con el 13.
- El 13 con 16, 3, 8.
- El 3 con el 13.
- El 8 con el 13 y con el 12.
- El 12 con el 8.

Estará formado por el 16 - 13 - 3 - 8 - 12

Segundo grupo:

- El 1 con el 10 y con el 5.
- El 10 con el 1.
- El 5 con el 1.

Estará formado por el 1 - 10 - 5

Tercer grupo:

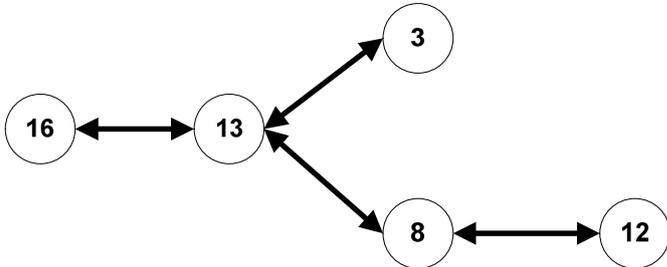
- El 19 con el 20.
- El 20 con el 19.

Estará formado por el 19 - 20.

3-Realizar el grafo asociado a cada agrupación. Se va a realizar el árbol de expansión mínima de cada grupo.

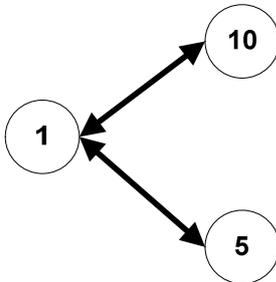
Grupo 1:

Figura 3. Árbol de expansión mínima del grupo 1



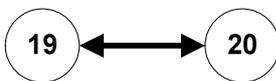
Grupo 2:

Figura 4. Árbol de expansión mínima del grupo 2



Grupo 3:

Figura 5. Árbol de expansión mínima del grupo 3



4-Determinar nodo de inicio y fin de la primera agrupación. Se comenzará por la red que contiene el Pol de inicio. Se parte de la tabla de distancias con penalización entre los Pols del grupo:

Tabla 8. Distancias con penalización entre los Pols del grupo 1

	16	13	3	8	12
16		0	5	5	10
13	0		0	0	5
3	5	0		5	5
8	5	0	5		0
12	10	5	5	0	

Se elimina como posibilidad el máximo por cada fila, excepto si es cero, de cara a no permitir los trayectos más largos.

Tabla 9. Distancias con penalización entre los Pols del grupo 1 eliminando máximo por fila

	16	13	3	8	12
16		0	5	5	∞
13	0		0	0	∞
3	∞	0		∞	∞
8	∞	0	∞		0
12	∞	5	5	0	

Se va a hacer un listado de todas las rutas permitidas que empiezan por el nodo de inicio (Pol 13), y se seleccionará aquella con menor recorrido. En caso de empate se escoge aleatoriamente.

Posibilidades:

13 16 3 ... Descartada por qué del 3 solo puede ir al 13.

13 16 8 12 3 con un recorrido total de 10.

13 3 ... Descartada por qué del 3 solo puede ir al 13.

13 8 12 3 ... Descartada por qué del 3 solo puede ir al 13.

En este caso la única ruta permitida será:

13 - 16 - 8 - 12 - 3

5-Determinar el siguiente grupo a unir con el nodo de fin del grupo que se acaba de determinar su orden: Se verá qué nodo de un grafo se une con el nodo de fin del otro grafo. Realizar tantas veces como sea necesario hasta que todos los grupos estén unidos.

Se van a analizar ahora las distancias desde el nodo de fin del grupo 1 (Pol 3) hasta los demás nodos de los otros grupos. Aquella que sea menor, señalará el siguiente grupo que se unirá con el primero. En caso de empate, se selecciona aquel grupo que tenga más Pol. En caso de empate dentro de un mismo grupo se seleccionará aleatoriamente.

Tabla 10. Distancias penalizadas desde el Pol 3 a los Pols del resto de grupos

	1	10	19	20	5
3	10	4	20	22	16

El mínimo corresponde al Pol número 10. Ahora habrá que poner el orden de los Pols a recorrer del grupo donde se encuentra el nodo 10 (Grupo 2), formado por los Pols 1, 5 y 10.

Se elimina como posibilidad el máximo por cada fila, excepto si es cero, de cara a no permitir los trayectos más largos. Se comienza con el Pol 10.

Tabla 11. Distancias con penalización entre los Pols del grupo 2 eliminando máximo por fila

	1	10	5
1		0	0
10	0		∞
5	0	∞	

La única ruta posible empezando por el Pol 10 es:
10 - 1 - 5

Se van a analizar ahora las distancias desde el nodo de fin del grupo 2 (Pol 5) hasta los demás nodos de los otros grupos. Aquella que sea menor, señalará el siguiente grupo que se unirá con el primero. En caso de empate, se selecciona aquel grupo que tenga más Pol. Ya solo queda por unir el grupo 3, formado por los Pols 19 y 20. Las distancias penalizadas desde el Pol 5 son:

Tabla 12. Distancias penalizadas desde el Pol 5 a los Pols del resto de grupos

	19	20
5	2	1

Orden Grupo 3:
20- 19

6-Unir todos los Pol para determinar la ruta turística personalizada. Quedará:
13 - 16 - 8 - 12 - 3 - 10 - 1 - 5 - 20 - 19

La distancia total recorrida será:

Tabla 13. Distancia total recorrida del ejemplo

	16	13	3	8	12	1	10	19	20	5
16		8	15	13	21	18	24	28	30	30
13	8		7	5	13	10	16	18	20	22
3	15	7		12	15	17	11	27	29	29
8	13	5	12		8	15	21	25	27	27
12	21	13	15	8		23	26	33	35	35
1	18	10	17	15	23		6	10	12	12
10	24	16	11	21	26	6		16	18	18
19	28	18	27	25	33	10	16		2	14
20	30	20	29	27	35	12	18	2		13
5	30	22	29	27	35	12	18	14	13	

Distancia total recorrida = 8 + 13 + 8 + 15 + 11 + 6 + 12 + 13 + 2 = 88

La distancia media entre dos Pols cualesquiera es de 18,5 mientras que la distancia media de la ruta propuesta por el algoritmo es de 9,8. Se ve que se logra un muy buen resultado, siendo la distancia media entre dos Pols prácticamente la mitad.

V. CONSIDERACIONES FINALES

Se ha presentado un nuevo algoritmo heurístico para obtener rutas personalizadas en el que se tienen en cuenta factores tan importantes como son:

- las tipologías de patrimonio asociadas a cada Pol.
- el estado de conservación de cada Pol.
- si el Pol es Patrimonio nacional.
- si el Pol es considerado *highlight*.
- las distancias entre cada uno de los Pol.
- los tiempos estimados de visita de cada Pol.

En base a estos factores se van a determinar los Pols a visitar en función la nueva ponderación introducida en este algoritmo asociada a cada Pol.

El usuario determinará los siguientes aspectos de la ruta:

- Tiempo estimado que tiene para la ruta.
- El punto de inicio de la misma, que debe ser un Pol.
- Las tipologías de patrimonio en qué está interesado.

El algoritmo va a agrupar a los Pols en función de las distancias entre ellos, y luego une estos grupos de Pols entre sí, encontrando una ruta personalizada que cumple con los siguientes criterios:

- Los Pols a visitar siempre estarán asociados a las tipologías que quiere el usuario.
- Se va a tener en cuenta la importancia que puede tener el Pol desde un punto de vista externo, como es el hecho de ser Patrimonio Nacional, así como interno debido a la calificación asociada al factor Highlight.
- Se considera el estado de conservación del Pol.
- La distancia a recorrer se logra que sea bastante cercana al óptimo, pero normalmente sin serlo, lo que logra que sea un algoritmo fácil de implementar y que no requiera muchos recursos, ni emplee tiempos demasiado grandes para una aplicación web o móvil.
- Se utiliza una estimación tanto de tiempo de estancia como de tiempos de desplazamiento

Todos estos criterios hacen que el nuevo algoritmo creado proporcione una ruta personalizada al usuario muy centrada en sus preferencias por un lado, y de fácil

implementación por el otro. Es de difícil comparación con otros algoritmos ya que tiene en cuenta factores distintos.

Con el presente algoritmo se logra obtener una ruta personalizada con los factores enumerados en las conclusiones. Se pretende implementar en el nuevo sistema de información turística que se tiene previsto desarrollar en el futuro de este proyecto.

Como futura línea de investigación se va a tener en cuenta en primer lugar la evaluación del presente algoritmo en un entorno real de un centro histórico de una ciudad todavía por determinar. También se va a considerar integrar y tener actualizados en la base de datos otros datos de los Pols, como son el precio y el horario de cada uno de los Pols, actualizando el algoritmo para incluirlos. Otra posibilidad que también se tiene presente de cara al futuro es la de cada Pol tenga dos tiempos asociados, uno a la visita por dentro del Pol, y otro a si sólo se quiere visitar por fuera. En este sentido sería muy interesante la personalización de los tiempos de visita de cada Pol por parte del usuario.

Otra opción que se quiere contemplar en el futuro es la posibilidad de ofertar al turista más una ruta, de manera que se puede elegir una ruta determinada de un conjunto propuesto en función de sus preferencias.

Con estas aportaciones se lograría una mejor adaptación al usuario, pero también implicaría una mayor complejidad del algoritmo en términos de recursos y tiempos necesarios para obtener la ruta personalizada.

AGRADECIMIENTOS

Este artículo se enmarca en el proyecto de investigación SIT-MAD, financiado por la Fundación Hergar.

BIBLIOGRAFÍA

- ASCHEUER, N., FISCHETTI, M., & GRÖTSCHEL, M. (2001). Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, 90(3), 475-506.
- CABALLERO, R., GONZÁLEZ, M., MOLINA, J., PELÁEZ, A., & RODRÍGUEZ, B. (2005). Planificación de rutas turísticas bajo un enfoque multicriterio. XIII Jornadas de ASEPUMA, La Coruña.
- DE CHOUDHURY, M., FELDMAN, M., AMER-YAHIA, S., GOLBANDI, N., LEMPEL, R., & YU, C. (2010). Constructing travel itineraries from tagged geo-temporal breadcrumbs. In *Proceedings of the 19th international conference on World wide web* (pp. 1083-1084). ACM.

- DESCAMPS-VILA, L., CASAS ROMA, J., CONESA CARALT, J., & PÉREZ NAVARRO, A. (2012a). Problemas en la implementación de algoritmos de routing de alta complejidad en dispositivos móviles: el caso Itiner@. VI Jornadas de SIG libre, Girona.
- _____ (2012b). Rutas turísticas personalizadas en dispositivos móviles sin necesidad de conexión a internet: Itiner@. VI Jornadas de SIG libre, Girona.
- GARCÍA, A., LINAZA, M. T., ARBELAITZ, O., & VANSTEENWEGEN, P. (2009). Intelligent Routing System for a Personalised Electronic Tourist Guide. En Höpken, W., Gretzel, U., Law, R. (Eds.), *Information and Communication Technologies in Tourism* (pp. 185-197). Springer Vienna. DOI. 10.1007/978-3-211-93971-0_16
- GARCÍA, A., LINAZA, M. T., ARBELAITZ, O., FUCHS, M., RICCI, F., & CANTONI, L. (2012). Evaluation of intelligent routes for personalised electronic tourist guides. na.
- GAVALAS, D., KENTERIS, M., KONSTANTOPOULOS, C., & PANTZIOU, G. (2012). An efficient algorithm for recommending personalized mobile tourist routes. *IET Software*, 6(4).
- GOYAL, S. K. (1971). The tree development method for solving the travelling-salesman problem. *The International Journal of Production Research*, 9(2), 239-246.
- HINCAPIÉ, R. A., PORRAS, C. A. R., & GALLEGO, R. A. (2004). Técnicas heurísticas aplicadas al problema del cartero viajante (TSP). *Scientia et Technica*, 1(24).
- KUO H., HORNG S.J., KAO TW, LIN T.L. ET AL. (2010) Hybrid swarm intelligence algorithm for the travelling salesman problem. *Expert Syst Appl* 27(3), 166–179.
- ALMEIDA, A. N. D., VALLE, A. G. D., & CRESPO-PEREIRA, D. (2012). ELOCONS: Un algoritmo de construcción de rutas eficiente para la pequeña y mediana empresa de distribución. *Dyna*, 87(2), 222-228.
- OCEGUEDA HERNÁNDEZ, V. C. (2012). *OntPersonal: ontología de personalización para ITINER@, un sistema generador de rutas turísticas basado en información semántica* (Trabajo final de Máster). Universitat Oberta de Catalunya
- SOUFFRIAUX, W., VANSTEENWEGEN, P., VERTOMMEN, J., BERGHE, G. V., & OUDHEUSDEN, D. V. (2008). A personalized tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence*, 22(10), 964-985.
- TAI, C. H., YANG, D. N., LIN, L. T., & CHEN, M. S. (2008). Recommending personalized scenic itinerary with geo-tagged photos. In Multimedia and Expo, 2008 IEEE International Conference on (pp. 1209-1212). IEEE.
- WALSHAW, C. (2002). A multilevel approach to the travelling salesman problem. *Operations Research*, 50(5), 862-877.
- YANG, T. & HO, L. (2006). The Comparison and Study of Shortest Path Algorithm in Heaps – Using a Taiwan Route Map as an Example. *The Journal of American Academy of Business*, 9(1). Cambridge.
- YANG, Y., DI, J., WANG, J., & LI, F. (2012). Optimized Traveling Route Scheme based on Improved Prim Algorithm. *International Journal of Digital Content Technology and its Applications*, 5(16).