

PLANIFICACIÓN DE VIAJES TURÍSTICOS CON PREFERENCIAS Y RESTRICCIONES DIFUSAS

J. Brito, A. Expósito, J.A. Moreno

Departamento de Ingeniería Informática y de Sistemas

Universidad de La Laguna, E-38271 La Laguna, España

ABSTRACT

Tourists plan trips based on their personal preferences and the point of interest information at destination. This paper propose the formulation of optimization problem for their multi-day planning sightseeing where the preferences of the tourists on the points of interest and travel time constraints are fuzzy. The soft computing methodology proposed to solve the problem integrates fuzzy optimization methods with a constructive metaheuristic. Computational results on instances from the team orienteering problem benchmarks are analyzed.

KEYWORDS: optimización difusa, planificación de viajes turísticos, problema de orientación en equipos.

MSC: 90C70

RESUMEN

Los turistas planifican un viaje basándose en sus preferencias personales y en la información de puntos de interés del destino. Este trabajo propone un problema de planificación de rutas turísticas de múltiples días donde las preferencias sobre los puntos de interés y las restricciones de duración de las rutas son consideradas difusas. Una metodología de Soft Computing es propuesta para resolver este modelo que integra métodos de optimización difusa con una metaheurística constructiva.

1. INTRODUCCIÓN

El sector del turismo en la sociedad actual tiene un gran impacto, prácticamente en toda actividad productiva [3]. De acuerdo a los datos provenientes del World Travel & Tourism Council (WTTC), en este momento, el sector provee 266 millones de puestos de trabajo en el mundo, o de otra manera 1 de cada 11 puestos de trabajo son en el sector turístico. El aumento de la demanda y la satisfacción de las necesidades del turista hace que la denominada industria del turismo tenga un crecimiento continuo. La heterogeneidad de turistas y sus distintos perfiles durante la visita o estancia exigen disfrutar de una variada oferta de servicios e iniciativas. A fin de lograr una industria competitiva en los destinos turísticos, las diferentes ofertas se deben integrar orientados al cliente, como estrategia diferencial del destino, tematizadas o simplemente como recursos propios de ocio. Generalmente, multitud de puntos de interés (PI) o atracciones turísticas se encuentran dispersos a lo largo del destino turístico. Estos puntos representan el foco de atención e interés del turista por multitud de factores. Algunos de ellos pueden ser por su valor histórico, cultural o belleza natural. Desafortunadamente, las estancias que realizan los turistas en el punto de destino se encuentran limitadas en tiempo. Es por ello que se considera un reto maximizar la satisfacción de las demandas de los turistas que llegan a un destino para visitar determinados PI en una estancia limitada en el tiempo.

El diseño y desarrollo de sistemas de planificación de viajes turísticos es un área de interés para el sector del turismo y la ingeniería informática. Los turistas planifican un viaje basándose en sus preferencias personales y en la información de PI del destino. Como se apuntaba anteriormente, en general, no es posible visitar todos los PI en un periodo limitado. La selección de los PI a visitar durante las vacaciones y el establecimiento de las rutas con la secuencia de puntos a visitar cada día, dentro de las limitaciones temporales, es un problema de interés. Se pretende encontrar el conjunto más adecuado de PI a visitar, considerando los múltiples criterios en conflicto derivados de las preferencias del turista y sus limitaciones temporales. De acuerdo a esta descripción, el problema de la generación de rutas turísticas personalizadas ha sido definido como Tourist Trip Design Problem (TTDP) [9]. En este problema disponemos de una serie de PI candidatos con una serie de atributos asociados (localización, tiempo de visita, horarios, etc.). Para resolver este problema necesitamos conocer el tiempo de viaje entre los diferentes PI, generalmente información multi-modal dado que los turistas pueden usar distintos

medios de transporte; interesa la información sobre el beneficio, recompensa o puntuación asociada a cada PI como función de la importancia o preferencia objetiva y subjetiva dada por el turista a los mismos; además el número de rutas a generar, asociada con cada uno de los días y periodo de estancia del turista; Así como el tiempo máximo que un turista desea invertir en las visitas a los PI en cada ruta. Con esta información se puede resolver el TTDP, generando las rutas asociadas a las visitas de los PI, las preferencias de interés establecidas y respetando las restricciones temporales consideradas.

La formulación y formalización más simple asociada al TTDP es el Problema de Orientación (Orienteering Problem, OP) o el Problema de Orientación en Equipo (Team Orienteering Problem, TOP) [5]. TOP introducido por Chao, Golden, and Wasil [1], en contraste al OP que solo considera una ruta, es competente para planificar simultáneamente un cierto número de rutas. TOP es un problema de planificación de rutas donde el objetivo es maximizar el total de recompensas recibidas al visitar localizaciones. además en este problema de optimización combinatoria, la duración total de cada ruta está limitada y no todas las localizaciones tienen que ser visitadas. Estos problemas de optimización en la medida que capturan características reales del problema, en cuanto al número y el tamaño de las variables, se vuelven más complejos y pertenecen a la clase de problemas NP-duros. En estos casos la incapacidad de las aproximaciones exactas para resolver problemas de mediana y larga escala hacen que heurísticas y metaheurísticas se consideren adecuados para la búsqueda de soluciones óptimas. Vansteenwegen [6] ofrece una visión general de la investigación existente sobre OP y TOP, incluyendo métodos de solución.

Como se describía anteriormente, los itinerarios formados a partir del conjunto de PI deben tener en cuenta entre otros, el tiempo de ir de un lugar a otro, la duración de la visita en cada PI y el tiempo total de cada ruta [7]. La planificación resultante tiene dos importantes características, la información disponible del destino es a menudo inconexa e imprecisa y hay un alto grado de flexibilidad por parte del usuario para planificar el tiempo de duración de las rutas y el conjunto de itinerarios. Es evidente que existe incertidumbre en muchos de los datos de entrada (el tiempo de ir de un lugar a otro, tiempo de visita, horarios, etc.), los cuales dependen de las condiciones del entorno y el estado del tráfico, carreteras o del tiempo. También la especificación de objetivos y restricciones que definen el problema, como son la expresión de preferencias de los clientes o la duración de la ruta tienen vaguedad e imprecisión. En ambos casos, incertidumbre asociada a los términos lingüísticos utilizados en la expresión de los mismos. En este contexto es necesario investigar modelos y metodologías que soporten nuevas aplicaciones en la organización y planificación de la experiencia de los usuarios, las cuales pueden incorporar imprecisión y flexibilidad.

La Soft Computing se presenta como una familia adecuada de modelos y métodos que dan respuesta a estas características. La Soft Computing es una asociación de sinergias derivadas de diferentes metodologías, entre las que destacan la lógica difusa, las redes neuronales, el razonamiento aproximado y las metaheurísticas [2]. En particular, los sistemas y conjuntos difusos proporcionan una aproximación metodológica adecuada para tratar con la incertidumbre producto de la naturaleza imprecisa de la información y las decisiones. Igualmente las metaheurísticas son un buen referente dado que ofrecen estrategias y soluciones eficientes que se integran con otras herramientas de Soft Computing, facilitando soluciones aproximadas a problemas complejos del mundo real [2]. La hibridación inteligente de estas metodologías produce mecanismos de razonamiento y métodos de búsqueda que nos permiten combinar el conocimiento del dominio en el que estemos trabajando con los datos experimentales que tengamos, para obtener nuevas herramientas de computación flexibles con las que se resuelven problemas complejos, muy difíciles de resolver con otros enfoques menos tolerantes.

En líneas generales, este trabajo propone un problema de planificación de rutas turísticas de múltiples días donde las preferencias sobre los puntos de interés y las restricciones de duración de las rutas son consideradas difusas. Este modelo difuso propuesto es transformado en un modelo de optimización equivalente usando las ideas desarrolladas por Verdegay et al. [10]. Esta aproximación proporciona métodos para resolver estos problemas, en un camino directo y simple, obteniéndose soluciones que son coherentes con su naturaleza difusa. Respecto de la formulación del problema se propone hacer uso de una variante del TOP para afrontar la problemática de planificación en múltiples rutas. En cuanto a la resolución del mismo, se aplica la metaheurística constructiva denominada Greedy Randomized Adaptive Search Procedure (GRASP).

El resto del trabajo está organizado como sigue. La siguiente sección describe la formulación del modelo. La sección 3 explica la aproximación de optimización difusa para resolver esta variante difusa del TOP. En la sección 4 presentamos la metaheurística GRASP para encontrar soluciones. En la sección

5 es descrita la experimentación computacional y sus resultados. Finalmente, se incluyen unas conclusiones en la última sección.

2. FORMULACIÓN DEL MODELO DIFUSO

En el TOP dado un número de n localizaciones, cada una con una puntuación asociada, y un equipo de m rutas, el objetivo consiste en maximizar la puntuación total en un periodo de tiempo. En nuestro caso la puntuación que representa preferencias, es imprecisa y la restricción de limitación de tiempo es flexible, es decir admite una cierta tolerancia. Ambos casos pueden ser expresados en términos difusos, como números difusos y restricciones difusas, respectivamente. Las localizaciones son identificadas por un índice i , $i = 1, 2, \dots, n$. Los índices 1 y n representan la localización de inicio y final, las cuales son las mismas para cada rutas. Cada ruta es representada por un índice k , $k = 1, 2, \dots, m$. La puntuación obtenida en cada localización es s_i . En este modelo las puntuaciones son números difusos \tilde{s}_i . Cada localización tiene un tiempo de estancia r_i . El tiempo de viaje desde el nodo i al nodo j se denota por t_{ij} . El tiempo límite de duración de cada ruta es T_{max}^k .

El modelo contiene tres conjuntos de variables de decisión u_i^k , x_{ij}^k y y_i^k , $i, j = 0, 1, \dots, n$, $k = 1, 2, \dots, m$; u_i^k define la posición de la localización i en la ruta k ; $x_{ij}^k = 1$ si y solo si la ruta k va desde la localización i a j y $x_{ij}^k = 0$ en otro caso; $y_i^k = 1$ si la localización i es visitada en la ruta k y $y_i^k = 0$ en otro caso. El TOP es formulado como un problema de programación lineal como sigue:

Maximizar

$$\sum_{k=1}^m \sum_{i=2}^{n-1} \tilde{s}_i y_i^k \quad (1)$$

Sujeto a:

$$\sum_{k=1}^m \sum_{j=2}^{n-1} x_{1j}^k = \sum_{k=1}^m \sum_{i=2}^{n-1} x_{in}^k = m \quad (2)$$

$$\sum_{k=1}^m y_i^k \leq 1 \quad i = 2, \dots, n-1 \quad (3)$$

$$\sum_{i=1}^{n-1} x_{il}^k = \sum_{j=2}^n x_{lj}^k = y_l^k \quad l = 2, \dots, n-1, k = 1, \dots, m \quad (4)$$

$$\sum_{i=1}^{n-1} (r_i y_i^k + \sum_{j=2}^n t_{ij} x_{ij}^k) \leq_f T_{max}^k \quad k = 1, \dots, m, \quad (5)$$

$$2 \leq u_i^k \leq n \quad i = 2, \dots, n, k = 1, \dots, m \quad (6)$$

$$u_i^k - u_j^k + 1 \leq (n-1)(1 - x_{ij}^k) \quad k = 1, \dots, m, i, j = 2, \dots, n \quad (7)$$

$$x_{ij}^k, y_i^k \in \{0, 1\} \quad k = 1, \dots, m, i, j = 1, \dots, n. \quad (8)$$

La ecuación (1) representa la función objetivo que maximiza el total de la puntuación o recompensa obtenida. La restricción (2) garantiza que cada ruta empieza en la localización 1 y finaliza en la localización n . Las restricciones (3) establecen que toda localización sea visitada al menos una vez. Las restricciones (4) aseguran que, si una localización es visitada en una ruta, es precedida y seguida exactamente por otra en la misma ruta. Las restricciones (5) establecen el límite de tiempo de duración de cada ruta. Nota que el símbolo \leq_f denota que las restricciones son consideradas difusas. Las restricciones (6) y (7) son necesarias para evitar subrutas. Finalmente (8) establece las condiciones de las variables.

3. APROXIMACIÓN DE OPTIMIZACIÓN DIFUSA

En la sección previa formulamos un modelo TOP como un problema de programación lineal difusa con coeficientes difusos en la función objetivo y restricciones difusas. La aproximación de optimización

difusa propuesta [4] proporciona métodos para resolver el TOP con términos difusos. La metodología consiste primero en encontrar soluciones factibles, es decir resolver el problema de optimización con restricciones difusas, y segundo, resolver el problema con coeficientes difusos en la función objetivo. En el caso de restricciones difusas, la región factible puede ser definida con un conjunto difuso, con una determinada función de pertenencia. En la factibilidad difusa no es necesario que una solución difusa satisfaga completamente y de forma precisa las restricciones. En particular, Verdegay [11], usando el teorema de representación para conjuntos difusos, prueba que las soluciones para el caso de funciones lineales puede ser obtenida a partir del siguiente modelo auxiliar:

$$\begin{aligned} \text{Maximizar} \quad & z = cx \\ \text{subject to} \quad & Ax \leq b + \tau(1 - \alpha) \\ & x \geq 0, \alpha \in [0, 1] \end{aligned} \quad (9)$$

donde $\tau = (\tau_1, \tau_2, \dots, \tau_m) \in \mathfrak{R}^m$ es el vector de nivel de tolerancia.

En nuestro caso, utilizamos esta aproximación para obtener un modelo equivalente que resuelve el TOP con restricciones difusas reemplazando (5) por las siguientes restricciones:

$$\sum_{i=1}^{n-1} (r_i y_i^k + \sum_{j=2}^n t_{ij} x_{ij}^k) \leq_f T_{max}^k + \tau^k(1 - \alpha), \quad k = 1, \dots, m, \quad (10)$$

Así, podemos utilizar este nuevo modelo para obtener un rango de soluciones variando α .

El siguiente paso es resolver un modelo con coeficientes difusos en la función objetivo. Una cantidad vaga e imprecisa puede ser modelada haciendo uso del concepto de número difuso. Para la comparación de números difusos se necesita un método de ordenación difuso. El método propone el uso de una función ordenadora que permite la comparación, lo cual facilita la maximización de la función objetivo. Así, la función objetivo (1) puede ser reemplazada por:

$$\sum_{k=1}^m \sum_{i=2}^{n-1} g(\tilde{s}_i) y_i^k \quad (11)$$

En este trabajo usamos números difusos triangulares para modelar la imprecisión en las puntuaciones que expresan preferencias, el modelo más simple de número difuso con un soporte y moda dados. Así un decisor puede estimar subjetivamente estas puntuaciones, utilizando sus preferencias y el conocimiento disponible sobre los puntos de interés, como números triangulares $Tr(T_{min}, T_{mod}, T_{max})$ asignando los intervalos $[T_{min}, T_{max}]$ de los límites mínimos y máximos de puntuación y el valor T_{mod} más razonable.

Existen múltiples enfoques y técnicas disponibles para la ordenación y comparación de números difusos. En este trabajo utilizaremos un método de comparación contrastado en la literatura y con fácil interpretación, el tercer índice de Yager. Ese corresponde con los métodos basados en la definición de una función ordenadora, utiliza una función ordenadora f que permite establecer que $\tilde{u}_1 \leq \tilde{u}_2$ cuando $f(\tilde{u}_1) \leq f(\tilde{u}_2)$

Aplicando este método a números difusos triangulares $\tilde{u} = Tr(u_1, u_2, u_3)$ y $\tilde{v} = Tr(v_1, v_2, v_3)$ se obtiene el siguiente resultado:

$$\tilde{u} \leq_Y \tilde{v} \quad \text{si} \quad u_1 + 2u_2 + u_3 \leq v_1 + 2v_2 + v_3 \quad (12)$$

$$(13)$$

Utilizando el método de ordenación propuesto se obtiene la siguiente función objetivo:

$$\sum_{k=1}^m \sum_{i=2}^{n-1} (s_i^1 y_i^k + 2s_i^2 y_i^k + s_i^3 y_i^k) \quad (14)$$

donde \tilde{s}_i es un número triangular difuso $Tr(s_i^1, s_i^2, s_i^3)$

4. SOLUCIÓN GRASP

Con el objetivo de resolver el modelo anterior proponemos hacer uso de la metaheurística GRASP. GRASP es una metaheurística multi-arranque en dos fases para optimización combinatoria propuesta por Feo and Resende [8]. Básicamente, ésta se encuentra compuesta de una fase constructiva y una fase de mejora a través de una búsqueda local. En el primer paso una solución factible es construida en la fase constructiva. En el siguiente paso el vecindario de dicha solución es explorado hasta que un mínimo local es encontrado en la búsqueda local. El pseudocódigo mostrado en el Algoritmo 1 ilustra las principales fases de GRASP donde el parámetro *maxIterations* es el número máximo de iteraciones del procedimiento. La fase constructiva del GRASP es mostrada en el Algoritmo 2.

Algorithm 1 GRASP

```
function GRASP(maxIterations)  
  readInput()  
  for it = 1 to maxIterations do  
    begin  
      solution = GRASPConstructPhase()  
      solution = localSearch(solution)  
      updateSolution(solution, bestSolution)  
    end
```

El mecanismo de construcción de soluciones genera una solución paso a paso añadiendo de manera aleatoria un nuevo elemento proveniente de una lista de candidatos Restricted Candidate List (RCL) a la solución actual en construcción. Durante este proceso se mantiene la factibilidad de la solución. La lista de candidatos es generada con todos los posibles PI candidatos a formar parte de la solución.

Algorithm 2 GRASP - Construction phase

```
function GRASPConstructPhase(maxIterations)  
  Let  $x_0$  be an initial empty partial solution. Set  $j = 0$ .  
  repeat  
    begin  
      Construct the restricted candidate list RCL with the nearest available PIs to the last element  $PI_j$   
      Choose at random an element  $PI_{j+1}$  from RCL  
      Update the partial solution by  $x_{j+1} = x_j + \{PI_{j+1}\}$   
      Set  $j = j + 1$   
    end  
  until there is no element to include in the partial solution
```

Ésta se encuentra ordenada de acuerdo a la evaluación de la inserción de cada candidato en la solución parcial. Dicha evaluación se realiza a través de una función greedy. En este sentido, la lista es ordenada de manera descendente de acuerdo a la puntuación asociada a cada PI o en orden ascendente de acuerdo al tiempo de viaje, con la puntuación más alta o el tiempo de viaje más bajo en la primera posición de la lista de candidatos. La RCL es generada limitando en tamaño esta lista, formando una lista con los mejores candidatos.

A continuación, la fase de búsqueda local es aplicada con el objetivo de mejorar la solución. Normalmente, un algoritmo de búsqueda local trabaja iterativamente reemplazando la solución actual por una mejor solución encontrada en su vecindario. El procedimiento termina cuando no se encuentra una mejor solución en su vecindario. El Algoritmo 3 muestra el pseudocódigo básico del algoritmo de búsqueda local. Nuestra propuesta de búsqueda local hace uso de movimientos de intercambio entre PIs de diferentes rutas con el objetivo de reducir el tiempo de viaje total de ruta. De manera que se exploran todos los vecindarios y la solución actual es reemplazada por una mejor encontrada. En caso de que este primer paso sea capaz de reducir el tiempo total de desplazamiento de ruta, la búsqueda local intenta insertar un nuevo PI en la solución con el objetivo de maximizar la puntuación final. En términos generales, la fase constructiva y la búsqueda local intenta maximizar la puntuación asociada a la solución. Este proceso iterativo de dos fases continua hasta alcanzar el criterio de finalización impuesto.

Algorithm 3 GRASP - Local Search Improvement Phase

```
function localSearch(solution)  
repeat  
  begin  
    Find the best neighbor of solution  
    If  $f(\text{neighbor}) < f(\text{solution})$  do  $\text{solution} = \text{neighbor}$   
  end  
until no better neighbor is found  
return solution
```

5. EXPERIMENTACIÓN Y RESULTADOS

Esta sección describe los resultados de la experimentación computacional realizada, cuya finalidad es evaluar la bondad de la aproximación propuesta y su comportamiento cuando es usada para resolver el TOP con coeficientes difusos en la función objetivo y restricciones difusas.

En la experimentación se han utilizado seis instancias diferentes con el propósito de valorar el comportamiento. Las instancias utilizadas son parte de las instancias propuestas por el grupo de Tsiligrídes para el TOP. Específicamente en esta experimentación, instancias de los sets 1 y 2 han sido usadas. Dichas instancias proveen los datos de las posiciones de cada una de las localizaciones y la puntuación asociada a éstas. El número máximo de rutas de la solución también está incluida. En concreto, se han elegido instancias de tamaños 32 para el set 1 y 21 para el set 2. El número de localizaciones usada y el tiempo máximo por ruta varía de acuerdo a la instancia específica. Los detalles relativos a las instancias se puede ver la Tabla 1.

Instancia	Tamaño Rutas		Max. Tiempo
Set1			
<i>p1.2.q</i>	32	2	40.0
<i>p1.3.n</i>	32	3	23.3
<i>p1.4.r</i>	32	4	21.2
Set2			
<i>p2.3.c</i>	21	3	7.7
<i>p2.3.g</i>	21	3	10.7
<i>p2.4.k</i>	21	4	11.2

Tabla 1: Instancias propuestas por Tsiligrídes utilizadas en la experimentación

El nivel de tolerancia aplicado en la restricción de tiempo máximo de duración de la ruta es del 15% del máximo establecido, variando los valores de α en 0.0, 0.2, 0.4, 0.6, 0.8, y 1.0. además, la experimentación se ha realizado utilizando dos criterios para ordenar la RCL en el procedimiento GRASP, distancia y preferencia. El GRASP fue ejecutado 100 veces para cada instancia y criterio utilizado. Los resultados experimentales son presentados en las Tablas 2 y 3. La Tabla 2 contiene los resultados para el set 1 y la Tabla 3 para el set 2.

Instancias/ Alphas	Ordenadas por tiempo de viaje				Ordenadas por puntuación			
	Mejor		Media		Mejor		Media	
	Puntos	Tiempo	Puntos	Tiempo	Puntos	Tiempo	Puntos	Tiempo
p1.2.q								
$\alpha = 0.0$	557	79.5651	503.06	86.9984	488	81.1659	356.91	89.2527
$\alpha = 0.2$	557	76.4906	493.46	84.9098	459	82.6273	340.33	86.9833
$\alpha = 0.4$	557	76.3279	493.46	84.9098	419	78.4912	329.66	84.5675
$\alpha = 0.6$	548	73.8586	473.9	80.8122	419	78.3142	321.63	82.4763
$\alpha = 0.8$	528	72.5024	458.44	78.5176	390	76.1102	311.45	79.8964
$\alpha = 1.0$	537	69.7837	450.65	76.3227	379	73.7052	299.58	77.5354
p1.3.n								
$\alpha = 0.0$	557	68.296	496.28	74.6006	428	73.5801	320.94	77.0679
$\alpha = 0.2$	557	62.8477	487.92	72.3623	388	71.0056	309.51	75.0324
$\alpha = 0.4$	528	63.2642	475.24	70.2125	399	68.3635	305.87	72.7552
$\alpha = 0.6$	508	61.6263	466.76	69.0324	388	66.4739	298.01	70.8119
$\alpha = 0.8$	499	60.5314	453.19	66.6983	350	64.7004	279.24	68.7615
$\alpha = 1.0$	497	59.7324	442.21	65.0466	339	61.8559	278.11	66.8751
p1.4.r								
$\alpha = 0.0$	557	72.5156	548.91	81.723	437	85.9482	409.93	82.1798
$\alpha = 0.2$	557	71.3264	544.57	80.8979	437	83.9941	401.33	89.7034
$\alpha = 0.4$	557	69.8098	544.48	80.3718	468	81.8712	385.42	87.5239
$\alpha = 0.6$	557	71.1368	534.82	78.6585	457	80.3369	379.52	85.4478
$\alpha = 0.8$	557	69.7452	528.84	77.4451	477	79.1568	357.42	83.0532
$\alpha = 1.0$	557	68.6597	523.42	75.8872	517	75.6559	347.95	80.3631

Tabla 2: Resultados de las instancias del set 1

Instancias/ Alphas	Ordenadas por tiempo de viaje				Ordenadas por puntuación			
	Mejor		Media		Mejor		Media	
	Puntos	Tiempo	Puntos	Tiempo	Puntos	Tiempo	Puntos	Tiempo
p2.3.c								
$\alpha = 0.0$	580	20.5834	556.2	23.6049	540	21.9527	443	24.4075
$\alpha = 0.2$	580	20.4015	539	22.3302	560	21.5639	438.2	23.4101
$\alpha = 0.4$	580	20.1872	533	21.5867	560	21.1244	435.4	22.7722
$\alpha = 0.6$	560	17.0472	509.8	20.8059	560	18.8301	488.6	21.7059
$\alpha = 0.8$	520	17.0472	502.8	20.0716	520	18.6417	486.8	21.3093
$\alpha = 1.0$	520	16.4762	488.4	19.4187	520	16.4762	489.6	19.9951
p2.3.g								
$\alpha = 0.0$	729	28.4727	692.41	33.3719	407	33.6866	342.62	34.6838
$\alpha = 0.2$	729	27.2814	686.02	32.5706	387	33.4487	333.65	34.2335
$\alpha = 0.4$	769	25.4712	668.78	31.9678	427	32.4379	351.07	33.6526
$\alpha = 0.6$	749	23.6949	672.64	31.1572	427	30.9211	354.07	32.6936
$\alpha = 0.8$	798	23.6949	656.18	29.8263	489	30.2205	374.27	31.8143
$\alpha = 1.0$	769	21.9635	628.38	28.2921	509	28.9669	406	30.7442
p2.4.k								
$\alpha = 0.0$	896	34.0561	860.87	43.9567	716	44.0036	555.86	46.327
$\alpha = 0.2$	896	33.2563	848.87	42.9975	676	43.3123	537.86	45.9003
$\alpha = 0.4$	896	32.7576	835.02	41.4816	696	42.9876	540.55	45.3562
$\alpha = 0.6$	896	33.0474	829.31	40.8078	656	42.6318	524	44.6924
$\alpha = 0.8$	896	32.0175	816.21	39.154	676	41.4511	524.77	43.8279
$\alpha = 1.0$	896	31.0515	809.04	37.9391	587	39.7018	477.85	42.2547

Tabla 3: Resultados de las instancias del set 2

En la Tablas 2 y 3 se presentan los valores de la función objetivo obtenida maximizando la puntuación total de las soluciones. Así se puede apreciar que si hacemos uso del tiempo como criterio de ordenación de la lista de candidatos se obtienen mejores soluciones que utilizando la puntuación. Esto es así porque con el criterio de tiempo, el procedimiento puede insertar más localizaciones en las rutas antes de violar la restricción de tiempo máximo de duración.

Como ejemplo, en las Figuras 1 y 2, se muestran gráficas comparativa para la instancia p1.2.q y en las Figuras 3 y 4 las gráficas comparativas para la instancia p1.3.n.

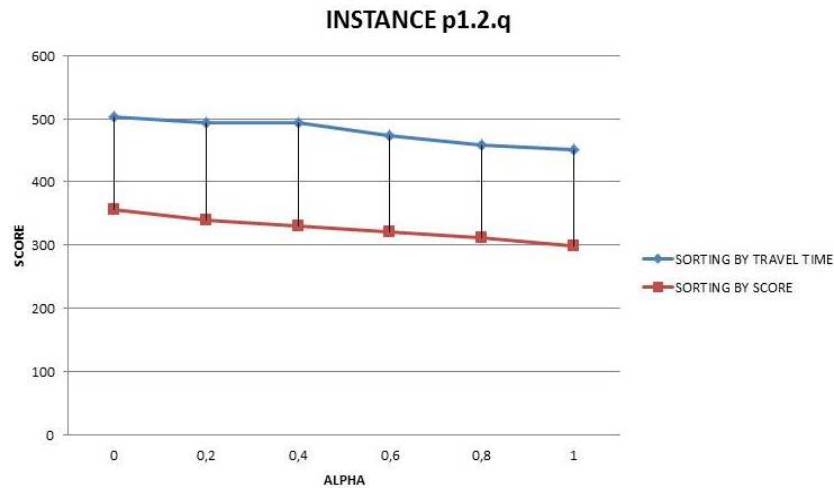


Figura 1: Comparativa de ordenaciones en base a la puntuación para la instancia p1.2.q

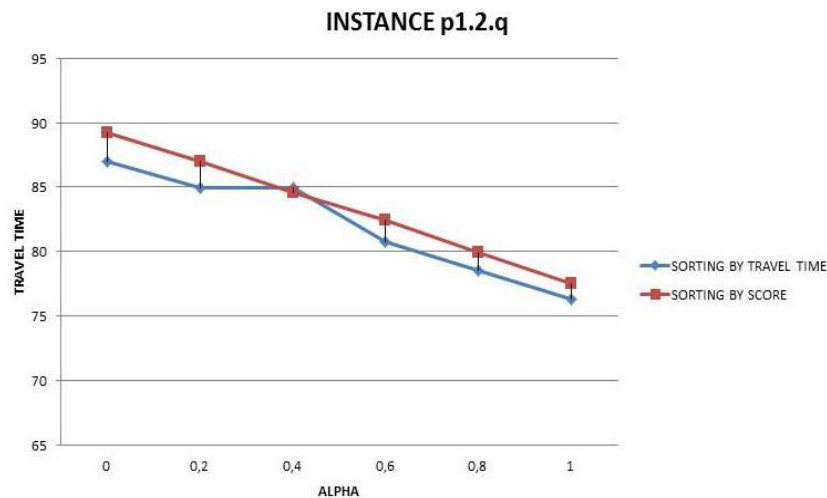


Figura 2: Comparativa de ordenaciones en base al tiempo de desplazamiento para la instancia p1.2.q

En las Figuras 1 y 2 se aprecia la comparativa para los distintos métodos de ordenación de la lista de candidatos. En el caso de la Figura 1 dicha comparativa se realiza en base a las puntuaciones obtenidas y respecto de la Figura 2 en base al tiempo de desplazamiento total obtenido. Estos datos pertenecen a la instancia p1.2.q. Adicionalmente, esta comparativa se realiza considerando los diferentes valores de α utilizados en la experimentación computacional. Referente a las Figuras 3 y 4, se muestran datos similares, con la salvedad de que pertenecen a la instancia p1.3.n.

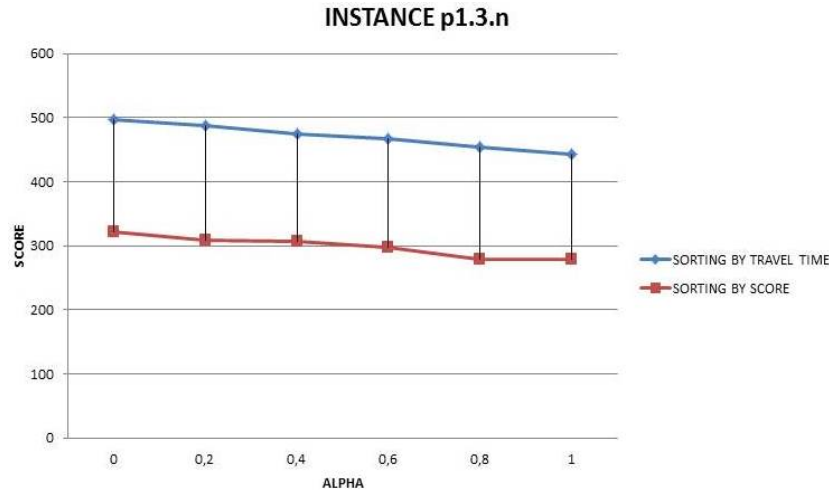


Figura 3: Comparativa de ordenaciones en base a la puntuación para la instancia p1.3.n

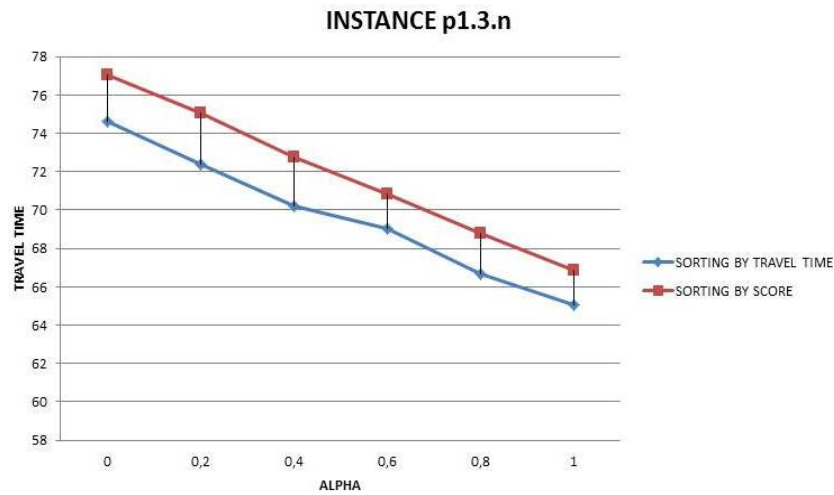


Figura 4: Comparativa de ordenaciones en base al tiempo de desplazamiento para la instancia p1.3.n

En base a los datos obtenidos para las instancias p1.2.q y p1.3.n, se aprecia en las Figuras 1, 2, 3 y 4, que la diferencia entre cada solución obtenida con distintos métodos de ordenación es mayor si se compara a la puntuación obtenida que al tiempo de desplazamiento total. Este hecho se aprecia independientemente del valor de α .

6. CONCLUSIONES

En este estudio presentamos una aproximación de Soft Computing para abordar un problema de planificación de viajes turísticos el cual es modelado como un TOP con preferencias y restricciones difusas. La experimentación computacional confirma que la aproximación propuesta es factible para resolver este modelo. Aplicando esta metodología un conjunto de soluciones diferentes son obtenidas, consistentes con su naturaleza difusa.

Futuros trabajos extenderán la experimentación con otras instancias, entre las cuales algunas pueden ser casos reales. También se estudiará el comportamiento de otras metaheurísticas, los tipos de números difusos utilizados para modelar y la inclusión de otros elementos difusos en el modelo. Una de las líneas de trabajo futuro interesantes y prioritaria es plantear el modelo multiobjetivo. Aunque la función objetivo planteada en el modelo es maximizar la puntuación obtenida en los PI, hemos encontrado que minimizando el tiempo de viaje en las rutas es posible agregar un mayor número de PI a

la solución y de esta manera aumentar también la puntuación total obtenida. Esta versión multiobjetivo consideraría maximizar la puntuación total obtenida y minimizar el tiempo de viaje en la función objetivo.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad de España (proyectos TIN2012-32608 and TIN2015-70226-R). Airam Expósito-Márquez agradece al Gobierno de Canarias el soporte financiero recibido a través del programa de formación del personal investigador. Agradecer además al IUDR (Instituto Universitario de Desarrollo Regional) el apoyo proporcionado.

RECEIVED: JULY, 2016
REVISED: DECEMBER, 2016

REFERENCIAS

- [1] I.CHAO, B. G. and E.WASIL [1996]: The team orienteering problem **European Journal of Operational Research**, 88:464–474.
- [2] J. L. VERDEGAY, R. R. YAGER, P. P. B. [2008]: On heuristics as a fundamental constituent of soft computingfuzzy sets and systems **Fuzzy Sets and Systems**, 159:846–855.
- [3] LAI, I. K. W. and HITCHCOCK, M. [2015]: Importance performance analysis in tourism: A framework for researchers. **Tourism Management**, 48, 10,:242–267.
- [4] M. DELGADO, J.L. VERDEGAY, M. V. [1989]: A general model for fuzzy linear programming **Fuzzy Sets and Systems**, 29:21–29.
- [5] P. VANSTEENWEGEN, D. V.-O. [2007]: The mobile tourist guide: an or opportunity **OR Insight**, 20:21–27.
- [6] P. VANSTEENWEGEN, W. SOUFFRIAU, D. V.-O. [2011a]: The orienteering problem: a survey **European Journal of Operational Research**, 209:1–10.
- [7] P. VANSTEENWEGEN, W. S. [2011b]: Trip planning functionalities: state of the art and future **Information Technology and Tourism**, 12:305–315.
- [8] T. FEO, M. R. [1995]: Greedy randomized adaptive search procedures **Journal Global Optimization**, 6:109–133.
- [9] VANSTEENWEGEN, P. and OUDHEUSDEN, D. V. [2007]: The mobile tourist guide: An or opportunity **Operational Research Insight**, 20:21–27.
- [10] VERDEGAY, J. [1982]: Fuzzy mathematical programming **in: Fuzzy Information and Decision Processes**, 12, pages 231–237.
- [11] VERDEGAY, J. [1995]: Fuzzy optimization: models, methods and perspectives **in: Proceeding 6thIFSA-95 World Congress**, pages 39–71.