Un enfoque metaheurístico para el Problema de Rutas de Orientación en Equipos con Tiempos de Servicio y Nodos Obligatorios e Incompatibles

C. Tobar-Fernández[1,4], A.D. López-Sánchez[2], J. Sánchez-Oro[1], and K. Sörensen[3]

[1] Departamento de Informática, Universidad Rey Juan Carlos
C/ Tulipán s/n, 28933 Móstoles, Madrid, España
c.tobar.2024@alumnos.urjc.es
[2] Departamento de Economía, Métodos Cuantitativos e Historia Económica,
Universidad Pablo de Olavide
Ctra. de Utrera km 1, 41013 Sevilla, España
adlope@upo.es
[3] Grupo de Investigación ANT/OR,
Universidad de Amberes, Amberes, Bélgica
kenneth.sorensen@uantwerpen.be
[4] Departamento de Investigación Operativa, OGA
C. Judería, 2, Planta 1, Módulo 3A, 41900 Camas, Sevilla, España
cristina.tobar@oga.ai

Resumen Este trabajo aborda el Problema de Rutas de Orientación en Equipos con Tiempos de Servicio, Nodos Obligatorios e Incompatibilidades físicas y lógicas (TOP-ST-MIN-PL), una variante del Problema de Rutas de Orientación en Equipos (TOP) que integra restricciones relevantes para contextos logísticos con múltiples vehículos. Para su resolución, se propone un algoritmo metaheurístico híbrido GRASP–ALNS (Greedy Randomized Adaptive Search Procedure – Adaptive Large Neighborhood Search), en el que una fase constructiva aleatorizada genera soluciones iniciales factibles y una búsqueda adaptativa en vecindarios grandes las mejora mediante operadores de destrucción y reparación. El problema tiene como objetivo maximizar el beneficio total obtenido por la flota, garantizando el cumplimiento de las restricciones de duración máxima de ruta, tiempos de servicio, visita obligatoria de ciertos nodos e incompatibilidades entre clientes. La evaluación computacional se realiza sobre un subconjunto representativo de instancias de referencia de la literatura y compara el desempeño del enfoque propuesto con un método exacto basado en planos de corte. Los resultados muestran que bajo un tiempo de ejecución de 1000s GRASP–ALNS obtiene soluciones factibles en 142 de las 150 instancias analizadas, frente a 99 alcanzadas por el método exacto, reduciendo el tiempo medio de cómputo de 382,16 a 3,89 segundos. Asimismo, el algoritmo presenta un gap medio del 2,55 %, lo que evidencia su capacidad para generar soluciones de alta calidad en tiempos reducidos. Estos resultados ponen de manifiesto la robustez y eficiencia del enfoque propuesto, especialmente en

instancias de mayor complejidad, donde los métodos exactos presentan mayores limitaciones computacionales.

Keywords: Optimización combinatoria · Team Orienteering Problem · Metaheurísticas híbridas · GRASP · ALNS · Eficiencia algorítmica

1. Introducción

El Problema de Rutas de Orientación en Equipos (TOP, por sus siglas en inglés Team Orienteering Problem), propuesto por primera vez por Chao et al. (5), es un conocido problema de optimización combinatoria en el que se debe planificar un conjunto de rutas para una flota de vehículos con el fin de maximizar la recompensa total recogida al visitar un subconjunto de nodos en una red, dentro de un límite de tiempo o distancia establecido para cada ruta. A diferencia del problema clásico de rutas de vehículos, el TOP normalmente no permite visitar todos los nodos debido a las restricciones de flota y de tiempo, o distancia. Sin embargo, no visitar nodos con alta recompensa o aquellos ubicados cerca de nodos ya visitados puede suponer la pérdida de oportunidades, por lo que la planificación de rutas resulta fundamental. El TOP tiene aplicaciones prácticas en distintos escenarios logísticos, en los que múltiples agentes deben realizar tareas dentro de los límites impuestos por la flota existente y el tiempo o la distancia máximos permitidos (8).

En este trabajo se resuelve una extensión del TOP denominada Problema de Rutas de Orientación en Equipos con Tiempos de Servicio y Nodos Obligatorios e Incompatibles (TOP-ST-MIN, por sus siglas en inglés Team Orienteering Problem with Service Times and Mandatory and Incompatible Nodes) propuesta por Guastalla et al. (6). Esta variante incorpora de forma conjunta tres elementos relevantes: tiempos de servicio en los nodos, nodos obligatorios e incompatibilidades entre nodos.

En primer lugar, cada nodo requiere un tiempo fijo de servicio, por lo que la duración de una ruta no depende únicamente de los tiempos de desplazamiento, sino también del tiempo necesario para atender a los nodos visitados. En segundo lugar, algunos nodos son obligatorios y, por tanto, deben ser visitados por la flota. No se impone como restricción el uso de todos los vehículos disponibles. Cada nodo, ya sea obligatorio u opcional, puede ser visitado como máximo una vez y su recompensa asociada se tendrá en cuenta en la función objetivo.

Por último, el problema contempla incompatibilidades físicas entre nodos, representadas mediante la prohibición de ciertos arcos dirigidos en el grafo. Además de estas incompatibilidades físicas, también se consideran incompatibilidades lógicas simétricas entre nodos, con el fin de ofrecer un análisis más completo y reflejar situaciones frecuentes en la literatura. Estas incompatibilidades establecen que nodos de distinto tipo no pueden ser atendidos en una misma ruta. Por ello en el presente trabajo nos centraremos en el estudio y resolución del TOP-ST-MIN-PL, donde PL indica la consideración conjunta de incompatibilidades físicas y lógicas.

El resto del trabajo se organiza de la siguiente manera. En la Sección 2 se revisan los principales antecedentes relacionados con el TOP y sus variantes, prestando especial atención a los enfoques exactos y metaheurísticos propuestos en la literatura. En la Sección 3 se describe la metodología de resolución planteada, incluyendo la definición formal del problema y el esquema híbrido GRASP–ALNS desarrollado para su resolución. A continuación, en la Sección 4 se presentan los experimentos computacionales realizados, así como la comparación entre el método propuesto y el enfoque exacto de referencia. Finalmente, en la Sección 5 se resumen las principales conclusiones del trabajo y se señalan posibles líneas de investigación futura.

2. Trabajo relacionado

El TOP tiene su origen en el problema clásico conocido como el Problema de Rutas Orientado (en inglés Orienteering Problem), introducido por Tsiligirides. (11). Como se señaló en la introducción, el TOP fue formalmente definido por Chao et al. (5). En ese mismo trabajo, propusieron también uno de los primeros enfoques heurísticos para resolver el TOP, considerando tiempos de viaje y recompensas deterministas. Su procedimiento se basa en una estrategia de mejora record-to-record, y los resultados computacionales mostraron que la heurística era eficiente y superaba de forma consistente a los métodos disponibles en aquel momento.

Desde entonces, el TOP ha recibido una atención considerable en la literatura. Entre los trabajos heurísticos más relevantes, Hao Tang et al. (10) propusieron una búsqueda tabú integrada en un procedimiento de memoria adaptativa, en el que se alternan búsquedas sobre vecindarios pequeños y grandes. Este enfoque explora tanto soluciones factibles como infactibles mediante estrategias de generación de vecindarios de carácter aleatorio y voraz. En una línea similar, Archetti et al. (2) analizaron distintos enfoques metaheurísticos para el TOP, incluyendo dos variantes generalizadas de búsqueda tabú y de VNS (Algoritmo de Búsqueda en Vecindarios Variables, en inglés, Variable Neighborhood Search). Su estudio puso de manifiesto la estrecha interdependencia entre las decisiones de selección de clientes y la construcción de las rutas.

Frente a estos enfoques aproximados, Boussier et al. (4) desarrollaron uno de los primeros métodos exactos para solucionar el TOP, basado en un esquema de Ramificación y generación de columnas. Su algoritmo incorpora reglas de ramificación específicas del problema, así como técnicas de aceleración que permiten resolver de forma óptima instancias con hasta 100 clientes. Además, los autores destacaron la utilidad de este tipo de enfoques exactos como referencia para evaluar la calidad de las soluciones obtenidas mediante heurísticas y metaheurísticas. Posteriormente, se han propuesto otros algoritmos exactos para el TOP; sugerimos los siguientes artículos: (7), (3) y (1).

En la literatura reciente, el TOP-ST-MIN se ha resuelto implementando un modelo exacto basado en planos de corte, complementado con nuevas desigualdades válidas y técnicas de separación (6). Además, se han propuesto distintas

instancias de referencia para evaluar el impacto de las características específicas del problema, como los tiempos de servicio, los nodos obligatorios y las incompatibilidades entre nodos.

Las principales contribuciones de este trabajo son las siguientes: (i) el desarrollo de un enfoque metaheurístico para resolver el TOP-ST-MIN; (ii) la comparación computacional con el modelo exacto existente; y (iii) la mejora del desempeño computacional, especialmente en instancias de mayor complejidad.

3. Metodología de resolución

3.1. Definición del problema

El TOP-ST-MIN se define sobre un grafo completo inducido por el conjunto de $|V| = n$ nodos, $V = \{0, 1, \ldots, n-1\}$, donde el nodo 0 representa el depósito de origen, $n-1$ el depósito de destino y $C = \{1, \ldots, n-2\}$ el conjunto de clientes. Cada cliente $i \in C$ está definido por sus coordenadas (x_i, y_i), un beneficio $p_i \geq 0$, un tiempo de servicio $s_i \geq 0$ y un indicador de obligatoriedad $c_i \in \{0, 1\}$.

Se dispone de m vehículos idénticos, cada uno con una duración máxima de ruta, $T_{\text{máx}}$. El tiempo de desplazamiento entre los nodos i y j se calcula usando la distancia euclídea d_{ij}. El tiempo de una ruta es la suma de los tiempos de desplazamiento entre nodos consecutivos más los tiempos de servicio de los clientes visitados.

Como comentamos en la introducción, el problema incorpora dos familias de restricciones de compatibilidad:

- Compatibilidad física: para cada par (i, j), el parámetro $a^F_{ij} \in \{0, 1\}$ indica si los nodos i y j pueden ser consecutivos en la misma ruta.
- Compatibilidad lógica: el parámetro $a^L_{ij} \in \{0, 1\}$ indica si los nodos i y j pueden coexistir en la misma ruta.

La función objetivo consiste en maximizar el beneficio total:

$$\text{máx } Z = \sum_{k=1}^{m} \sum_{i \in C} p_i \, z_{ik}, \tag{1}$$

donde $z_{ik} = 1$ si el cliente i es visitado por el vehículo k, y 0 en caso contrario.

Las restricciones principales del problema establecen que cada ruta debe iniciarse en el depósito de origen y finalizar en el depósito de destino. Asimismo, la duración total de cada ruta no puede superar el límite máximo permitido, denotado por $T_{\text{máx}}$. Además, todos los clientes obligatorios, es decir, aquellos clientes $i \in C$ tales que $c_i = 1$, deben ser visitados exactamente una vez. Finalmente, toda solución debe respetar las compatibilidades físicas y lógicas definidas para los clientes y vehículos a lo largo de cada ruta.

3.2. Esquema algorítmico

La metodología propuesta sigue un esquema híbrido GRASP–ALNS, por sus siglas en inglés de Greedy Randomized Adaptive Search Procedure combinado con Adaptive Large Neighborhood Search, Procedimiento de búsqueda adaptativa aleatorizada y voraz combinado con Búsqueda Adaptativa en Vecindarios Grandes. El algoritmo maestro ejecuta N_{GRASP} iteraciones externas; en cada una genera una solución inicial mediante una heurística constructiva aleatorizada, la mejora mediante ALNS y conserva la mejor solución factible encontrada.

Algorithm 1 Esquema híbrido GRASP–ALNS

Require: Instancia $\mathcal{I}$, N_{GRASP}, N_{ALNS}, α, seed
Ensure: Mejor solución factible $S^{\star}$

$S^{\star} \leftarrow \emptyset$
for $t = 1$ **to** N_{GRASP} **do**
  $S \leftarrow \textsc{Constructive}(\mathcal{I}, \alpha, \mathrm{seed})$
  $S \leftarrow \mathrm{ALNS}(S, N_{\mathrm{ALNS}}, \mathrm{seed})$
  if S es factible **y** $f(S) > f(S^{\star})$ **then**
    $S^{\star} \leftarrow S$
  end if
end for
return $S^{\star}$

Se emplea una heurística constructiva aleatorizada que genera una solución formada por un conjunto de rutas. Dado que la flota es homogénea y todas las rutas son análogas, la construcción se realiza de forma iterativa ruta a ruta. Para cada vehículo, se inicializa una ruta que comienza en el depósito de origen y finaliza en el depósito de destino, y se van incorporando clientes de forma secuencial mientras existan candidatos factibles.

Dicha heurística consta de dos fases. En primer lugar, se construye la solución considerando únicamente los clientes obligatorios, es decir, aquellos candidatos no visitados para los que se cumple $c_i = 1$ y que además son factibles respecto a las restricciones de tiempo y compatibilidad. Una vez agotadas las posibles inserciones de clientes obligatorios, se procede a una segunda fase en la que se incorporan clientes opcionales, definidos por $c_i = 0$, siempre que satisfagan las mismas condiciones de factibilidad temporal y de compatibilidad.

En cada paso de inserción se evalúan los candidatos factibles mediante una función de prioridad basada en la relación beneficio–coste temporal. Para cada candidato i se calcula el incremento de tiempo ΔT_i al insertarlo en la ruta actual, y se define el ratio normalizado:

$$g_i = \frac{\tilde{p}_i}{\tilde{\Delta T}_i}, \tag{2}$$

donde $\tilde{p}_i$ y $\tilde{\Delta T}_i$ denotan los valores normalizados en el conjunto de candidatos del paso actual. En caso de que $\tilde{\Delta T}_i = 0$, se redefine $\tilde{\Delta T}_i = 10^{-1}$ para evitar divisiones por cero.

La aleatorización se introduce mediante una lista restringida de candidatos, controlada por el parámetro $\alpha \in [0, 1]$. Sean $g^{\text{máx}}$ y $g^{\text{mín}}$ el mayor y menor valor de prioridad entre los nodos candidatos factibles de la iteración actual. La lista restringida de candidatos se define como:

$$\text{RCL} = \{i : g_i \geq g^{\text{máx}} - \alpha \left(g^{\text{máx}} - g^{\text{mín}}\right)\}.$$

A continuación, se selecciona aleatoriamente un nodo de la RCL y se incorpora a la ruta actual. Valores bajos de α inducen un comportamiento más voraz, mientras que valores altos permiten una mayor diversificación al ampliar el conjunto de candidatos potencialmente seleccionables.

Tras la construcción inicial, cada solución se somete a un procedimiento ALNS (9), que itera operadores de destrucción y reparación sobre vecindarios amplios. En cada iteración se selecciona un operador de destrucción y un operador de reparación mediante un mecanismo probabilístico basado en pesos, los cuales se actualizan dinámicamente en función del rendimiento obtenido por cada operador.

En cada iteración se eliminan q clientes, con $q \sim \mathcal{U}\{q_{\text{mín}}, q_{\text{máx}}\}$ (en este trabajo se han considerado $q_{\text{mín}} = 1$ y $q_{\text{máx}} = 10$). La destrucción afecta a clientes opcionales mayoritariamente. Se consideran cinco operadores de destrucción:

- Eliminación aleatoria de clientes opcionales
- Eliminación de los últimos q clientes de una ruta
- Eliminación según menor ratio $p_i/\Delta T_i$
- Eliminación completa de una ruta
- Eliminación de clientes relacionados con un nodo semilla, donde la relación entre el nodo semilla s y un candidato i se mide como $\phi(s, i) = w_1 \cdot d_{si} + w_2 \cdot |p_s - p_i|$.

Tras la destrucción, un operador de reparación reconstruye parcialmente la solución. En todos los casos, la inserción exige factibilidad temporal, compatibilidad física de los arcos de entrada y salida, y compatibilidad lógica con todos los clientes ya presentes en la ruta. Se consideran cinco operadores de reconstrucción:

- Inserción en la primera posición factible los nodos obligatorios en primer lugar.
- Inserción en la posición de menor incremento de tiempo
- Inserción siguiendo el orden ratio beneficio/distancia
- Inserción conjunta según ratio beneficio–tiempo
- Inserción basada en el criterio de arrepentimiento, es decir para cada cliente no visitado i se calculan las dos mejores posiciones de inserción factibles, con costes Δ_1 y Δ_2. La función de arrepentimiento se define como $r_i = \Delta_2 - \Delta_1$, y se inserta prioritariamente el cliente con mayor valor de r_i.

La solución candidata obtenida tras aplicar un par destrucción–reparación se evalúa mediante un criterio de aceptación tipo *Simulated Annealing*. Sea S la solución actual y S' la solución candidata. Si S' mejora la función objetivo, la solución se acepta directamente. En caso contrario, puede aceptarse con probabilidad $P(\text{aceptar } S') = \exp(-\frac{\Delta}{T})$, donde Δ representa el deterioro de la solución candidata y T es la temperatura actual. La temperatura parte de un valor inicial T_0 y se reduce progresivamente durante la búsqueda, lo que permite aceptar con mayor facilidad soluciones no mejorantes al inicio, favoreciendo la exploración, y hacer el criterio más restrictivo conforme avanza el algoritmo.

Después de evaluar el movimiento, se actualizan los pesos de los operadores seleccionados mediante una regla suavizada: $\pi_j \leftarrow (1-\rho)\pi_j + \rho\sigma$, donde π_j es el peso del operador, ρ es el factor de reacción y σ la recompensa asignada según el resultado obtenido. Se asignan mayores recompensas cuando la solución mejora la mejor solución global, recompensas intermedias cuando mejora la solución actual o es aceptada sin mejorar, y recompensas menores cuando el movimiento es rechazado. Esta actualización se realiza de forma independiente para los operadores de destrucción y reparación, favoreciendo en iteraciones posteriores aquellos que han mostrado mejor rendimiento.

4. Experimentos Computacionales

Con el objetivo de realizar una comparación directa con el estado del arte, se llevó a cabo una evaluación computacional preliminar sobre un subconjunto representativo de las instancias utilizadas en la literatura y proporcionadas por Guastalla et al. (6). Además, la disponibilidad del ejecutable del algoritmo desarrollado por Guastalla et al. (6) facilitó la realización de los experimentos y la validación de los resultados presentados en este trabajo.

Todos los experimentos se realizaron en un equipo con Windows 11 de 64 bits, equipado con un procesador 13th Gen Intel® Core™ i5-13500H, 12 núcleos físicos y 16 procesadores lógicos a 2.60 GHz, 32 GB de RAM a 5200 MT/s y una unidad SSD de 477 GB. El algoritmo propuesto fue implementado en Python 3.14.2.

La comparación se realizó sobre un conjunto representativo correspondiente al 20 % de las instancias proporcionadas. Para cada instancia, el método de referencia se ejecutó con un límite máximo de tiempo de 1000 segundos.

Para cada instancia resuelta mediante nuestro enfoque, se ejecutó el esquema GRASP–ALNS con una semilla fija y con el parámetro $\alpha = 0{,}5$. La elección de este valor corresponde a un análisis preliminar de sensibilidad destinado a estudiar el efecto de la aleatorización en la fase constructiva del algoritmo. En particular, se consideraron los valores $\alpha \in \{0{,}0; 0{,}1; \ldots; 1{,}0\}$. Para cada valor de α se realizaron $R = 100$ réplicas independientes sobre un subconjunto representativo de instancias. De forma análoga, los valores $N_{\text{GRASP}} = N_{\text{ALNS}} = 100$ se adoptaron por defecto tras un análisis computacional preliminar orientado a estudiar la relación entre tiempo de cómputo y calidad de la solución. En la fase ALNS, la temperatura inicial se fijó en $T_0 = 50$ y se redujo geométricamente

según la regla $T \leftarrow 0{,}995 \cdot T$ hasta alcanzar un umbral mínimo de $T_{\text{mín}} = 10^{-3}$. Finalmente, en el operador de eliminación de clientes que usa la relación entre el nodo semilla y el candidato como una función se utilizaron los pesos $w_1 = 0{,}7$ y $w_2 = 0{,}3$, asignando mayor importancia a la proximidad o relación estructural entre elementos frente al componente secundario considerado en dicho operador.

La Tabla 1 recoge un resumen agregado de los resultados por conjunto de instancias donde la columna *Conjunto* identifica el grupo de instancias analizado, mientras que *Inst.* indica el número de instancias pertenecientes a cada conjunto. Las columnas *Fact.Exacto* y *Fact.GRASP–ALNS* muestran, respectivamente, el número de instancias para las que el método exacto obtiene el óptimo certificado y el número de instancias para las que GRASP–ALNS obtiene una solución factible. Las columnas *T.Exacto* y *T.GRASP–ALNS* calculan el tiempo medio de resolución, en segundos, para cada método dentro de cada conjunto. Por último, la columna *Gap* representa la diferencia porcentual media entre las soluciones obtenidas por GRASP–ALNS y un valor de referencia. Cuando el método exacto obtiene una solución factible, dicha solución corresponde al óptimo certificado y se utiliza como referencia. En aquellas instancias en las que el método exacto no obtiene solución factible dentro del límite de tiempo, se emplea como referencia el mejor valor conocido disponible (*Best Known Solution*, BKS). Por tanto, el gap se calcula únicamente en las instancias para las que existe un valor de referencia válido.

La Tabla 1 muestra que en los tres primeros conjuntos, ambos enfoques logran resolver todas las instancias de forma factible; sin embargo, GRASP–ALNS alcanza tiempos medios de resolución claramente inferiores en Set1 y Set3, manteniendo además gaps medios reducidos. La superioridad del enfoque GRASP–ALNS se hace más evidente a partir de Set4, donde las instancias presentan una mayor dificultad. En estos conjuntos, el método exacto alcanza con frecuencia el límite de tiempo de 1000 segundos y no consigue encontrar soluciones factibles en un número considerable de casos. En cambio, GRASP–ALNS mantiene tiempos de ejecución muy reducidos y obtiene soluciones factibles en una proporción mucho mayor de instancias. Este comportamiento es especialmente relevante en Set4, Set6 y Set7, donde el método exacto muestra importantes limitaciones, mientras que el enfoque heurístico propuesto conserva una alta capacidad de resolución.

Cuadro 1. Resumen de la comparación entre el método exacto y GRASP–ALNS por conjunto de instancias.

Conjunto	Inst.	Fact.Exacto	Fact.GRASP–ALNS	T.Exacto	T.GRASP–ALNS	Gap
Set1	24	24	24	6,87	1,37	3,51 %
Set2	15	15	15	0,60	0,96	1,21 %
Set3	24	24	24	13,46	1,55	1,66 %
Set4	24	0	20	1000,00	7,82	1,38 %
Set5	24	21	24	208,86	3,47	4,67 %
Set6	24	12	24	597,45	5,07	2,67 %
Set7	15	3	11	898,36	7,06	1,51 %
Total	**150**	**99**	**142**	**382,16**	**3,89**	**2,55 %**

La Tabla 2 presenta la comparación en términos agregados. La columna *Método* identifica el enfoque evaluado, es decir, el método exacto o el algoritmo GRASP–ALNS. La columna *Instancias* indica el número total de instancias consideradas en la comparación. La columna *Factibles* cuenta cuántas de esas instancias fueron resueltas obteniendo al menos una solución factible. Por su parte, *Tiempo medio* representa el tiempo promedio de cómputo, expresado en segundos, empleado por cada método. La columna *Gap medio* recoge la diferencia porcentual media entre las soluciones obtenidas por GRASP–ALNS y las soluciones de referencia del método exacto.

Como se observa en la Tabla 2, el método GRASP–ALNS obtiene soluciones factibles en 142 de las 150 instancias consideradas, mientras que el método exacto alcanza factibilidad en 99 instancias, todo esto en el límite de tiempo establecido. Además, el tiempo medio de cómputo del enfoque propuesto es sustancialmente inferior, con un valor medio de 3,89 segundos frente a los 382,16 segundos registrados por el método exacto lo que supone una reducción del 99 %. En términos de calidad de solución, GRASP–ALNS presenta un gap medio del 2,55 %, lo que indica que, en promedio, las soluciones obtenidas se encuentran próximas a las del método exacto en aquellas instancias en las que la comparación es posible. Estos resultados sugieren que el enfoque heurístico propuesto ofrece una alternativa computacionalmente eficiente, especialmente en instancias en las que el método exacto no logra obtener una solución factible dentro del límite de tiempo establecido.

Cuadro 2. Resumen de la comparación entre el método exacto y GRASP–ALNS.

Método	Instancias	Factibles	Tiempo medio	Gap medio
Exacto	150	99	382,16	–
GRASP–ALNS	150	142	3,89	2,55 %

5. Conclusiones

En este trabajo se ha propuesto un enfoque heurístico basado en GRASP–ALNS para abordar el Problema de Rutas de Orientación en Equipos con Tiempos de Servicio y Nodos Obligatorios e Incompatibles, donde dichas incompatibilidades pueden ser físicas o Llógicas (TOP-ST-MIN-PL), con el objetivo de obtener soluciones factibles y de buena calidad en tiempos de cómputo reducidos. Los resultados computacionales preliminares muestran que el enfoque propuesto presenta un comportamiento competitivo frente al método exacto considerado como referencia.

La comparación realizada evidencia que GRASP–ALNS alcanza soluciones factibles en el 94 % de las instancias analizadas, mientras que el método exacto solo consigue factibilidad en el 66 % de ellas. Esta diferencia pone de manifiesto la mayor robustez del enfoque heurístico, ya que es capaz de proporcionar soluciones admisibles en un número considerablemente mayor de casos dentro del límite de

tiempo establecido. Además, el enfoque GRASP–ALNS reduce de forma muy significativa los tiempos de cómputo. En promedio, el método exacto requiere 382,16 segundos, mientras que GRASP–ALNS obtiene soluciones en apenas 3,89 segundos.

En términos de calidad de solución, el gap medio obtenido por GRASP–ALNS es de 2,55 %, lo que indica que las soluciones generadas se mantienen próximas a las soluciones de referencia cuando la comparación con el método exacto es posible. Por tanto, el método propuesto no solo mejora la eficiencia y la factibilidad, sino que también conserva una calidad de solución competitiva.

Como líneas de trabajo futuro, se propone profundizar en la calibración automática de los parámetros del algoritmo, con el fin de mejorar aún más su rendimiento y adaptabilidad a distintos tipos de instancias. Asimismo, sería interesante incorporar nuevos operadores de destrucción y reparación dentro del esquema ALNS, de modo que el algoritmo pueda explorar el espacio de soluciones de manera más diversa y efectiva. Una vez realizado esto, se pretende resolver el conjunto completo de instancias de la literatura.

Acknowledgments. J. Sánchez-Oro agradece la financiación recibida por el "Ministerio para la Transformación Digital y de la Función Pública" mediante la ayuda con referencia TSI-100930-2023-3, por el "Ministerio de Economía y Competitividad" mediante la ayuda con referencia PID2021-125709OA-C22, y por la Comunidad Autónoma de Madrid, proyecto CIRMA-CM (TEC-2024/COM-404).

A.D. López-Sánchez agradece la financiación recibida por el Ministerio de Ciencia e Innovación de España a través del Proyecto PID2022-139543OB-C41, así como por el Programa FEDER Andalucía 2021-2027 y por la Consejería de Universidad, Investigación e Innovación de la Junta de Andalucía, mediante el Plan Propio de Investigación y Transferencia (2023-2026) de la Universidad Pablo de Olavide (UPO), referencia PPI2307.

C. Tobar-Fernández agradece la financiación recibida de la "Agencia Estatal de Investigación" a través de las "Ayudas para contratos para la formación de doctores y doctoras en empresas y otras entidades (Doctorados Industriales) 2024" (referencia DIN2024-013339), mediante la empresa QOSIT CONSULTING SL.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Bibliografía

[1] Almeida, L.: Optimization algorithms for the steiner team orienteering problem (2019), https://api.semanticscholar.org/CorpusID:215880566

[2] Archetti, C., Hertz, A., Speranza, M.G.: Metaheuristics for the team orienteering problem. Journal of Heuristics **13**(1), 49–76 (2007)

[3] Bianchessi, N., Mansini, R., Speranza, M.G.: A branch-and-cut algorithm for the team orienteering problem. International Transactions in Operational Research **25**(2), 627–635 (2018)

[4] Boussier, S., Feillet, D., Gendreau, M.: An exact algorithm for team orienteering problems. 4OR **5**(3), 211–230 (2007)

[5] Chao, I.M., Golden, B.L., Wasil, E.A.: The team orienteering problem. European Journal of Operational Research **88**(3), 464–474 (1996)

[6] Guastalla, A., Aringhieri, R., Hosteins, P.: The team orienteering problem with service times and mandatory & incompatible nodes. Computers & Operations Research **183**, 107170 (2025)

[7] Keshtkaran, M., Ziarati, K., Bettinelli, A., Vigo, D.: Enhanced exact solution methods for the team orienteering problem. International Journal of Production Research **54**(2), 591–601 (2016)

[8] Orlis, C., Bianchessi, N., Roberti, R., Dullaert, W.: The team orienteering problem with overlaps: An application in cash logistics. Transportation Science **54**(2), 470–487 (2020)

[9] Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. Computers & operations research **34**(8), 2403–2435 (2007)

[10] Tang, H., Miller-Hooks, E.: A tabu search heuristic for the team orienteering problem. Computers & Operations Research **32**(6), 1379–1407 (2005). https://doi.org/https://doi.org/10.1016/j.cor.2003.11.008, https://www.sciencedirect.com/science/article/pii/S0305054803003265

[11] Tsiligirides, T.: Heuristic methods applied to orienteering. Journal of the Operational Research Society **35**(9), 797–809 (1984)