

Optimización de Rutas de Mantenimiento: Resolución del VRP con Ganancias Constantes Aplicado a Mantenimiento de Instalaciones

Autor: Emilio Castro Castro

Tesis doctoral UDC / 2021

Director: Alejandro García del Valle

Tutor: Alejandro García del Valle

Programa de doctorado en Ingeniería Naval e Industrial



UNIVERSIDADE DA CORUÑA



UNIVERSIDADE DA CORUÑA

Ferrol, 23 de abril de 2021

D. Alejandro García del Valle, Catedrático de Universidad del Departamento de Empresa de la Universidade da Coruña,

CERTIFICA:

Que la memoria titulada:

“Optimización de Rutas de Mantenimiento: Resolución del VRP con Ganancias Constantes Aplicado a Mantenimiento de Instalaciones.”

Ha sido realizada por **D. Emilio Castro Castro** bajo mi dirección en el departamento de Ingeniería Naval e Industrial de la Universidade da Coruña, y constituye la Tesis que presenta para optar al grado de doctor.

Fdo. Alejandro García del Valle

Director de la Tesis Doctoral

Quiero dedicarles este trabajo a mis hijos Emilio José, Cristina y Mercedes

Agradecimientos

Quisiera agradecer a mi familia y a mis amigos su confianza, que me hayan soportado, tolerado y acompañado todos estos años. "I get high with a little help from my friends" (With a little help from my friends, The Beatles).

A mi profesor y amigo el Catedrático Dr. D. Alejandro García del Valle quisiera agradecer su paciencia, comprensión y humanidad.

"I thank whatever gods may be for my unconquerable soul" (Invictus, William Ernest Henley).

RESUMO

As empresas mantedoras de patrimonio distribuído teñen un amplo mercado: cadeas de tendas, entidades bancarias, edificios da administración, gasolinerías, centros de saúde, iluminación pública, mobilidade urbana, etcétera. Estas empresas teñen un importante punto de mellora: a optimización dos roteiros de mantemento.

Preséntanse dous algoritmos que optimizan os roteiros de mantemento dun caso real. Os resultados do primeiro algoritmo compáranse cos das tres variantes do segundo. O primeiro está baseado no algoritmo k -NN, con tres fases posteriores nas que se pretenden implementan mantementos para mellorar o roteiro. O segundo algoritmo ten tres variantes, a primeira conta cunha fase na que se organizan os roteiros e nunha segunda optimízanse, á segunda variante implementáronse dúas fases nas que se pretende mellorar os roteiros para que na fase de optimización final obtéñanse mellores resultados, a terceira variante é similar á segunda, na que se supón que, mediante un hipotético software de predición, coñecemos de antemán as incidencias que serán xeradas nun horizonte temporal de tres días.

Os algoritmos programáronse nun software empregado frecuentemente nas empresas: Microsoft Excel.

RESUMEN

Las empresas mantenedoras de patrimonio distribuido tienen un amplio mercado: cadenas de tiendas, entidades bancarias, edificios de la administración, gasolineras, centros de salud, alumbrado público, movilidad urbana, etcétera. Estas empresas tienen un importante punto de mejora: la optimización de las rutas de mantenimiento.

Se presentan dos algoritmos que optimizan las rutas de mantenimiento de un caso real. Los resultados del primer algoritmo se comparan con los de las tres variantes del segundo. El primero está basado en el algoritmo k-NN, con tres fases posteriores en las que se implementan mantenimientos para mejorar la ruta. El segundo algoritmo tiene tres variantes, la primera cuenta con una fase en la que se organizan las rutas y en una segunda se optimizan, a la segunda variante se implementaron dos fases en las que se pretende mejorar las rutas para que en la fase de optimización final se obtengan mejores resultados, la tercera variante es similar a la segunda, en la que se supone que, mediante un hipotético software de predicción, conocemos de antemano las incidencias que serán generadas en un horizonte temporal de tres días.

Se han programado los algoritmos en un software de uso frecuente en las compañías: Microsoft Excel.

ABSTRACT

Facility Management Industry is moving towards a highly competitive market having as end-users' commercial entities, financial services, manufacturing, retail, healthcare, food and beverage, transportation, hospitality, leisure and logistics. These companies could achieve an important improvement in terms of reduction in the upfront cost through the optimization of the maintenance routes.

On this base, this report presents the implementation of two algorithms in a real case, comparing the results of the first algorithm with the three variants of the second one. The first algorithm (base on k-NN algorithm) has three subsequent phases used to implement the maintenance to improve the route. The second algorithm shows three variants: the first one, has an organization routes phase and an optimization routes phase. In the second variant two new phases were implemented to boost better results in the optimization routes phase. The third variant is similar to the second one but, with a hypothetical prediction software, we are able to establish the issues we could have in a three days horizon time.

The algorithms have been programmed into a frequently used software in companies: Microsoft Excel.

PRÓLOGO

Durante mis estudios del Máster Universitario en Investigación en Tecnologías Navales e Industriales conocí al Dr. D. Alejandro García del Valle que impartía las materias de optimización e inteligencia artificial. En ese momento se inició la aventura que en este documento pretendo presentar.

A lo largo de mi vida profesional, en una multinacional, he adquirido una experiencia de más de diez años en el mantenimiento de instalaciones en patrimonio distribuido y posteriormente he implementado, durante más de cinco años, planes de mantenimiento en contratos, entre los que se encuentran los de patrimonio distribuido.

Las empresas con patrimonio distribuido son aquellas que cuentan con numerosos centros de trabajo distribuidos por la geografía, tales como cadenas de tiendas, entidades bancarias, mutuas, aseguradoras, edificios de la administración, gasolineras, centros de salud, alumbrado público, movilidad urbana, etc.

Estas entidades han de mantener sus instalaciones por obligación administrativa, para mantener unos mínimos de confort y operatividad y para que no se reduzca la vida útil de los elementos.

Son conocidas las ventajas de la externalización de las actividades no estratégicas de las empresas, por lo que las empresas con patrimonio distribuido pueden obtener importantes ahorros al externalizar, entre otras, las actividades de mantenimiento de sus instalaciones.

Por otro lado, las empresas de servicios han encontrado un nicho de mercado en las empresas con patrimonio distribuido, dado que la dispersión geográfica de los centros de trabajo obliga a una importante especialización en la operativa.

El objeto del mantenimiento de edificios es la aseguración del valor de los bienes a mantener y que éstos, dentro de los límites de la vida útil, mantengan las prestaciones iniciales (UNE-EN 15331, 2012).

Así mismo, los objetivos principales del mantenimiento son (UNE-EN 13306, 2011):

- Asegurar la disponibilidad de los elementos a mantener.
- Considerar los requisitos de seguridad hacia las personas y los obligatorios de cada elemento.

- Tener en cuenta los impactos medioambientales.
- El mantenimiento de la durabilidad de los elementos, considerando los costes.

Distinguiremos entre otros los siguientes tipos de mantenimiento:

- Mantenimiento preventivo es aquel que se realiza a intervalos establecidos, cuyo objeto es la reducción de la probabilidad de fallos o de la degradación de un elemento.
- Mantenimiento correctivo es el que se realiza tras el reconocimiento de una avería, con el fin de devolver al elemento la función que se le requiere. Puede ser diferido si no se acomete inmediatamente después de detectarse la avería, en función de unas reglas dadas o puede ser inmediato en caso contrario.

Los ordenadores son herramienta fundamental en casi cualquier ámbito. Uno de los principales problemas con los que se encuentran las compañías es la compra de licencias de software especializado. Teniendo en cuenta esta circunstancia he querido utilizar un programa implantado en la mayor parte de las empresas, Microsoft Excel, de modo que sea posible implementar la solución con un mínimo coste.

Gracias al trabajo profesional que he desarrollado, dispongo de una base de datos de un contrato, en la que figuran todos los mantenimientos realizados a lo largo de un año, con las fechas de sus solicitudes. Esta base de datos será sobre la que se realizará el estudio de las soluciones que se plantean.

No se presenta la comparación de los resultados obtenidos en el presente trabajo con los reales, puesto que el equipo de trabajo que desempeñaba estas funciones también realizaba otras, durante ese periodo de tiempo, de las que no se dispone de información.

ÍNDICE

Resumo	VII
Resumen	IX
Abstract	XI
Prólogo	XIII
Tablas	XVII
Figuras	XIX
Gráficas	XXI
Abreviaturas	XXIII
1. Introducción y Objetivos	1
2. Revisión Bibliográfica	5
2.1. El problema de orientación (OP)	9
2.2. Problema del tour rentable (PTP)	15
2.3. El problema del viajante de comercio con recogida de premios (PCTSP)	17
2.4. El problema del equipo de orientación (TOP)	19
2.5. TSP multi-criterio	22
2.6. Variantes al VRPP	24
3. Planteamiento	33
3.1. Ingresos	34
3.2. Tiempo de trabajo	34
3.3. Costes de los equipos de mantenimiento	35
3.4. Costes de subcontratación	37
3.5. Costes por penalizaciones	38
3.6. Funciones objetivo y restricciones	38
3.7. Definición del problema objeto del estudio	41
4. Métodos propuestos	47
4.1. Algoritmo basado en vecino más cercano	48
4.2. Algoritmo simple	62

4.3. Algoritmo con reordena	76
4.4. Algoritmo con reordena y predicciones.....	92
5. Experimentación y resultados	105
5.1. Experimentación.....	105
5.2. Resultados	107
6. Conclusiones y futuras líneas de investigación	113
6.1. Conclusiones	113
6.2. Futuras líneas de investigación	118
7. Bibliografía	121

TABLAS

Tabla 1. Fechas máximas de resolución	93
Tabla 2. Denominación de los métodos estudiados	106
Tabla 3. Media y error relativo de los tiempos parciales.....	107
Tabla 4. Media y error relativo de los tiempos totales y costes	110
Tabla 5. Media de importes medios de las funciones objetivo	114
Tabla 6. Relación de costes obtenidos según el método y la función objetivo utilizados.....	115
Tabla 7. Resultados de los métodos con las dos mejores funciones objetivo.....	117

FIGURAS

Figura 1. Distribución geográfica de los centros.....	42
Figura 2. Matriz de tiempos de desplazamiento	49
Figura 3. Detalle del calendario	50
Figura 4. Detalle de la tabla de órdenes de trabajo a ejecutar.....	50
Figura 5. Detalle de la hoja Estudio.....	51
Figura 6. Diagrama de flujo de la fase Resolución	52
Figura 7. Detalle de la lista de incidencias L_1	53
Figura 8. Detalle de la lista de tareas L_2	53
Figura 9. Detalle de la tabla que calculará la ruta.....	54
Figura 10. Diagrama de flujo de la fase Vecino más cercano.....	54
Figura 11. Diagrama de flujo de la fase Externalizar	55
Figura 12. Diagrama de flujo de la fase Centro visitado	57
Figura 13. Diagrama de flujo de la fase Centros mínimo desplazamiento	59
Figura 14. Diagrama de flujo de la fase Centros cercanos.....	61
Figura 15. Diagrama de flujo de la fase Registro	62
Figura 16. Diagrama de flujo de la fase Resolución del algoritmo simple	65
Figura 17. Detalle de las listas L_1 y L_2	66
Figura 18. Diagrama de flujo de la fase Estudio del algoritmo simple.....	66
Figura 19. Detalle de la hoja Reparto, donde se aprecian la lista de incidencias L_1 y la tabla de cálculo	68
Figura 20. Detalle de los parámetros de Solver en la fase Reparto	69
Figura 21. Diagrama de flujo de la fase Reparto del algoritmo simple	70
Figura 22. Detalle de la hoja Externalizar, donde se aprecian la lista de incidencias y la tabla de cálculo	71
Figura 23. Diagrama de flujo de la fase Externalizar del algoritmo simple	72
Figura 24. Detalle de la hoja OptimizarH, donde se aprecian las listas L_0 y L_2 y la tabla de cálculo.....	73
Figura 25. Diagrama de flujo de la fase OptimizarH del algoritmo simple	75
Figura 26. Diagrama de flujo de la fase Resolución	77
Figura 27. Diagrama de flujo de la fase Estudio	78
Figura 28. Diagrama de flujo de la fase Reparto.....	80
Figura 29. Diagrama de flujo de la fase Externalizar	82

Figura 30. Detalle de la tabla con todas las posibles secuencias de horas y de la tabla con las posibles permutaciones de las secuencias de horas propuesta en Reparto	83
Figura 31. Detalle de la hoja Transferir, donde se aprecia la lista de incidencias y la tabla que define las restricciones de orden	84
Figura 32. Detalle de la hoja Transferir, donde se aprecia la generación de las permutaciones posibles	85
Figura 33. Diagrama de flujo de la fase Transferir.....	86
Figura 34. Detalle de la hoja Reordena, donde se aprecian las listas L_1 y la tabla de cálculo	87
Figura 35. Detalle de los parámetros de Solver en la fase Reordena	87
Figura 36. Diagrama de flujo de la fase Transferir.....	89
Figura 37. Diagrama de flujo de la fase Transferir.....	91
Figura 38. Diagrama de flujo de la fase Resolución.....	92
Figura 39. Detalle de la hoja Estudio, donde se aprecian las listas L_1 , L_2 y L_3	94
Figura 40. Diagrama de flujo de la fase Estudio.....	94
Figura 41. Diagrama de flujo de la fase Reparto	96
Figura 42. Diagrama de flujo de la fase Externalizar.....	98
Figura 43. Diagrama de flujo de la fase Transferir.....	100
Figura 44. Diagrama de flujo de la fase Reordena	102
Figura 45. Diagrama de flujo de la fase OptimizarH.....	104

GRÁFICAS

Gráfica 1. Comportamiento de las funciones objetivo según el método utilizado	113
Gráfica 2. Comportamiento de los métodos utilizados según la función objetivo.....	115
Gráfica 3. Resultados de los métodos con las dos mejores funciones objetivo	117

ABREVIATURAS

ABC	(Artificial Bee Colony) Colonia artificial de hormigas
AB-k-NN	Algoritmo propuesto en este trabajo, basado en vecino más cercano
ACO	(Ant Colony Optimization) Algoritmo de optimización mediante colonias de hormigas
ALNS	(Adaptive Large Neighbourhood Search) Búsqueda adaptativa de grandes vecindarios
BTP	(One period Bus Touring Problem) El problema de un periodo del autobús turístico
CPCTSP	(Capacitated Prize Collecting Travelling Salesman Problem) El problema del viajante de comercio con recogida de premios y restricciones de capacidad
CPLEX	Algoritmo solucionador estándar de números enteros mixtos
CPTP	(Capacited Profitable Tour Problem) PTP teniendo en cuenta la capacidad de los vehículos
CTOP	(Capacitated Team Orienteering Problem) TOP teniendo en cuenta la capacidad de los vehículos
CTOP-IS	(Capacitated Team Orienteering Problem with Incomplete Service) CTOP en el que la prestación del servicio contratado no se ejecuta en su totalidad
CVRP	(Capacitated Vehicle Routing Problem) VRP teniendo en cuenta la capacidad de los vehículos
ELS	(Evolutionary Local Search) Algoritmo de búsqueda local evolutiva
EPI	Equipos de Protección Individual
GLS	(Guided Local Search) Algoritmos de búsqueda local guiada
GOP	(Generalized Orienteering Problem) El problema de orientación generalizada
GRASP	(Greedy Randomized Adaptive Search Procedure) Procedimiento de búsqueda adaptativa aleatoria codiciosa
k-NN	(k-Nearest Neighbor) Algoritmo del vecino más cercano
KP	(Knapsack Problem) El problema de la mochila
MA	(Memetic Algorithm) Algoritmo híbrido memético
MS	Microsoft
MTG	(Mobile Tourist Guide) El problema de la guía turística móvil

MTMCP	(Multiple Tour Maximun Collection Problem) El problema de múltiples tours de recolección máxima
MuPOPTW	(Multi-Period Orienteering Problem with multiple Time Windows) OP de múltiples periodos con múltiples ventanas de tiempo
MVP	(Multiobjetive Vending Problem) El problema de venta multiobjetivo
NP-hard	(Non deterministic Polynomial time hard problem) Problema no determinista de tiempo polinomial difícil
OP	(Orienteering Problem) El problema de orientación
OPmTW	(Orienteering Problem with multiple Time Windows) OP con múltiples ventanas de tiempo
OPSP	(Orienteering Problem with Stochastic Profits) OP con beneficios estocásticos
OPSTS	(Orienteering Problem with Stochastic Travel and Service times) OP en el que los tiempos de viaje y de servicio son estocásticos
OPTW	(Orienteering Problem with Time Windows) OP con ventanas temporales
PCTSP	(Prize Collecting Traveling Salesman Problem) El problema del viajante de comercio con recogida de premios
POI	(Point Of Interest) Punto de interés
PRED -	Algoritmo, propuesto en este trabajo, con reordena y predicciones, en el que el redondeo es a menos
PRED +	Algoritmo, propuesto en este trabajo, con reordena y predicciones, en el que el redondeo es a más
PSO	(Particle Swarm Optimization) Algoritmo de optimización con enjambre de partículas
PTP	(Profitable Tour Problem) El problema del tour rentable
PTSP	(Probabilistic Traveling Salesman Problem) El problema del viajante probabilístico
REOR -	Algoritmo, propuesto en este trabajo, con reordena, en el que el redondeo es a menos
REOR +	Algoritmo, propuesto en este trabajo, con reordena, en el que el redondeo es a más
SIMP	Algoritmo simple, propuesto en este trabajo
SSTSP	(Selective Travelling Salesperson Problem with Stochastic service times, travel times, and travel costs) El problema del viajante selectivo, en la que son estocásticos tanto los tiempos de servicio, como los tiempos y gastos de viaje
STSP	(Selective Traveling Salesman Problem) TSP selectivo

STSPW	(Stochastic Traveling Salesman Problem with Time Windows) El problema del viajante estocástico con ventanas de tiempo
SVRP	(Stochastic Vehicle Routing Problem) VRP con tiempos de desplazamiento estocásticos
TDO	(Time Dependent Orienteering) El problema de Orientación dependiente del tiempo
TOP	(Team Orienteering Problem) El Problema del equipo de orientación
TOPTW	(Team Orienteering Problem with Time Windows) TOP con ventanas temporales
TSP	(Traveling Salesman Problem) El problema del viajante
TSPP	(Traveling Salesman Problems with Profits) TSP con ganancias
TTDP	(Tourist Trip Design Problem) El problema de diseño del viaje turístico
VMI	(Vendor Managed Inventory) El problema de la gestión del stock de proveedores
VNS	(Variable Neighbourhood Search) Algoritmo de búsqueda variable en el vecindario
VRP	(Vehicle Routing Problem) El problema de rutas de vehículos
VRPP	(Vehicle Routing Problem with Profits) VRP con ganancias
VRPTW-P	(Vehicle Routing and scheduling Problem with Time Windows-Probabilistic) VRP probabilístico con ventanas temporales

1. INTRODUCCIÓN Y OBJETIVOS

Cada empresa con patrimonio distribuido establece sus estrategias referentes a la externalización de las tareas de mantenimiento, de modo que, mediante un pliego de condiciones, las mantenedoras oferten dichos servicios de forma satisfactoria.

Es por este motivo que no se puede hablar de un contrato tipo, pero sí podemos suponer unas condiciones generales típicas, aunque no se den en todos los casos. En esta generalización, basada en la experiencia, suelen resolverse tanto la contratación del mantenimiento preventivo como la del correctivo.

El mantenimiento preventivo, como ya visto, es programable.

El mantenimiento correctivo suele dividirse en tres tipos en función de la urgencia, dados unos niveles que dependen de la instalación averiada. Por lo que, en caso de que surgiese una avería con el mayor nivel de urgencia (prioridad alta) el mantenimiento correctivo es inmediato, el plazo de atención suele ser de horas; si la avería tuviese un tipo de urgencia medio (prioridad media) el mantenimiento correctivo puede ser diferido, con un plazo de atención de hasta un día y en el caso de una avería con un nivel menor de urgencia (prioridad baja), consecuentemente también puede diferirse, con un plazo de atención de hasta varios días.

Para asegurarse unos niveles mínimos de servicio, las empresas con patrimonio distribuido, en algunos casos, suelen establecer unas penalizaciones que se aplican ante la no atención de los mantenimientos. Los importes de las penalizaciones suelen determinarse en función del perjuicio recibido.

Así mismo, puede incluirse la posibilidad de rescisión del contrato en el caso de una acumulación de las penalizaciones.

Como segunda generalización podríamos establecer tres tipos de contratos de mantenimiento:

- Tipo 1: En el que se factura por intervención de mantenimiento.
- Tipo 2: En el que se factura por intervención de mantenimiento aplicándose penalizaciones por las tareas de mantenimiento no ejecutadas.
- Tipo 3: En el que se factura una cuota por el servicio contratado y se aplican penalizaciones por las tareas de mantenimiento no ejecutadas.

- Tipo 4: En el que se factura una cuota por el servicio contratado sin que se apliquen penalizaciones. Este tipo de contrato es en el que se basará el estudio.

Las empresas mantenedoras que prestan sus servicios a las empresas con patrimonio distribuido suelen dividir en zonas las superficies geográficas a mantener, en las que se definen una o varias bases. Así mismo, asignan unos recursos limitados para la prestación del servicio. Estos recursos suelen ser equipos de trabajo, formados por un operario y una furgoneta dotada tanto con las herramientas, como por los repuestos necesarios para ejecutar los mantenimientos.

Se asigna a cada base los equipos de trabajo que se estiman suficientes para atender los centros ubicados en las distintas zonas, teniendo en cuenta tanto el número de centros a mantener, como su dispersión geográfica. Desde estas bases parte diariamente cada equipo para atender tanto los mantenimientos preventivos como los correctivos, finalizando su jornada laboral en la base de donde partió.

Es evidente que una buena planificación de las rutas recorridas por cada equipo de trabajo mejora los resultados de la empresa teniendo en cuenta los siguientes factores:

- Reducción del número de equipos de trabajo.
- Reducción del importe de las penalizaciones.
- Satisfacción del cliente.

Para poder programar correctamente las planificaciones se debe tener en cuenta, entre otras variables, las siguientes:

- La jornada laboral de cada equipo de trabajo es de 8 horas.
- El número de horas totales anuales de trabajo de los operarios varía en función del convenio colectivo provincial.
- Los tiempos de desplazamiento entre los centros a mantener no son constantes.
- Los tiempos necesarios para localizar un estacionamiento para las furgonetas son variables.
- Los tiempos de operación de los mantenimientos tanto preventivos como correctivos no son constantes.
- Algunas operaciones de mantenimiento es necesario que sean ejecutadas por más de un operario.
- Los tiempos necesarios para que se completen los stocks de los repuestos de las furgonetas, así como los tiempos necesarios para la adquisición de algún repuesto fuera de lo común.

Los objetivos que se plantean en este trabajo son:

- Diseño de un algoritmo de referencia, que genere las rutas de mantenimiento de los contratos de mantenimiento tipo 4.
- Diseño de un algoritmo que optimice las rutas de mantenimiento de los contratos de mantenimiento tipo 4, que mejore los resultados del algoritmo de referencia.
- Programación de los algoritmos en Microsoft Excel.

2. REVISIÓN BIBLIOGRÁFICA

En la investigación operativa, siempre que se habla de planificaciones de rutas se piensa en el Problema del Viajante, TSP (Traveling Salesman Problem). Éste es uno de los problemas más estudiados (Laporte, 1992 a), considerándose NP-hard (Gutin & Punnen, 2007).

El TSP consiste en encontrar una ruta para un comercial de ventas, de modo que, partiendo de una localidad de origen, recorra una serie de ciudades definidas previamente, con el objetivo de que la distancia recorrida sea mínima (Gutin & Punnen, 2007).

La primera ocasión en la que se expone el problema con este nombre es en 1949 por Robinson (Gutin & Punnen, 2007).

El TSP se puede definir (Laporte, 1992 a), (Gutin & Punnen, 2007) del siguiente modo:

Sea $G = (V, A)$ un grafo, donde $V = \{0, \dots, n\}$ es el conjunto de vértices y sea A el conjunto de arcos o aristas que unen los vértices. Los vértices $i = 1, \dots, n$ se corresponden con los clientes, siendo el vértice 0 la ciudad de origen.

Se asocia a cada arista $(i, j) \in A$ un coste no negativo c_{ij} , que representa el coste de desplazarse entre los vértices i y j . No se suelen admitir bucles, es decir arcos (i, i) , por lo que se asocia a estos arcos un coste infinito: $c_{ii} = \infty$, para todo $i \in V$. Sea $C = (c_{ij})$ la matriz de distancias (o costes, o tiempos de desplazamiento) asociada a A .

El TSP consiste en encontrar un circuito Hamiltoniano de distancia mínima, en el que se recorran todos los vértices de V una sola vez.

Si la matriz C es simétrica, esto es, si $c_{ij} = c_{ji}, \forall i, j \in V$, se dice que el TSP es simétrico, en caso contrario es asimétrico. Así mismo se dice que cumple la inecuación triangular, si y solo si $c_{ij} + c_{jk} \geq c_{ik}, \forall i, j, k \in V$, condición que se cumple en los problemas Euclídeos (Laporte, 1992 a), (Gutin & Punnen, 2007).

Sea x_{ij} una variable binaria asociada a cualquier arco (i, j) , que toma el valor 1 si y solo si el arco $(i, j) \in V$ pertenece a la solución óptima y toma el valor 0 en caso contrario. El TSP puede plantearse del siguiente modo (Laporte, 1992 a), (Gutin & Punnen, 2007):

$$\text{Min} \sum_{i \neq j} c_{ij} x_{ij} \tag{1}$$

$$\sum_{j=1}^n x_{ij} = 1; i = 1, \dots; n \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1; j = 1, \dots; n \quad (3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad (4)$$

$$\forall S \subset V; 2 \leq |S| \leq n - 2 \quad (5)$$

$$x_{ij} \in \{0,1\} \quad (6)$$

$$i, j = 1, \dots, n; i \neq j \quad (7)$$

La función objetivo (1) pretende minimizar el coste del tour. Esta función está sujeta a la ecuación (2) que obliga a que las aristas que entran en los vértices lo hagan en una sola ocasión y con la ecuación (3) se consigue que de todos los vértices solo salga una arista. La ecuación (4) impide la formación de subtours. Esta limitación puede plantearse de distintos modos, según se verá más adelante. En (5) se define S como subconjunto de V , cuyo número elementos han de ser mayor o igual a 2 y menor o igual a $n - 2$. Con ella se indica que para cualquier subconjunto de nodos S , debe de existir, al menos un arco que salga de él (Dantzig, et al., 1954). Las restricciones (6) y (7) determinan que x_{ij} sea binario y que i y j no pueden ser iguales, perteneciendo ambos a V .

Una generalización del TSP es el problema de rutas de vehículos, VRP (The Vehicle Routing Problem), que consiste en encontrar un ciclo Hamiltoniano de coste mínimo, para una serie de vehículos (Christofides, 1976), (Laporte, 1992 b), (Toth & Vigo, 2002), (Golden, et al., 2008).

El VRP, desde su planteamiento por Dantzig y Ramset en 1958, es también uno de los problemas de investigación operativa más estudiados (Toth & Vigo, 2002).

La versión básica del VRP es el CVRP (Capacitated VRP), en el que son deterministas las demandas de los clientes, que son atendidos por un conjunto de K vehículos, todos iguales, con una capacidad determinada. El objetivo del

CVRP es minimizar los desplazamientos de los K vehículos que partiendo de un almacén recorrerán todos los clientes, para finalizar cada tour en el almacén de origen (Toth & Vigo, 2002).

De modo similar al TSP definimos el VRP: sea $G = (V, A)$ un grafo, donde $V = \{0, \dots, n\}$ es el conjunto de vértices y sea A el conjunto de aristas. Los vértices $i = 1, \dots, n$ corresponden con los clientes, siendo el vértice 0 el almacén.

Se asocia a cada arista $(i, j) \in V$ un coste no negativo c_{ij} . Sea x_{ij} una variable binaria, que tomará el valor 1 si el arco (i, j) pertenece a la solución óptima y el valor 0 en caso contrario (Laporte, 1992 b), (Toth & Vigo, 2002).

$$\text{Min } \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (8)$$

$$\sum_{i \in V} x_{ij} = 1; \forall j \in V \setminus 0 \quad (9)$$

$$\sum_{j \in V} x_{ij} = 1; \forall i \in V \setminus 0 \quad (10)$$

$$\sum_{i \in V} x_{i0} = K \quad (11)$$

$$\sum_{j \in V} x_{0j} = K \quad (12)$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S); \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (13)$$

$$x_{ij} \in \{0,1\}; \forall i, j \in V \quad (14)$$

La función objetivo (8) minimiza los costes de desplazamiento. Las ecuaciones (9) y (10) limitan que a cada vértice solo entre y salga un arco, las (11) y (12) imponen que el número de arcos que entran y salen del almacén coincida con el número de vehículos. La restricción (13), además de limitar la capacidad, elimina los subtours, por lo que para cualquier subconjunto $S \subset V$, éste debe de ser atravesado por un número mayor de arcos que $r(S)$, que es el mínimo

número de vehículos necesarios para visitar todos los clientes en la solución óptima.

El VRP en la realidad presenta múltiples limitaciones particulares, por lo que el problema es subdividido y clasificado en diversas variantes que pueden aparecer operativamente (Toth & Vigo, 2002), (Golden, et al., 2008).

Una característica del TSP y del VRP consiste en que todos los clientes tienen que ser atendidos, por lo que no existe un valor asociado a la visita. No obstante, existen variantes de estos problemas en los que se propone la elección de los clientes visitados en función de la recaudación que dicha visita aporte. Estos problemas son conocidos como el TSP con ganancias (TSPs with Profits), (Feillet, et al., 2005), o problema de rutas con beneficios (Archetti, et al., 2014c).

El problema objeto de esta tesis puede ser considerado como un TSP o VRP con dos criterios, cuyos objetivos son opuestos: por un lado, maximizar la recolección de los ingresos asociados a las visitas y por otro minimizar los gastos de viaje (Feillet, et al., 2005), (Archetti, et al., 2014c).

Las distintas variantes del TSP o VRP que pueden englobar nuestro problema, que han ido surgiendo, pueden clasificarse en los siguientes problemas (Feillet, et al., 2005), (Archetti, et al., 2014c):

- El problema de orientación, OP (Orienteering Problem), cuyo objetivo es maximizar los ingresos y como restricción limita la duración de la ruta. El problema se plantea para un solo vehículo.
- Problema del tour rentable, PTP (Profitable Tour Problem), en el que la diferencia entre los ingresos y el coste es la función objetivo que se maximiza. El problema está planteado para un único vehículo.
- El problema del viajante de comercio con recogida de premios, PCTSP (Prize Collecting Traveling Salesman Problem), que tiene como función objetivo minimizar el coste, pero limitando la ruta a unos ingresos dados. Al igual que los anteriores solo contempla un único vehículo.
- El problema del equipo de orientación, TOP (Team Orienteering Problem), es un OP para varios vehículos.

No existe una clara definición del PTP ni del PCTSP en la literatura (Archetti, et al., 2014c), por lo que se utilizarán las definiciones dadas por (Feillet, et al., 2005).

Al menos existen cuatro revisiones literarias sobre el problema de rutas con beneficios (Feillet, et al., 2005), (Vansteenwegen, et al., 2011), (Archetti, et al., 2014c), (Vansteenwegen & Gunawan, 2019).

Se propone el estudio del OP, PTP y PCTSP teniendo en cuenta los tiempos de operación. Así mismo, en este trabajo, se define la complejidad de cada uno de estos tres problemas sin tener en cuenta dichos tiempos de intervención (Angelelli, et al., 2014).

Ha de mencionarse que, en la mayor parte de los planteamientos de resolución de estos problemas con dos criterios, se ponderan y combinan linealmente. Keller (1985) y Kekker & Goldman (1988) han intentado resolver el problema bi-criterio y multicriterio, definiendo el TSP multi-criterio (Feillet, et al., 2005).

2.1. El problema de orientación (OP)

El OP (Orienteering Problem), (Chao, et al., 1996b) se basa en el juego con dicho nombre, donde los jugadores, individualmente, partiendo de un punto inicial, disponen de un tiempo máximo para recorrer una serie de puntos de control. Cada jugador, al visitar un punto de control, consigue una puntuación previamente conocida. Los jugadores han de llegar hasta el punto de control final antes de que finalice un tiempo máximo. Con el tiempo disponible no es posible que se visiten todos los puntos de control. El objetivo del juego es conseguir la mayor puntuación posible.

El OP (Vansteenwegen, et al., 2011) es también conocido con los siguientes nombres: el problema vendedor ambulante selectivo (Laporte & Martello, 1990), (Gendreau, et al., 1998a), (Thomadsen & Stidsen, 2003), el problema de la máxima colección (Kataoka & Morito, 1988), (Butt & Cavalier, 1994) y el problema ladrón de bancos (Arkin, et al., 1998), (Awerbuch, et al., 1998) que se define como la elección del recorrido de los bancos a atracar, sabiendo el dinero que contienen, maximizando el dinero robado, pero teniendo como limitación que el vehículo no puede repostar, puesto que en la gasolinera se avisaría a la policía.

El OP en el que el punto de partida y el final coinciden fue definido como el problema de la gira de orientación (Ramesh, et al., 1992), (Mansini, et al., 2006).

Se ha demostrado que OP es NP-hard (Golden, et al., 1987). Tomando como base el TSP, que es NP-hard, se cuestiona si en el OP existirá un tiempo T con el que daría tiempo visitar todos los nodos, se ajusta la puntuación de cada vértice al valor 1 y el tiempo máximo a T . Si al resolverse el problema la puntuación obtenida es igual al número de vértices, se habrá transformado el OP en un TSP.

Podemos expresar el OP del siguiente modo: sea un conjunto de N vértices y sea S_i la puntuación asignada a cada uno de ellos. Se han identificado los vértices de salida V_1 y de llegada V_n . Los tiempos que se tarda en desplazarse entre los vértices t_{ij} son conocidos. Se dispone de un tiempo máximo $T_{máx}$ para realizar un recorrido que se inicia en el vértice de salida, que recorra una serie de vértices para obtener las puntuaciones asociadas a ellos y finalice en el vértice de llegada, obteniéndose la mayor puntuación posible.

El problema puede definirse del siguiente modo, sea $G = (V, A)$ un grafo, siendo el conjunto de vértices $V = \{V_1, \dots, V_n\}$ y siendo A el grupo de arcos $a_{ij} \in A$ que unen dichos vértices y se definen los tiempos t_{ij} como los tiempos necesarios para desplazarse por cada uno de los arcos, $a_{ij} \in A$. A cada vértice $V_i \in V$ se le asigna una puntuación no negativa S_i .

El objetivo del OP consiste en determinar un camino Hamiltoniano $G' \subseteq G$, sobre un subconjunto de vértices $V' \subseteq V$, al que pertenecen los vértices de salida V_1 y de llegada V_n , en el que la suma total de los tiempos de desplazamiento entre los vértices del camino no supere el tiempo máximo definido $T_{máx}$, de modo que la suma de las puntuaciones de los vértices visitados sea máxima.

Se define la variable binaria x_{ij} , para el camino G' , que tomará el valor 1 si se recorre el arco a_{ij} o tomará el valor 0 en caso contrario. Así mismo, se definen las variables u_i , que tomarán el valor de la posición del vértice V_i en el camino G' . (Feillet, et al., 2005):

$$\text{Max} \sum_{i=2}^{N-1} \sum_{j=2}^N S_i x_{ij} \quad (15)$$

$$\sum_{j=2}^N x_{ij} = \sum_{i=2}^{N-1} x_{iN} = 1 \quad (16)$$

$$\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^N x_{kj} \leq 1; \forall k = 2, \dots; N - 1 \quad (17)$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^N t_{ij} x_{ij} \leq T_{max} \quad (18)$$

$$2 \leq u_i \leq N; \forall i = 2; \dots; N \quad (19)$$

$$u_i - u_j + 1 \leq (N - 1)(1 - x_{ij}); \forall i, j = 2; \dots; N \quad (20)$$

$$x_{ij} \in \{0,1\}; \forall i, j = 1; \dots; N \quad (21)$$

La ecuación (15) es la función objetivo que maximiza la puntuación total adquirida, teniendo las siguientes restricciones: mediante la ecuación (16) nos aseguramos que la gira se inicia en V_1 y finaliza en V_n , la ecuación (17) nos garantiza que los vértices son visitados una sola vez, la restricción que impide que no se supere el tiempo máximo permitido viene dada por la ecuación (18), garantizamos que no existan subtours mediante las ecuaciones (19) y (20) (Miller, et al., 1960). Por último, la restricción (21) establece que las variables x_{ij} sean binarias.

La diferencia con el TSP es que no tienen que ser visitados todos los clientes y que la función objetivo consiste en maximizar el beneficio de la recaudación (Vansteenwegen & Gunawan, 2019).

El OP es un problema de elección multinivel, en un primer nivel hay que elegir un subconjunto de vértices a visitar y en un segundo nivel hay que resolver el TSP, que viene a ser el encontrar el camino Hamiltoniano más corto para el subconjunto de vértices definidos en el primer nivel. Ambos niveles están estrechamente relacionados. Por ejemplo, si no es posible resolver el TSP de un conjunto de vértices, porque se supera el tiempo máximo, será necesario la eliminación de alguno de dichos vértices. Por otro lado, puede que sea posible añadir algún nuevo vértice para alguna solución del TSP, intentando maximizar la puntuación obtenida (Chao, et al., 1996b).

Podría decirse que resolver el OP sería como resolver el problema de la mochila (Knapsack Problem - KP), para elegir los vértices a visitar y posteriormente determinar el orden de la ruta mediante un TSP (Vansteenwegen & Gunawan, 2019), (Vansteenwegen, et al., 2011).

Como revisión bibliográfica del OP se puede consultar (Vansteenwegen & Gunawan, 2019), (Gunawan, et al., 2016), (Vansteenwegen, et al., 2011).

La mayor parte de los estudios consideran estocástica la demanda de los clientes. En algunos casos se estudia el problema teniendo en cuenta que los tiempos de los viajes son estocásticos debido a las variaciones del tráfico (Laporte, et al., 1992), (Malandraki & Daskin, 1992), (Taniguchi, et al., 1999), (Lau, et al., 2012).

Para resolver el problema, en los años 80, se utilizaban dos enfoques: un método estocástico que utiliza la simulación Montecarlo para generar gran cantidad de rutas, eligiéndose como solución la de mayor puntuación y una segunda línea en la que se busca la creación de rutas dentro de unos sectores en los que se divide el área geográfica. Estos sectores están delimitados por dos círculos concéntricos y un determinado arco. Para estudiar distintas posibilidades se giran los ejes $\pi/2$. La propuesta pretende minimizar los desplazamientos. Los sectores van cambiando al variarse los radios de los círculos y la longitud del arco (Tsiligrídes, 1984). Tsiligrídes, mediante un segundo algoritmo, mejora su solución aplicando tres pasos finales: aplica en primer lugar un algoritmo heurístico que reduce la trayectoria de la que se parte, en segundo lugar, se intentan insertar nuevos vértices y finalmente, manteniendo la viabilidad del trayecto, se intercambian vértices de dentro de la ruta con otros que han quedado fuera, con la intención de mejorar la puntuación del viaje.

Se propone un método en tres etapas para la solución del OP (Golden, et al., 1987), en la primera se propone una ruta, aplicándose a cada punto un valor en función de un centro de gravedad calculado, en la segunda se van intercambiando vértices de dentro de la ruta con vértices que han quedado fuera, buscando reducir el trayecto, por último, se recalcula el centro de gravedad para mejorar la trayectoria. Es decir, en la primera etapa se genera una ruta, buscando los vértices de máxima puntuación, pero solo los que, al incluirse, el tiempo de viaje no supere el tiempo máximo. En una segunda etapa se mejora la ruta aplicando un algoritmo de intercambio que busca disminuir el tiempo de viaje, para finalmente ir incluyendo nuevos nodos, sin que se supere el tiempo máximo de viaje. En la tercera etapa se calcula el centro de gravedad de la ruta y observa la relación entre la puntuación de cada vértice con su distancia al centro de gravedad. Se intenta mejorarla la ruta suponiendo que se incluyen nuevos vértices, teniendo en cuenta la nueva posición de su centro de gravedad.

Tomando la idea del centro de gravedad de Tsiligrídes se incorpora un algoritmo con capacidad de aprendizaje (Golden, et al., 1988).

Para la resolución del OP, también se propone la relajación de Lagrange, introduciéndose en la función objetivo las restricciones que impiden superar el tiempo máximo y la formación de subtour (Kataoka & Morito, 1988).

Como métodos exactos, debemos de mencionar a (Laporte & Martello, 1990), quienes proponen un método de ramificación y acotamiento que se aplicó en un problema con 290 vértices. Dos métodos se proponen en la tesis de (Sokkappa, 1990), uno basado en el método de ramificación, vinculándolo

al problema de la mochila y el segundo, un heurístico basado en el propuesto por (Golden, et al., 1988).

Un algoritmo de cuatro fases es propuesto por (Ramesh & Brown, 1991). En primer lugar, se crea una ruta, posteriormente se insertan y eliminan vértices para mejorar la puntuación, en la tercera fase se eliminan vértices y se insertan otros para reducir la trayectoria, finalmente se inserta el mayor número posible de vértices a la trayectoria. Las tres últimas fases se aplican repetidamente para la búsqueda de una solución de gran calidad.

Un algoritmo exacto, basado en la relajación de Lagrange, utilizado para resolver problemas con 150 vértices, fue propuesto (Ramesh, et al., 1992). RS Pillai, en su tesis, presentada en 1992 propone un procedimiento exacto basado en el método de planos de corte. Utiliza un método relajado, en el que observa las restricciones no cumplidas, para ir añadiéndolas en un segundo algoritmo, y así sucesivamente hasta que se hayan cumplido todas las restricciones.

Se propuso un algoritmo basado en redes neuronales (Wang, et al., 1995), que emplearon una red neuronal Hopfield continua.

Un algoritmo heurístico en dos fases es presentado (Chao, et al., 1996b), en la fase inicial se genera una elipse en la que los puntos de salida y llegada son sus focos y en el que la longitud del eje mayor se corresponde con el tiempo máximo disponible. Se generará una ruta teniendo en cuenta los puntos interiores a la elipse y en la fase final de mejora, se espera encontrar una ruta con una puntuación casi óptima. Inicialmente se genera una primera ruta que contenga el punto con mayor distancia entre el punto de inicio y el de fin. Se van incrementando puntos con un método codicioso que pretende ir aumentando el número puntos a visitar, de modo que sea mínimo el incremento del coste de viaje, sin tener en cuenta el incremento en la puntuación obtenida, hasta que se supera el límite de tiempo. Se construyen nuevas rutas con los puntos restantes, hasta que se utilicen todos los puntos dentro de la elipse. De entre estas rutas propuestas se elige la de mayor puntuación y tomaremos este valor como solución inicial. En la fase de intercambio de puntos se intentará mejorar la solución presentada en la primera etapa intercambiando puntos de la solución óptima con los de otra ruta, pero con el mínimo coste posible, es decir, cada punto hay que insertarlo entre los puntos adecuados y no se podrá superar el tiempo máximo determinado. Si se superase este límite, el punto se incluirá en una nueva ruta. Estos intercambios pueden hacer que la ruta óptima deje de serlo ya que otra ruta puede tener ahora mayor puntuación. Con posterioridad se estudiará si es posible reordenar la posición de los vértices de modo que se ahorre desplazamiento, para una nueva inserción si fuese posible.

Hubo varias propuestas utilizando la combinación de Lagrange con un método simplex dual (Leifer & Rosenwein, 1994), (Kataoka, et al., 1998). También se plantean un método basado en la herramienta CPLEX (Millar & Kiragu, 1997).

Se propuso varios métodos exactos basados en la ramificación y acotamiento (Fischetti, et al., 1998), (Gendreau, et al., 1998a).

En 1998 se proponen varios métodos aproximados para resolver el OP (Gendreau, et al., 1998b), (Awerbuch, et al., 1998).

Para resolver el OP también se utilizaron algoritmos genéticos, la aplicación propuesta era muy costosa computacionalmente, se utilizaron diferentes representaciones, funciones de penalización simple y operadores de cruce (Tasgetiren & Smith, 2000).

Se presentó un algoritmo genético en el que se implementa una función de penalización de adaptación, con la que se descartan soluciones no factibles (Tasgetiren, 2001).

También se propone una solución mediante colonias de hormigas y una segunda mediante la búsqueda tabú (Liang, et al., 2002).

Otro enfoque, donde se presenta el primer algoritmo con factor de aproximación constante, se plantea en (Blum, et al., 2007). (Chekuri, et al., 2012) muestran soluciones de aproximación para el problema dirigido y no dirigido.

Una solución mediante metaheurístico del enjambre de partículas es presentada por (Sevkli & Sevilgen, 2010).

Un algoritmo de ramificación y corte se plantea para su resolución (Archetti, et al., 2014c).

(Campos, et al., 2014) presentan cuatro métodos GRASP que solucionan el OP.

Se propone una solución mediante un algoritmo memético – GRASP (MenGRASP) en (Marinakis, et al., 2015).

Otra solución es presentada mediante un algoritmo basado en la agrupación en clústeres (Dbscan) y la búsqueda adaptativa de grandes vecindarios (ALNS) (Santini, 2019).

Se empleó el OP para optimizar la entrega del combustible para calefacción para una lista de clientes, la puntuación a obtener en cada entrega coincide con la prioridad con la que cada cliente hace su solicitud de entrega. El gerente de la distribuidora de combustible escoge diariamente el conjunto de clientes a ser atendidos (Golden, et al., 1984).

El OP también se utiliza para aplicaciones turísticas en las que se escoge unos puntos de interés a visitar (POI), que se ponderan según la satisfacción que daría su visita al usuario de la aplicación. La aplicación ordenaría las visitas de modo que resultase el máximo beneficio del usuario, cumpliendo las limitaciones de tiempo (Verbeeck, et al., 2014), (Vansteenwegen & Gunawan, 2019).

Se utiliza el OP para resolver el problema de las empresas que gestionan el stock de sus proveedores, Vendor Managed Inventory (VMI) (Vansteenwegen & Mateo, 2014).

Se realizaron experimentos con camiones de reparto y de recogida comparándose con soluciones óptimas probabilísticas, analizándose los costes, los tiempos de funcionamiento totales y los impactos ambientales (Taniguchi & Shimamoto, 2004).

2.2. Problema del tour rentable (PTP)

El PTP consiste en encontrar una ruta para un vehículo que maximice la diferencia entre los premios recogidos en los vértices visitados y el coste total del viaje.

Sea c_{ij} el coste asociado al desplazamiento por el arco $(i, j) \in A$ y un premio p_i está asociado a cada nodo $i \in V$. Sean las variables binarias, y_i , que tomará el valor 1 si V_i es visitada y 0 en caso contrario y la variable x_{ij} ya definida en el OP (Archetti, et al., 2014c):

$$\text{Max} \sum_{i \in V \setminus \{0\}} p_i y_i - \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (22)$$

$$\sum_{j \in V} x_{ij} = y_i; \forall i \in V \quad (23)$$

$$\sum_{i \in V} x_{ij} = y_j; \forall j \in V \quad (24)$$

$$y_0 = 1 \quad (25)$$

$$\sum_{(i,j) \in \delta^+(S)} x_{ij} \geq y_h; \forall S \subset V; 0 \in S; \forall h \in V \setminus S \quad (26)$$

$$x_{ij} \in \{0,1\}; \forall i, j \in A \quad (27)$$

$$y_i \in \{0,1\}; \forall i \in V \quad (28)$$

La función objetivo definida en (22) maximiza la diferencia entre los ingresos y los costes. Las restricciones (23) y (24) obligan a que para cada vértice visitado entre un arco y salga otro. La ecuación (25) obliga a que el almacén o base sea visitado en primer lugar. Si denotamos como $\delta^+(S)$ el conjunto de arcos que salen de S , la restricción (26) elimina la formación de subtours. Finalmente, las restricciones (27) y (28) hacen binarias las variables x_{ij} e y_i .

La función objetivo a maximizar es la diferencia entre lo recaudado y los costes (Vansteenwegen & Gunawan, 2019).

La diferencia con el TSP es que en éste hay que visitar todos los clientes mientras que, en el PTP, puesto que nuestra restricción de tiempo nos impide visitarlos todos, escogemos los clientes a visitar en función del beneficio que se consiga al realizar la visita. El PTP se diferencia del OP en que el coste de viaje, que era una restricción en el OP, ahora pasa a ser parte de la función objetivo (Vansteenwegen & Gunawan, 2019).

El PTP es un NP-hard (Vansteenwegen & Gunawan, 2019).

Se define el problema del PTP y sus variantes, además de mencionar métodos para su resolución, en (Aksen & Aras, 2005).

Para la resolución del PTP se han realizado diversas propuestas referentes a resultados de aproximación (Bienstock, et al., 1993), (Goemans & Williamson, 1995), (Goemans, 2009), (Nguyen & Nguyen, 2010).

Se ha propuesto un algoritmo de aproximación que aplica el árbol de Steiner, suponiéndose que satisface la desigualdad triangular (Nguyen & Nguyen, 2012), resolviéndose el problema asimétrico mediante un algoritmo de aproximación basado en el algoritmo Held-Karp (Nguyen, 2013).

2.3. El problema del viajante de comercio con recogida de premios (PCTSP)

El PCTSP (the Prize Collecting Travelling Salesman Problem) fue introducido por Balas y Matin (Balas & Martin, 1985), como una variante del VRP, en el que se penaliza al vendedor por cada ciudad que no visita. Posteriormente Balas estudió el problema, sus propiedades y distintos métodos de resolución (Balas, 1989).

Dado un diágrafo $G = (V, A)$, donde $V = \{1, 2, \dots, n\}$ es el conjunto de vértices, suponiéndose que el nodo 1 es el almacén o ciudad base y donde A es el conjunto de arcos, con n elementos. Un coste c_{ij} está asociado a cada arco $(i, j) \in A$, un premio p_i está asociado a cada nodo $i \in V$ y una penalización γ_i debe ser pagada si no se visita el nodo $i \in V$. Para el nodo 1 se considera que $p_1 = 0$ y que $\gamma_1 = +\infty$.

El PCTSP consiste en encontrar un ciclo $Z = (V_z, A_z)$, $V_z \subseteq V$, $A_z \subseteq A$, con $1 \in V_z$, en el que se visitarán los nodos en una sola ocasión, de modo que se minimizará la suma de costes de los nodos que se visiten junto con la suma de las penalizaciones asociadas a los vértices no visitados, teniendo en cuenta que la suma de premios recaudados no debe ser menor a un valor dado B .

Se utilizan las variables binarias x_{ij} e y_i definidas anteriormente, por lo que el PCTSP puede formularse:

$$\text{Min} \sum_{i \in V} \sum_{j \in V \setminus i} c_{ij} x_{ij} + \sum_{i \in V} \gamma_i (1 - y_i) \quad (29)$$

$$\sum_{j \in V \setminus i} x_{ij} = y_i; \forall i \in V \quad (30)$$

$$\sum_{i \in V \setminus j} x_{ij} = y_j; \forall j \in V \quad (31)$$

$$y_1 = 1 \quad (32)$$

$$\sum_{i \in V} p_i y_i \geq B \quad (33)$$

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq y_h; \forall h \in V \setminus 1; \forall S \subset V \quad (34)$$

$$x_{ij} \in \{0,1\}; \forall i, j \in A \quad (35)$$

$$y_i \in \{0,1\}; \forall i \in V \quad (36)$$

La ecuación (29) representa la función objetivo, en la que se pretende minimizar la suma de costes de los arcos incluidos en la ruta más la suma de las penalizaciones referentes a los nodos no visitados. Las ecuaciones (30) y (31) nos aseguran la continuidad del ciclo, la (32) obliga a que la gira se inicie en el almacén o ciudad base. La ecuación (33) obliga a que la suma de los premios recaudados no sea inferior al valor predeterminado B . La restricción (34), además de impedir la generación de subrutas, obliga a que el almacén o ciudad base pertenezca a la ruta. Finalmente, las ecuaciones (35) y (36) limitan las variables para que sean binarias.

Uno de los primeros estudios que se realizó sobre el PCTSP con penalizaciones lo efectuó (Balas, 1995).

En el estudio de (Balas, 1999) se definen nuevas restricciones, consiguiéndose soluciones para el caso del PCTSP con penalizaciones.

Otros autores consideran el PCTSP sin penalizaciones, por lo que la función objetivo a minimizar serán solamente los costes de desplazamiento, teniendo en cuenta que la ruta debe de generar un beneficio mínimo (Vansteenwegen & Gunawan, 2019).

La diferencia con el TSP es que no se deben de visitar todos los clientes y que la función objetivo consiste en minimizar las penalizaciones y costes de desplazamiento. Las diferencias con el OP es que en este segundo se maximiza el beneficio, en lugar de minimizar los costes totales. Finalmente, se diferencia del PTP en que los beneficios no son parte de la función objetivo si no una restricción (Vansteenwegen & Gunawan, 2019).

Para facilitar el cálculo, en muchos estudios se considera el tiempo de operación como parte del tiempo de desplazamiento (Vansteenwegen & Gunawan, 2019).

El PCTSP es un NP-hard (Vansteenwegen & Gunawan, 2019).

Se define el problema del PCTSP y sus variantes, además de mencionar métodos para su resolución en (Aksen & Aras, 2005)

Para su resolución, se propone un algoritmo de ramificación y poda, en el que, para cada nodo de la ramificación, se imponen y restringen arcos (Fischetti & Toth, 1988).

El primer algoritmo de aproximación para la PCTSP con penalizaciones, no dirigido, es desarrollado por (Balas, 1989), dicho algoritmo fue mejorado por (Awerbuch, et al., 1998), basándose ambos estudios en problema del k -árbol de peso mínimo.

(Dell'Amico, et al., 1995) estudian los límites inferiores para el problema asimétrico.

Una de las primeras soluciones heurísticas para el PCTSP con penalizaciones se presenta en (Dell'Amico, et al., 1998), donde aprovechando un límite inferior, utiliza un heurístico de Lagrange para obtener una solución factible.

Más recientemente, se propuso un algoritmo de ramificación y corte para el PCTSP (Bérubé, et al., 2009a).

2.4. El problema del equipo de orientación (TOP)

Si el juego de orientación se jugase por equipos, TOP (Team Orienteering Problem), el objetivo consistiría en maximizar la suma total de las puntuaciones obtenidas por cada miembro del equipo (Chao, et al., 1996a). El problema del equipo de orientación también es conocido como el problema de la máxima recolección de la gira múltiple (Butt & Cavalier, 1994).

El TOP puede ser enunciado así: para un equipo de P jugadores, la búsqueda de P caminos Hamiltonianos, tal que la suma de las puntuaciones obtenidas por los P jugadores, al recorrer los vértices o puntos de control, sea máxima, todos ellos sujetos a no superar el tiempo máximo establecido. Definiremos las variables binarias x_{ijp} , que tomarán el valor 1 en el caso que, para un camino p , tras visitar el vértice V_i se visite el V_j , en caso contrario tomarán el valor 0. Las variables binarias y_{ip} tomarán el valor 1 si, en un camino p , se visita el vértice V_i , en caso contrario tomarán el valor 0. Así mismo, se definen las variables u_{ip} , que toman el valor de la posición del vértice V_i dentro del camino p :

$$\text{Max} \sum_{p=1}^P \sum_{i=2}^{N-1} S_i y_{ip} \quad (37)$$

$$\sum_{p=1}^P \sum_{j=2}^N x_{ijp} = \sum_{p=1}^P \sum_{i=1}^{N-1} x_{iNp} = P \quad (38)$$

$$\sum_{p=1}^P y_{kp} \leq 1; \forall k = 2, \dots, N-1 \quad (39)$$

$$\sum_{i=1}^{N-1} y_{ikp} = \sum_{j=2}^N x_{kjp} = 1; \forall k = 2, \dots, N-1; \forall p = 1, \dots, P \quad (40)$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^N t_{ij} x_{ijp} \leq T_{max}; \forall p = 1, \dots, P \quad (41)$$

$$2 \leq u_{ip} \leq N; \forall i = 2, \dots, N; \forall p = 1, \dots, P \quad (42)$$

$$u_{ip} - u_{jp} + 1 \leq (N-1)(1 - x_{ijp}); \forall i, j = 2, \dots, N; \forall p = 1, \dots, P \quad (43)$$

$$x_{ijp}, y_{ip} \in \{0,1\}; \forall i, j = 1; \dots, N; \forall p = 1, \dots, P \quad (44)$$

La función objetivo (37) pretende maximizar las puntuaciones obtenidas por los jugadores, sujetas a las siguientes restricciones: que las rutas se inicien y finalicen en los vértices inicial V_1 y final V_N (38), que cada vértice solo es visitado en una ocasión (39), que cada ruta sea conexa (40), que para cada camino no se supere el tiempo máximo asignado (41), las restricciones (42) y (43) evitarán que se formen subgras y la (44) garantiza que sean binarias las variables x_{ijp} e y_{ip} .

La formulación del OP y del OTP no tiene grandes diferencias, nótese que para convertir un OP en OTP solo habría que crear un arco ficticio, con coste cero, que una el vértice de salida con el de llegada.

Puesto que el OP es NP-hard, el TOP tiene la misma consideración (Vansteenwegen & Gunawan, 2019).

En la revisión bibliográfica de (Gunawan, et al., 2016) se presentan las soluciones aportadas.

El primer heurístico para el TOP es un algoritmo de construcción sencilla (Butt & Cavalier, 1994), que se plantea para la resolución del MTMCP, presentando la solución tres características: las soluciones son factibles (si existen), las soluciones son buenas, incluso óptimas en muchos casos. El coste temporal de resolución no fue elevado para problemas de 15 vértices.

Otra propuesta heurística fue presentada por (Chao, et al., 1996a), capaz de resolver problemas de hasta 102 vértices de forma rápida. Ese mismo año, los mismos autores presentan una nueva heurística de construcción (Chao, et al., 1996b).

Un método exacto fue presentado por Gueguen (Methodes de résolution exacte verter Problèmes de Tournées de véhicules. Estas de Doctorat, Ecole Centrale París 1999). Con posterioridad, basándose en el algoritmo de ramificación y precio, se presenta una nueva solución exacta (Boussier, et al., 2007).

Una propuesta que combina la búsqueda tabú con un proceso de adaptación entre barrios grandes y pequeños, durante el periodo de mejora, fue estudiado por (Tang & Miller-Hooks, 2005a).

(Archetti, et al., 2007) proponen la combinación de la búsqueda tabú con un algoritmo de búsqueda local variable.

La optimización mediante colonias de hormigas, ACO (Ant Colony Optimization) también fue aplicada para la resolución del TOP (Ke, et al., 2008). En su trabajo presentan cuatro métodos deterministas y aleatorios, concurrentes y simultáneos.

Con posterioridad se presenta el estudio que combina, sobre varios algoritmos, la búsqueda local guiada, GLS (Guided Local Search), a la que se añade un segundo algoritmo heurístico que diversifique la búsqueda para explorar un mayor número de áreas en el espacio de las soluciones (Vansteenwegen, et al., 2009a).

Se presenta el concepto de la re-vinculación heurística para resolver el TOP (Souffriau, et al., 2010). Se presenta una variedad lenta y otra rápida en el mencionado trabajo.

Un algoritmo genético híbrido se utilizó para resolver el TOP, en el que utiliza un procedimiento Split óptimo para la evaluación de los cromosomas y empleó técnicas de búsqueda local para las mutaciones (Bouly, et al., 2010). A este algoritmo híbrido se le llamó algoritmo Memético, MA (Memetic Algorithm).

Varios algoritmos de ramificación y acotamiento se presentan en (Dang, et al., 2013) y (Bianchessi, et al., 2018). Otro algoritmo que resuelve el TOP plantea una solución de corte de rama y premio del cliente visitado (Poggi, et al., 2010).

Posteriormente, se propuso emplear el algoritmo de optimización con enjambre de partículas, PSO (Particle Swarm Optimization), basado en el algoritmo de optimización memética (MA) (Dang, et al., 2011).

Otra solución planteada se basa en el recocido simulado con múltiples arranques (Lin, 2013).

También se ha estudiado el TOP con distintos tipos de vehículos, resolviendo el problema con algoritmos de aproximación (Xu, et al., 2020).

El TOP se utiliza para aplicaciones turísticas en las que se escoge unos puntos de interés a visitar (POI), que se ponderan según la satisfacción que daría su visita al usuario de la aplicación. La aplicación ordenaría las visitas de modo que resultase el máximo beneficio al usuario cumpliendo las limitaciones de tiempo (Vansteenwegen & Gunawan, 2019), (Verbeeck, et al., 2014).

Se ha planteado el TOP como el problema para resolver la solicitud online de transporte y servicios (Vansteenwegen & Gunawan, 2019).

2.5. TSP multi-criterio

Los primeros estudios sobre bi-criterios en el TSP fueron realizados por Keller, concretamente en el llamado MVP (the Multiobjective Vending Problem) (Keller, 1985), (Keller & Goodchild, 1988).

Si consideramos un TSP en el que se obtenga una recompensa R cada vez que se visite un nodo, teniendo en cuenta que tenemos una penalización (coste o tiempo de viaje) P en cada desplazamiento, podemos plantearnos si merece la pena que se visite todos los nodos.

Podemos formular el MVP como:

$$\text{Max } Z = (Z_1, Z_2) \quad (45)$$

$$Z_1 = \sum_{i=1}^m R_i; Z_2 = \sum_{i=1}^m P_{i,i+1} \quad (46)$$

$$R_i > 0; P_{i,i+1} > 0 \quad (47)$$

$$S_1 = S_{m+1} = S_d \quad (48)$$

$$S_i \in \{1, \dots, N\}; \forall i \quad (49)$$

$$S_i = S_j; \forall i, j; i \neq j \quad (50)$$

La ecuación (45) representa el objetivo general Z , que es una combinación de las funciones Z_1 y Z_2 . La relación entre las funciones Z_1 y Z_2 no se puede establecer hasta que se haya definido la función objetivo general.

La ecuación (46) establece que la función Z_1 representa a la suma de las recompensas R_i obtenidas al visitar los nodos y que a función Z_2 se corresponde a la suma de las penalizaciones ligadas a los desplazamientos, representándose por $p_{i,i+1}$, la penalización debida al desplazamiento desde el nodo i al nodo $i + 1$. Tanto las recompensas como las penalizaciones no pueden ser negativas (47). La restricción (48) obliga a que la ruta se inicie y finalice en el nodo base. S_i representa el elemento i -ésimo de la secuencia de vectores S y n es el número de nodos visitados.

No es posible que se generen subtours con esta formulación.

Se ha asumido que es un problema simétrico.

Con posterioridad se ha planteado el problema multicriterio (Ehrgott, 2000), basándose en el mismo planteamiento: sea G un grafo, la resolución del problema consiste en encontrar un ciclo Hamiltoniano $V \subseteq G$ tal que la suma de los pesos en C sea mínima:

$$\text{Min } (f_1(V), \dots, f_g(V)) \quad (51)$$

$$f_g(V) = \sum_{e \in C} w_q(v) \quad (52)$$

Donde v representan los nodos y w_q los pesos asociados a ellos.

La versión multi-objetivo de la OP fue estudiada (Schilde, et al., 2009). Este problema se deriva del problema de la planificación de las rutas turísticas individuales en una ciudad. Cada punto de interés (POI) en una ciudad ofrece diferentes beneficios dentro de diferentes categorías (por ejemplo, cultura, ir de compras, etc.). Cada turista tiene diferentes preferencias, para las diferentes categorías, al seleccionar y visitar los puntos de interés (por ejemplo, museos, iglesias). De ahí surge una situación de decisión multi-objetivo. Se presentan dos métodos de resolución, un algoritmo de optimización con colonias de hormigas y un método de búsqueda en barrios variables, aplicándoseles a ambos una hibridación mediante procedimientos de caminos re-encadenados.

El método para la resolución de los problemas multi-objetivo de optimización combinatoria, presentado por (Bérubé, et al., 2009b), consiste en transformar todas las funciones objetivo menos una en restricciones. Presentan una heurística, que acelera la resolución al utilizar la información de otros problemas previos.

Otra propuesta del estudio del TSPP como un bi-objetivo utiliza el óptimo de Pareto, para lo cual emplea una búsqueda local combinada con un algoritmo evolutivo múltiple-objetivo (Jozefowicz, et al., 2008).

2.6. Variantes al VRPP

A continuación, se presentan algunas variantes del problema de rutas para vehículos con ganancias.

El OP con ventanas de tiempo fue estudiado inicialmente por (Kantor & Rosenwein, 1992), el OPTW (The Orienteering Problem with Time Windows) que se trata de un OP en el que los nodos solo pueden ser visitados en un intervalo específico de tiempo. En este trabajo se proponen un algoritmo heurístico para su resolución. (Righini & Salani, 2009) presentan un algoritmo exacto, basado un algoritmo bidireccional delimitado por otro con programación dinámica decreciente. Otra solución planteada es una heurística codiciosa y un algoritmo de rama y poda (Erkut & Zhang, 1996).

Se define el OPTW como un OP en el que los servicios deben de recibirse entre una hora inicial y una hora final, definida por cada cliente o vértice del problema. Este problema se puede aplicar para la recogida de basuras, rutas de autobuses escolares, etc. Se puede dividir el problema en tres etapas: la elección de los clientes a incluirse en la ruta, la organización de la secuencia de la ruta y la programación de entrega de los clientes (Kantor & Rosenwein, 1992). Se propone para su resolución un algoritmo heurístico en árbol, que genera

numerosas rutas viables, para finalmente escoger la ruta con mayor puntuación y aplicarle un algoritmo de inserción de vértices, utilizando un método similar al de (Laporte & Martello, 1990).

Se presentan tres enfoques para resolver el OPTW, la primera basada en programación dinámica exacta, un segundo planteamiento con un algoritmo de programación dinámica con relajación del estado del espacio y finalmente una reducción de la relajación del estado del espacio (Righini & Salani, 2008).

En muchos artículos se considera que las ventanas temporales solo restringen el inicio del servicio y no la finalización del mismo, planteamiento que no se ajusta a la realidad en muchos casos. Para solventar esta situación se puede considerar el tiempo de operación como parte del tiempo de desplazamiento (Vansteenwegen & Gunawan, 2019).

Puede suceder que haya múltiples ventanas temporales, por ejemplo, para la resolución de problemas turísticos, donde una función tiene distintos horarios de pase. En estos casos el problema se conoce como OP con múltiples ventanas de tiempo OPmTW (Vansteenwegen & Gunawan, 2019).

El OP en el que hay que visitar un determinado conjunto de vértices obligatorios se define como el TSP selectivo, STSP (Selective Traveling Salesman Problem), (Gendreau, et al., 1998a). El STSP consiste en obtener el máximo de ingresos al diseñarse rutas en las que se visitarán los vértices obligatorios, pero sin exceder un coste máximo establecido. Para un STSP simétrico de 300 vértices se propone como solución un algoritmo de ramificación y corte.

Un problema más general que el OP es el llamado problema de un periodo del autobús turístico, BTP (One period Bus Touring Problem) (Deitch & Ladany, 2000). El objetivo de este problema es maximizar el atractivo de la ruta turística que se diseña para un autobús que ha de visitar una serie de lugares y rutas panorámicas. Los autores asumen que el problema es similar al TOP, donde el ingreso de cada visita se equipara con el interés del punto turístico visitado. Propusieron una heurística que resuelve de modo muy rápido el problema para 11 autobuses, con una solución próxima a la exacta y se compara con las heurísticas propuestas por (Tsiligirides, 1984).

El problema multi-objetivo del vendedor, MVP (The Multiobjective Vending Problem) ha de escoger la ruta a realizar teniendo en cuenta los ingresos que le repercutiría visitar cada cliente y los costes de cada viaje, por lo que el problema también puede ser considerado como OP. Se propone un algoritmo de dos etapas, la primera puede tener dos enfoques, el primero, determinista, asigna una puntuación de conveniencia a cada vértice no perteneciente al camino, seleccionando el que mayor puntuación tuviese, la segunda opción es estocástica, normalizando la puntuación que se asigna a cada vértice. En el

primer paso se inserta un vértice y se considera la posibilidad de no eliminar ninguno o eliminar uno o dos para mejorar la puntuación. En la segunda etapa se eliminan una nube de vértices y se inserta un conjunto para intentar incrementar la puntuación total (Keller & Goodchild, 1988).

Otra variante es el llamado problema de orientación dependiente del tiempo, TDO (Time Dependent Orienteering) (Fomin & Lingas, 2002). En el TDO el coste del desplazamiento no depende de a dónde viajar, sino de cuando hacerlo. Para resolver problemas de hasta 20 nodos se propone un algoritmo heurístico y un algoritmo de ramificación y acotamiento.

En el OP con beneficios estocásticos, OPSP (Orienteering Problem with Stochastic Profits), se debe encontrar un tour que visita un subconjunto de vértices, dentro de un límite de tiempo especificado previamente, con el objetivo de obtener el máximo de ingresos (Ilhan, et al., 2008). La clave de este problema radica en seleccionar el conjunto de clientes a visitar en función de la urgencia de su pedido, pero generando la ruta del modo más eficiente. Para su resolución se propuso un algoritmo exacto bi-objetivo.

Una variante del OP, es el llamado el problema de orientación generalizada, GOP (Generalized Orienteering Problem) (Wang, et al., 2008), (Silberholz & Golden, 2010). Supongamos el problema OP en el que se limita la distancia total que el viaje puede durar. En el OP, la puntuación total se obtiene sumando los ingresos asociados a cada punto visitado en el trayecto, en el GOP la puntuación total es una función no lineal basada en los puntos visitados. El GOP es, por lo tanto, un problema de optimización combinatoria con múltiples objetivos, con una función objetivo no lineal que se desea maximizar. A cada ciudad se le asigna varias funciones que se denominan atributos. Cualquiera de estas funciones puede ser la generadora de la puntuación si se produce la visita en la ruta a generarse. La función global para optimizar es una función de estos atributos que generan las puntuaciones.

El GOP es una versión generalizada de la OP en la que cada ciudad se le asigna un número de diferentes atributos. Por este motivo consideran que el GOP es un problema más flexible que el OP.

El problema del equipo de orientación teniendo en cuenta la capacidad, CTOP (The Capacitated Team Orienteering Problem), es estudiado por (Archetti, et al., 2009). En este problema la demanda está asociada con cada cliente y cada vehículo tiene una capacidad máxima. El objetivo es maximizar el ingreso total recogido, respetando las restricciones de capacidad de los vehículos y de duración máxima de cada ruta. En el mismo artículo se estudia el CPTP, proponiéndose un algoritmo ramificación y precio que resuelve problemas de hasta 100 clientes, este algoritmo es una adaptación del propuesto

por (Boussier, et al., 2007). Es uno de los primeros métodos exactos para este tipo de problemas, según el estudio presentado por Gueguen, *Methodes de résolution exacte verter Problèmes de Tournées de véhicules*, como tesis doctoral en París en 1999.

Se estudia una variante del CTOP en el que la prestación del servicio contratado no se ejecuta en su totalidad, sino parcialmente (CTOP-IS) (Archetti, et al., 2013b) que proponen un algoritmo de ramificación y precio.

Otras variantes de la CTOP se estudian en (Archetti, et al., 2014b) y (Archetti, et al., 2014a). En ambas se permiten entregas parciales, pero sirviéndose a todos los clientes, presentándose un algoritmo exacto de ramificación y precio y varios algoritmos heurísticos.

Se proponen algoritmos heurísticos para la resolución del problema CTOP en (Archetti, et al., 2007) y en (Archetti, et al., 2013a). En este último trabajo, se presenta un algoritmo de ramificación y poda, que, a pesar de su simplicidad, calcula soluciones de alta calidad, mejorando los resultados presentados en (Archetti, et al., 2009).

Más recientemente se estudia el OP en el que se penaliza al no atenderse a cada cliente o vértice, teniendo en cuenta además que los tiempos de viaje y de servicio son estocásticos, OPSTS (The Orienteering Problem with Stochastic Travel and Service times). Consideremos un negocio en el que haya que realizarse una entrega o la prestación de un servicio en la dirección de sus clientes. Supongamos que se desconocen las peticiones o encargos que van a surgir, además no podrán ser atendidas todas ellas, de modo que se obtendrá un beneficio por las tareas realizadas y una penalización por las no atendidas. Con el fin de maximizar el beneficio de la empresa se elegirán que clientes atender. Se supondrá además que los costes de desplazamiento entre los distintos clientes también son cambiantes, debido a las condiciones del tráfico (Campbell, et al., 2011). Se propusieron tres métodos exactos para tres casos particulares.

Así mismo se estudió el OP donde el beneficio de la visita es estocástico, OPSP (Orienteering Problem with Stochastic Profits), proponiendo un algoritmo exacto para problemas pequeños y uno genético para problemas con mayor número de centros (Ilhan, et al., 2008).

La variante del TOP con ventanas temporales (TOPTW) ha recibido considerable atención de la comunidad científica (Vansteenwegen, et al., 2009b). Una guía turística electrónica personalizada ayuda a los turistas en la planificación de su viaje. El problema de planificación, que necesita ser resuelto en tiempo real, puede ser modelado como un problema de equipo de orientación con ventanas de tiempo (TOPTW). En el TOPTW, se da un conjunto de lugares, cada uno con una puntuación, un tiempo de servicio y una ventana de

tiempo. El objetivo es maximizar la suma de las puntuaciones recogidas en un número fijo de rutas. Las rutas permiten visitar lugares en el momento adecuado y están limitadas con un tiempo máximo. Se propone un algoritmo que incluye un paso de inserción que permite escapar de óptimos locales.

Para la resolución de TOPTW se planteó el empleo de colonias de hormigas (Montemanni & Gambardella, 2009). El método se aprovecha de la generalización jerárquica del problema original, pero lo combinan con el algoritmo de las colonias de hormigas, obteniéndose unos resultados computacionales que demuestran la efectividad del método. Este método fue mejorado por (Gambardella, et al., 2012), que utilizan el Enhanced Ant Colony System.

Otros enfoques para la resolución del TOPTW se han implementado obteniéndose buenos resultados (Tricoire, et al., 2010), que estudian el problema de orientación de múltiples periodos con múltiples ventanas de tiempo, MuPOPTW (Multi-period Orienteering Problem with multiple Time Windows), que convierten el OP en un sub-problema con características reales, añadiéndole las limitaciones existentes para un caso concreto de la industria, proponiéndose un algoritmo exacto para su resolución.

El recocido simulado también es utilizado para resolver el TOPTW (Lin & Vincent, 2012).

Una nueva propuesta para resolver TOPTW se presenta en (Labadie, et al., 2012), donde se propone aplicar un VNS que explore ciertas zonas concretas en lugar de barrios enteros, es decir, de forma granular, de modo que se mejore la eficacia sin que ésta se pierda en la búsqueda, basado en (Toth & Vigo, 2003), que presentan un método en el que combinan la búsqueda Tabú con la búsqueda granular.

La simulación del comportamiento de las abejas europeas es la base del método heurístico ABC presentado por (Cura, 2014) para la resolución del TOPTW.

Se plantean distintas soluciones heurísticas para resolver el TOPTW en (Gunawan, et al., 2017), (Gunawan, et al., 2018), (Hu & Lim, 2014), (Lin & Vincent, 2012).

Para la resolución del MuPOPTW se aplicó el algoritmo de la búsqueda variable en el vecindario, VNS (Variable Neighbourhood Search) (Labadie, et al., 2011). La propuesta combina el procedimiento de búsqueda codiciosa aleatoria adaptativo (GRASP) con la búsqueda local evolutiva (ELS). Con ELS, empleándose un mecanismo de mutación, se generan múltiples soluciones distintas. Cada una de estas soluciones se mejora mediante un procedimiento de búsqueda local. Es decir, GRASP ofrece múltiples soluciones de partida al ELS.

El OPTW y TOPTW se utilizan para aplicaciones turísticas en las que se escoge unos puntos de interés a visitar (POI), que se ponderan según la satisfacción que daría su visita al usuario de la aplicación. La aplicación ordenaría las visitas de modo que resultase el máximo beneficio al usuario cumpliendo las limitaciones de tiempo (Vansteenwegen & Gunawan, 2019), (Verbeeck, et al., 2014).

Se estudió el PTP en conjunto con el PCTSP, mostrándose límites inferiores para el problema asimétrico, además, mediante experimentos computacionales se intenta demostrar que se puede resolver con exactitud el PTP asimétrico (Dell'Amico, et al., 1995).

En el problema del viajante de comercio con recogida de premios y restricciones de capacidad, CPCTSP (Capacitated Prize Collecting Travelling Salesman Problem), se pretende reducir al máximo los costes de viaje y las penalizaciones que se originen por no visitar a clientes, pero asegurándose los ingresos mínimos sin sobrepasar la capacidad del vehículo (Tang & Wang, 2008), proponiéndose una heurística de búsqueda interactiva.

Se estudia el PTP teniendo en cuenta la capacidad de los vehículos en el CPTP (Jepsen, 2011). Para la resolución del CPTP se propone un algoritmo de ramificación y corte para un problema con 800 clientes.

Se resuelve el PTP con restricciones de capacidad con una búsqueda Tabú adaptativa (Stavropoulou, et al., 2019).

El problema del viajante probabilístico, PTSP (Probabilistic Traveling Salesman Problem) también fue estudiado (Jaillet, 1988), (Teng, et al., 2004). Consiste en resolver el TSP en el que no ha de visitarse todos los vértices, sino a un conjunto de ellos siendo este conjunto estocástico (Bertsimas, et al., 1990).

También se presentó el problema del viajante de comercio con limitaciones de tiempo, con viajes estocásticos y resolviéndose mediante un algoritmo en L (Teng, et al., 2004). Otro caso particular es el problema del viajante selectivo en la que son estocásticos tanto los tiempos de servicio, como los tiempos y gastos de viaje, SSTSP (The Selective Travelling Salesperson Problem with Stochastic service times, travel times, and travel costs). Se propuso para su resolución un método exacto y uno heurístico (Tang & Miller-Hooks, 2005b).

Se estudia el problema del viajante estocástico con ventanas de tiempo, STSPTW (Stochastic Traveling Salesman Problem with Time Windows), donde los tiempos de viaje y de servicio son estocásticos, proponiendo una programación dinámica que resuelve problemas con 80 clientes (Jula, et al., 2006).

Una variante del PCTSP no dirigido, es aquel cuyo tiempo de servicio está asociado con cada una de las visitas, presentándose un algoritmo polinómico para su resolución (Angelelli, et al., 2014).

Otra variante del PCTSP es estudiada como el problema de recogida y entrega, con cargas divididas, beneficios y múltiples ventanas de tiempo (Jaballah, et al., 2021).

El problema de múltiples tours de recolección máxima, MTMCP (The Multiple Tour Maximun Collection Problem), se diferencia del TSP puesto que no es posible visitar a todos los clientes en el tiempo máximo permitido. Al visitarse a un cliente se obtiene una recompensa, pero no es posible que se visite al mismo cliente en más de una ocasión dentro del mismo tour, que parte y finaliza en el mismo punto llamado almacén. Es decir, al buscarse la recolección máxima de premios, el problema se puede definir como el TOP, presentándose en este trabajo la primera solución exacta (Butt & Ryan, 1999).

El VRP probabilístico con ventanas temporales (VRPTW-P) es modelado para minimizar los costes totales, que comprenden la suma de los costes fijos, costes de operación y las penalizaciones por la llegada fuera de la ventana temporal (Taniguchi, et al., 2001). Los tiempos de viaje entre los nodos se calculan estadísticamente, basándose en numerosas mediciones reales. Es decir, se pretende modelizar una aproximación de la realidad, teniendo en cuenta las siguientes condiciones:

- a) Un vehículo se le permite hacer múltiples rutas por día.
- b) A cada cliente se le debe asignar solamente una ruta de un vehículo y todas las mercancías de cada cliente deben cargarse en el vehículo a la misma hora.
- c) El peso total de las mercancías de una ruta no debe exceder la capacidad del vehículo.
- d) Un vehículo debe ser operado dentro del tiempo de funcionamiento establecido.

En el CVRP con tiempos de desplazamiento estocásticos, SVRP (Stochastic Vehicle Routing Problem) se buscan que todos los tours se realicen en un tiempo máximo determinado. Se propuso la creación de un catálogo de soluciones mediante simulación de Montecarlo (Kenyon & Morton, 2003).

El estudio de la influencia del tiempo de viaje se estudia en el CVRP con ventanas temporales (Ando & Taniguchi, 2006).

Gracias al creciente uso de Internet en los terminales telefónicos móviles, se están desarrollando aplicaciones turísticas on-line, por lo que el estudio de la optimización de las rutas turísticas se ha disparado.

El TTDP es conocido como el problema de diseño del viaje turístico (The Tourist Trip Design Problem), donde en primer lugar habrá de escogerse los puntos a visitar, los puntos de interés POIs y se planificará una ruta, teniendo en cuenta los horarios y tiempos de desplazamiento entre dichos puntos, sin que se supere el tiempo máximo disponible. Esta planificación de ruta se realiza online, en tiempo real, mediante las guías turísticas móviles, MTG (Mobile Tourist Guide) (Vansteenwegen & Van Oudheusden, 2007).

3. PLANTEAMIENTO

Las empresas mantenedoras deben de maximizar sus beneficios económicos, B^o . El beneficio es la diferencia entre los ingresos I y los costes C , siempre referidos a la moneda del país donde opera.

Parece, de este modo, que sea el beneficio la función objetivo del problema objeto de esta tesis, que deberá ser maximizada.

$$\text{Max } B^o = \text{Max } (I - C) \quad (53)$$

Dado que podemos encontrar cuatro tipos de contratos de mantenimiento, como se ha indicado en el capítulo 1, deberemos definir los Ingresos I y los costes C , para cada uno de ellos. Puesto que la matriz de costes de desplazamiento, tanto para el TSP y el VRP, pueden referenciarse a costes, distancias o tiempos, en nuestro caso utilizaremos la variable tiempo, por lo que se utilizará para el cálculo de los costes.

Tanto para el cálculo de las funciones objetivo como para poder plantear las restricciones, han de considerarse varias circunstancias que se comentarán a continuación.

Como también se indicó en el capítulo 1, las mantenedoras asignan las áreas geográficas a una serie de equipos de mantenimiento, pudiéndonos encontrar áreas con un solo equipo o áreas con más de uno.

Cada una de las bases de estas áreas geográficas coincide con un centro de trabajo, es decir, en la base puede haber tareas de mantenimiento preventivo o correctivo.

En un centro de trabajo puede haber más de un mantenimiento correctivo pendiente, además de un mantenimiento preventivo. A las tareas de mantenimiento pendientes las llamaremos incidencias.

Sea $G = (V, A)$ un grafo, donde $V = \{1, \dots, n\}$ es el conjunto de vértices y sea A el conjunto de aristas. Los vértices $i = 1, \dots, n$ se corresponden con los clientes, siendo el V_1 y V_n la base. De este modo las rutas se podrán iniciar y finalizar en la base.

Se asocia a cada arista $(i, j) \in A$ un tiempo de desplazamiento no negativo t_{ij} , que se corresponde con el tiempo para desplazarse desde V_i a V_j .

Sea x_{ij} una variable binaria, que tomará el valor 1 si el arco (i, j) pertenece a la solución óptima y el valor 0 en caso contrario.

Sea z_{pi} una variable que tomará el valor del número de tareas de mantenimiento preventivo pendientes en V_i , que se ejecutarán en la ruta. Análogamente sea z_{ci} la variable que tomará el valor del número de tareas de mantenimiento correctivo pendientes en V_i , que se ejecutarán en la ruta.

A cada orden de trabajo se le asigna una ventana temporal (A_i, B_i) , de tal modo que la visita al nodo V_i , para resolver las incidencias existentes, solo se podrá realizar entre su fecha de creación A_i y su fecha máxima de resolución B_i . Al nodo V_n se le asigna una incidencia con una ventana temporal con los valores $(0, \infty)$. Obsérvese que en el nodo V_1 no se le asigna este criterio ya que es sobre este nodo al que se asignarán las incidencias de la base.

Finalmente, se define la variable s_i que indica el inicio del servicio en el vértice V_i .

3.1. Ingresos

En el caso que se facturase por intervención de mantenimiento realizada (supuesto más general), podemos adaptar el planteamiento de PTP, del capítulo 2. Si para cada tarea de preventivo y de correctivo realizadas se obtiene un ingreso p_{pi} y p_{ci} respectivamente, los ingresos totales serán la suma de todos los mantenimientos preventivos y correctivos ejecutados:

$$I = \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (p_{pi}z_{pi} + p_{ci}z_{ci}) \quad (54)$$

3.2. Tiempo de trabajo

Los convenios colectivos provinciales de las empresas siderometalúrgicas son los que se aplican a los operarios que desempeñan las tareas de mantenimiento. En estos convenios se establece un número de horas máximas de trabajo diario y anual. El número máximo de horas que un operario puede trabajar diariamente suele ser de 8, pero no podemos establecer un valor para las horas anuales, puesto que varía en función de la provincia.

Definiremos los tiempos t_{ij} como los tiempos de desplazamiento de los operarios, desde un punto i hasta el punto j . Estos puntos o lugares son, tanto

la base, desde donde comienzan y finalizan su jornada laboral, como los lugares donde se encuentran los equipamientos o sistemas que han de mantener, es decir, los centros de trabajo de las empresas con patrimonio distribuido.

Así mismo se definen los tiempos de operación de mantenimiento, como los tiempos necesarios para ejecutar las tareas de mantenimiento de los equipos o sistemas, por lo que, distinguiremos para los mantenimientos preventivos un tiempo de operación t_{pi} y para los correctivos t_{ci} .

Por tanto, teniendo en cuenta que cada operario puede trabajar un máximo de 8 horas diarias, se debe cumplir:

$$\sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) \leq 8 \quad (55)$$

Hay que mencionar que los convenios colectivos admiten que los operarios realicen horas extras, aunque no se tendrá en cuenta para el presente estudio.

3.3. Costes de los equipos de mantenimiento

Son muchos los costes que se han de contemplar, como el absentismo del personal, ropa de trabajo y EPIs, herramientas, pérdidas de tiempo, etc., pero para no complicar en demasía el problema objeto de esta tesis, como criterio, se tendrán en cuenta los principales, que son los costes de mano de obra, los costes de alquiler de las furgonetas y los costes de gasoil. Si se quisiese realizar un estudio más cercano a la realidad en lo que a costes se refiere, no habría más que implementarlos.

Los costes de mano de obra se suelen calcular en base al salario, que suele estar definido en el convenio colectivo provincial, al que se aplican los seguros sociales a cargo de la empresa mantenedora. Si al importe que la empresa paga por cada operario se divide por el número de horas anuales que éste debe trabajar, se obtendrá C_{or} como el coste hora de cada operario r . Por lo que, para cada operario se definen sus costes como indica la ecuación (56).

$$C_o \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) \quad (56)$$

Del mismo modo, si se divide el alquiler, que la mantenedora asume por una furgoneta anualmente, entre el número de horas anuales que el operario

debe trabajar se obtendrá el coste hora de cada furgoneta r : C_{fr} . Debemos mencionar que es obvio que se pueda optimizar este coste, dado que estamos suponiendo que cada operario dispone de una furgoneta, pero éstas pueden trabajar a tres turnos. Además, una furgoneta no tiene por qué estar detenida mientras un operario está de vacaciones, fin de semana o festividad. No obstante, se supondrá, para el coste de la furgoneta el definido, C_{fr} , puesto que no influye en el presente estudio. En cualquier caso, se podría calcular el coste real y sustituirlo en la ecuación si quisiera acercarse en mayor grado a la realidad. Se asumirán unos costes para cada vehículo según la ecuación (57):

$$C_f \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) \quad (57)$$

Por último, debemos considerar los costes del gasoil necesario para el desplazamiento de los equipos de mantenimiento. Consideraremos el consumo medio de cada furgoneta para ciudad y carretera Q_m , que suelen facilitar los fabricantes. Así mismo, se supondrá que la velocidad media de las furgonetas para ciudad y carretera v_m será la resultante que haga coincidir los consumos teóricos para ciudad y para carretera, teniendo en cuenta las velocidades máximas en cada una de ellas. Si llamamos C_g al coste del litro de gasoil, el coste del combustible, en función del tiempo de desplazamiento C_c será:

$$C_c = Q_m v_m C_g \quad (58)$$

Por lo que los costes de combustible, de cada furgoneta, se definen en la ecuación (59):

$$C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \quad (59)$$

Si sumamos los costes de las ecuaciones (56), (57) y (59) obtendremos los costes totales por operario, según la ecuación (60):

$$C_m = (C_o + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \quad (60)$$

Estos costes se pueden aplicar para el caso de que el equipo de trabajo pueda derivarse a otros quehaceres, a los que se les repercutiría los costes en

base a la misma ecuación, de modo que complete la jornada laboral. Por ejemplo, si tras trabajar 5 horas en un contrato, fuese enviado a otro contrato de mantenimiento las 3 horas restantes de su jornada laboral.

Si sucediera que no es posible que otros contratos asuman horas sueltas, pero si días completos, deberemos hacernos cargo de los costes de las jornadas completas en las que se trabaje en nuestro contrato, según la ecuación (61):

$$C_m = 8(C_o + C_f) + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \quad (61)$$

Si consideramos que el equipo de trabajo está destinado exclusivamente para este servicio, es decir, que no podemos trasladarlo para otras tareas distintas al problema objeto del estudio, diariamente debemos asumir el coste de toda la jornada, tanto de la mano de obra como de la furgoneta. Por este motivo estos costes no varían, por lo que deben extraerse de la función objetivo. Por tanto, los costes variables, en función del tiempo, serán los referentes al combustible debido al desplazamiento, esto es:

$$C_m = C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \quad (62)$$

3.4. Costes de subcontratación

Si no fuese posible realizar las incidencias existentes en un momento dado, sin superarse los plazos máximos de resolución de las mismas, es factible que se subcontrate alguna de dichas tareas.

Los costes repercutidos por la subcontrata son los que hemos visto en el apartado 3.3, sobre los que se aplicará un margen bruto contra la venta, M_b .

Puesto que las mantenedoras solo admitirán a las subcontratas los costes reales, según la ecuación (60), éstas suelen aplicar márgenes altos para compensar las pérdidas de tiempo. Por tanto, los costes de subcontratación C_s , se calcularán por:

$$C_s = \frac{1}{1 - M_b} \left[(C_o + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij}t_{ij} \right] \quad (63)$$

3.5. Costes por penalizaciones

Hemos visto que para el contrato Tipo 2, explicado en el capítulo 1, el cliente puede aplicar penalizaciones por las tareas de mantenimiento no ejecutadas. Sea y_{pi} una variable que tomará el valor del número de tareas de mantenimiento preventivo pendientes en V_i , que no se ejecutarán en la ruta, cuyas fechas de finalización máximas sean menores o iguales que la del día en que se realice la ruta. Análogamente sea y_{ci} la variable que tomará el valor del número de tareas de mantenimiento correctivo pendientes en V_i , que no se ejecutarán en la ruta, cuyas fechas de finalización máximas sean menores o iguales que la del día en que se realice la ruta. Si consideramos unas penalizaciones por cada preventivo y correctivo no realizado, γ_{pi} y γ_{ci} , respectivamente, los costes por penalizaciones C_p , serán:

$$C_p = \sum_{i=1}^{n-1} (\gamma_{pi}y_{pi} + \gamma_{ci}y_{ci}) \quad (64)$$

3.6. Funciones objetivo y restricciones

Como se ha mencionado en el capítulo 1, se contemplan tres posibles tipos de contratos, por lo que estudiaremos la función objetivo de cada uno de ellos, contempladas para un solo vehículo.

En este apartado consideraremos los costes del equipo de mantenimiento según la ecuación (60), es decir solo asumimos los costes de las horas que trabaja el operario, aunque se podría aplicar tanto los que figuran en (61) como los de (62).

Recodemos que en el contrato Tipo 1 se factura por intervención de mantenimiento por lo que los ingresos los marca (54), los costes del equipo de mantenimiento vienen de (60) y los de subcontratación figuran en (63). En este tipo de contrato no se aplican penalizaciones.

Por lo que podríamos definir el problema del siguiente modo:

$$\begin{aligned}
Max \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (p_{pi}z_{pi} + p_{ci}z_{ci}) - (C_o & \\
+ C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) - C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} & \\
- \frac{1}{1 - M_b} \left[(C_o + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) \right. & \\
\left. + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \right] & \quad (65)
\end{aligned}$$

$$\sum_{j=2}^n x_{1j} = \sum_{i=1}^{n-1} x_{in} = 1; \forall i, j \in V \quad (66)$$

$$\sum_{i=1}^{n-1} x_{ik} = \sum_{j=2}^n x_{kj} \leq 1; \forall k = (2, \dots, n-1) \quad (67)$$

$$A_i \leq s_i; \forall i = (1, \dots, n) \quad (68)$$

$$s_i \leq B_i; \forall i = (1, \dots, n) \quad (69)$$

$$s_i + t_{ij} - s_j \leq L(1 - x_{ij}); \forall i, j = (1, \dots, n) \quad (70)$$

$$\sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) \leq 8 \quad (71)$$

$$x_{ij} \in \{0,1\}; \forall i, j \in V \quad (72)$$

La función objetivo (65), maximiza los beneficios, es decir la diferencia entre ingresos y costes totales. La restricción (66) obliga a que la ruta se inicie y finalice en la base del origen y la (67) asegura la conectividad de la ruta y que cada vértice solo sea visitado una vez. Se acota el inicio del servicio con (68) y (69), para que esté dentro de la ventana temporal. Con la restricción (70) nos aseguramos de que la ruta está encadenada si la constante L tiene un valor mayor o igual a la jornada laboral. Que la jornada laboral no exceda de 8 horas está limitada por (71). Finalmente, la ecuación (72) restringe los valores de la variable binaria.

En contrato tipo 2, en el que se factura por intervención de mantenimiento aplicándose penalizaciones por las tareas de mantenimiento no ejecutadas, los ingresos vienen dados por (54), los costes del equipo de mantenimiento se definieron en (60), los de subcontratación figuran en (63) y los costes por penalizaciones se vieron en (64):

$$\begin{aligned}
Max \quad & \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (p_{pi}z_{pi} + p_{ci}z_{ci}) - (C_o \\
& + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) \\
& - C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \\
& - \frac{1}{1 - M_b} \left[(C_o + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) \right. \\
& \left. + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \right] - \sum_{i=1}^{n-1} (\gamma_{pi}y_{pi} + \gamma_{ci}y_{ci})
\end{aligned} \quad (73)$$

Se aplican las restricciones de la (66) a la (72); **Error! No se encuentra el origen de la referencia.** a la funcione objetivo (73).

En el contrato tipo 3, se factura una cuota por el servicio contratado y se aplican penalizaciones por las tareas de mantenimiento no ejecutadas. Al ser constantes los ingresos deben salir de la función objetivo que debe minimizar los costes:

$$\begin{aligned}
\text{Min } (C_o + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \\
+ \frac{1}{1 - M_b} \left[(C_o + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi}z_{pi} + t_{ci}z_{ci}) \right. \\
\left. + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \right] + \sum_{i=1}^{n-1} (\gamma_{pi}y_{pi} + \gamma_{ci}y_{ci}) \quad (74)
\end{aligned}$$

A la función objetivo (74) también se le aplican las restricciones de la (66) a la (72).

Si fuese necesario definir las funciones objetivo y las restricciones para una zona con m vehículos no habría dificultad. Respecto a las funciones objetivo, no se tendría más que sumar los ingresos y costes de los m vehículos, permaneciendo constantes tanto los costes de subcontratación como de penalizaciones. Las restricciones tendrían que aplicarse a los m vehículos.

3.7. Definición del problema objeto del estudio

El contrato tipo 4, según se definió en el apartado 1, es aquel por el que se cobra una cuota fija sin que se apliquen penalizaciones por las incidencias no realizadas.

Siguiendo los razonamientos del apartado anterior, la función objetivo consistiría en minimizar nuestros costes.

Se dispone de una base de datos de las incidencias reales de un año de una empresa con patrimonio distribuido, cuyos centros están ubicados en la provincia de Lugo.

Se definirá el tipo de contrato 4 y se intentará optimizar el caso real que a continuación se describe.



Figura 1. Distribución geográfica de los centros

El alcance del contrato consiste en el mantenimiento de 34 centros repartidos por la provincia de Lugo. Los tiempos de respuesta establecidos son de 3 horas para avisos urgentes, 24 horas para los que tengan prioridad media y 72 horas para los de prioridad baja. Además, cada uno de los centros deberá ser visitado cada dos meses para realizar los mantenimientos preventivos. Es decir, tenemos demandas estocásticas para el caso de mantenimientos correctivos y deterministas para el caso de preventivos, con ventanas temporales.

Como criterio para el desarrollo de este estudio se tomarán constantes los tiempos de operación, estableciéndose la necesidad de invertir 2 horas para cada mantenimiento preventivo y 1 hora para los correctivos.

La matriz de tiempos de desplazamiento es asimétrica. Se dispone de un solo equipo de trabajo para atender el contrato.

Se plantea el servicio de modo que se procederá a subcontratar las tareas que no puedan ejecutarse con los medios propios. No se ha tenido en cuenta la restricción de las 8 horas por jornada de la subcontrata, porque no influye en la optimización de los recursos propios. En caso de querer considerar este factor, no habría más que incluirla y calcular las rutas óptimas que serán subcontratadas.

En el caso real que vamos a estudiar, por las circunstancias de la mantenedora, no es posible que otros contratos puedan asumir horas sueltas, pero sí días completos, por lo que los costes del equipo de mantenimiento vienen dados por (61). Los costes de subcontratación se definieron en la ecuación (63).

Sumando ambas ecuaciones se obtendría la función objetivo que se ha de minimizar:

$$\begin{aligned} \text{Min } C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \\ + \frac{1}{1 - M_b} \left[(C_o + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi} z_{pi} + t_{ci} z_{ci}) \right. \\ \left. + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \right] \end{aligned} \quad (75)$$

La función objetivo (75) estará sujeta a las restricciones (66) a (72).

El contrato tipo 4 se diferencia del PCTSP en que no tiene que generar un mínimo de ganancias puesto que los ingresos son constantes.

Minimizar la función objetivo, matemáticamente planteada, no tiene mayor dificultad. Si los tiempos de desplazamiento y los de operación fueran nulos la función objetivo resultante sería 0.

Este tipo de contratos suele tener una cláusula de rescisión por incumplimiento de las incidencias, de esta manera las mantenedoras están obligadas a realizar los mantenimientos. Se propone añadir la siguiente restricción:

$$\sum_{j=2}^n x_{ij} = 1; \forall z_{pi} \geq 1: s_i = B_i \text{ y } \forall z_{ci} \geq 1: s_i = B_i; \forall i = (1, \dots, n-1) \quad (76)$$

Con la restricción (76) conseguimos que sea visitado todo vértice V_i en el que exista alguna incidencia, de preventivo o correctivo, con fecha de finalización, según su ventana temporal, igual al día en el que se está recorriendo la ruta.

Con esta restricción, solo se irían resolviendo las incidencias cuyo plazo de resolución vaya a finalizar, dejando las incidencias para el último día, por lo que se tendrá que subcontratar los mantenimientos preventivos, dado que tienen la misma ventana temporal, con el incremento de coste que conlleva.

Podría añadirse alguna restricción que obligue a que se realicen las incidencias o se podría distribuir homogéneamente los mantenimientos preventivos a lo largo del bimestre, pero estos criterios limitarían la optimización del problema.

Los métodos utilizados para la resolución se exponen en el capítulo 4.

Si el recurso tuviese que salir para atender una ruta, cuyo tiempo total no se aproximase a 8 horas, se plantea la duda de si resultaría más rentable subcontratar dicha ruta.

Para la toma de la decisión de la conveniencia de la subcontratación del servicio o realizarla por medios propios, se plantea la siguiente ecuación que compara los costes propios, que se obtienen en (61) y los costes de subcontratación, que vienen de (63):

$$\begin{aligned}
8(C_o + C_f) + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \\
\geq \frac{1}{1 - M_b} \left[(C_o + C_f) \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi} z_{pi} + t_{ci} z_{ci}) \right. \\
\left. + C_c \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \right] \quad (77)
\end{aligned}$$

Por lo que, despejando el tiempo total de la ruta, nos queda una función que depende de los tiempos de desplazamiento:

$$\begin{aligned}
8(1 - M_b) + M_b \frac{C_c}{C_o + C_f} \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \\
\geq \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi} z_{pi} + t_{ci} z_{ci}) \quad (78)
\end{aligned}$$

Para poder acotar el tiempo, supongamos, en primer lugar, que no hubiese desplazamiento porque en el centro base hay incidencias suficientes para ocupar toda la jornada laboral:

$$8(1 - M_b) \geq \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{pi} z_{pi} + t_{ci} z_{ci}) \quad (79)$$

Supongamos en segundo lugar que debemos desplazarnos a un irreal centro alejado, al menos, 4 horas de viaje. En tal caso los tiempos de operación serían nulos:

$$\frac{8(1 - M_b)(C_o + C_f)}{C_o + C_f + C_c M_b} \geq \sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} t_{ij} \quad (80)$$

Asumiendo los importes que figuran en el convenio colectivo de aplicación para los trabajadores del metal de la provincia de Lugo para el año 2019 (Lugo, 2020), sin tener en cuenta los seguros sociales, para un oficial de tercera:

$$C_o = 9,78 \text{ €/h} \quad (81)$$

Si, además, suponemos unos costes de combustible y de furgoneta:

$$\begin{aligned} C_f &= 1,42 \text{ €/h} \\ C_c &= 2,73 \text{ €/h} \end{aligned} \quad (82)$$

Operando sobre (79) y (80), suponiendo un margen bruto de la subcontrata del 40%, obtendremos que los tiempos totales de trabajo, que determinan la decisión de la subcontratación, se encuentran entre 4,8 y 4,4 horas. Supondremos inicialmente que, una ruta es más rentable realizarla con los medios propios que con los subcontratados si el tiempo total de trabajo es mayor o igual a 4,5 horas.

Si se dispusiese de un histórico de tiempos de desplazamiento o se utilizaran los datos que se obtengan de los experimentos, se podría aproximar este tiempo mínimo.

Por todo ello, se decide no registrar ninguna ruta a realizar por los medios propios por debajo de 4,5 horas. Efectivamente, si hubiese incidencias cuyo plazo de resolución caducasen el día para el que se ha calculado la ruta, dichas tareas se subcontratarán, en caso contrario no se ejecutarán, a la espera de que llegue el último día para su resolución o que aparezcan nuevas incidencias que permitan que la ruta supere las 4,5 horas, por lo que se propone la siguiente restricción:

$$\sum_{i=1}^{n-1} \sum_{j=2}^n x_{ij} (t_{ij} + t_{pi} z_{pi} + t_{ci} z_{ci}) \geq 4,5 \quad (83)$$

Cuando sea el caso de no poder resolver las incidencias con los recursos propios, como hemos comentado, se externalizará la subruta de coste mínimo

que nos permita cumplir los compromisos. Así mismo, se entregará a la subcontrata el orden de dicha subruta, que minimice los desplazamientos.

4. MÉTODOS PROPUESTOS

Se pretende resolver el problema día a día, de modo que se generen las rutas óptimas que se deberán ir ejecutando.

Se ha decidido utilizar un software disponible por la mayor parte de empresas, que tiene un coste más que razonable, para que no resulte una limitación para las empresas mantenedoras. El programa elegido es MS Excel.

Los métodos que se presentan tienen, en primer lugar, que resolver las incidencias con fecha de caducidad igual a la del día en que se está ejecutando el algoritmo, intentando mejorar las rutas con las restantes tareas.

En primer lugar, se describirá un algoritmo que se ha diseñado partiendo de la base del algoritmo del vecino más cercano, que será utilizado como algoritmo de comparación. Se ha desarrollado sobre la aplicación MS Excel utilizando el lenguaje **VBA** (**V**isual **B**asic for **A**pplications).

Se presenta un segundo algoritmo que, mediante el complemento Solver de Excel, tras analizar si es necesario subcontratar alguna de las incidencias, definiendo cuáles y en qué orden deberán de resolverse, posteriormente intentará mejorar la ruta a realizar con medios propios, añadiendo tareas con fecha de caducidad superior a la del día en la que se ejecuta el algoritmo.

Con este planteamiento ejecutaríamos una primera etapa, en la que solo se tendrán en cuenta las incidencias cuyo plazo de finalización esté en un horizonte temporal de 4 días. Se analizará si es posible realizarlas con los medios propios o será necesario subcontratar, en tal caso la ruta subcontratada será de coste mínimo. A cada una de las rutas, en la segunda etapa, se intentará ampliar la jornada laboral incluyendo mantenimientos preventivos, con fecha límite de resolución fuera del horizonte temporal, de modo que, si hubiese que asumir el coste completo del operario, como queda reflejado en la ecuación (61), que éste realice el mayor número de ordenes de trabajo posible. Si se diese el caso alguna ruta no superase las 4,5 horas no se realizaría con medios propios, si fuese posible se aguardaría a que apareciesen nuevas incidencias para completar la ruta, en caso contrario se subcontrataría.

Respecto a la función objetivo de la segunda etapa, aunque intuitivamente se puede definir como maximizar la ejecución de las incidencias, minimizando el coste de desplazamiento, se plantean varias opciones, como se explicará en el capítulo 5.

Como mejora al segundo algoritmo, también utilizando Solver, se presenta un tercero en el que se han implementado dos fases, en las que se pretende

reordenar la ruta a realizar por medios propios, de modo que sea posible incrementar mayor número de tareas a añadir en las rutas, para mejorarlas.

Finalmente, se ha querido estudiar como influiría en el resultado del problema si se dispusiese del conocimiento de las tareas que serán creadas a futuro. Aunque aún no existe un software capaz de predecir con suficiente fiabilidad las averías que serán generadas, no se ha querido perder la oportunidad de estudiar en el cuarto algoritmo si, con esta nueva información, es posible mejorar el resultado del estudio.

4.1. Algoritmo basado en vecino más cercano

Se ha programado en la aplicación Excel un algoritmo para resolver el problema, que optimizará las rutas utilizando el algoritmo del vecino más cercano.

Se ha planteado de modo que en primer lugar se intentará resolver todas las incidencias cuyo plazo de atención sea desde 3 horas hasta de un máximo de un día.

El algoritmo consta de las siguientes fases:

- Resolución: donde se encuentra la macro del libro donde se inicia el algoritmo, que finalizará cuando se hayan recorrido todas las fechas de la base de datos.
- Vecino más cercano: que definirá las fechas del horizonte temporal dentro de los días laborables: hoy, H y mañana, M. Buscará las incidencias que caducarán dentro de esta horquilla de fechas, es decir con fecha de resolución $F_r \leq M$, para crear una ruta que, partiendo de la base, visite todos los centros con incidencias a ejecutar, utilizando el algoritmo del vecino más cercano. También buscará todas las restantes órdenes de trabajo que caducarán fuera de la horquilla temporal.
- Externalizar: Si la ruta resultante de la fase anterior superase las 8 horas de trabajo, paulatinamente eliminará de la lista de incidencias las que se encontrasen en los centros más alejado a la base, hasta que se cumpliera con dicha restricción, registrando las órdenes de trabajo a externalizar.
- Centros visitados: se intentará incluir en la ruta propuesta las incidencias existentes en la segunda lista, contemplando solamente aquellas que pertenezcan a centros que se visiten en la ruta.

- Centros mínimo desplazamiento: que pretende que se añadan a la ruta ordenes de trabajo de la segunda lista, teniendo en cuenta solamente aquellas que existan en centros que se encuentren ubicados muy cercanamente a la trayectoria de la ruta.
- Centros cercanos: se intentará añadir órdenes de trabajo pertenecientes a la segunda lista, teniendo en cuenta solamente aquellas que existan en centros que se encuentren ubicados cercanamente a la trayectoria de la ruta.
- Registro: se registrarán las incidencias pertenecientes a la ruta si solo si el tiempo total de ejecución supere el mínimo indicado.

El libro de Excel cuenta con las siguientes hojas:

- TIEMPOS: donde se encuentran la matriz de tiempos de desplazamiento entre los distintos centros de trabajo.

	L01	L02	L03	L04	L05	L06	L07	L08	L09	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20	L21	L22	L23	L24	L25	L26	L27	L28	L29	L30	L31	L32	L33	L34
L01	000	006	003	005	005	009	033	034	041	055	057	110	118	122	123	128	118	113	049	039	059	055	032	023	033	041	050	052	104	059	058	054	042	032
L02	005	000	001	003	003	011	033	034	042	057	100	111	120	123	125	129	118	114	051	041	102	058	034	025	034	038	052	054	106	101	100	052	040	030
L03	004	009	000	002	006	009	034	035	045	059	101	113	122	125	127	131	119	117	052	042	104	057	034	023	033	037	051	053	105	100	058	051	039	029
L04	004	009	004	000	004	008	032	033	041	055	100	110	118	122	123	128	117	113	051	041	103	055	032	021	031	035	049	051	103	058	057	049	037	027
L05	007	006	002	005	000	011	030	031	040	054	058	108	117	120	122	126	115	112	049	039	100	057	033	024	034	039	051	053	105	100	059	053	041	031
L06	012	010	007	008	009	000	035	036	043	058	100	112	121	124	126	130	120	115	051	041	102	053	030	020	029	037	047	049	101	056	055	052	041	030
L07	035	034	031	032	030	034	000	022	030	044	046	058	107	110	112	116	105	101	103	048	116	112	048	039	049	057	107	109	121	116	114	114	102	051
L08	037	036	032	033	031	036	022	000	015	028	031	043	052	055	057	101	049	046	048	029	116	113	050	041	050	058	108	110	122	117	116	115	103	051
L09	045	044	041	042	040	043	030	015	000	015	018	031	039	043	044	049	056	034	035	022	111	121	057	048	058	106	116	118	130	125	123	122	110	059
L10	100	059	055	056	054	058	044	029	016	000	013	025	028	037	039	043	056	028	037	032	124	137	112	103	113	120	130	132	144	139	138	137	125	114
L11	102	101	057	100	058	100	045	031	017	012	000	018	023	030	031	036	048	021	025	033	115	128	106	105	115	122	132	134	146	141	140	139	127	116
L12	113	112	108	109	107	111	057	042	029	025	016	000	021	019	021	025	038	022	042	039	124	144	123	116	126	133	143	145	157	152	151	150	138	127
L13	118	117	114	115	113	116	103	048	034	023	021	019	000	019	021	025	038	032	047	044	134	149	128	121	131	139	149	151	203	158	156	155	143	132
L14	130	129	126	127	125	128	115	100	047	043	033	023	024	000	010	014	027	040	059	057	142	202	140	134	143	151	201	203	215	210	209	207	155	144
L15	127	126	123	124	122	125	112	057	044	040	030	020	021	012	000	007	019	037	056	054	139	159	137	130	140	148	158	200	212	207	206	204	152	141
L16	133	132	129	129	128	131	117	103	049	045	036	026	027	017	007	000	019	043	102	059	145	205	143	136	146	154	204	206	218	213	211	210	158	147
L17	145	144	141	141	140	143	129	115	101	057	048	038	039	029	019	019	000	055	114	111	157	216	155	148	158	206	216	217	229	224	223	222	210	159
L18	118	117	113	114	112	116	102	047	034	029	023	025	038	037	038	043	055	000	031	044	108	146	125	121	130	138	148	150	202	157	156	155	143	131
L19	052	051	048	050	048	050	103	047	036	037	024	043	048	055	056	101	113	031	000	026	101	114	053	056	105	113	123	125	137	132	131	130	118	106
L20	043	042	038	041	039	041	046	028	021	032	023	040	046	052	053	058	110	043	026	000	056	109	047	046	055	103	113	135	127	122	121	120	108	056
L21	103	102	058	101	100	100	116	115	109	124	114	128	138	140	142	146	159	110	100	055	000	051	045	059	111	119	129	131	143	138	137	139	127	116
L22	100	058	055	056	057	052	112	113	122	137	127	144	150	156	157	202	157	145	114	108	051	000	033	050	103	110	120	122	134	129	128	130	118	107
L23	036	025	032	033	034	029	049	050	059	113	105	122	129	134	136	140	134	124	053	047	045	032	000	027	036	048	058	100	112	107	106	107	055	044
L24	026	025	022	024	018	040	041	049	104	106	118	127	130	132	136	125	121	057	047	059	050	026	000	014	023	033	035	047	042	041	051	045	034	
L25	039	032	030	028	031	025	047	048	056	110	113	125	133	137	138	143	132	128	104	054	109	059	037	011	000	015	027	029	041	036	035	044	047	041
L26	039	038	039	035	039	036	058	059	107	122	124	136	145	148	150	154	143	139	115	105	120	110	047	023	018	000	038	040	052	047	046	044	032	037
L27	052	051	048	047	050	044	105	106	115	129	132	144	152	156	157	202	150	147	123	113	127	117	054	031	025	034	000	013	021	020	019	035	043	100
L28	055	054	052	050	053	047	108	109	118	132	135	147	155	159	200	205	153	150	126	116	130	121	057	034	028	037	013	000	026	014	014	027	035	053
L29	106	105	103	102	105	058	120	121	130	144	146	158	207	210	212	216	205	201	137	128	142	132	108	045	040	049	020	026	000	032	032	048	055	114
L30	102	101	058	057	100	054	115	116	125	139	142	154	202	206	207	212	200	157	133	123	137	127	104	041	035	044	020	013	032	000	010	030	038	056
L31	101	100	058	056	059	053	115	116	124	138	141	153	202	205	207	211	200	156	132	122	136	127	103	040	035	044	020	014	033	009	000	025	033	051
L32	053	052	052	048	052	051	112	113	122	136	138	150	159	202	204	208	157	153	129	120	137	128	104	051	045	043	036	028	049	031	026	000	014	032
L33	041	040	041	037	041	039	100	101	110	124	127	139	147	151	152	157	145	142	118	108	125	116	053	043	047	031	044	036	057	039	034	015	000	020
L34	031	030	031	027	031	028	049	051	100	114	117	129	137	141	142	147	137	132	108	058	115	106	043	035	045	037	102	054	115	056	051	032	020	000

Figura 2. Matriz de tiempos de desplazamiento

- CALENDARIO: figuran todos los días, desde la fecha de inicio hasta la fecha de fin de la base de datos. Estas fechas se han clasificado en función de si son laborables o no. En esta hoja se registrará las horas totales de las rutas propuestas, así como los tiempos de desplazamiento.

DIA	LABORABLE	T TOTAL	T DESPLAZA	TT EXTER	T DESP EXTER
01/01/2009	N				
02/01/2009	S	6,206	0,206		
03/01/2009	N				
04/01/2009	N				
05/01/2009	S	7,517	0,517		
06/01/2009	N				
07/01/2009	S				
08/01/2009	S	7,560	1,560		
09/01/2009	S	7,661	1,661		
10/01/2009	N				
11/01/2009	N				
12/01/2009	S				
13/01/2009	S	7,942	1,942		
14/01/2009	S	5,326	1,326		

Figura 3. Detalle del calendario

- LUGO INCIDENCIAS: en esta hoja se encuentran todas las órdenes de trabajo, numeradas, indicándose su prioridad y su centro de trabajo. Para cada una de ellas se calcula la fecha límite para su resolución. Según se vaya ejecutando el algoritmo propuesto, se registrará la fecha en la que se propone su ejecución.

Nº OT	FECHA	CENTRO	PROVINCIA	PRIORIDAD	FECHA LIM RES	FECHA EJECUCIÓN
22	01-ene-09	L22	LUGO	0. Preventivo	02-mar-09	
23	01-ene-09	L23	LUGO	0. Preventivo	02-mar-09	14-ene-09
24	01-ene-09	L24	LUGO	0. Preventivo	02-mar-09	08-ene-09
25	01-ene-09	L25	LUGO	0. Preventivo	02-mar-09	08-ene-09
26	01-ene-09	L26	LUGO	0. Preventivo	02-mar-09	08-ene-09
27	01-ene-09	L27	LUGO	0. Preventivo	02-mar-09	
28	01-ene-09	L28	LUGO	0. Preventivo	02-mar-09	
29	01-ene-09	L29	LUGO	0. Preventivo	02-mar-09	
30	01-ene-09	L30	LUGO	0. Preventivo	02-mar-09	
31	01-ene-09	L31	LUGO	0. Preventivo	02-mar-09	
32	01-ene-09	L32	LUGO	0. Preventivo	02-mar-09	
33	01-ene-09	L33	LUGO	0. Preventivo	02-mar-09	09-ene-09
34	01-ene-09	L34	LUGO	0. Preventivo	02-mar-09	09-ene-09
35	05-ene-09	L01	LUGO	1. Baja	09-ene-09	05-ene-09
36	07-ene-09	L02	LUGO	1. Baja	12-ene-09	09-ene-09
37	09-ene-09	L03	LUGO	3. Alta	09-ene-09	09-ene-09
38	13-ene-09	L04	LUGO	1. Baja	16-ene-09	14-ene-09
39	14-ene-09	L05	LUGO	3. Alta	14-ene-09	14-ene-09

Figura 4. Detalle de la tabla de órdenes de trabajo a ejecutar

- ESTUDIO: donde se buscarán las listas de incidencias.
- RUTA: donde se realizarán las rutas.

HOY	31/12/2009	365	MAÑANA	04/01/2010	369	
INCIDENCIAS PENDIENTES						
Nº OT	FECHA	CENTRO	PROVINCIA	PRIORIDAD	FECHA LIMITE R	URGENTE
10	01/01/2009	L10	LUGO	0. Preventivo	02/03/2009	NO
12	01/01/2009	L12	LUGO	0. Preventivo	02/03/2009	NO
13	01/01/2009	L13	LUGO	0. Preventivo	02/03/2009	NO
14	01/01/2009	L14	LUGO	0. Preventivo	02/03/2009	NO
15	01/01/2009	L15	LUGO	0. Preventivo	02/03/2009	NO
16	01/01/2009	L16	LUGO	0. Preventivo	02/03/2009	NO
17	01/01/2009	L17	LUGO	0. Preventivo	02/03/2009	NO
18	01/01/2009	L18	LUGO	0. Preventivo	02/03/2009	NO
19	01/01/2009	L19	LUGO	0. Preventivo	02/03/2009	NO
21	01/01/2009	L21	LUGO	0. Preventivo	02/03/2009	NO
22	01/01/2009	L22	LUGO	0. Preventivo	02/03/2009	NO
27	01/01/2009	L27	LUGO	0. Preventivo	02/03/2009	NO
28	01/01/2009	L28	LUGO	0. Preventivo	02/03/2009	NO
29	01/01/2009	L29	LUGO	0. Preventivo	02/03/2009	NO
30	01/01/2009	L30	LUGO	0. Preventivo	02/03/2009	NO
31	01/01/2009	L31	LUGO	0. Preventivo	02/03/2009	NO
32	01/01/2009	L32	LUGO	0. Preventivo	02/03/2009	NO
41	19/01/2009	L07	LUGO	1. Baja	22/01/2009	NO
42	19/01/2009	L03	LUGO	2. Media	20/01/2009	SI
43	19/01/2009	L06	LUGO	3. Alta	19/01/2009	SI

Figura 5. Detalle de la hoja Estudio

Así mismo el libro de Excel cuenta con tantos módulos como fases tiene el algoritmo, llamados del mismo modo que éstas, donde se albergan las macros que conforman el algoritmo que presentamos.

Se procede a explicar con detalle cada una de las fases:

A) Fase Resolución

La fase resolución es la que inicia y finaliza el algoritmo, contando con las siguientes etapas:

- A.1. Se inicia el algoritmo cuya fecha de hoy es un día anterior a la de inicio de la base de datos.
- A.2. Se indicará el valor $h = 4.5$ horas, el tiempo de una ruta por debajo del cual no será registrado, como hemos visto en el apartado 3.7.
- A.3. Se define la fecha final F_f , que coincide con la última fecha de la base de datos.
- A.4. En caso de que $F_f \geq M$, la fecha que figura para mañana pasará a la fase Estudio.
- A.5. Si $F_f < M$ se finalizará el algoritmo.

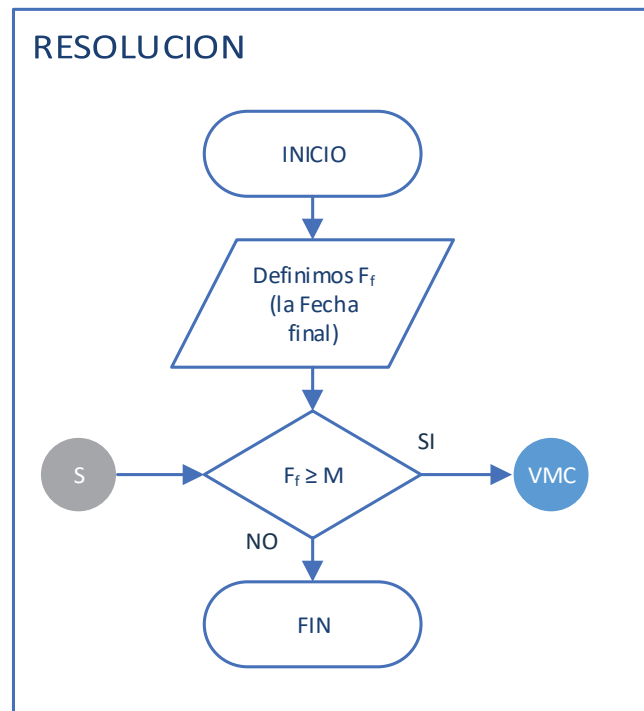


Figura 6. Diagrama de flujo de la fase Resolución del algoritmo basado en k-NN

B) Fase Vecino más cercano

Durante este proceso se avanzará en el calendario, se buscarán las incidencias susceptibles de ejecución y se generará una primera ruta R . Sus etapas son como sigue:

- B.1. Se pasará de día, es decir hoy, H , tomará el valor de mañana, M y en la hoja CALENDARIO se buscarán el valor del siguiente día laborable, para definir M .
- B.2. En la hoja ESTUDIO, se realizará una lista, buscando en la hoja LUGOINCIDENCIAS, con aquellas incidencias con fecha de creación $F_c \leq H$ y con fecha de resolución $F_r \leq M$, a la que denominaremos L_1 , listado de órdenes de trabajo a ejecutar.

FECHA		POS CALEND			1 B. Inc.	2 Inc a H Ruta	
HOY	19/01/2009	19	Cambio día		RESETEAR		
MAÑANA	20/01/2009	20	Día a Día		Año		
6 Externa. 7 C. Visitado							
INCIDENCIAS PENDIENTES							
Nº OT	FECHA	CENTRO	PROVINCIA	PRIORIDAD	FECHA LIMITE R	DISTAN A L01	POS INC
42	19/01/2009	L03	LUGO	2. Media	20/01/2009	0,002997685	42
43	19/01/2009	L06	LUGO	3. Alta	19/01/2009	0,008344907	43
27	01/01/2009	L27	LUGO	0. Preventivo	02/03/2009	0,036238426	27

Figura 7. Detalle de la lista de incidencias L_1

B.3. También se buscarán en LUGOINCIDENCIAS las órdenes de trabajo con $F_c \leq H$ y $F_r > M$, que serán las órdenes de trabajo que no haya que realizar dentro de nuestro horizonte temporal. A ésta la llamaremos L_2 , listado de otras órdenes de trabajo, que figurará en la hoja ESTUDIO.

INCIDENCIAS OTRAS TAREAS							C	MIN	DESP	D	ULT RUTA SI 2	H	D	ULT RUTA SI 1	H
Nº OT	FECHA	CENTRO	PROVINCIA	PRIORIDAD	FECHA LIMITE R	C	MIN	DESP	D	ULT RUTA SI 2	H	D	ULT RUTA SI 1	H	
10	01/01/2009	L10	LUGO	0. Preventivo	02/03/2009					0,06307					
12	01/01/2009	L12	LUGO	0. Preventivo	02/03/2009					0,07208					
13	01/01/2009	L13	LUGO	0. Preventivo	02/03/2009					0,07588					
14	01/01/2009	L14	LUGO	0. Preventivo	02/03/2009					0,08428					
15	01/01/2009	L15	LUGO	0. Preventivo	02/03/2009					0,08218					
16	01/01/2009	L16	LUGO	0. Preventivo	02/03/2009					0,08620					
17	01/01/2009	L17	LUGO	0. Preventivo	02/03/2009					0,09449					
18	01/01/2009	L18	LUGO	0. Preventivo	02/03/2009					0,07549					
19	01/01/2009	L19	LUGO	0. Preventivo	02/03/2009					0,05800					
21	01/01/2009	L21	LUGO	0. Preventivo	02/03/2009					0,06209					
22	01/01/2009	L22	LUGO	0. Preventivo	02/03/2009					0,05616					

Figura 8. Detalle de la lista de tareas L_2

B.4. Se llevará la lista L_1 a la hoja RUTA. Se creará una tabla que será la herramienta para el cálculo de la ruta, que tendrá, en primer lugar, una fila en la que figurará el centro base, desde donde parte la ruta, posteriormente tantas filas como figuren en la lista L_1 , donde se irán ordenando las incidencias en función del tiempo de desplazamiento, entre el centro de trabajo de la línea previa y el centro de trabajo de cada incidencia, finalmente, habrá una última línea que también estará asignada a la base. Tanto la primera línea como la última no tienen asociados tiempos de operación, solo tienen como finalidad que la ruta se inicie y finalice en la base. La tabla, mediante fórmulas, propondrá una ruta R , que partiendo de la base ira visitando aquellos centros con incidencias de la lista L_1 , utilizando el algoritmo del vecino más cercano, k-NN. Así

mismo se calculará el tiempo total y el tiempo de desplazamiento de la ruta.

3 Otras tareas	4 Tabla	5 K-NN					T TOTAL	5,881
8 C min Desp.	9 C Cercano	10 Regist					T DESPLAZA	1,881

	CENTROS ACTIVOS	ORDEN RUTA	Nº OT	ORIGEN	DESTINO	DISTANCIA	DESPLAZAM	TIEMPO OP
	L01	0	11	L01	L01	-		
1	L03	1	42	L01	L03	0,05777778	1	
2	L06	2	43	L03	L06	0,16222222	1	
3	L27	3	27	L06	L27	0,79111111	2	
	L01		F1	L27	L01	0,86972222		

Figura 9. Detalle de la tabla que calculará la ruta

- B.5. Finalmente se dará orden para que se ejecuten las macros del módulo Externalizar.

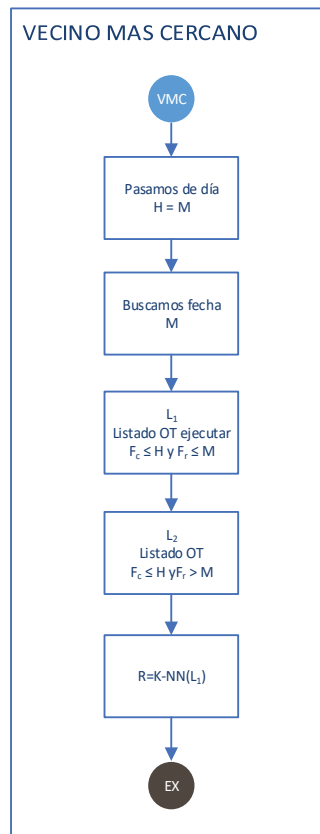


Figura 10. Diagrama de flujo de la fase Vecino más cercano del algoritmo basado en k-NN

C) Fase Externalizar

Si la ruta R , superase las 8 horas de jornada laboral, se eliminarán incidencias de la ruta hasta que ésta no supere las mencionadas 8 horas. Los pasos a seguir en esta fase son:

- C.1. Si la ruta excede las 8 horas, $T > 8$, buscaremos en L_1 la primera incidencia del centro más alejado de la base y la eliminaremos de L_1 . Se creará una tercera lista de incidencias L_3 , donde se irán incluyendo los órdenes de trabajo eliminados de L_1 . Se pasará el algoritmo k-NN sobre los centros de L_1 , obteniéndose una nueva ruta R .
- C.2. Volveremos sobre el apartado anterior hasta que el tiempo de la ruta no supere las 8 horas.
- C.3. Se pasará el algoritmo k-NN sobre los centros de L_3 para generar la ruta a externalizarse R_e .
- C.4. Registraremos R_e como ruta a externalizada y pasamos a la fase Centros visitados.

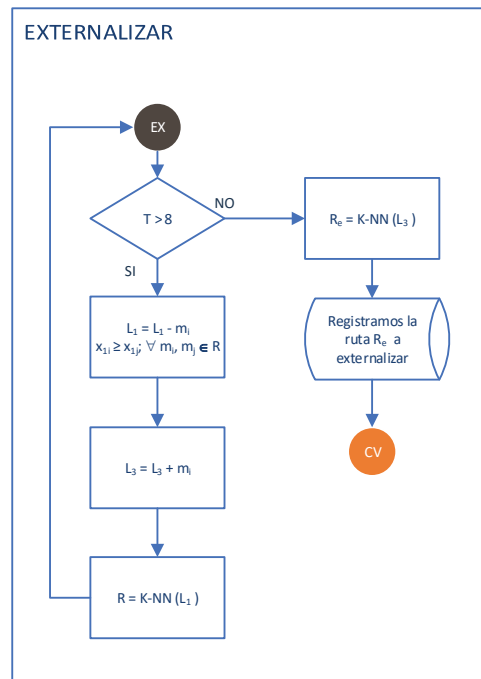


Figura 11. Diagrama de flujo de la fase Externalizar del algoritmo basado en k-NN

D) Fase Centros visitados

Es posible que, sin exceder la jornada laboral de 8 horas, se pueda incluir en la ruta R , que viene de la fase anterior, alguna incidencia perteneciente a L_2 , pero solo aquellas de centros que se visiten en la ruta R .

- D.1. Si el tiempo de la ruta supera las 7 horas de trabajo, $T > 7$, no se podrá incluir ninguna incidencia puesto que las de menor tiempo de operación, las de correctivo, son de 1 hora, mientras que las de preventivo son de 2 horas, por lo que se pasaría a la fase Registro.
- D.2. En el caso que no se superasen las 6 horas, se buscaría en L_2 las incidencias preventivas de centros que se visiten en R . Se irán incluyendo en L_1 una a una estas incidencias de preventivo m_p , siguiendo el orden de la ruta, es decir, primero las del primer centro a visitar, si existiesen órdenes de trabajo, posteriormente las del segundo centro a visitar y sucesivamente hasta el último centro de la ruta. A este proceso lo llamaremos CV . Cada vez que se incluya una incidencia en L_1 , se le pasará el algoritmo k-NN para obtenerse una nueva ruta R' . Si R' no supera las 8 horas de trabajo, $T' < 8$, admitimos la nueva ruta, $R = R'$, para volver al punto anterior, en caso contrario se mantiene la ruta R , en la que no se ha incluido ninguna nueva incidencia, para regresar al punto D.1. Para el caso de no existir ninguna incidencia de preventivo, con las condiciones descritas, se pasaría al punto siguiente.
- D.3. En el caso que se superasen las 6 horas o no hubiese ninguna incidencia a incluir en el punto anterior, se buscaría en L_2 las incidencias correctivas de centros que se visiten en R . Se irán incluyendo en L_1 una a una estas incidencias de correctivo m_c , siguiendo el orden de la ruta, es decir, primero las del primer centro a visitar, si existiesen órdenes de trabajo, posteriormente las del segundo centro a visitar y sucesivamente hasta el último centro de la ruta. A este proceso lo llamaremos CV . Cada vez que se incluya una incidencia en L_1 , se le pasará el algoritmo k-NN para obtenerse una nueva ruta R' . Si R' no supera las 8 horas de trabajo, $T' < 8$, admitimos la nueva ruta, $R = R'$, para volver al punto D.1, en caso contrario se mantiene la ruta R , en la que no se ha incluido ninguna nueva incidencia, para ejecutar la fase Registro. Para el caso de no existir ninguna incidencia de correctivo se pasaría a la fase Centros mínimo desplazamiento.

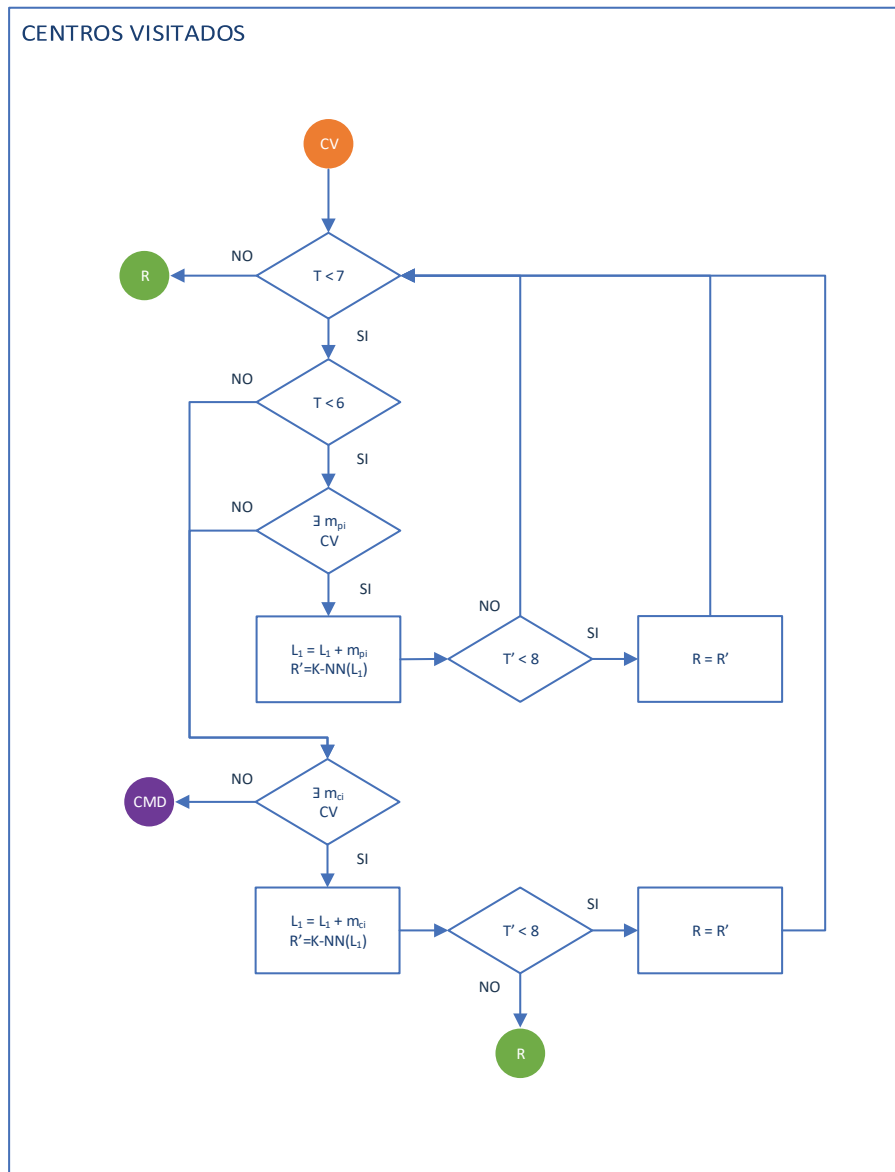


Figura 12. Diagrama de flujo de la fase Centro visitado del algoritmo basado en k-NN

E) Fase Centros mínimo desplazamiento

Es posible que, sin exceder la jornada laboral de 8 horas, se pueda incluir en la ruta R , que viene de la fase anterior, alguna incidencia perteneciente a L_2 , pero solo aquellas de centros muy próximos a la ruta R .

- E.1. Si el tiempo de la ruta supera las 7 horas de trabajo, $T > 7$, no se podrá incluir ninguna incidencia puesto que las de menor tiempo de operación, las de correctivo, son de 1 hora, mientras que las de preventivo son de 2 horas, por lo que se pasaría a la fase Registro.
- E.2. En el caso que no se superasen las 6 horas, se buscaría en L_2 las incidencias preventivas de centros que cumplan:

$$t_{1i} \leq t_{1j} \text{ y } t_{ij} \leq t_{1j}; \forall V_i \notin R; \forall V_j \in R \quad (84)$$

De entre las incidencias cuyos centros cumplan la ecuación (84), se irán incluyendo en L_1 una a una estas incidencias de preventivo m_p , comenzando por las de menor tiempo de desplazamiento t_{ij} siguiendo el orden inverso de la ruta, es decir, primero las del último centro a visitar, si existiesen órdenes de trabajo, posteriormente las del penúltimo centro a visitar y sucesivamente hasta la base. A este proceso lo llamaremos *CMD*, se le pasará el algoritmo k-NN para obtenerse una nueva ruta R' . Si R' no supera las 8 horas de trabajo, $T' < 8$, admitimos la nueva ruta, $R = R'$, para volver al punto anterior, en caso contrario se mantiene la ruta R , en la que no se incluye ninguna nueva incidencia, para regresar al punto E.1. Para el caso de no existir ninguna incidencia de preventivo, con las condiciones descritas, se pasaría al punto siguiente.

- E.3. En el caso que se superasen las 6 horas o no hubiese ninguna incidencia a incluir en el punto anterior, se buscaría en L_2 las incidencias de correctivo cuyos centros cumplan la ecuación (84). Se irán incluyendo en L_1 una a una estas incidencias de correctivo m_c , comenzando por las de menor tiempo de desplazamiento t_{ij} siguiendo el orden inverso de la ruta, es decir, primero las del último centro a visitar, si existiesen órdenes de trabajo, posteriormente las del penúltimo centro a visitar y sucesivamente hasta la base. A este proceso lo llamaremos *CMD*. Cada vez que se incluya una incidencia en L_1 , se le pasará el algoritmo k-NN para obtenerse una nueva ruta R' . Si R' no supera las 8 horas de trabajo, $T' < 8$, admitimos la nueva ruta, $R = R'$, para volver al punto E.1, en caso contrario se mantiene la ruta R , en la que no se ha incluido ninguna nueva incidencia, para ejecutar la fase Registro. Para el caso de no existir ninguna incidencia de correctivo se pasaría la fase Centros cercanos.

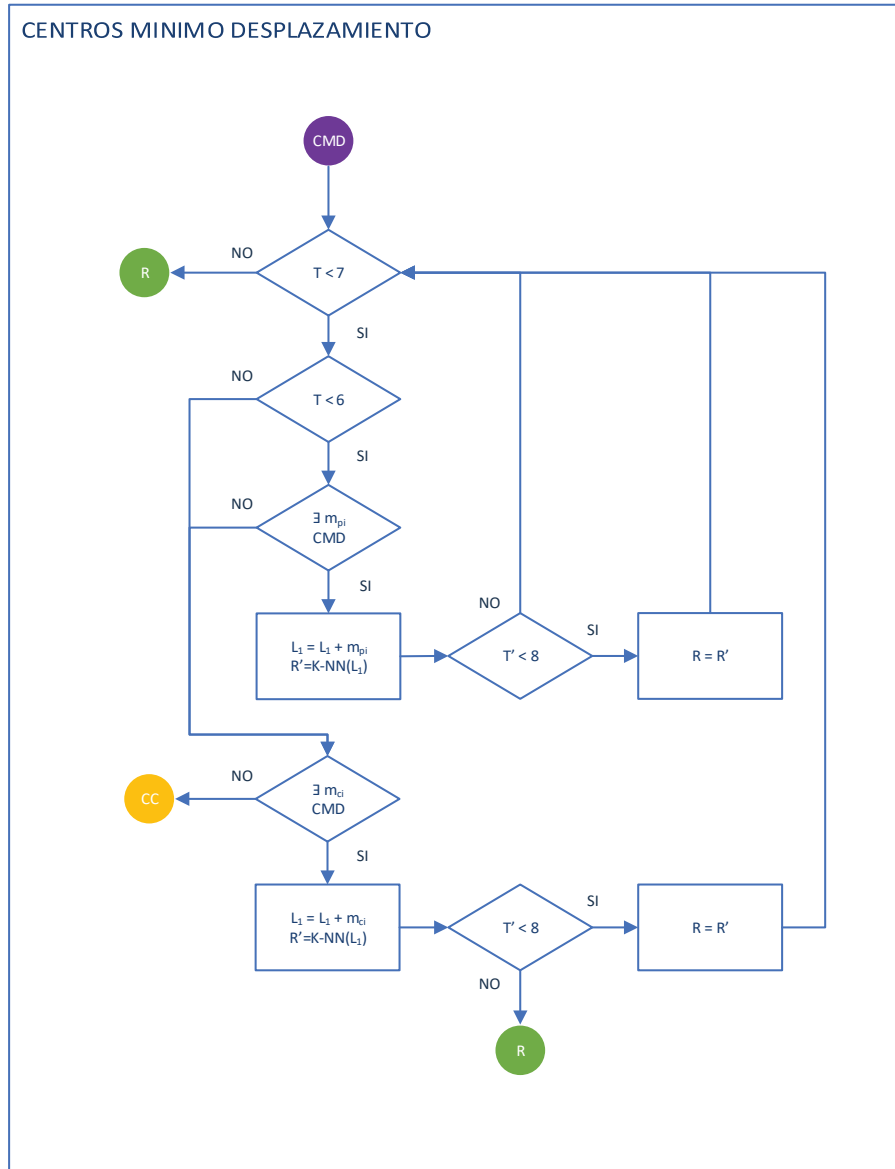


Figura 13. Diagrama de flujo de la fase Centros mínimo desplazamiento del algoritmo basado en k-NN

F) Fase Centros cercanos

Es posible que, sin exceder la jornada laboral de 8 horas, se pueda incluir en la ruta R , que viene de la fase anterior, alguna incidencia perteneciente a L_2 , pero solo aquellas de centros próximos a la ruta R .

- F.1. Si el tiempo de la ruta supera las 7 horas de trabajo, $T > 7$, no se podrá incluir ninguna incidencia puesto que las de menor tiempo de operación, las de correctivo, son de 1 hora, mientras que las de preventivo son de 2 horas, por lo que se pasaría a la fase Registro.
- F.2. En el caso que no se superasen las 6 horas, se buscaría en L_2 las incidencias preventivas de centros que cumplan:

$$t_{ik} \leq t_{jk}; \forall V_i, V_j \notin R; \forall V_k \in R \quad (85)$$

De entre las incidencias cuyos centros cumplan la ecuación (85), se irán incluyendo en L_1 una a una estas incidencias de preventivo m_p , comenzando por las de menor tiempo de desplazamiento t_{ij} siguiendo el orden inverso de la ruta, es decir, primero las del último centro a visitar, si existiesen órdenes de trabajo, posteriormente las del penúltimo centro a visitar y sucesivamente hasta la base. A este proceso lo llamaremos CC , se le pasará el algoritmo k-NN para obtenerse una nueva ruta R' . Si R' no supera las 8 horas de trabajo, $T' < 8$, admitimos la nueva ruta, $R = R'$, para volver al punto anterior, en caso contrario se mantiene la ruta R , en la que no se ha incluido ninguna nueva incidencia, para regresar al punto F.1. Para el caso de no existir ninguna incidencia de preventivo, con las condiciones descritas, se pasaría al punto siguiente.

- F.3. En el caso que se superasen las 6 horas o no hubiese ninguna incidencia a incluir en el punto anterior, se buscaría en L_2 las incidencias de correctivo cuyos centros cumplan la ecuación (85). Se irán incluyendo en L_1 una a una estas incidencias de correctivo m_c , comenzando por las de menor tiempo de desplazamiento t_{ij} siguiendo el orden inverso de la ruta, es decir, primero las del último centro a visitar, si existiesen órdenes de trabajo, posteriormente las del penúltimo centro a visitar y sucesivamente hasta la base. A este proceso lo llamaremos CC . Cada vez que se incluya una incidencia en L_1 , se le pasará el algoritmo k-NN para obtenerse una nueva ruta R' . Si R' no supera las 8 horas de trabajo, $T' < 8$, admitimos la nueva ruta, $R = R'$, para volver al punto F.1, en caso contrario se mantiene la ruta R , en la que no se ha incluido ninguna nueva incidencia, para ejecutar la fase Registro. Para el caso de no existir ninguna incidencia de correctivo también se pasaría la fase Registro.

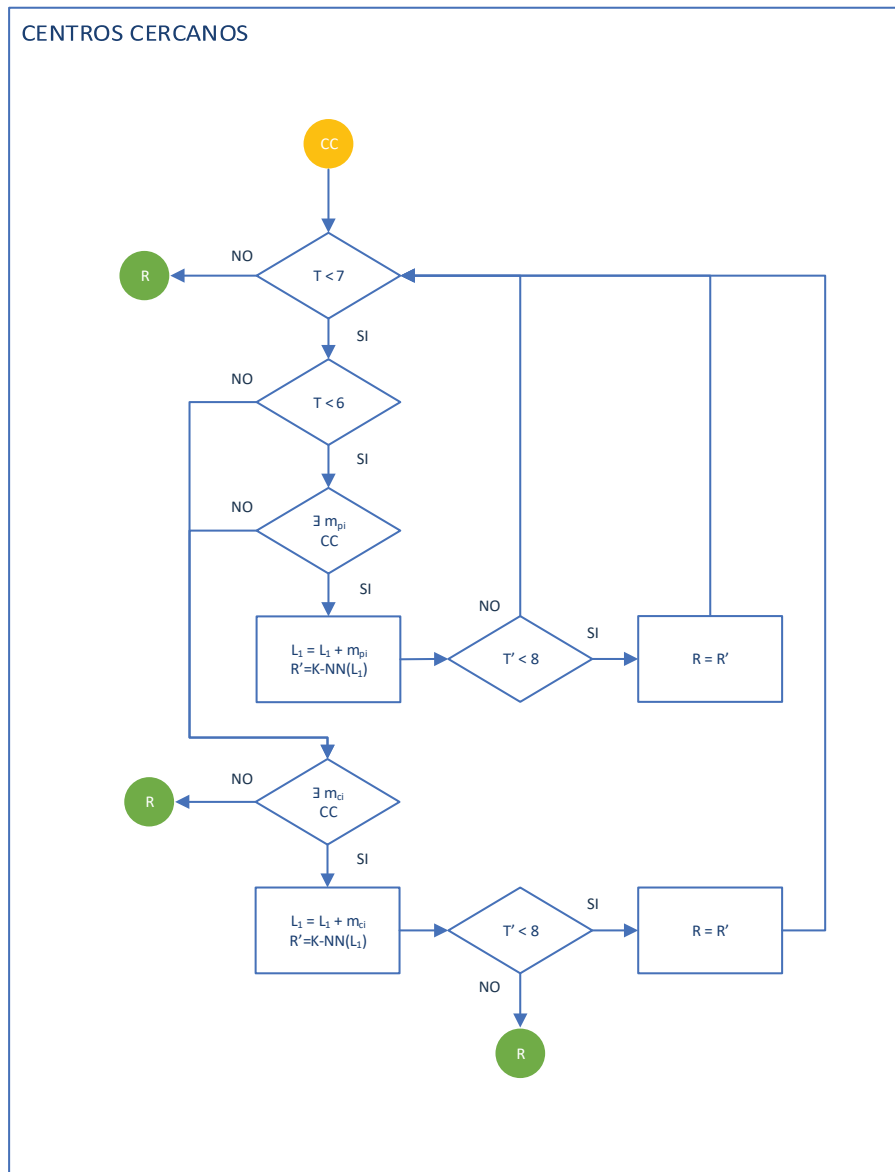


Figura 14. Diagrama de flujo de la fase Centros cercanos del algoritmo basado en k-NN

G) Fase Registro

En esta fase se estudia si se puede registrar la ruta propuesta, quedando del siguiente modo:

- G.1. Se registrará la ruta R , propuesta en alguna de las fases anteriores y se volverá a la fase Resolución si $T \geq h$, es decir el tiempo empleado en la ruta propuesta es mayor o igual a h , que es el número de horas mínimo por ruta para registrarse. En caso negativo se pasaría a la fase Resolución para pasar de día y continuar.

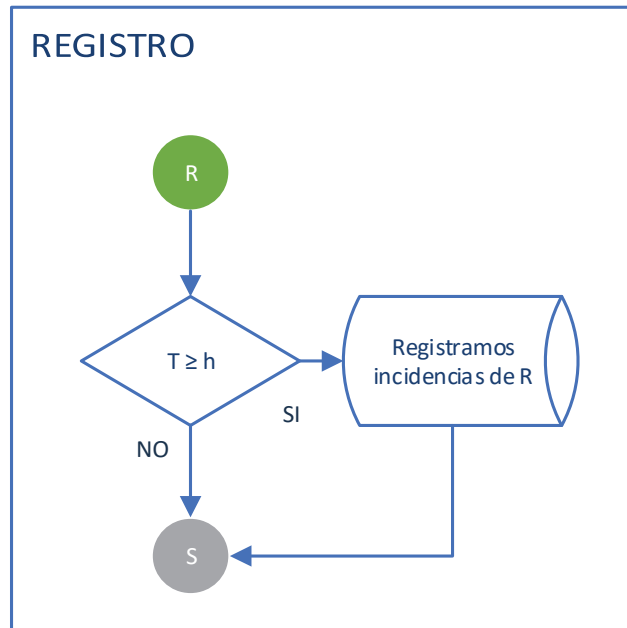


Figura 15. Diagrama de flujo de la fase Registro del algoritmo basado en k-NN

4.2. Algoritmo simple

Se propone optimizar las rutas de mantenimiento mediante la aplicación Microsoft Excel empleando el algoritmo Evolucionista del complemento Solver.

Puesto que puede ser creada una orden de trabajo con plazo de ejecución de hasta 3 días, el estudio se extenderá desde hoy, fecha en la que se inicia el algoritmo, hasta el tercer día posterior.

El algoritmo propuesto consta de varias fases:

- Resolución: donde se encuentra la macro del libro donde se inicia el algoritmo, que finalizará cuando se hayan recorrido todas las fechas de la base de datos.
- Estudio: que pasará la fecha al siguiente día laborable y se definirá las fechas del horizonte temporal dentro de los días laborables: hoy H , mañana M , pasado P y siguiente S , así como se buscarán las incidencias que caducarán dentro de esta horquilla de fechas, es decir con fecha de resolución $F_r \leq S$
- Reparto: teniendo en cuenta los recursos disponibles, buscará si es posible realizar las órdenes de trabajo con fecha de resolución menor o igual S en los cuatro días de horizonte temporal, mediante la ejecución de Solver. Las restricciones contempladas son de tiempo R_t , que impiden que en ninguno de los cuatro días se puedan exceder de 8 horas de jornada y las restricciones de orden R_o , con las que se fuerza a que las incidencias no se puedan realizar antes de su fecha de creación F_c y no se supere el plazo de resolución cada una de ellas, es decir no podrán visitarse posteriormente a la fecha de resolución máxima F_r . La función objetivo es minimizar los tiempos de desplazamiento T_d .
- Externalizar: si no fuese posible resolver todas las órdenes de trabajo en la fase de Reparto, en Externalizar se externalizará la ejecución de alguna de ellas, de tal modo que las restantes si puedan ser cerradas por los recursos disponibles, teniendo en cuenta además que la ruta externalizada, que partirá y terminará en el centro base, ha de tener un mínimo coste. Se utilizará Solver, siendo su función objetivo minimizar el tiempo de ejecución de las tareas subcontratadas, aplicándose las restricciones de tiempo R_t y de orden R_o .
- OptimizarH: se pretende optimizar el día de hoy, tomando el orden de las órdenes de trabajo propuestas en Reparto, implementando las tareas de preventivo pendientes cuyo plazo de resolución está fuera del horizonte temporal, utilizando Solver, sujeto a las restricciones de tiempo R_t y de orden R_o . En esta fase se estudiarán distintas funciones objetivo, de modo que la jornada laboral de nuestros recursos sea la máxima.

El libro de Excel cuenta con las siguientes hojas:

- TIEMPOS: donde se encuentran la matriz de tiempos de desplazamiento entre los distintos centros de trabajo.
- CALENDARIO: figuran todos los días, desde la fecha de inicio hasta la fecha de fin, de la base de datos. Estas fechas se han clasificado en función de si son laborables o no. En esta hoja se registrará las horas

totales de las rutas propuestas, así como los tiempos de desplazamiento.

- LUGO INCIDENCIAS: en esta hoja se encuentran todas las órdenes de trabajo, numeradas, indicándose su prioridad y su centro de trabajo. Para cada una de ellas se calcula la fecha límite para su resolución. Según se vaya ejecutando el algoritmo propuesto, se registrará la fecha en la que se propone su ejecución.
- REPARTO: donde se estudiará la fase Reparto.
- EXTERNALIZAR: donde se estudiará la fase Externalizar.
- OPTIMIZARH: donde se estudiará la fase OptimizarH.

Así mismo el libro de Excel cuenta con tantos módulos como fases tiene el algoritmo, llamados del mismo modo que éstas, donde se albergan las macros que conforman el algoritmo que presentamos.

Se procede a explicar con detalle cada una de las fases:

A) Fase Resolución

La fase resolución es la que inicia y finaliza el algoritmo, contando con las siguientes etapas:

- A.1. Se inicia el algoritmo cuya fecha de hoy es un día anterior a la de inicio de la base de datos.
- A.2. Se indicará el valor $h = 4,5$ horas, por debajo del cual el tiempo total de una ruta no será registrado, como hemos visto en el apartado 3.7.
- A.3. Se define la fecha final F_f , que coincide con la última fecha de la base de datos.
- A.4. En caso de que $F_f \geq M$, la fecha que figura para mañana, pasará a la fase Estudio.
- A.5. Si $F_f < M$ se finalizará el algoritmo.

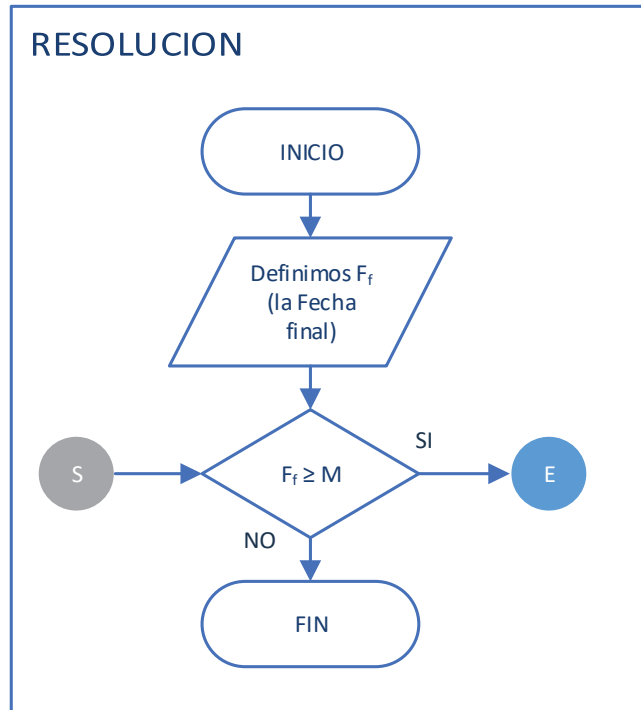


Figura 16. Diagrama de flujo de la fase Resolución del algoritmo simple

B) Fase Estudio

Durante este proceso se avanzará en el calendario y se buscarán las incidencias susceptibles de ejecución. Sus etapas son como sigue:

- B.1. Se pasará de día, es decir H tomará el valor de M y en la hoja CALENDARIO se buscarán los valores de los siguientes 3 días laborables, para definir M , P y S .
- B.2. Se realizará una lista, buscando en la hoja LUGOINCIDENCIAS, con aquellas incidencias con $F_c \leq H$ y $F_r \leq S$, a la que denominaremos L_1 , listado de órdenes de trabajo a ejecutar.
- B.3. También se buscarán en LUGOINCIDENCIAS las órdenes de trabajo que $F_c \leq H$ y $F_r > S$, que serán las órdenes de trabajo de preventivo que no haya que realizar dentro de nuestro horizonte temporal. A ésta la llamaremos L_2 , listado de órdenes de trabajo de preventivo.
- B.4. Finalmente se dará orden para que se ejecuten las macros del módulo Reparto.

FECHA	29/01/2009	30	1ª PREDICCIÓN	30/01/2009	31			RESETEAR	RESOLUCIÓN	1 DÍA			
			2ª PREDICCIÓN	02/02/2009	34								
			3ª PREDICCIÓN	03/02/2009	35								
L1	INCIDENCIAS HORIZONTE TEMPORAL					L2	PREVENTIVOS PENDIENTES						
TIPO	Nº OT	FECHA	CENTRO	PROVINCIA	PRIORIDAD	FECHA LIMITE B	TIPO	Nº OT	FECHA	CENTRO	PROVINCIA	PRIORIDAD	FECHA LIMITE B
R	45	24/01/2009	L08	LUGO	1. Baja	29/01/2009	R	7	01/01/2009	L07	LUGO	0. Preventivo	02/03/2009
R	46	24/01/2009	L02	LUGO	1. Baja	29/01/2009	R	11	01/01/2009	L11	LUGO	0. Preventivo	02/03/2009
R	47	26/01/2009	L09	LUGO	1. Baja	29/01/2009	R	12	01/01/2009	L12	LUGO	0. Preventivo	02/03/2009
R	49	27/01/2009	L05	LUGO	1. Baja	30/01/2009	R	13	01/01/2009	L13	LUGO	0. Preventivo	02/03/2009
							R	14	01/01/2009	L14	LUGO	0. Preventivo	02/03/2009
							R	15	01/01/2009	L15	LUGO	0. Preventivo	02/03/2009
							R	16	01/01/2009	L16	LUGO	0. Preventivo	02/03/2009
							R	17	01/01/2009	L17	LUGO	0. Preventivo	02/03/2009
							R	18	01/01/2009	L18	LUGO	0. Preventivo	02/03/2009
							R	19	01/01/2009	L19	LUGO	0. Preventivo	02/03/2009
							R	20	01/01/2009	L20	LUGO	0. Preventivo	02/03/2009
							R	21	01/01/2009	L21	LUGO	0. Preventivo	02/03/2009
							R	22	01/01/2009	L22	LUGO	0. Preventivo	02/03/2009
							R	23	01/01/2009	L23	LUGO	0. Preventivo	02/03/2009
							R	29	01/01/2009	L29	LUGO	0. Preventivo	02/03/2009

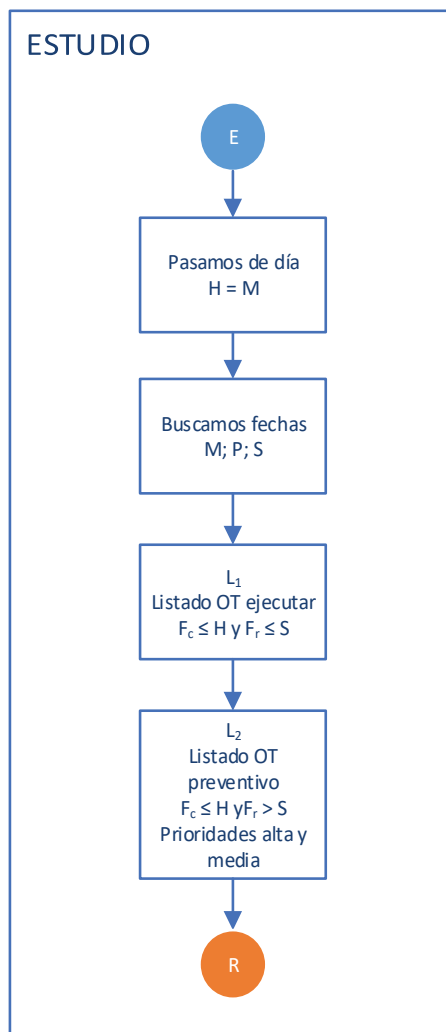
Figura 17. Detalle de las listas L_1 y L_2 

Figura 18. Diagrama de flujo de la fase Estudio del algoritmo simple

C) Fase Reparto

En esta fase se pretende ver si con los recursos disponibles seríamos capaces de realizar las incidencias, sujetos a las restricciones R_t y R_o .

Esta fase se divide en las siguientes etapas:

- C.1. Traeremos a la hoja REPARTO las fechas H , M , P y S .
- C.2. Si se diese el caso que L_1 y L_2 están vacías significa que no tenemos ninguna incidencia que podamos ejecutar hoy, por lo que volveremos a la fase Resolución para avanzar de día.
- C.3. Si sucediese que L_1 está vacía, pero si hubiese incidencias en L_2 , enviaríamos L_2 a la hoja OPTIMIZARH y ejecutaríamos dicha fase. Como veremos más adelante, en la fase OptimizarH se conservarán la secuencia propuesta en Reparto y se intentará intercalar incidencias de L_2 . Al no existir tareas en L_1 solo se tendrá que optimizar el día de hoy ejecutando tareas de L_2 .
- C.4. La siguiente posibilidad que se plantea es que L_1 no esté vacía, dicha lista será pegada, en la hoja REPARTO, bajo tres líneas identificadas con número de orden de trabajo $F1$, $F2$ y $F3$ que representarán las hipotéticas ordenes de trabajo que finalizarán las jornadas de H , M y P , respectivamente, todas ellas asociadas al centro base. Efectivamente, cada una de las jornadas ha de iniciarse y finalizarse en la base, por lo que podemos identificar las hipotéticas ordenes de trabajo que iniciará las jornadas de cada día como $I1$, $I2$, $I3$ e $I4$, ubicadas en la base. Así mismo se identifica como $F1$, $F2$, $F3$ y $F4$ a las incidencias que finalizarán cada ruta, también situadas en la base. Ninguna de estas hipotéticas incidencias tiene tiempo de operación. Con ellas conseguimos iniciar las rutas y acabar las rutas en la base. Pretendemos buscar el orden de las incidencias de L_1 , que se colocaran entre el inicio Ii y fin Fi de cada día. Si escribiésemos la secuencia de las rutas de los cuatro días, se iniciaría con $I1$, aparecerían las incidencias a realizar el primer día, seguiría $F1$ e $I2$, al finalizarse la ruta del día de hoy e iniciarse la de mañana y así sucesivamente hasta acabarse con un $F4$. Dado que en esta secuencia coinciden correlativamente $F1$ con $I2$, $F2$ con $I3$ y $F3$ con $I4$, se prescinde de $I2$, $I3$ e $I4$ para el cálculo de las rutas suponiendo que $F1$ finaliza el primer día e inicia el segundo, sucediendo lo mismo con $F2$ y $F3$.

The screenshot shows a spreadsheet titled 'REPARTO'. On the left, there is a table with columns 'FECHA', 'OF', and 'ODOS'. Below it, a table lists incidents with columns: 'LINEA', 'CENTRO', 'FECHA INICIO', 'FECHA FIN', 'FECHA LIMITE D', 'LIMITE IN', 'VARIABLES', 'LIMITE SUP', and 'ODOS'. On the right, a larger table is used for calculations, with columns: 'ALTA TABLA', 'IDENTIFICADOR', 'OPERACION', 'FECHA', 'ORIGEN', 'DESTINO', 'DESPLAZAM', 'TIEMPO OP', 'ACUMULADO', 'TOTAL DIA', and 'ACUM OPERACION'. The table contains numerical data for various routes and operations.

Figura 19. Detalle de la hoja Reparto, donde se aprecian la lista de incidencias L_1 y la tabla de cálculo

A la lista de incidencias se les añaden las columnas Límite inferior y Límite superior, que se calculan con fórmulas mediante la tablita situada a la derecha de las fechas, que establecerá las restricciones de orden. La columna Variables serán las variables que tomará Solver. Las dos últimas columnas nos indicarán si se han respetado las restricciones de orden en la secuencia de rutas obtenidas como solución.

- C.5. Se creará una tabla base del cálculo, con tantas filas como las de la lista de incidencias de esta hoja. La primera fila tras los encabezados y la última, con números de orden de trabajo $I1$ y $F4$, marcarán el inicio de la ruta del primer día y la finalización de la ruta el cuarto día. En la primera columna se numeran las filas de las posiciones que pueden ser tomadas por las incidencias de la lista. Tras una columna auxiliar para el cálculo, nos presenta el orden de la ruta, a continuación, los números de las incidencias incluidas en las rutas. Las columnas Origen y Destino indicarán desde que centro se parte y hasta que centro se desplazará, presentándose en la columna desplazamiento el tiempo que conllevaría cada uno, siendo buscados en la hoja DESPLAZAMIENTOS, mediante fórmulas. La columna tiempo de operación indica para cada línea el tiempo estimado para ejecutar la orden de trabajo. Las últimas columnas indican el tiempo total acumulado para cada línea, es decir, tiempo de desplazamiento más el de operación. El tiempo total para cada día se presentará solo en las líneas en que se finalicen las jornadas y finalmente se muestra el tiempo de operación acumulado por cada línea.

La tabla busca, para cada una de sus líneas numeradas, las incidencias en la lista cuya variable coincida con el valor de la línea de la tabla.

- C.6. Se programará el Solver, definiéndose la función objetivo, T_d tiempo de desplazamiento, que se pretende minimizar. Las variables son las celdas de la columna variable de la lista, como hemos

visto. Se añaden las restricciones de tiempo y de orden, además las variables han de ser números enteros y distintos. Finalmente se indica que el método de resolución es el Evolucionista.

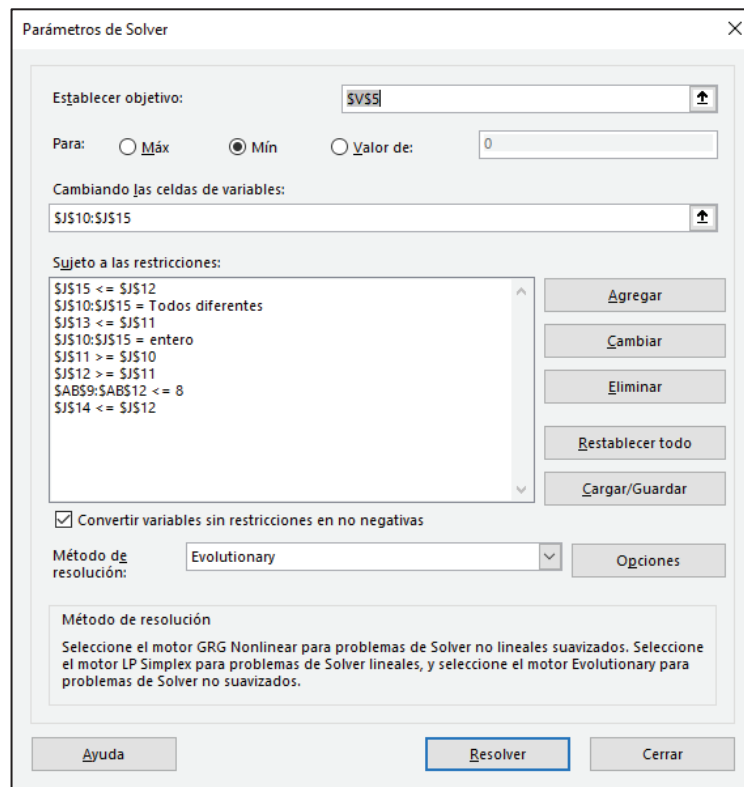


Figura 20. Detalle de los parámetros de Solver en la fase Reparto

- C.7. Se ejecutará el Solver
- C.8. Si no se cumpliesen las restricciones de orden y/o las de tiempo en la solución propuesta se ejecutaría la fase Externalización. En cambio, si se cumpliesen ambas llevaríamos a la hoja OPTIMIZARH tanto las incidencias de las tablas como la secuencia horas trabajadas para el día de hoy y se iniciaría la fase OptimizarH.

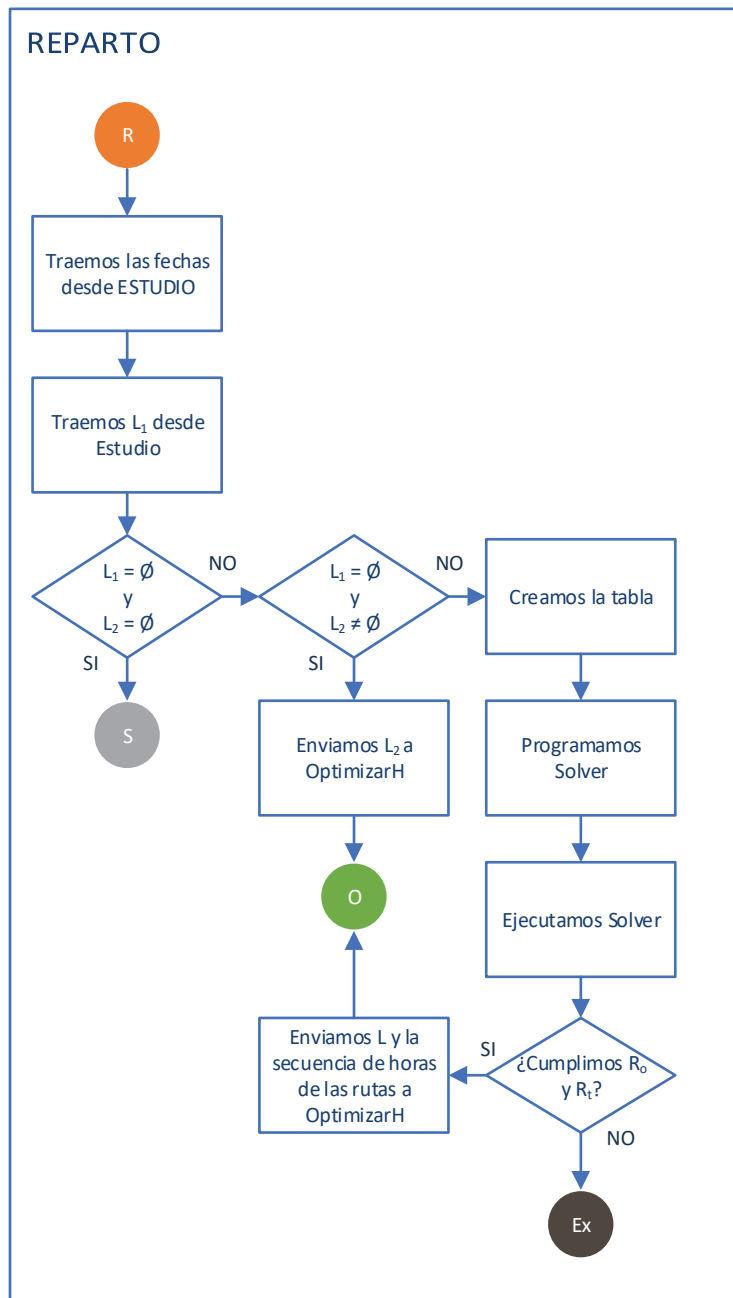


Figura 21. Diagrama de flujo de la fase Reparto del algoritmo simple

D) Fase Externalizar

Partiendo del mismo sistema que el empleado en Reparto, se crean las hipotéticas órdenes de trabajo *F1*, *F2* y *F3*, añadiéndose un *F0* que corresponderá a la finalización de la ruta a externalizar.

Las etapas de esta fase son:

- D.1. Se lleva a la hoja EXTERNALIZAR las incidencias que figuran en REPARTO, colocándolas bajo las hipotéticas órdenes de trabajo *Fi*.
- D.2. Se crea la tabla de modo similar a la creada en la etapa Reparto, en las que además de las incidencias figuran las hipotéticas órdenes de trabajo *F0*, *F1*, *F2* y *F3*.
- D.3. Se programa el Solver con las mismas restricciones que en Reparto. Al incluirse *F0*, donde finalizaría la ruta a externalizar, que deberá de colocarse por encima de *F1*, todas las incidencias son libres de poder ser externalizadas y se cumplirán las restricciones de orden. La función objetivo será minimizar el tiempo total de la ruta que finaliza en *F0*, de modo que sea mínimo el coste de las tareas a externalizar.
- D.4. Se ejecuta el Solver.
- D.5. Se registran las tareas cuya ruta finaliza en *F0*, como externalizadas.
- D.6. Se enviará la lista de las incidencias no externalizadas a Reparto y se ejecuta esa fase.

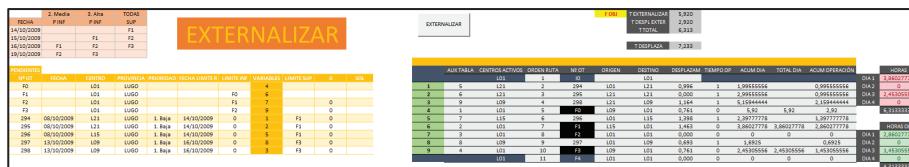


Figura 22. Detalle de la hoja Externalizar, donde se aprecian la lista de incidencias y la tabla de cálculo

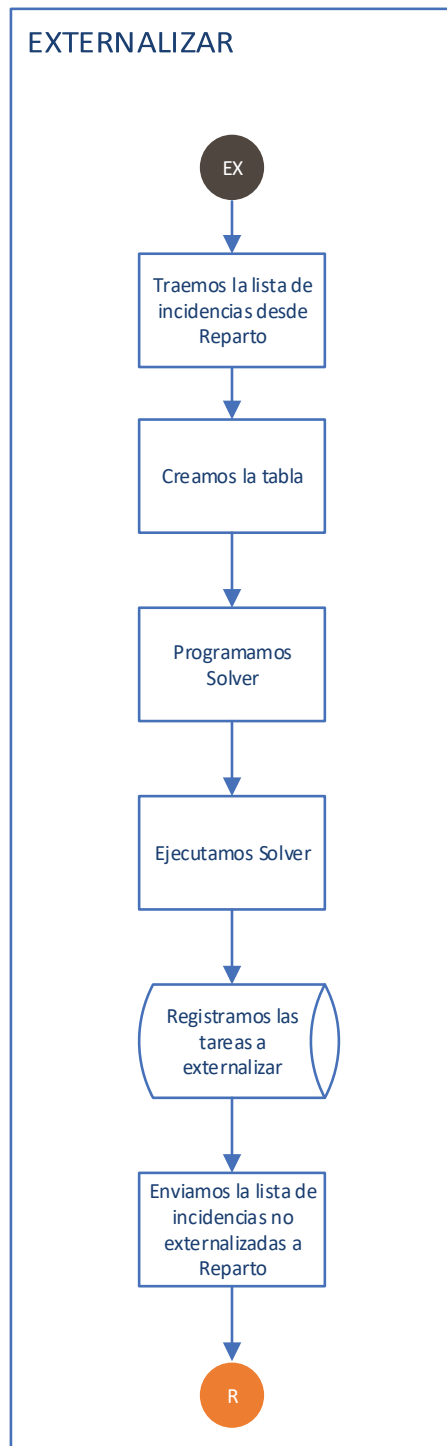


Figura 23. Diagrama de flujo de la fase Externalizar del algoritmo simple

E) Fase OptimizarH

Finalmente intentaremos optimizar el día de hoy, partiendo de la lista de incidencias que se han importado, según el orden establecido en Reparto. A esta lista la denominaremos L_0 .

También ha sido importada la lista de incidencias de preventivo L_2 .

Se seguirán las siguientes etapas:

- E.1. Se importa, a la hoja OPTIMIZARH, la fecha de hoy, H .
- E.2. Si existen incidencias en L_0 , pero no en L_2 , no podría mejorarse la ruta ya que no existen mantenimientos preventivos pendientes de realizar, por lo que se vería si $T_1 \geq h$, es decir el tiempo empleado en la ruta propuesta en Reparto es mayor o igual a h , que es el número de horas mínimo por ruta para registrarse. En caso afirmativo se registra la ruta no mejorada y en caso negativo se pasaría a la fase Resolución para pasar de día y continuar.
- E.3. Habiendo incidencias en L_0 y L_2 , si sucediese que la ruta propuesta por Reparto superase las 6 horas, $T_1 > 6$, ésta no se podría mejorar ya que no tiene capacidad para asumir 2 horas, resultantes de ejecutar un preventivo, por lo que se grabaría la ruta que se ha importado y se pasaría a la fase Resolución.
- E.4. En caso de poderse mejorar la ruta importada, se crearía una tabla similar a la de Reparto. En esta tabla se ubicarán los datos de las incidencias siguiendo el orden propuesto. Se dejarán filas libres numeradas entre las líneas fijas I1, las incidencias de la secuencia y F1. Entre las filas fijas se dejarán tantas filas libres como resulte el valor del redondeo a menos, sin decimales, de la mitad del resultado de restarle a 8 el valor redondeado a más de T_1 . Estas filas podrán ser ocupadas o no por órdenes de trabajo de preventivo.

The screenshot shows a spreadsheet titled 'OPTIMIZARH'. It contains several tables and a calculation table. The top part shows incident lists with columns for 'INCIDENCIA', 'RUT', 'FECHA', 'CATEGORIA', 'TIPO', 'ESTADO', 'FECHA DE INICIO', 'FECHA DE FIN', 'FECHA DE VENCIMIENTO', 'FECHA DE REALIZACION', 'FECHA DE PREVISION', and 'FECHA DE VIGENCIA'. Below these are two more incident lists, one for 'PREVENTIVO' and another for 'ORDEN DE TRABAJO'. On the right side, there is a calculation table with columns 'RUT', 'ORDEN DE TRABAJO', 'FECHA', 'CATEGORIA', 'TIPO', 'ESTADO', 'FECHA DE INICIO', 'FECHA DE FIN', 'FECHA DE VENCIMIENTO', 'FECHA DE REALIZACION', 'FECHA DE PREVISION', and 'FECHA DE VIGENCIA'. The calculation table has rows numbered 1 to 4, with values in the 'FECHA DE REALIZACION' and 'FECHA DE PREVISION' columns.

Figura 24. Detalle de la hoja OptimizarH, donde se aprecian las listas L_0 y L_2 y la tabla de cálculo

Al programarse Solver, tenemos en cuenta que las variables, asociadas a la lista de incidencias preventivas L_2 , deben de ser ampliadas ya que tenemos más huecos en la tabla que órdenes de trabajo se pudiesen ejecutar sin superar las 8 horas de jornada.

Se han estudiado varias funciones objetivo puesto que, llegados a este punto, sería deseable que cada operario ejecutase el mayor número de horas posible, pero minimizando los desplazamientos. Los datos estudiados serán expuestos en el capítulo 5.

- E.5. Si sucediese que la propuesta obtenida por Solver superase las 8 horas, $T_0 > 8$, al no cumplirse la restricción de tiempo R_t , no se podría registrar, por lo que analizaríamos si $T_1 > h$, en caso afirmativo se registraría la ruta mejorada por OptimizarH y en caso negativo se ejecutaría Resolución para continuar con el algoritmo.

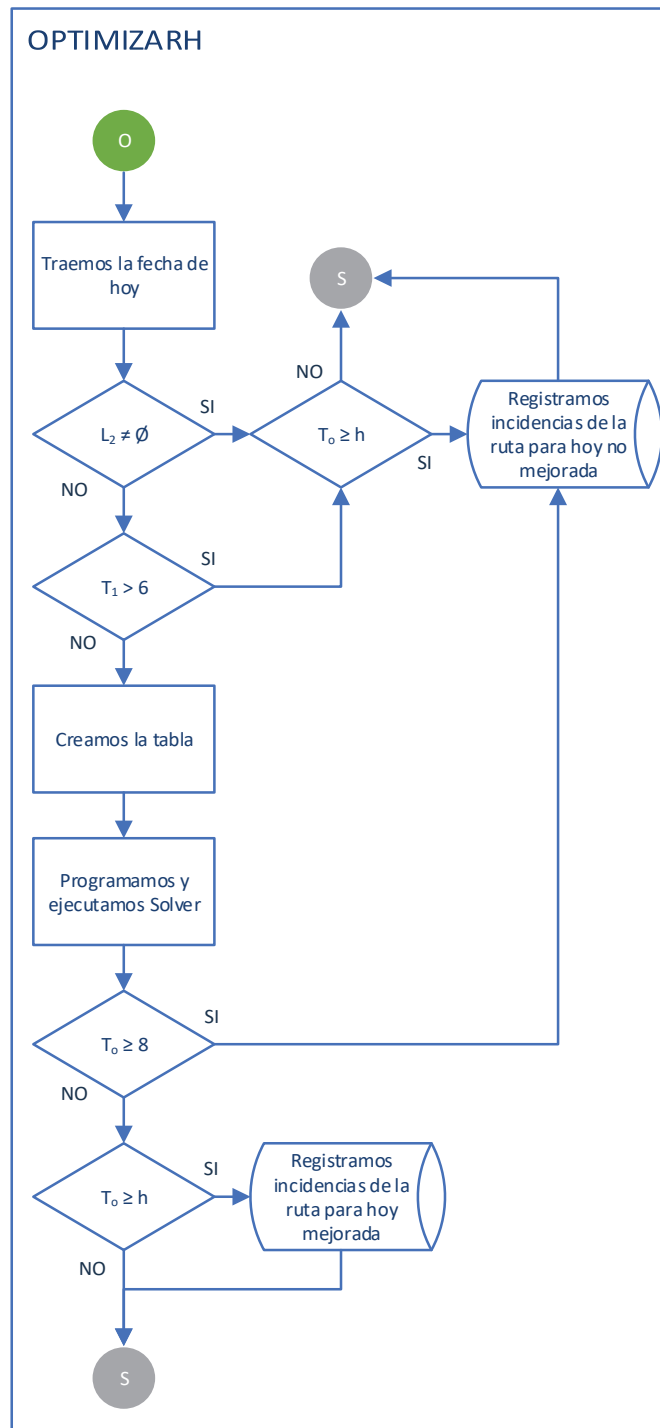


Figura 25. Diagrama de flujo de la fase OptimizarH del algoritmo simple

4.3. Algoritmo con reordena

Tomando como base el Algoritmo Simple, se han añadido dos nuevas fases:

- Transferir: puesto que puede haber en L_2 , órdenes de trabajo de preventivo, con F_r fuera del horizonte temporal, es posible que se puedan implementar en las rutas, para optimizarlas en OptimizarH. Como estas órdenes de trabajo tienen un tiempo de operación de 2 horas, se buscará la posibilidad de reordenar las incidencias incluidas en las rutas, de modo que en sus jornadas de trabajo se puedan incluir más órdenes de trabajo de preventivo. Se estudiará la secuencia de centros, en la fase Reparto o Externalizar, que se proponen se visiten durante los cuatro días y se presentarán todas las posibilidades de reordenarlas, cumpliendo las restricciones orden de R_o .
- Reordena: si fuese el caso que hubiese alguna opción, en Transferir, para reordenar las rutas, mediante Solver se buscará si es viable alguna de estas opciones, cumpliendo las restricciones de tiempo R_t y de orden R_o . La función objetivo minimiza los tiempos de desplazamiento T_d .

Así mismo, se han añadido dos hojas al libro:

- TRANSFERIR: donde se estudiará la fase Transferir.
- REORDENA: donde se estudiará la fase Reordena.

Por lo que también se cuentan con dos nuevos módulos.

Por lo que las fases quedan como sigue:

A) Fase Resolución

La fase resolución es la que inicia y finaliza el algoritmo, contando con las siguientes etapas:

- A.1. Se inicia el algoritmo cuya fecha de hoy es un día anterior a la de inicio de la base de datos.
- A.2. Se indicará el valor $h = 4,5$ horas, por debajo del cual el tiempo total de una ruta no será registrado, como hemos visto en el apartado 3.7.
- A.3. Se define la fecha final F_f , que coincide con la última fecha de la base de datos.
- A.4. En caso de que $F_f \geq M$, la fecha que figura para mañana pasará a la fase Estudio.
- A.5. Si $F_f < M$ se finalizará el algoritmo.

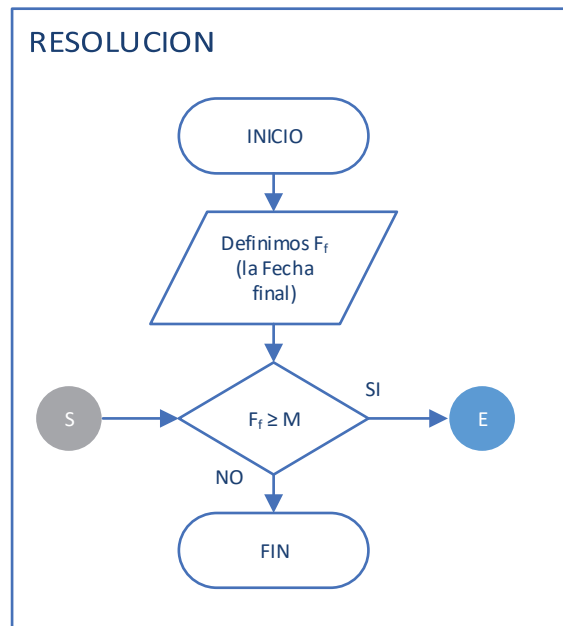


Figura 26. Diagrama de flujo de la fase Resolución del algoritmo con reordena

B) Fase Estudio

Durante este proceso se avanzará en el calendario y se buscarán las incidencias susceptibles de ejecución. Sus etapas son como sigue:

- B.1. Se pasará de día, es decir H tomará el valor de M y en la hoja CALENDARIO se buscarán los valores de los siguientes 3 días laborales, para definir M , P y S .
- B.2. Se realizará una lista, buscando en la hoja LUGOINCIDENCIAS, con aquellas incidencias con $F_c \leq H$ y $F_r \leq S$, a la que denominaremos L_1 , listado de órdenes de trabajo a ejecutar.
- B.3. También se buscarán en LUGOINCIDENCIAS las órdenes de trabajo que $F_c \leq H$ y $F_r > S$, que serán las órdenes de trabajo de preventivo que no haya que realizar dentro de nuestro horizonte temporal. A ésta la llamaremos L_2 , listado de órdenes de trabajo de preventivo.
- B.4. Finalmente se dará orden para que se ejecuten las macros del módulo Reparto.

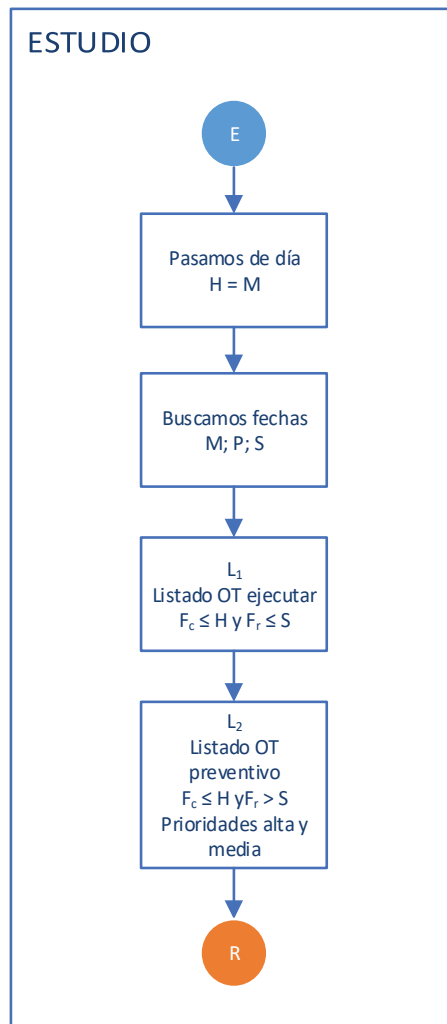


Figura 27. Diagrama de flujo de la fase Estudio del algoritmo con reordena

C) Fase Reparto

En esta fase se pretende ver si con los recursos disponibles seríamos capaces de realizar las incidencias, sujetos a las restricciones R_t y R_o .

Esta fase se divide en las siguientes etapas:

- C.1. Traeremos a la hoja REPARTO las fechas H , M , P y S , además de la lista L_1 , que figuran en la hoja ESTUDIO.

- C.2. Si se diese el caso que L_1 y L_2 están vacías significa que no tenemos ninguna incidencia que podamos ejecutar hoy, por lo que volveremos a la fase Resolución para avanzar de día.
- C.3. Si sucediese que L_1 está vacía, pero si hay incidencias en L_2 , enviaríamos L_2 a la hoja OPTIMIZARH y ejecutaríamos dicha fase, como se hizo en la fase C.3 del algoritmo anterior.
- C.4. La siguiente posibilidad que se plantea es que L_1 no esté vacía, dicha lista será pegada en la hoja, de modo similar a lo expuesto en el punto C.4 del algoritmo anterior.
- C.5. Se creará una tabla base del cálculo, de igual modo que la explicada en el punto C.5 del algoritmo anterior.
- C.6. Se programará el Solver, definiéndose la función objetivo, T_d tiempo de desplazamiento, que se pretende minimizar. Las variables que son las celdas de la columna variable de la lista, como hemos visto. Se añaden las restricciones de tiempo y de orden, así como que las variables han de ser números enteros y distintos. Finalmente se indica que el método de resolución es el Evolucionista.
- C.7. Se ejecutará el Solver
- C.8. Si no se cumpliesen las restricciones de orden y/o las de tiempo en la solución propuesta se ejecutaría la fase Externalización. En cambio, si se cumpliesen ambas restricciones llevaríamos, a la hoja TRANSFERIR, tanto las incidencias de las tablas como la secuencia horas trabajadas para cada uno de los días y se iniciaría la fase Transferir.

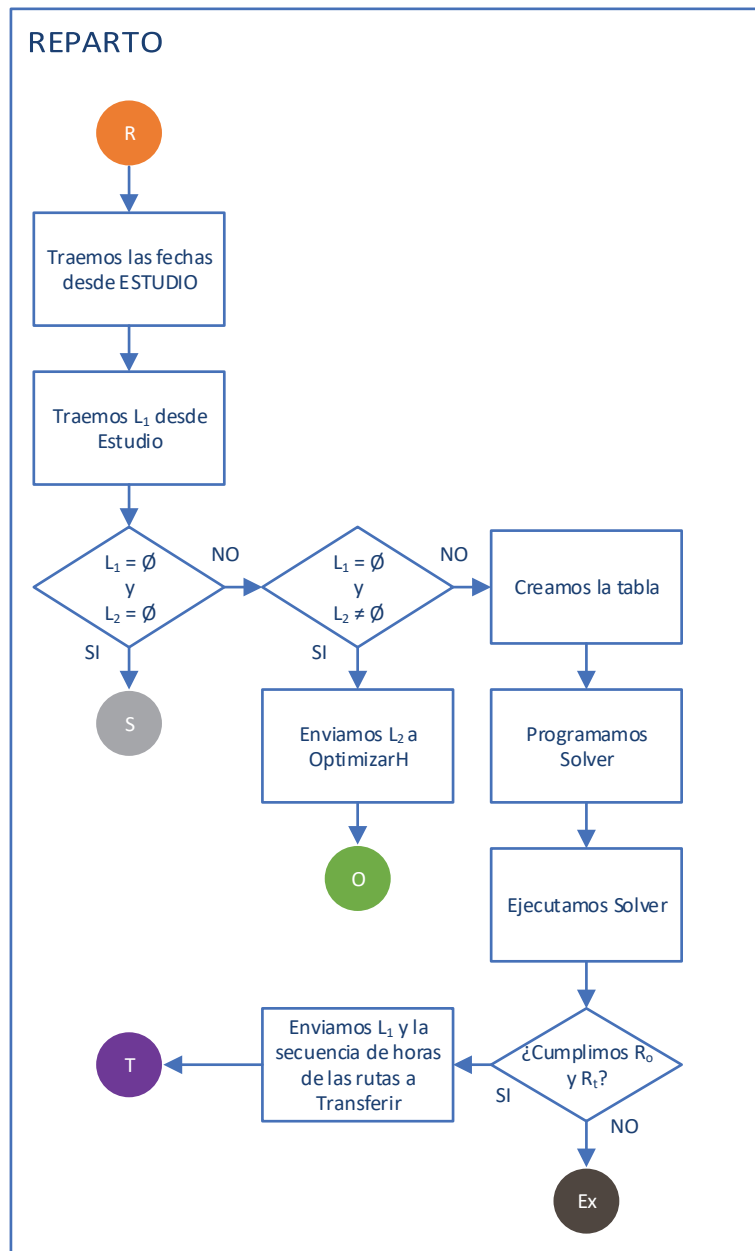


Figura 28. Diagrama de flujo de la fase Reparto del algoritmo con reordena

D) Fase Externalizar

Esta fase es exactamente igual a su análoga del algoritmo anterior.

Las etapas de esta fase son:

- D.1. Se lleva a la hoja EXTERNALIZAR las incidencias que figuran en REPARTO, colocándolas bajo las hipotéticas órdenes de trabajo F_i .
- D.2. Se crea la tabla de modo similar a la creada en la etapa Reparto.
- D.3. Se programa el Solver, con las mismas restricciones que en Reparto, de este modo todas las incidencias son libres de poder ser externalizadas y se cumplirán las restricciones de orden. La función objetivo será minimizar el tiempo total de la ruta que finaliza en F_0 , de modo que será mínimo el coste de las tareas a externalizar.
- D.4. Se ejecuta el Solver.
- D.5. Se registran las tareas que finalizan en F_0 , como externalizadas.
- D.6. Se enviará la lista de las incidencias no externalizadas a Reparto y se ejecutaría esa fase.

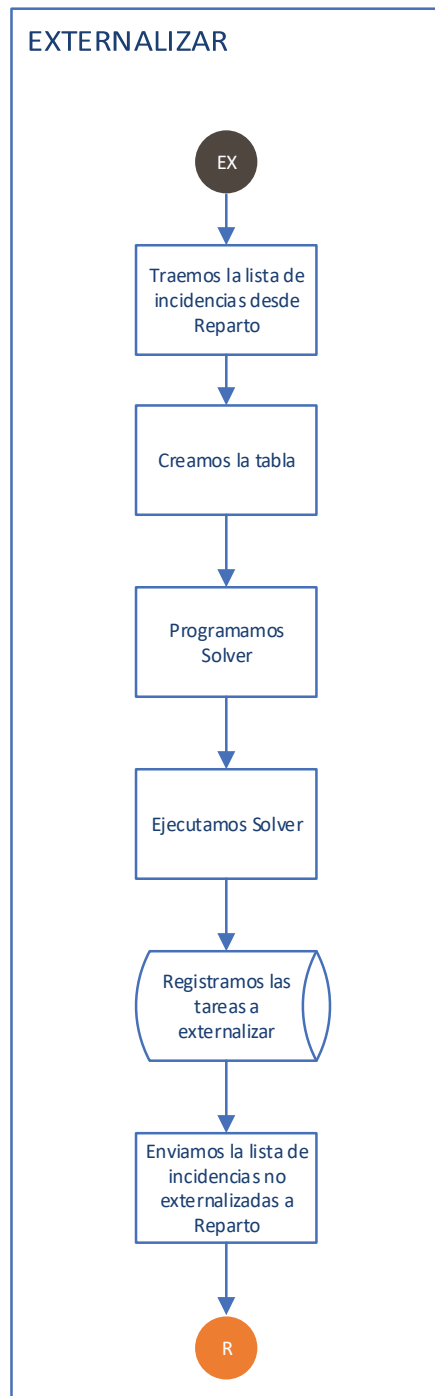


Figura 29. Diagrama de flujo de la fase Externalizar del algoritmo con reordena

E) Fase Transferir

Se pretende ver si dentro de la secuencia de horas para los cuatro días propuestas en Estudio, existe alguna combinación que admita, en los cuatro días, mayor número de lotes de 2 horas, que es el tiempo de operación de las incidencias pertenecientes a L_2 , que se intentarán implementar a las rutas en la fase OptimizarH.

Para ello, previamente se ha estudiado todas las posibilidades de secuencias de horas de trabajo para los cuatro días, desde 0001 hasta 7777, analizándose cuántos grupos de 2 horas pueden incrementarse sin que se supere para cada jornada las 7 horas. Se ha tomado como límite diario 7 horas para, al menos, que haya una hora para desplazarse entre centros. No se ha incluido en el estudio la secuencia 0000 porque en ella no hay incidencias que reordenar.

Posteriormente se relacionaron todas las secuencias que tengan la misma suma de horas de cada los 4 días.

La hoja TRANSFERIR presenta una primera tabla, en la que en su tercera columna se encuentran todas las posibles secuencias de horas, en la columna Suma de Horas se indica la suma de horas de trabajo y en Número de horas Secuencia se indica cuantos grupos de 2 horas podríamos incrementar a esta secuencia, en las columnas que tiene como encabezado Mejoras Máximas e Incremento Mejora se hace mención al número de mejoras máximas que podríamos realizar si redistribuyésemos los trabajos de la secuencia y por último la diferencia entre esta mejora máxima y la que figura en la columna Números de horas Secuencia. Se ha observado que puede haber mejoras de una o dos horas, según la secuencia.

Nº H SECUENCIA	SUMA DE HORAS	Nº	MEJORAS MAX	INCREMENTO MEJORA
12	1	0001	12	0
11	2	0002	12	1
11	3	0003	12	1
10	4	0004	12	2
10	5	0005	11	1
9	6	0006	11	2
9	7	0007	10	1
12	1	0010	12	0
12	2	0011	12	0
11	3	0012	12	1
11	4	0013	12	1
10	5	0014	11	1
10	6	0015	11	1
9	7	0016	10	1
9	8	0017	10	1
11	2	0020	12	1
11	3	0021	12	1
10	4	0022	12	2
10	5	0023	11	1
9	6	0024	11	2

MEJORA	22,2%	11,1%
Nº PERMUTACIONES	4	60
PREVENT POSIBLES	11	10
RUTA 1	1113	0015
RUTA 2	1131	0033
RUTA 3	1311	0051
RUTA 4	3111	0105
RUTA 5		0114
RUTA 6		0123
RUTA 7		0132
RUTA 8		0141
RUTA 9		0150
RUTA 10		0213
RUTA 11		0231
RUTA 12		0303
RUTA 13		0312

Figura 30. Detalle de la tabla con todas las posibles secuencias de horas y de la tabla con las posibles permutaciones de las secuencias de horas propuesta en Reparto

Existe una segunda tabla, donde figura la secuencia de horas obtenida en Reparto, que serán redondeadas a más o a menos, en función del método que se emplee, así como el número de preventivos que se podrían incluir. Bajo esta tabla se presenta otra que nos muestra las posibles secuencias de horas que mejorarían en uno o dos los preventivos que podríamos implementar.

Será objeto de estudio independiente, para cada función objetivo que se propongan en el capítulo 5, la mencionada opción del redondeo a más y a menos en la secuencia de horas propuestas por Reparto.

La fase Transferir presenta las siguientes etapas:

- E.1. Si no existiese ninguna permutación que mejorase la secuencia propuesta de Reparto, se llevaría la secuencia a OptimizarH y se ejecutaría esta fase.
- E.2. Si al menos existiese una permutación que pudiese mejorar la secuencia, en la lista de incidencias se escribirían las ecuaciones que determinarían las restricciones de orden. Buscando en la tabla superior izquierda se definiría el tipo de restricción que tiene cada incidencia y posteriormente la tabla superior derecha nos indicará con un número 1 los días que se pueda ejecutar y con un 0 los que no. Estos 1 o 0, se multiplicarán por el tiempo de operación de cada incidencia, de modo que en cada línea figurará el tiempo de operación asignado a cada uno de los días en los que puede ejecutarse.

L. INFERIOR		LIMITE SUPERIOR				D1 D2 D3 D4			
O	F1	F2	F3						
A	B	C	D		1	1	1	1	
E	F	G			1				1
H	I					1	1		
J							1	1	1

INCIDENCIAS PENDIENTES									LÍMITES				
TIPO	Nº OT	FECHA	CENTRO	PROVINCIA	PRIORIDAD	FECHA LIMITE R	LIMITE INF	VARIABLES	LIMITE SUP	1ª	2ª	3ª	4ª
R	45	24/01/2009	L08	LUGO	1. Baja	29/01/2009	0	2	F3	1	1	1	0
R	46	24/01/2009	L02	LUGO	1. Baja	29/01/2009	0	4	F3	1	1	1	0
R	47	26/01/2009	L09	LUGO	1. Baja	29/01/2009	0	3	F3	1	1	1	0
R	48	26/01/2009	L10	LUGO	1. Baja	29/01/2009	0	1	F3	1	1	1	0
R	49	27/01/2009	L05	LUGO	1. Baja	30/01/2009	0	1	F3	1	1	1	0
R	50	27/01/2009	L11	LUGO	1. Baja	30/01/2009	0	1	0	1	1	1	1

Figura 31. Detalle de la hoja Transferir, donde se aprecia la lista de incidencias y la tabla que define las restricciones de orden

- E.3. Mediante una macro se realizarán todas las permutaciones, de la secuencia de horas de los cuatro días, teniendo en cuenta las restricciones de orden y se observará si alguna de ellas coincide con las que figuran en la tabla de mejoras.
- E.4. En caso de no haber posibilidad de mejora se llevaría la secuencia de Reparto a OptimizarH y se ejecutaría esta fase. En cambio, si existiese una permutación que mejorase la secuencia se ejecutaría la fase Reordenar. Además de la secuencia de horas propuesta, se calcula para cada una de ellas la secuencia de incidencias resueltas por días. Estas dos secuencias no coincidirán para el caso de haberse incluido algún preventivo en las rutas propuestas, recordemos que los preventivos tienen un tiempo de operación de 2 horas, en lugar de 1 hora, que es el que correspondería para el caso de ser correctivo. Será esta secuencia de número de órdenes de trabajo realizadas las que será base de la siguiente etapa.

MEJORAS H		MEJORAS D1		MEJORAS D2		MEJORAS D3		MEJORAS D4		ENTRADA		ENTRADA		ENTRADA		ENTRADA		ENTRADA		ENTRADA		ENTRADA		ENTRADA		ENTRADA		ENTRADA		ENTRADA	
1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111
1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111
1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111
1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111	1111

Figura 32. Detalle de la hoja Transferir, donde se aprecia la generación de las permutaciones posibles

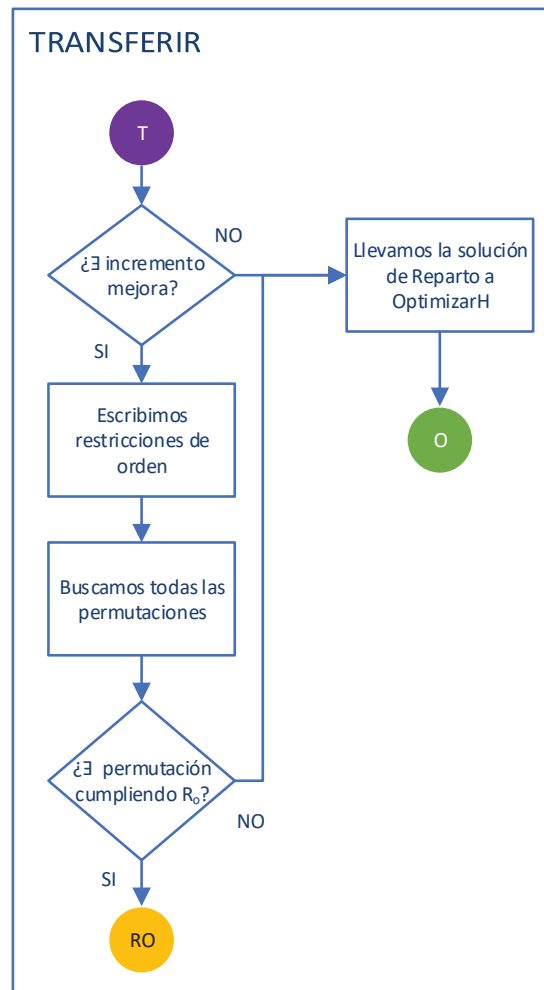


Figura 33. Diagrama de flujo de la fase Transferir del algoritmo con reordena

F) Fase Reordena

Al haberse encontrado, en Transferir, al menos una alternativa a la secuencia obtenida en Reparto y cumpliendo estas alternativas con las restricciones de orden R_o , se pretende averiguar si alguna de estas propuestas puede cumplir con las restricciones de tiempo R_t .

Se intentará que, mediante Solver, se encajen las incidencias en alguna de las propuestas, de modo que se cumplan las dos restricciones.

Las etapas de Reordena son:

- F.1. Traemos la lista de incidencias desde transferir, L_1 .
- F.2. Traemos las secuencias de número de órdenes de trabajo propuestas en Transferir, que están ordenadas de mayor a menor incremento de mejora.

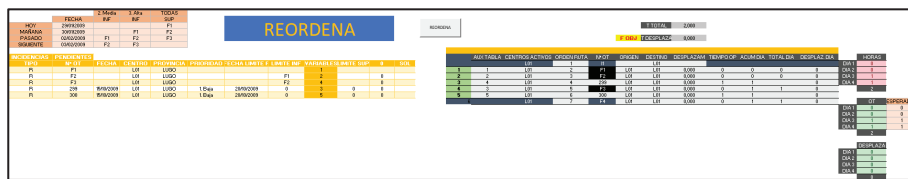


Figura 34. Detalle de la hoja Reordena, donde se aprecian las listas L_1 y la tabla de cálculo

- F.3. Creamos un bucle que recorrerá cada una de las secuencias propuestas.
- F.4. Se crea la tabla base del cálculo, se programa y ejecuta Solver. La tabla tiene una estructura y funcionamiento similar a la de Reparto. Para que se cumpla la secuencia de número de órdenes de trabajo es necesario añadirle nuevas restricciones.

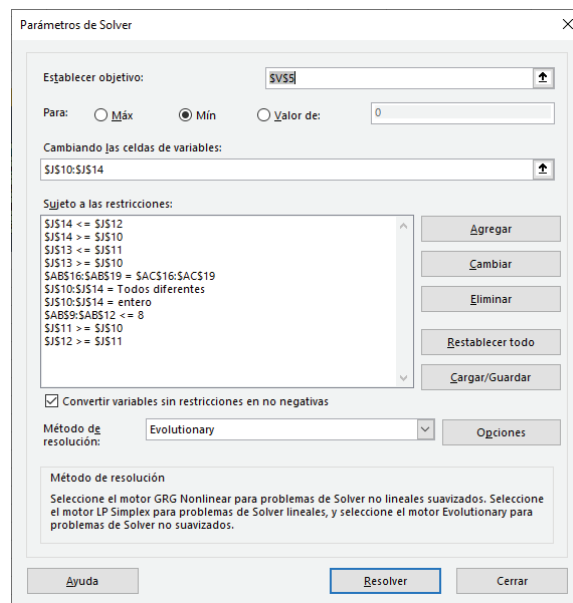


Figura 35. Detalle de los parámetros de Solver en la fase Reordena

En unas celdas, mediante fórmulas, se calcula cuantas incidencias se ejecutarían al darse la solución y a su derecha figura la secuencia de número de órdenes de trabajo que deberían resultar. La nueva restricción obligará a igualarse ambos rangos de celdas. La función objetivo es minimizar los tiempos de desplazamiento.

- F.5. Si, mientras se ejecuta el bucle, Solver encuentra una solución factible la enviaría a la hoja OptimizarH y ejecutaría dicha fase.
- F.6. Una vez finalizado en bucle sin una solución factible, se envía la propuesta de Reparto a OptimizarH y se ejecuta esa fase.

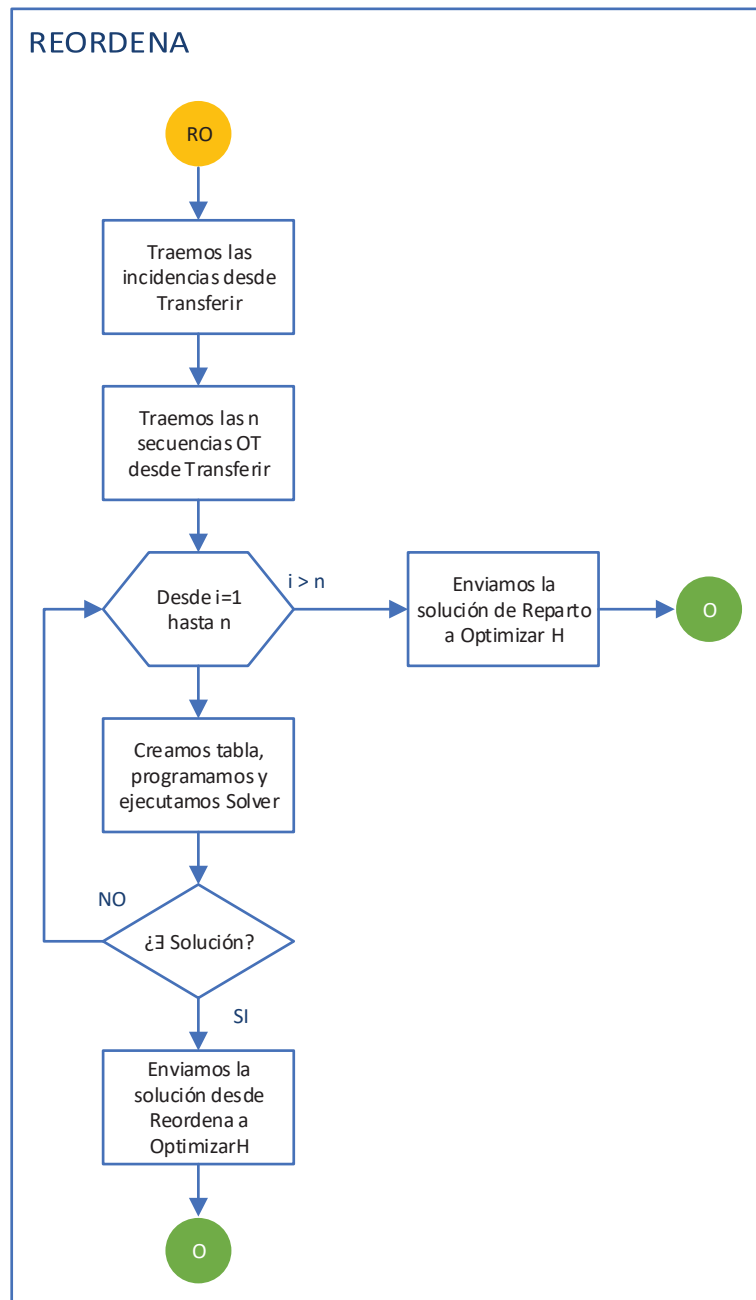


Figura 36. Diagrama de flujo de la fase Reordena del algoritmo con reordena

G) Fase OptimizarH

Esta fase es exactamente igual a su análoga del algoritmo anterior, siguiendo las siguientes etapas:

- G.1. Se importa, a la hoja OPTIMIZARH, la fecha de hoy, H .
- G.2. Si existen incidencias en L_0 , pero no en L_2 , no podría mejorarse la ruta ya que no existen mantenimientos preventivos pendientes de realizar, por lo que se vería si $T_1 \geq h$, es decir, el tiempo empleado en la ruta propuesta en Reparto es mayor o igual a h , que es el número de horas mínimo por ruta para registrarse. En caso afirmativo se registra la ruta no mejorada y en caso negativo se pasaría a la fase Resolución para pasar de día y continuar.
- G.3. Habiendo incidencias en L_0 y L_2 , si sucediese que la ruta propuesta por Reparto superase las 6 horas, $T_1 > 6$, ésta no se podría mejorar ya que no tiene capacidad para asumir 2 horas, resultantes de ejecutar un preventivo, por lo que se grabaría la ruta que se ha importado y se pasaría a la fase Resolución.
- G.4. En caso de poderse mejorar la ruta importada, se crearía una tabla similar a la de Reparto. En esta tabla se ubicarán los datos de las incidencias siguiendo el orden propuesto. Se dejarán filas libres numeradas entre las líneas fijas $I1$, las incidencias de la secuencia y $F1$. Entre las filas fijas se dejarán tantas filas libres como resulte el valor del redondeo a menos, sin decimales, de la mitad del resultado de restarle a 8 el valor redondeado a más de T_1 . Estas filas podrán ser ocupadas o no por órdenes de trabajo de preventivo.
 Al programarse Solver, tenemos en cuenta que las variables, asociadas a la lista de incidencias preventivas L_2 , deben de ser ampliada ya que tenemos más huecos en la tabla que órdenes de trabajo se pudiesen ejecutar sin superar las 8 horas de jornada.
 Se han estudiado varias funciones objetivo puesto que, llegados a este punto, sería deseable que cada operario ejecutase el mayor número de horas posible, pero minimizando los desplazamientos. Los datos estudiados serán expuestos en el capítulo 5.
- G.5. Si sucediese que la propuesta obtenida por Solver superase las 8 horas, $T_0 > 8$, al no cumplirse la restricción de tiempo R_t no se podría registrar, por lo que analizaríamos si $T_1 > h$, en caso afirmativo se registraría la ruta mejorada por OptimizarH y en caso negativo se ejecutaría Resolución para continuar con el algoritmo.

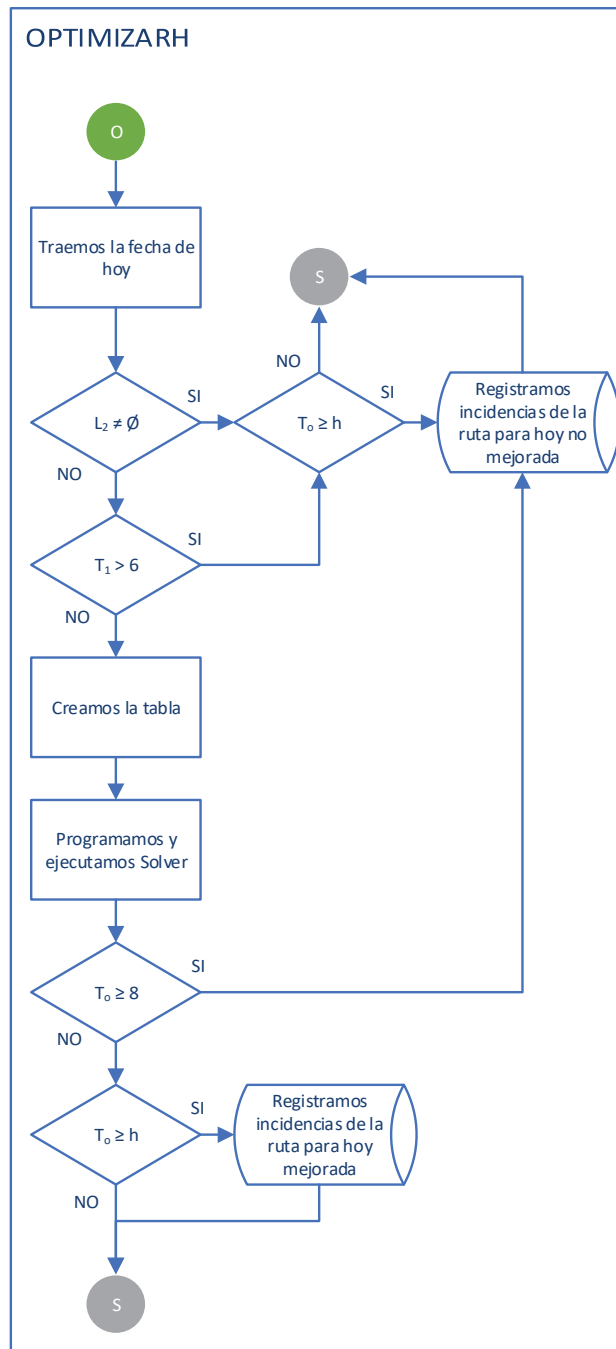


Figura 37. Diagrama de flujo de la fase OptimizarH del algoritmo con reordena

4.4. Algoritmo con reordena y predicciones

En este método se incluye una nueva lista de órdenes de trabajo, en la que figurarán las incidencias que serán creadas en los días posteriores a hoy, dentro del horizonte temporal. Con este planteamiento se quiere estudiar como influiría en la optimización de rutas si contáramos con un software fiable para predecir incidencias futuras.

El algoritmo propuesto consta de las mismas fases que en el descrito en el algoritmo anterior.

Se procede a explicar con detalle cada una de las fases:

A) Fase Resolución

La fase resolución es la que inicia y finaliza el algoritmo, contando con las siguientes etapas:

- A.1. Se inicia el algoritmo cuya fecha de hoy es un día anterior a la de inicio de la base de datos.
- A.2. Se indicará el valor $h = 4,5 \text{ horas}$, por debajo del cual el tiempo total de una ruta no será registrado, como hemos visto en el apartado 3.7.
- A.3. Se define la fecha final F_f , que coincide con la última fecha de la base de datos.
- A.4. En caso de que $F_f \geq M$, la fecha que figura para mañana pasará a la fase Estudio.
- A.5. Si $F_f < M$ se finalizará el algoritmo.

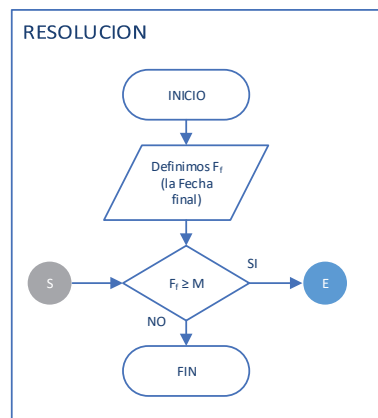


Figura 38. Diagrama de flujo de la fase Resolución del algoritmo con reordena y predicciones

B) Fase Estudio

Durante este proceso se avanzará en el calendario y se buscarán las incidencias susceptibles de ejecución. Sus etapas son como sigue:

- B.1. Se pasará de día, es decir H tomará el valor de M y en la hoja CALENDARIO se buscarán los valores de los siguientes 3 días laborales, para definir M , P y S .
- B.2. Se realizará una lista, buscando en la hoja LUGOINCIDENCIAS, con aquellas incidencias con $F_c \leq H$ y $F_r \leq S$, a la que denominaremos L_1 , listado de órdenes de trabajo a ejecutar. También se buscarán en LUGOINCIDENCIAS las órdenes de trabajo que $F_c \leq H$ y $F_r > S$, que serán las órdenes de trabajo de preventivo que no haya que realizar dentro de nuestro horizonte temporal. A ésta la llamaremos L_2 , listado de órdenes de trabajo de preventivo.
- B.3. Se buscará una tercera lista que se definirá como L_3 , listado de predicciones de órdenes de trabajo, que son aquellas incidencias que, un hipotético software podría estimar que se generarán a futuro, que no serán otras que las órdenes de trabajo que serán creadas con fecha posterior a hoy, pero dentro de nuestro horizonte temporal, con prioridades alta y media si son creadas mañana o pasado y prioridad alta en el caso de crearse en siguiente. A estas incidencias las marcaremos con una P de predicción.

El motivo de elegir estas órdenes de trabajo se justifica con la explicación de la Tabla 1: En el día de hoy podemos encontrarnos con incidencias que pueden tener como último día de resolución hoy, mañana, pasado o siguiente, identificadas por el número de días que quedaría de plazo. Mañana pueden aparecer incidencias con plazos de 3 y 24 horas, se omiten las que puedan aparecer con plazo de 3 días ya que están fuera del horizonte temporal. Lo mismo sucede con pasado, en cambio para el día siguiente solo tendremos en cuenta las incidencias con plazo de resolución de 3 horas.

Las incidencias que figuran en L_3 deberán ser ejecutadas dentro del horizonte temporal.

Tabla 1. Fechas máximas de resolución

		Último día resolución			
		H	M	P	S
Días	H	0	1	2	3
	M		0	1	
	P			0	1
	S				0

B.4. Finalmente se dará orden para que se ejecuten las macros del módulo Reparto.

The screenshot shows an Excel spreadsheet with several data tables. The tables contain columns for 'ID', 'FECHA', 'TIPO', 'ESTADO', 'RESOLUCION', and 'LISA'. The data is organized into three distinct sections, corresponding to lists L1, L2, and L3 mentioned in the caption.

Figura 39. Detalle de la hoja Estudio, donde se aprecian las listas L_1 , L_2 y L_3

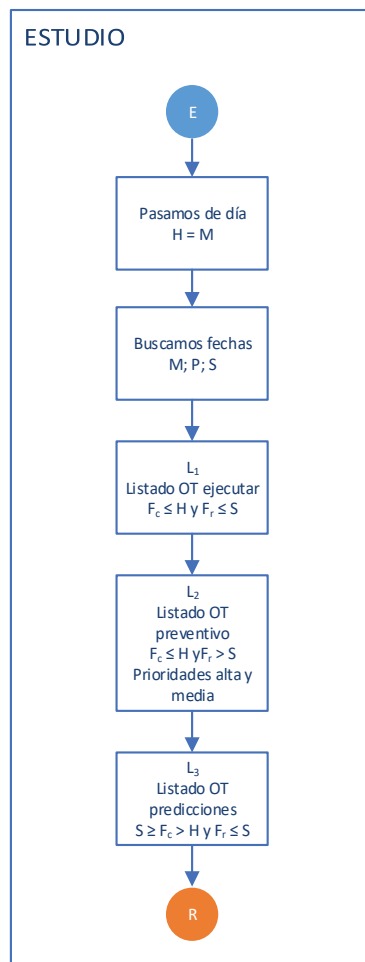


Figura 40. Diagrama de flujo de la fase Estudio del algoritmo con reordena y predicciones

C) Fase Reparto

En esta fase se pretende ver si con los recursos disponibles seríamos capaces de realizar las incidencias de las listas L_1 y L_3 , sujetos a las restricciones R_t y R_o .

Esta fase se divide en las siguientes etapas:

- C.1. Traeremos a la hoja REPARTO las fechas H , M , P y S , además de las listas L_1 y L_3 , que figuran en la hoja ESTUDIO.
- C.2. Si se diese el caso que L_1 y L_2 están vacías significa que no tenemos ninguna incidencia que podamos ejecutar hoy, por lo que volveremos a la fase Resolución para avanzar de día. Nótese que, aunque pueda haber órdenes de trabajo en L_3 , al ser predicciones aún no se han creado y no podrán ejecutarse hoy.
- C.3. Si sucediese que L_1 está vacía, pero si hay incidencias en L_2 , enviaríamos L_2 a la hoja OPTIMIZARH y ejecutaríamos dicha fase, como se hizo en la fase C.3 de los algoritmos anteriores.
- C.4. La siguiente posibilidad que se plantea es que L_1 no esté vacía, por lo que será pegada junto a L_3 en la hoja, de modo similar a lo expuesto en el punto C.4 de los algoritmos anteriores.
- C.5. Se creará una tabla base del cálculo, de igual modo que la explicada en el punto C.5 de los algoritmos anteriores.
- C.6. Se programará el Solver, definiéndose la función objetivo, T_d tiempo de desplazamiento, que se pretende minimizar. Las variables que son las celdas de la columna variable de la lista, como hemos visto. Se añaden las restricciones de tiempo y de orden, así como que las variables han de ser números enteros y distintos. Finalmente se indica que el método de resolución es el Evolucionista.
- C.7. Se ejecutará el Solver
- C.8. Si no se cumpliesen las restricciones de orden y/o las de tiempo en la solución propuesta se ejecutaría la fase Externalización. En cambio, si se cumpliesen ambas restricciones, llevaríamos, a la hoja TRANSFERIR, tanto las incidencias de las tablas, como la secuencia horas trabajadas para cada uno de los días y se iniciaría la fase Transferir.

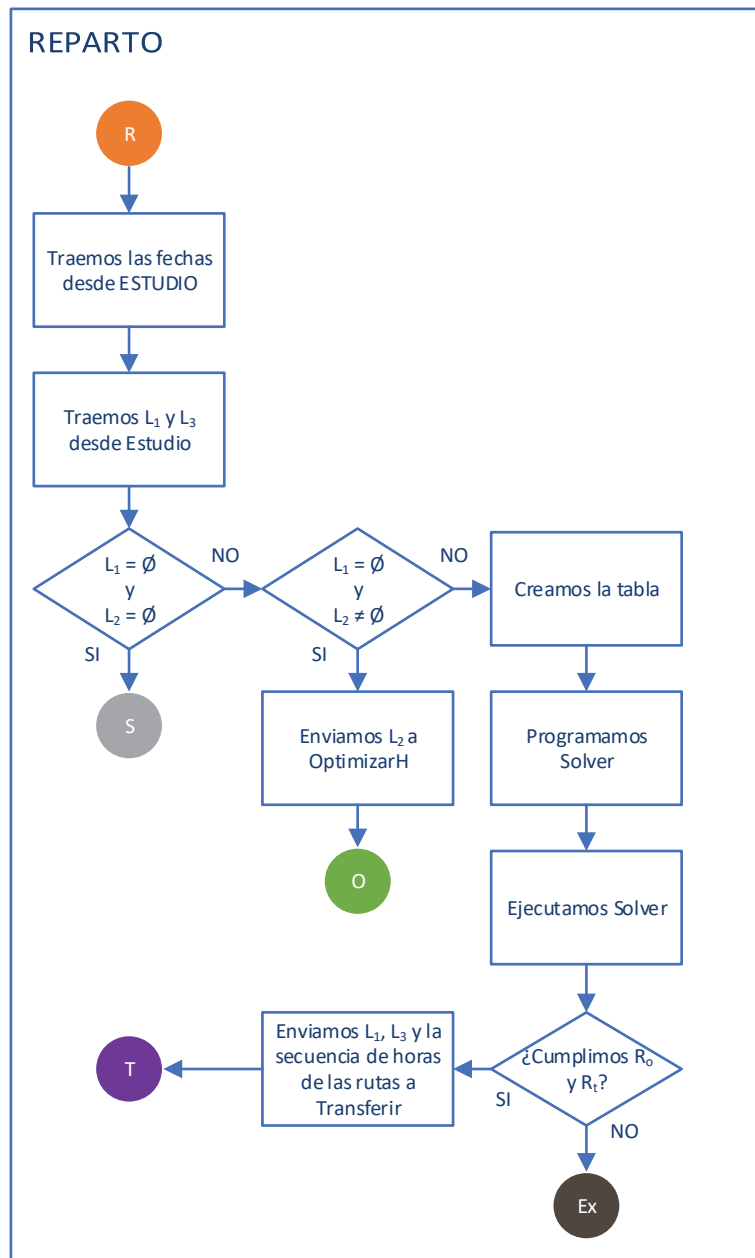


Figura 41. Diagrama de flujo de la fase Reparto del algoritmo con reordena y predicciones

D) Fase Externalizar

Esta fase es exactamente igual a su análoga del algoritmo anterior.

Las etapas de esta fase son:

- D.1. Se lleva a la hoja EXTERNALIZAR las incidencias que figuran en REPARTO, colocándolas bajo las hipotéticas órdenes de trabajo F_i .
- D.2. Se crea la tabla de modo similar a la creada en la etapa Reparto.
- D.3. Se programa el Solver, con las mismas restricciones que en Reparto, de este modo todas las incidencias reales son libres de poder ser externalizadas y se cumplirán las restricciones de orden. La función objetivo será minimizar el tiempo total de la ruta que finaliza en F_0 , de modo que será mínimo el coste de las tareas a externalizar.
- D.4. Se ejecuta el Solver.
- D.5. Se registran las tareas que finalizan en F_0 , como externalizadas.
- D.6. Se enviará la lista de las incidencias no externalizadas a Reparto y se ejecutaría esa fase.

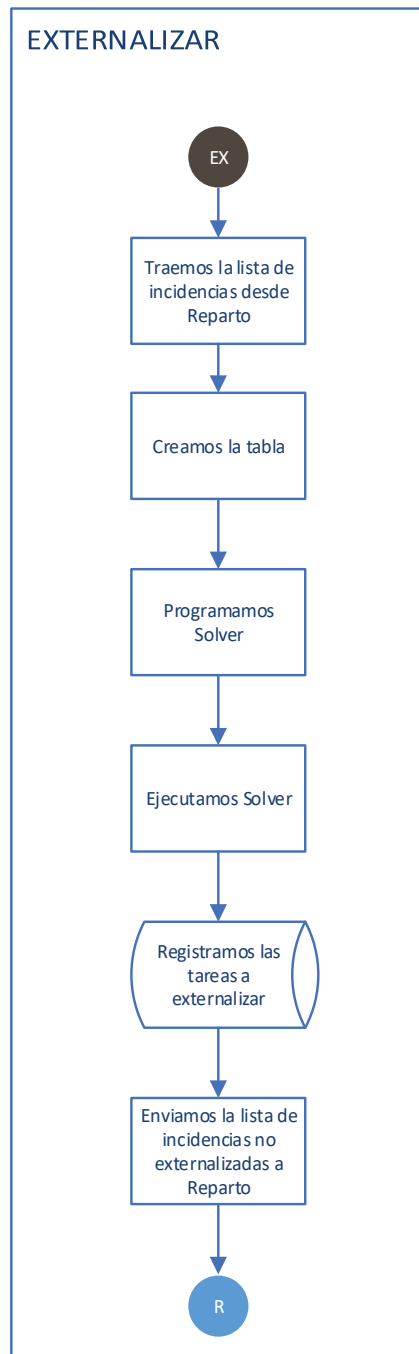


Figura 42. Diagrama de flujo de la fase Externalizar del algoritmo con reordena y predicciones

E) Fase Transferir

Esta fase es exactamente igual a su análoga expuesta en el algoritmo anterior.

Será objeto de estudio independiente, para cada función objetivo que se estudie en el capítulo 5, la posibilidad de redondeo a más y a menos en la secuencia de horas propuestas por Reparto.

La fase Transferir presenta las siguientes etapas:

- E.1. Si no existiese ninguna permutación que mejorase la secuencia propuesta de Reparto, se llevaría su secuencia a OptimizarH y se ejecutaría esta fase.
- E.2. Si al menos existiese una permutación que pudiese mejorar la secuencia, en la lista de incidencias se escribirían las ecuaciones que determinarían las restricciones de orden, de forma análoga a lo expuesto en el punto E.2 del algoritmo anterior. En cada línea figurará el tiempo de operación asignado a cada uno de los días en los que pueda ejecutarse.
- E.3. Mediante una macro se realizarán todas las permutaciones de la secuencia de horas de los cuatro días, teniendo en cuenta las restricciones de orden y se observará si alguna de ellas coincide con las que figuran en la tabla de mejoras.
- E.4. En caso de no haber posibilidad de mejora se llevaría la secuencia de Reparto a OptimizarH y se ejecutaría esta fase.
Si existiese una permutación que mejorase la secuencia se ejecutaría la fase Reordenar. Además de la secuencia de horas propuesta, se calcula para cada una de ellas la secuencia de incidencias resueltas por días. Como hemos visto, estas dos secuencias no coincidirán para el caso de haberse incluido en las rutas algún preventivo, que tiene un tiempo de operación de 2 horas.

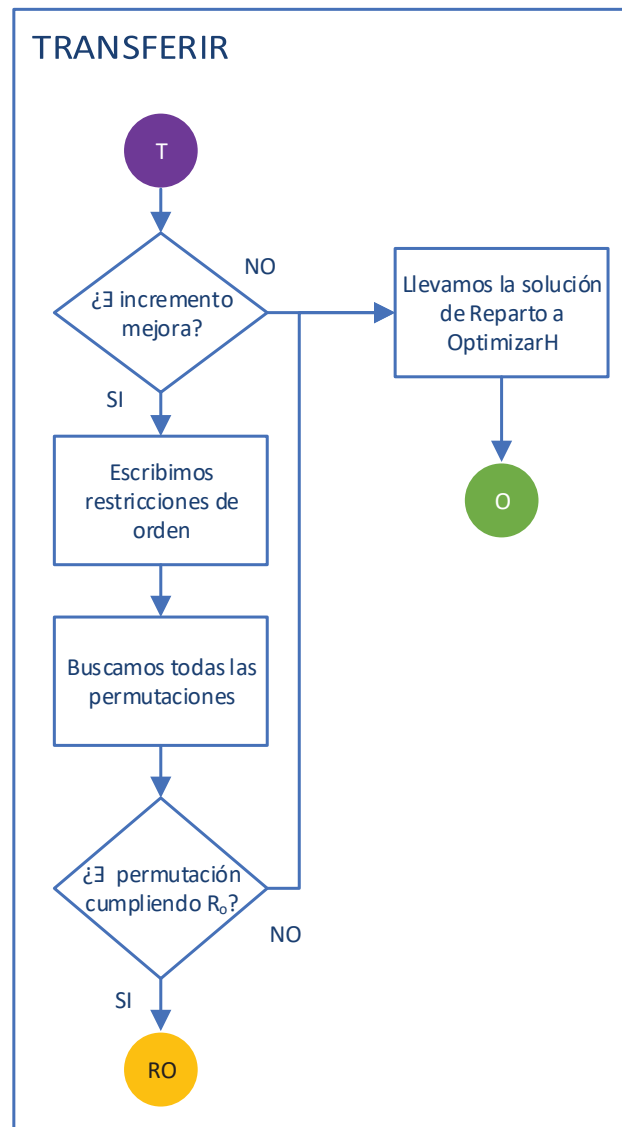


Figura 43. Diagrama de flujo de la fase Transferir del algoritmo con reordena y predicciones

F) Fase Reordena

La etapa Reordena es conceptualmente igual a la misma fase del algoritmo anterior, la única diferencia es que en la lista de incidencias aparecerán las listas L_1 y L_3 .

Las etapas de Reordena son:

- F.1. Traemos la lista de incidencias desde transferir, L_1 y L_3 .
- F.2. Traemos las secuencias de número de órdenes de trabajo propuestas en Transferir, que están ordenadas de mayor a menor incremento de mejora.
- F.3. Creamos un bucle que recorrerá cada una de las secuencias propuestas.
- F.4. Se crea la tabla base del cálculo, se programa y ejecuta Solver. La tabla tiene una estructura y funcionamiento similar a la de Reparto. Para que se cumpla la secuencia de número de órdenes de trabajo es necesario añadirle nuevas restricciones. En unas celdas, mediante fórmulas, se calcula cuantas incidencias se ejecutarían en la tabla y a su derecha figura la secuencia de número de órdenes de trabajo. La nueva restricción obligará a igualarse ambos rangos de celdas. La función objetivo es minimizar los tiempos de desplazamiento.
- F.5. Si, mientras se ejecuta el bucle, Solver encuentra una solución factible la enviaría a la hoja OptimizarH y ejecutaría dicha fase.
- F.6. Una vez finalizado en bucle sin una solución factible, se envía la propuesta de Reparto o Externalizar a OptimizarH y se ejecuta esta fase.

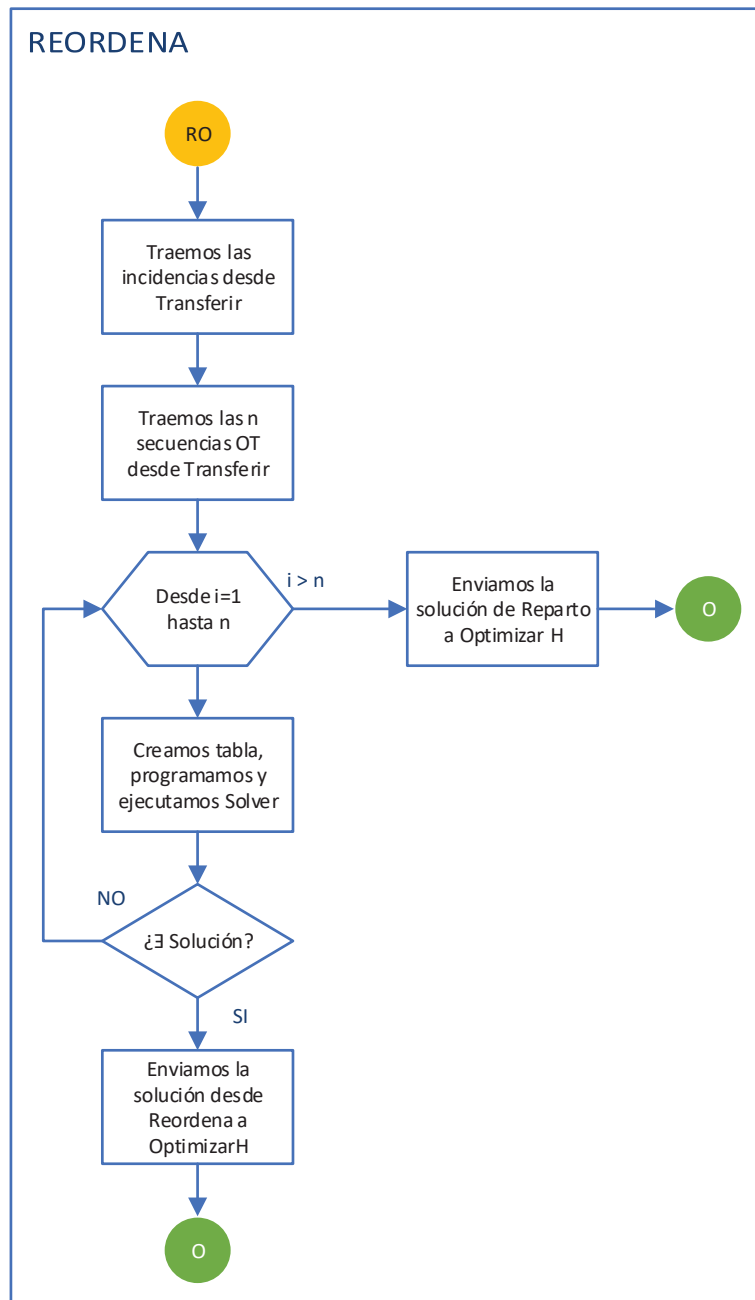


Figura 44. Diagrama de flujo de la fase Reordena del algoritmo con reordena y predicciones

G) Fase OptimizarH

Esta fase es exactamente igual a sus análogas de los algoritmos anteriores, siguiendo las siguientes etapas:

- G.1. Se importa, a la hoja OPTIMIZARH, la fecha de hoy, H .
- G.2. Si existen incidencias en L_0 , pero no en L_2 , no podría mejorarse la ruta ya que no existen mantenimientos preventivos pendientes de realizar, por lo que se vería si $T_1 \geq h$, es decir, el tiempo empleado en la ruta propuesta en Reparto es mayor o igual a h , que es el número de horas mínimo por ruta para registrarse. En caso afirmativo se registra la ruta no mejorada y en caso negativo se pasaría a la fase Resolución para pasar de día y continuar.
- G.3. Habiendo incidencias en L_0 y L_2 , si sucediese que la ruta propuesta por Reparto superase las 6 horas, $T_1 > 6$, ésta no se podría mejorar ya que no tiene capacidad para asumir 2 horas, resultantes de ejecutar un preventivo, por lo que se grabaría la ruta que se ha importado y se pasaría a la fase Resolución.
- G.4. En caso de poderse mejorar la ruta importada, se crearía una tabla similar a la de Reparto. En esta tabla se ubicarán los datos de las incidencias siguiendo el orden propuesto. Se dejarán filas libres numeradas entre las líneas fijas $I1$, las incidencias de la secuencia y $F1$. Entre las filas fijas se dejarán tantas filas libres como resulte el valor del redondeo a menos, sin decimales, de la mitad del resultado de restarle a 8 el valor redondeado a más de T_1 . Estas filas podrán ser ocupadas o no por órdenes de trabajo de preventivo. Al programarse Solver, tenemos en cuenta que las variables, asociadas a la lista de incidencias preventivas L_2 , deben de ser ampliada ya que tenemos más huecos en la tabla que órdenes de trabajo se pudiesen ejecutar sin superar las 8 horas de jornada. Se han estudiado varias funciones objetivo puesto que, llegados a este punto, sería deseable que cada operario ejecutase el mayor número de horas posible, pero minimizando los desplazamientos. Los datos estudiados serán expuestos en el capítulo 5.
- G.5. Si sucediese que la propuesta obtenida por Solver superase las 8 horas, $T_0 > 8$, al no cumplirse la restricción de tiempo R_t no se podría registrar, por lo que analizaríamos si $T_1 > h$, en caso afirmativo se registraría la ruta mejorada por OptimizarH y en caso negativo se ejecutaría Resolución para continuar con el algoritmo.

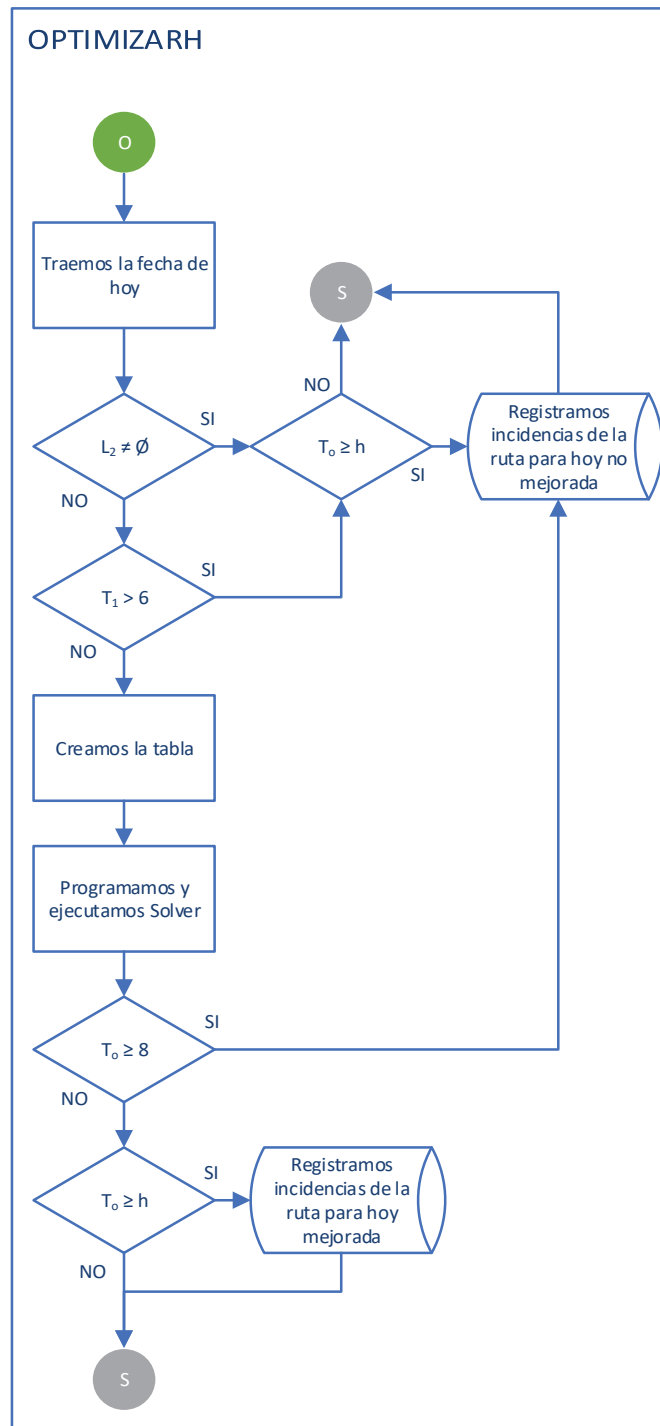


Figura 45. Diagrama de flujo de la fase OptimizarH del algoritmo con reordena y predicciones

5. EXPERIMENTACIÓN Y RESULTADOS

Como se ha mencionado en el capítulo anterior, se han utilizado distintas funciones objetivo en la fase OptimizarH, para los algoritmos basados en el complemento Solver de Excel. Estas funciones objetivo deberían de maximizar el número de incidencias a realizar, pero minimizando los desplazamientos.

5.1. Experimentación

Las dos primeras funciones objetivos más evidentes son:

$$\text{Max } \frac{N}{T_d} \quad (86)$$

$$\text{Max } \frac{T_o}{T_d} \quad (87)$$

Donde N es el número de órdenes de trabajo ejecutadas, T_d son los tiempos de desplazamiento y T_o son los tiempos de operación, necesarios para realizar las incidencias.

Dado que intuitivamente parece que debemos de primar la cantidad de órdenes de trabajo a realizar sobre la penalización de los desplazamientos, ya que por muy lejano que se encuentre un centro en el que haya una incidencia nos interesa ejecutarla, se plantean las siguientes funciones objetivo:

$$\text{Max } \frac{N T_o}{T_d} \quad (88)$$

$$\text{Max } \frac{10^{N T_o}}{T_d} \quad (89)$$

$$\text{Max } \frac{10^N}{T_d} \quad (90)$$

Finalmente, como comparación, se planteó el estudio de la siguiente función objetivo, en la que no se penalizan los desplazamientos:

$$\text{Max } N \quad (91)$$

Se comentó en el capítulo anterior, que en la fase Transferir, del algoritmo con reordena y del algoritmo con reordena y predicciones, tenían dos variantes, en función del redondeo a más y a menos en la secuencia de horas propuestas por las fases Reparto.

Los métodos estudiados son:

Tabla 2. Denominación de los métodos estudiados

Nomenclatura	Método
AB-k-NN	4.1. Algoritmo basado en vecino más cercano
SIMP	4.2. Algoritmo simple
REOR +	4.3. Algoritmo con reordena, en el que el redondeo es a más
REOR -	4.3. Algoritmo con reordena, en el que el redondeo es a menos
PRED +	4.4. Algoritmo con reordena y predicciones, en el que el redondeo es a más
PRED -	4.4. Algoritmo con reordena y predicciones, en el que el redondeo es a menos

Al ejecutarse en más de una ocasión los algoritmos que utilizan Solver, se observó que los resultados eran distintos, por lo que se decidió realizar diez experiencias con cada algoritmo y función objetivo. Para el caso del algoritmo de comparación, AB-k-NN, siempre se obtuvieron los mismos resultados.

5.2. Resultados

La Tabla 3 muestra la media y el error relativo de los días en que ha trabajado el recurso propio y sus tiempos de desplazamiento, además del número de horas de trabajo y los tiempos de desplazamiento que se han externalizado:

Tabla 3. Media y error relativo de los tiempos empleados

Algo- ritmo	F. obj.	Días r. propio		H. desplaz. r. propio		H. exter.		H. desplaz. exter.	
		Media	Error	Media	Error	Media	Error	Media	Error
AB-k-NN		138,00	0,00%	334,00	0,00%	36,85	0,00%	19,85	0,00%
SIMP	$\frac{10^N T_0}{T_d}$	138,90	0,52%	337,67	0,64%	47,15	9,59%	24,05	11,73%
	$\frac{10^N}{T_d}$	135,00	0,74%	328,04	1,13%	51,90	10,52%	25,30	11,81%
	$\frac{N T_0}{T_d}$	135,30	1,26%	326,67	1,25%	50,75	6,50%	23,75	5,66%
	$\frac{N}{T_d}$	134,30	1,86%	321,23	1,48%	49,48	25,61%	20,68	20,66%
	N	134,80	0,77%	374,73	0,99%	46,09	9,37%	22,89	10,85%
	$\frac{T_0}{T_d}$	133,80	2,84%	318,05	2,80%	46,69	25,87%	18,59	22,29%

Algoritmo	F. obj.	Días r. propio		H. desplaz. r. propio		H. exter.		H. desplaz. exter.	
		Media	Error	Media	Error	Media	Error	Media	Error
REOR +	$\frac{10^{N T_0}}{T_d}$	140,40	1,34%	340,11	1,50%	47,52	18,52%	22,12	14,11%
	$\frac{10^N}{T_d}$	136,40	1,17%	329,36	1,15%	45,53	11,53%	22,73	11,85%
	$\frac{N T_0}{T_d}$	135,60	1,03%	327,11	1,42%	53,32	9,31%	24,72	9,57%
	$\frac{N}{T_d}$	134,30	1,67%	318,89	1,85%	48,00	17,42%	19,70	14,23%
	N	135,10	1,11%	378,12	1,54%	45,16	10,85%	22,56	11,85%
	$\frac{T_0}{T_d}$	135,30	1,45%	320,72	2,21%	43,65	25,92%	17,75	23,72%
REOR -	$\frac{10^{N T_0}}{T_d}$	139,40	0,86%	339,68	1,19%	44,85	9,69%	22,85	11,13%
	$\frac{10^N}{T_d}$	134,80	0,89%	326,62	1,32%	50,43	8,74%	25,03	8,56%
	$\frac{N T_0}{T_d}$	135,50	0,81%	328,46	1,24%	53,22	10,41%	25,52	11,45%
	$\frac{N}{T_d}$	134,60	1,16%	320,63	1,07%	48,37	25,22%	20,97	20,97%
	N	134,40	0,74%	375,06	0,77%	47,16	9,83%	23,26	14,25%
	$\frac{T_0}{T_d}$	135,30	1,76%	322,83	1,71%	44,07	24,17%	18,97	18,43%

Algoritmo	F. obj.	Días r. propio		H. desplaz. r. propio		H. exter.		H. desplaz. exter.	
		Media	Error	Media	Error	Media	Error	Media	Error
PRED +	$\frac{10^N \tau_0}{T_d}$	141,40	1,07%	344,98	1,39%	41,25	12,78%	20,75	13,35%
	$\frac{10^N}{T_d}$	136,10	0,68%	331,57	1,28%	46,01	14,27%	22,91	10,76%
	$\frac{N T_0}{T_d}$	136,70	0,66%	333,68	1,06%	46,33	6,57%	22,33	8,68%
	$\frac{N}{T_d}$	134,70	1,46%	318,01	1,90%	41,75	28,06%	16,95	27,66%
	N	135,10	1,13%	373,91	1,07%	42,44	12,04%	21,94	11,88%
	$\frac{T_o}{T_d}$	136,30	1,35%	326,51	1,56%	32,95	38,63%	13,45	33,95%
PRED -	$\frac{10^N \tau_0}{T_d}$	140,20	0,46%	342,70	1,11%	38,43	6,36%	20,13	6,61%
	$\frac{10^N}{T_d}$	136,30	1,25%	331,86	0,96%	47,19	8,46%	23,79	9,74%
	$\frac{N T_0}{T_d}$	136,00	1,18%	328,86	1,34%	47,33	10,94%	22,53	14,10%
	$\frac{N}{T_d}$	135,30	1,15%	321,43	1,11%	39,35	20,67%	16,35	14,87%
	N	135,60	1,18%	376,32	1,41%	39,28	21,03%	19,58	21,40%
	$\frac{T_o}{T_d}$	135,10	2,28%	321,03	2,57%	40,74	31,43%	16,44	27,41%

Los errores en los tiempos y desplazamientos externalizados son considerables debido a que, al haber introducido la condición que impide que una ruta

sea realizada por debajo de las 4,5 horas de trabajo, dependiendo de las casuísticas que vayan desarrollando los algoritmos, durante su ejecución, la variabilidad de las tareas a externalizar es muy amplia.

En la Tabla 4. se muestran la media de la suma de los tiempos totales empleados, la media de la suma los tiempos de desplazamiento y la media de los costes totales, calculados al multiplicar los costes, definidos en el apartado 3.7, con cada uno de los tiempos.

Tabla 4. Media y error relativo de los tiempos totales y costes

Algoritmo	F. obj.	Horas		Horas desplaz.		Coste	
		Media	Error	Media	Error	Media	Error
AB-k-NN		967,84	0,00%	353,84	0,00%	13.949,79 €	0,00%
SIMP	$\frac{10^N T_0}{T_d}$	957,41	0,41%	361,72	0,73%	14.249,83 €	0,66%
	$\frac{10^N}{T_d}$	948,19	0,18%	353,34	0,69%	13.970,51 €	0,54%
	$\frac{N T_0}{T_d}$	945,36	0,70%	350,42	1,19%	13.965,00 €	0,79%
	$\frac{N}{T_d}$	936,45	0,35%	341,90	1,00%	13.823,73 €	1,06%
	N	996,71	0,41%	397,62	1,18%	13.961,69 €	0,79%
	$\frac{T_0}{T_d}$	929,52	1,16%	336,64	2,00%	13.709,62 €	1,45%

Algoritmo	F. obj.	Horas		Horas desplaz.		Coste	
		Media	Error	Media	Error	Media	Error
REOR +	$\frac{10^N T_0}{T_d}$	969,93	1,96%	362,23	0,64%	14.387,73 €	1,76%
	$\frac{10^N}{T_d}$	950,18	0,48%	352,09	0,82%	13.968,78 €	0,65%
	$\frac{N T_0}{T_d}$	950,46	0,33%	351,83	1,05%	14.044,86 €	0,53%
	$\frac{N}{T_d}$	933,51	0,54%	338,58	1,68%	13.785,54 €	0,58%
	N	998,04	0,47%	400,69	1,26%	13.978,93 €	0,79%
	$\frac{T_0}{T_d}$	935,96	0,48%	338,47	1,55%	13.789,98 €	0,59%
REOR -	$\frac{10^N T_0}{T_d}$	958,40	0,29%	362,53	0,82%	14.251,64 €	0,75%
	$\frac{10^N}{T_d}$	947,37	0,23%	351,64	0,97%	13.920,32 €	0,42%
	$\frac{N T_0}{T_d}$	950,56	0,59%	353,98	1,44%	14.041,41 €	0,92%
	$\frac{N}{T_d}$	938,93	0,42%	341,59	0,98%	13.829,52 €	1,08%
	N	997,39	0,39%	398,32	0,94%	13.948,46 €	0,52%
	$\frac{T_0}{T_d}$	935,72	0,37%	341,80	1,04%	13.809,05 €	0,73%

Algoritmo	F. obj.	Horas		Horas desplaz.		Coste	
		Media	Error	Media	Error	Media	Error
PRED +	$\frac{10^N T_0}{T_d}$	962,51	0,41%	365,73	0,93%	14.367,56 €	0,78%
	$\frac{10^N}{T_d}$	949,85	0,62%	354,48	1,03%	13.957,94 €	0,85%
	$\frac{N T_0}{T_d}$	952,66	0,39%	356,02	0,92%	14.020,43 €	0,56%
	$\frac{N}{T_d}$	933,45	0,55%	334,96	1,32%	13.690,44 €	0,87%
	N	993,63	0,42%	395,85	1,07%	13.914,03 €	0,69%
	$\frac{T_o}{T_d}$	936,78	0,45%	339,96	1,30%	13.676,98 €	0,91%
PRED -	$\frac{10^N T_0}{T_d}$	959,43	0,37%	362,83	1,06%	14.199,64 €	0,51%
	$\frac{10^N}{T_d}$	951,59	0,46%	355,65	0,80%	14.002,31 €	0,95%
	$\frac{N T_0}{T_d}$	950,24	0,49%	351,39	1,28%	13.964,38 €	0,98%
	$\frac{N}{T_d}$	934,33	0,38%	337,77	0,87%	13.705,96 €	0,68%
	N	992,85	1,16%	395,90	1,83%	13.895,98 €	1,43%
	$\frac{T_o}{T_d}$	934,45	0,60%	337,47	1,62%	13.713,33 €	0,95%

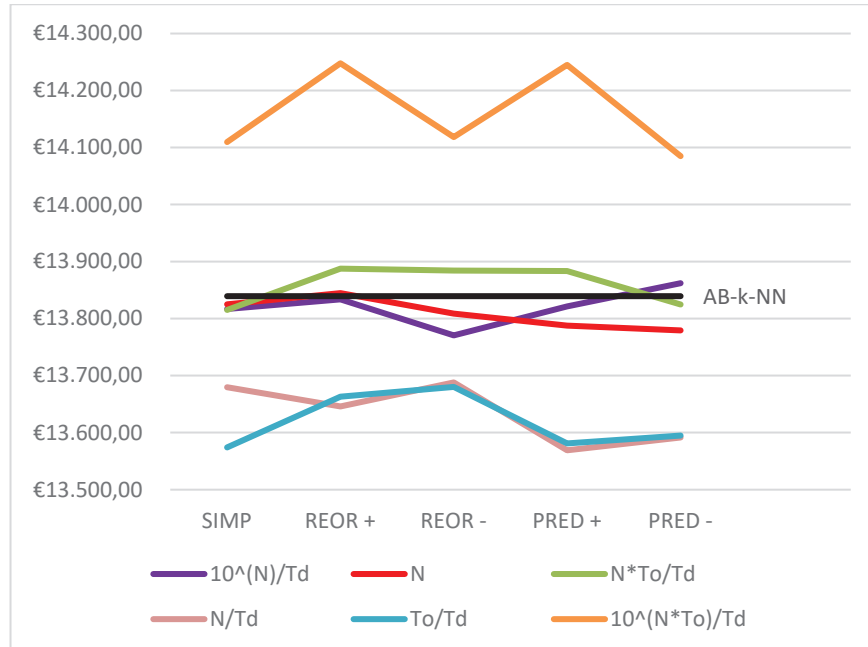
Tanto la media de tiempos como la media de los costes no muestran errores importantes. Los mencionados errores medios de los tiempos de externalización carecen de importancia ya que no influyen prácticamente en los errores de costes, que es la variable que se pretende minimizar.

6. CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

Habiéndose obtenido unos datos fiables en la experimentación, se procede a analizar, en el primer apartado de este capítulo, tanto los métodos como las funciones objetivo empleados y finalmente evaluar si se han conseguido los objetivos planteados. En un segundo apartado se exponen distintas líneas para futuras investigaciones.

6.1. Conclusiones

La Gráfica 1 muestra los importes medios que se obtienen, utilizando las distintas funciones objetivo, en relación con los distintos métodos. Se ha querido representar el importe obtenido por el método AB-k-NN, que servirá de base de comparación.



Gráfica 1. Comportamiento de las funciones objetivo según el método utilizado

Es evidente que solamente las funciones objetivo (86) y (87) mejoran los resultados de AB-k-NN.

La media de los importes medios de las funciones objetivo sobre los distintos métodos figura en la Tabla 5, ordenados de mayor a menor importe. Podemos observar como las funciones objetivo (88), (89) y (90), que pretendían favorecer la ejecución de incidencias sobre los tiempos de desplazamiento, presentan los peores resultados. Finalmente, la función objetivo (91), que maximiza la resolución de órdenes de trabajo sin tener en cuenta los tiempos de desplazamiento se sitúa como tercera mejor opción.

Tabla 5. Media de importes medios de las funciones objetivo

F. objetivo	Media	Error
$\frac{10^N T_0}{T_d}$	14.291,28 €	0,48%
$\frac{N T_0}{T_d}$	14.007,22 €	0,24%
$\frac{10^N}{T_d}$	13.963,97 €	0,14%
N	13.939,82 €	0,20%
$\frac{N}{T_d}$	13.767,04 €	0,40%
$\frac{T_0}{T_d}$	13.739,79 €	0,35%

La Gráfica 2 muestra los importes medios que se obtienen en los distintos métodos, según la función objetivo que utilicen.



Gráfica 2. Comportamiento de los métodos utilizados según la función objetivo

La Tabla 6 presenta, ordenados de mayor a menor importe, los costes en función del método y función objetivo utilizados, además del incremento porcentual respecto al mejor resultado.

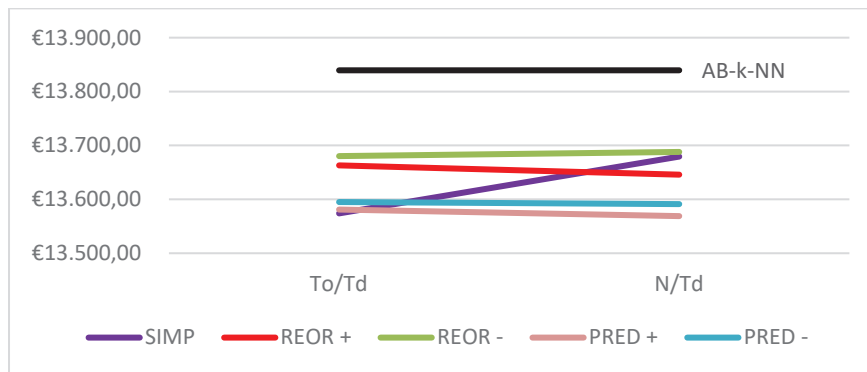
Tabla 6. Relación de costes obtenidos según el método y la función objetivo utilizados

Método	F. obj.	Importe	Incremento
AB-k-NN		13.949,79 €	1,99%
REOR -	$\frac{N}{T_d}$	13.829,52 €	1,12%

Método	F. obj.	Importe	Incremento
SIMP	$\frac{N}{T_d}$	13.823,73 €	1,07%
REOR -	$\frac{T_o}{T_d}$	13.809,05 €	0,97%
REOR +	$\frac{N}{T_d}$	13.785,54 €	0,79%
PRED -	$\frac{T_o}{T_d}$	13.713,33 €	0,27%
SIMP	$\frac{T_o}{T_d}$	13.709,62 €	0,24%
PRED -	$\frac{N}{T_d}$	13.705,96 €	0,21%
PRED +	$\frac{N}{T_d}$	13.690,44 €	0,10%
PRED +	$\frac{T_o}{T_d}$	13.676,98 €	0,00%

La diferencia de costes anuales presentados en la Tabla 6 no tiene gran relevancia.

Si nos centramos solamente en las dos funciones objetivo, según se observó en la Gráfica 1, que mejoran los resultados del algoritmo de comparación, AB-k-NN, la Gráfica 3 muestra los costes medios de los métodos, según dichas funciones objetivo.



Gráfica 3. Resultados de los métodos con las dos mejores funciones objetivo

La siguiente tabla muestra, ordenada de mayor a menor importe, los costes medios de los distintos métodos, según las funciones objetivos (86) y (87), que presentan los mejores resultados.

Tabla 7. Resultados de los métodos con las dos mejores funciones objetivo

Método	Importe medio	Error
AB-k-NN	13.949,79 €	0,00%
REOR -	13.819,28 €	0,07%
REOR +	13.787,76 €	0,02%
SIMP	13.766,68 €	0,41%
PRED -	13.709,65 €	0,03%
PRED +	13.683,71 €	0,05%

Podemos indicar que los métodos basados en el Solver tienen una pequeña pendiente en la Gráfica 3, con excepción del método Simple.

Queda demostrado que la mejor función objetivo a maximizar es $\frac{T_o}{T_d}$, seguida muy de cerca por $\frac{N}{T_d}$.

En cuanto a los métodos, el de mejor resultado es PRED + seguido, a poca distancia por PRED -.

Los dos métodos REOR – y REOR +, que en la fase de diseño de los algoritmos presentaban grandes esperanzas, han obtenido los peores resultados, aunque mejoran los del algoritmo de comparación.

Se concluye, con todo lo expuesto, que se han alcanzado los objetivos planteados en el capítulo 1.

Ha de mencionarse que los tiempos medios de ejecución del algoritmo más complejo, Algoritmo con reordena y predicciones, son razonables, ejecutado sobre un ordenador convencional, con procesador Intel Core i5 y 16 GB de memoria RAM, no supera el minuto de cálculo diario.

6.2. Futuras líneas de investigación

En el capítulo 1 se mencionó que las empresas mantenedoras dividían las superficies geográficas en zonas. Nuestro planteamiento se redujo a una zona, que disponía de un solo equipo de trabajo, como complemento a este estudio, se podrían adaptar los métodos planteados a otras zonas con varios equipos asignados.

Es probable que, si se hace un estudio global, en el que se definan distintas bases, cada una con sus recursos, sin que se asignen zonas, se puedan mejorar los resultados.

Aplicando las ideas aquí planteadas, podrían desarrollarse unas soluciones que puedan determinar la cantidad de recursos que se tienen que asignar a las bases de los contratos y/o qué centros resultan más rentable que sean elegidos como bases. Estos posibles estudios resultarían de gran utilidad para que las empresas de mantenimiento puedan dimensionar sus recursos.

Para el desarrollo del estudio se ha supuesto un tiempo mínimo para registrar las rutas resultantes, dada la rentabilidad de su subcontratación. Sería muy interesante afinar este tiempo mínimo utilizando métodos iterativos de aproximación y establecer un procedimiento para su aplicación en otras bases de datos, convenios colectivos, márgenes brutos de las subcontratas y costes de furgoneta y de gasoil.

Como hemos visto en el capítulo 5, se han planteado varias funciones objetivo. Los resultados obtenidos son posiblemente mejorables con otras funciones a plantear.

La mejora que presenta el tercer algoritmo, con la que se intenta reordenar las incidencias en las secuencias de tareas, aunque no dio los resultados esperados, podría servir para mejorar otros planteamientos.

El estudio se ha realizado sobre la base de datos disponible, que cuenta con unas concretas incidencias en un periodo de tiempo. Sería interesante disponer de otras bases de datos para analizar los comportamientos de los algoritmos y las funciones objetivo planteadas, para ver si es posible dar una solución a este tipo de contratos con otras bases de datos y otros costes unitarios.

Si se dispusiese de los tiempos reales de operación, se podría acercarse más a la realidad el estudio presentado. Puesto que, en la realidad, dichos tiempos son estocásticos, que dependen de los mantenimientos a realizar, en distintos equipos, en distintos centros, que presentan limitaciones de acceso. Los algoritmos deberían ser dinámicos, de modo que, una vez finalizada una incidencia, en función de las tareas pendientes, la ubicación del operario y la jornada laboral restante, vaya planificando las rutas.

Como complemento a la idea anterior, sería de gran interés la implementación de las pérdidas de tiempo útil de los equipos de trabajo: compras, aparcamiento, absentismos, disponibilidad de acceso a los equipos a mantener, etcétera.

Ya se mencionó en el apartado 3.2, que los convenios colectivos permiten que se realicen horas extras, estableciendo unas horas máximas anuales. Esta circunstancia también puede ser tomada en cuenta en futuros métodos de resolución, que deberán contemplar cuando es rentable que se realicen dichas horas extras, para los medios propios o los subcontratados.

7. BIBLIOGRAFÍA

Aksen, D. & Aras, N., 2005. *Customer Selection and Profit Maximization in Vehicle Routing Problems*. Annapolis, INFORMS.

Ando, N. & Taniguchi, E., 2006. Travel time reliability in vehicle routing and scheduling with time windows. 6(3-4)(293-311).

Angelelli, E., Bazgan, C., Speranza, M. G. & Tuza, Z., 2014. Complexity and approximation for Traveling Salesman Problems with profits. 531(54-65).

Archetti, C., Bianchessi, N. & Speranza, M. G., 2013a. Optimal solutions for routing problems with profits. *Discrete Applied Mathematics*, 161(4)(547-557).

Archetti, C., Bianchessi, N. & Speranza, M. G., 2013b. The capacitated team orienteering problem with incomplete service. 7(7)(1405-1417).

Archetti, C., Bianchessi, N., Speranza, M. G. & Hertz, A., 2014a. Incomplete service and split deliveries in a routing problem with profits. 63(2)(135-145).

Archetti, C., Bianchessi, N., Speranza, M. G. & Hertz, A., 2014b. The split delivery capacitated team orienteering problem. 63(1)(16-33).

Archetti, C., Feillet, D., Hertz, A. & Speranza, M. G., 2009. The capacitated team orienteering and profitable tour problems. 60(6)(831-842).

Archetti, C., Hertz, A. & Speranza, M. G., 2007. Metaheuristics for the team orienteering problem. 13(1)(49-76).

Archetti, C. y otros, 2014c. The Team Orienteering Arc Routing Problem. *INFORMS*, 48(3), pp. 442-457.

Arkin, E., Mitchell, J. & Narasimhan, G., 1998. *Resource-Constrained Geometric Network Optimisation*. s.l., s.n.

Awerbuch, B., Azar, Y., Blum, A. & Vempala, S., 1998. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. 28(1)(254-262).

Balas, E., 1989. The prize collecting traveling salesman problem. 19(6)(621-636).

Balas, E., 1995. The prize collecting traveling salesman problem: II. Polyhedral results. 25(4)(199-216).

Balas, E., 1999. New classes of efficiently solvable generalized traveling salesman problems. 86(529-558).

Balas, E. & Martin, C. H., 1985. Software package for scheduling the rounds of a rolling mill. Volumen 104.

Bertsimas, D. J., Jaillet, P. & Odoni, A. R., 1990. A priori optimization. 38(6)(1019-1033).

Bérubé, J. F., Gendreau, M. & Potvin, J. Y., 2009a. A branch - and - cut algorithm for the undirected prize collecting traveling salesman problem. 54(1)(56-67).

Bérubé, J. F., Gendreau, M. & Potvin, J. Y., 2009b. An exact ϵ -constraint method for bi-objective combinatorial optimization problems: Application to the Traveling Salesman Problem with Profits. 194(1)(39-50).

Bianchessi, N., Mansini, R. & Speranza, M. G., 2018. A branch - and - cut algorithm for the Team Orienteering Problem. *Intl. Trans. in Op*, Volumen 25, pp. 627-635.

Bienstock, D., Goemans, M. X., Simchi-Levi, D. & Williamson, D., 1993. A note on the prize collecting traveling salesman problem. 59(1-3)(413-420).

Blum, A. y otros, 2007. Approximation algorithms for orienteering and discounted-reward TSP. 37(2)(653-670).

Bouly, H., Dang, D. C. & Moukrim, A., 2010. A memetic algorithm for the team orienteering problem. 8(1)(49-70).

Boussier, S., Feillet, D. & Gendreau, M., 2007. An exact algorithm for the team orienteering problem. 5(211-230).

Butt, S. & Cavalier, T., 1994. A heuristic for the multiple tour maximum collection problem. 21(101-111).

Butt, S. E. & Ryan, D. M., 1999. An optimal solution procedure for the multiple tour maximum collection problem using column generation. 26(4)(427-441).

Campbell, A. M., Gendreau, M. & Thomas, B. W., 2011. The orienteering problem with stochastic travel and service times. 186(1)(61-81).

Campos, V., Martí, R., Sánchez-Oro, J. & Duarte, A., 2014. GRASP with path relinking for the orienteering problem. *J Oper Res Soc*, Volumen 65, pp. 1800-1813.

Chao, I., Golden, B. L. & Wasil, E. A., 1996a. Theory and Methodology - The Team Orienteering Problem. 88(464-474).

Chao, I., Golden, B. L. & Wasil, E. A., 1996b. Theory and Methodology - A fast and effective heuristic for the Orienteering Problem. 88(475-489).

Chekuri, C., Korula, N. & Pál, M., 2012. Improved algorithms for orienteering and related problems. 8(3)(23).

Christofides, N., 1976. The vehicle routing problem. 10 (1)(55-70).

Cura, T., 2014. An artificial bee colony algorithm approach for the team orienteering problem with time windows. *Computers & Industrial Engineering*, Volumen 74, pp. 270-290.

Dang, D. C., Guibadj, R. N. & Moukrim, A., 2011. A pso-based memetic algorithm for the team orienteering problem. Issue 471-480.

Dang, D.-C., Guibadj, R. N. & Moukrim, A., 2013. An effective PSO-inspired algorithm for the team orienteering problem. *European Journal of Operational Research*, 229(2), pp. 332-344.

Dantzig, G., Fulkerson, R. & Johnson, S., 1954. Solution of a large-scale traveling-salesman problem. 2(4)(393-410).

Deitch, R. & Ladany, S. P., 2000. The one-period bus touring problem: Solved by an effective heuristic for the orienteering tour problem and improvement algorithm. 127(1)(69-77).

Dell'Amico, M., Maffioli, F. & Sciomachen, A., 1998. A Lagrangian heuristic for the Prize Collecting Travelling Salesman Problem. 81(289-306).

Dell'Amico, M., Maffioli, F. & Värbrand, P., 1995. On Prize - collecting Tours and the Asymmetric Travelling Salesman Problem. 2(3)(297-308).

Ehrgott, M., 2000. Approximation algorithms for combinatorial multicriteria optimization problems. 7(1).

Erkut, E. & Zhang, J., 1996. The maximum collection problem with time - dependent rewards. 43(5)(749-763).

Feillet, D., Dejax, P. & Gendreau, M., 2005. Traveling salesman problems with profits. *Transportation science*, 39 (2)(188-205).

Fischetti, M., Gonzalez, J. J. S. & Toth, P., 1998. Solving the orienteering problem through branch-and-cut. 10(2)(133-148).

Fischetti, M. & Toth, P., 1988. An additive approach for the optimal solution of the prize collecting traveling salesman problem. Issue 319-343.

Fomin, F. V. & Lingas, A., 2002. Approximation algorithms for time-dependent orienteering. 83(2)(57-62).

Gambardella, L. M., Montemanni, R. & Weyland, D., 2012. Coupling ant colony systems with strong local searches. 220(3)(831-843).

Gendreau, M., Laporte, G. & Semet, F., 1998a. A branch-and-cut algorithm for the undirected selective traveling salesman problem. 32(4)(263-273).

Gendreau, M., Laporte, G. & Semet, F., 1998b. A tabu search heuristic for the undirected selective travelling salesman problem. 106(2)(539-545).

Goemans, M. X., 2009. Combining approximation algorithms for the prize-collecting TSP. Issue 0910.0553.

Goemans, M. X. & Williamson, D. P., 1995. A general approximation technique for constrained forest problems. 24(2)(296-317).

Golden, B., Assad, A. & Dahl, R., 1984. Analysis of a large-scale vehicle routing problem with an inventory component. 7(181-190).

Golden, B. L., Levy, L. & Vohra, R., 1987. The orienteering problem. 34(307-318).

Golden, B. L., Raghavan, S. & Wasil, E. A., 2008. *The vehicle routing problem: latest advances and new challenges*. USA: Springer Science & Business Media, LLC.

Golden, B. L., Wang, Q. & Liu, L., 1988. A multifaceted heuristic for the orienteering problem. 35(3)(359-366).

Gunawan, A., Lau, H. C. & Lu, K., 2018. ADOPT: Combining parameter tuning and Adaptive Operator Ordering for solving a class of Orienteering Problems. *Computers & Industrial Engineering*, Volumen 121, pp. 82-96.

Gunawan, A., Lau, H. C. & Vansteenwegen, P., 2016. Orienteering Problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2), pp. 315-332.

Gunawan, A., Lau, H. C., Vansteenwegen, P. & Kun, L., 2017. Well-tuned algorithms for the Team Orienteering Problem with Time Windows. *J Oper Res Soc*, Volumen 68, pp. 861-876.

Gutin, G. & Punnen, A. P., 2007. *The traveling salesman problem and its variations*. s.l.:Springer Science & Business Media.

Hu, Q. & Lim, A., 2014. An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, 22(2), pp. 276-286.

Ilhan, T., Iravani, S. M. & Daskin, M. S., 2008. The orienteering problem with stochastic profits. 40(4)(406-421).

Jaballah, A., Ramdane, W. & Cherif-Khettaf, 2021. *Multi-trip pickup and delivery problem, with split loads, profits and multiple time windows to model a real case problem in the construction industry*. Viena, s.n.

Jaillet, P., 1988. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. 36(6)(929-936).

Jepsen, M. K., 2011. Branch-and-cut and Branch-and-Cut-and-Price Algorithms for Solving Vehicle Routing Problems.

Jozefowicz, N., Glover, F. & Laguna, M., 2008. Multi-objective metaheuristics for the traveling salesman problem with profits. 7(2)(177-195).

Jula, H., Dessouky, M. & Ioannou, P. A., 2006. Truck route planning in nonstationary stochastic networks with time windows at customer locations. 7(1)(51-62).

Kantor, M. G. & Rosenwein, M. B., 1992. The orienteering problem with time windows. Issue 629-635.

Kataoka, S. & Morito, S., 1988. An algorithm for the Single Constraint Maximum Collection Problem. 31(4)(515-530).

Kataoka, S., Yamada, T. & Morito, S., 1998. Minimum directed 1-subtree relaxation for score orienteering problem. 104(1)(139-153).

Ke, L., Archetti, C. & Feng, Z., 2008. Ants can solve the team orienteering problem. 54(3)(648-665).

Keller, C. P., 1985. Multiobjective routing through space and time: The MVP and TDVP problems. Volumen Paper 1452.

Keller, C. P. & Goodchild, M., 1988. The multiobjective vending problem: A generalization of the traveling salesman problem. 15(4).

Kenyon, A. S. & Morton, D. P., 2003. Stochastic vehicle routing with random travel times. 37(1)(69-82).

Labadie, N., Mansini, R., Melechovský, J. & Calvo, R. W., 2012. The team orienteering problem with time windows: An lp-based granular variable neighborhood search. 220(1)(15-27).

Labadie, N., Melechovský, J. & Calvo, R. W., 2011. Hybridized evolutionary local search algorithm for the team orienteering problem with time windows. 17(6)(729-753).

Laporte, G., 1992 a. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59 (2)(231-247), pp. 59(2), 231-247.

- Laporte, G., 1992 b. The vehicle routing problem: An overview of exact and approximate algorithms. 59(3)(345-358).
- Laporte, G., Louveaux, F. & Mercure, H., 1992. The vehicle routing problem with stochastic travel times. 26(3)(161-170).
- Laporte, G. & Martello, S., 1990. The Selective Travelling Salesman Problem. 26(193-207).
- Lau, H. C. y otros, 2012. *Dynamic stochastic orienteering problems for risk-aware applications*. Catalina Island, Los Angeles, s.n.
- Leifer, A. C. & Rosenwein, M. B., 1994. Strong linear programming relaxations for the orienteering problem. 73(3)(517-523).
- Liang, Y. C., Kulturel-Konak, S. & Smith, A. E., 2002. *Meta heuristics for the orienteering problem*. s.l., IEEE, pp. 384-389.
- Lin, S.-W., 2013. Solving the team orienteering problem using effective multi-start simulated annealing. *Applied Soft Computing*, 13(2), pp. 1064-1073.
- Lin, S. W. & Vincent, F. Y., 2012. A simulated annealing heuristic for the team orienteering problem with time windows. 217(1)(94-107).
- Lugo, D. d., 2020. *Deputación de Lugo*. [En línea] Available at: <http://www.deputacionlugo.gal/sites/deputacionlugo.org/files/inline-files/26-07-2019.pdf#page=10> [Último acceso: 14 10 2020].
- Malandraki, C. & Daskin, M. S., 1992. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. 26(3)(185-200).
- Mansini, R., Pelizzari, M. & Wolfer, R., 2006. A Granular Variable Neighbourhood Search Heuristic for the Tour Orienteering Problem with Time Windows. Issue 02-52.
- Marinakis, Y., Politis, M., Marinaki, M. & Matsatsinis, N., 2015. *A Memetic-GRASP Algorithm for the Solution of the Orienteering Problem*. Lorraine, Springer.
- Millar, H. H. & Kiragu, M., 1997. A time-based formulation and upper bounding scheme for the selective travelling salesperson problem. 48(511-518).
- Miller, C. E., Tucker, A. W. & Zemlin, R. A., 1960. Integer programming formulations and traveling salesman problems. *J. ACM*, 7(4)(326-329.), p. 326-329.
- Montemanni, R. & Gambardella, L. M., 2009. An ant colony system for team orienteering problems with time windows. 34(287-306).

Nguyen, V. H., 2013. A primal-dual approximation algorithm for the Asymmetric Prize-Collecting TSP. 25(2)(265-278).

Nguyen, V. H. & Nguyen, T. T. T., 2010. Approximating the asymmetric profitable tour. *Electronic Notes in Discrete Mathematics*, Volumen 36, pp. 907-914.

Nguyen, V. H. & Nguyen, T. T. T., 2012. Approximating the asymmetric profitable tour. 4(3)(294-301).

Poggi, M., Viana, H. & Uchoa, E., 2010. *The Team Orienteering Problem: Formulations and Branch-Cut and Price*. Dagstuhl, Thomas Erlebach and Marco.

Ramesh, R. & Brown, K. M., 1991. An efficient four-phase heuristic for the generalized orienteering problem. 18(2)(151-165).

Ramesh, R., Yoon, Y. S. & Karwan, M. H., 1992. An optimal algorithm for the orienteering tour problem. 4(2)(155-165).

Righini, G. & Salani, M., 2008. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, Volumen 51, pp. 155-170.

Righini, G. & Salani, M., 2009. Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. 36(4)(1191-1203).

Santini, A., 2019. An adaptive large neighbourhood search algorithm for the orienteering problem. *Expert Systems with Applications*, Volumen 123, pp. 154-167.

Schilde, M., Doerner, K. F., Hartl, R. F. & Kiechle, G., 2009. Metaheuristics for the bi-objective orienteering problem. 3(3)(179-201).

Sevcli, A. & Sevilgen, F. E., 2010. StPSO: Strengthened particle swarm optimization. *Turk J Elec Eng & Comp Sci*, Volumen 6, pp. 1095-1114.

Silberholz, J. & Golden, B., 2010. The effective application of a new approach to the generalized orienteering problem.. 16(3)(393-415).

Sokkappa, P. R., 1990. *The cost-constrained traveling salesman problem (No. UCRL-LR-105145)*. s.l.:Lawrence Livermore National Lab., CA (USA).

Souffriau, W., Vansteenwegen, P., Berghe, G. V. & Van Oudheusden, D., 2010. A path relinking approach for the team orienteering problem. 37(11)(1853-1859).

Stavropoulou, F., Repoussis, P. P. & Tarantilis, C. D., 2019. The Vehicle Routing Problem with Profits and consistency constraints. *European Journal of Operational Research*, 274(1), pp. 340-356.

Tang, H. & Miller-Hooks, E., 2005a. A TABU search heuristic for the team orienteering problem. 32(1379-1407).

Tang, H. & Miller-Hooks, E., 2005b. Algorithms for a stochastic selective travelling salesperson problem. 56(4)(439-452).

Tang, L. & Wang, X., 2008. An iterated local search heuristic for the capacitated prize-collecting travelling salesman problem. Issue 590-599.

Taniguchi, E. & Shimamoto, H., 2004. Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. 12(3)(235-250).

Taniguchi, E., Thompson, R. G., Yamada, T. & Van Duin, R., 2001. *City Logistics*. s.l.:Network modelling and intelligent transport systems.

Taniguchi, E., Yamada, T. & Tamagawa, D., 1999. *Probabilistic vehicle routing and scheduling on variable travel times with dynamic traffic simulation*. Cairns, Queensland, Australia, In-International Conference on City Logistic, 1st, 1999.

Tasgetiren, M. F., 2001. A genetic algorithm with an adaptive penalty function for the orienteering problem. 4(2)(1-26).

Tasgetiren, M. F. & Smith, A. E., 2000. *A genetic algorithm for the orienteering problem*. s.l., IEEE, pp. 910-915.

Teng, S. Y., Ong, H. L. & Huang, H. C., 2004. An integer L-shaped algorithm for time-constrained traveling salesman problem with stochastic travel and service times. 21(02)(241-257).

Thomadsen, T. & Stidsen, T., 2003. The Quadratic Selective Travelling Salesman Problem. Volumen 17.

Toth, P. & Vigo, D., 2002. *The vehicle routing problem*. USA: Society for Industrial and Applied Mathematics.

Toth, P. & Vigo, D., 2003. The granular tabu search and its application to the vehicle-routing problem. 15(4)(333-346).

Tricoire, F., Romauch, M., Doerner, K. F. & Hartl, R. F., 2010. Heuristics for the multi-period orienteering problem with multiple time windows. 37(2)(351-367).

Tsiligirides, T., 1984. Heuristic methods applied to orienteering. Issue 797-809.

UNE-EN 13306, 2011. 13306. s.l.:UNE-EN.

UNE-EN 15331, 2012. 15331. s.l.:UNE-EN.

Vansteenwegen, P. & Gunawan, A., 2019. *Orienteering Problems - Models and Algorithms for Vehicle Routing Problems with Profits*. Cham: Springer.

Vansteenwegen, P. & Mateo, M., 2014. *An iterated local search algorithm for the single-vehicle cyclic inventory routing problem*. s.l.:European Journal of Operational Research.

Vansteenwegen, P., Souffriau, W., Berghe, G. V. & Van Oudheusden, D., 2009a. A guided local search metaheuristic for the team orienteering problem. 196(1)(118-127).

Vansteenwegen, P., Souffriau, W., Berghe, G. V. & Van Oudheusden, D., 2009b. Iterated local search for the team orienteering problem with time windows. 36(12)(3281-3290).

Vansteenwegen, P., Souffriau, W. & Van Oudheusden, D., 2011. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1), pp. 1-10.

Vansteenwegen, P. & Van Oudheusden, D., 2007. The mobile tourist guide: an OR opportunity. 20(3)(21-27).

Verbeeck, C., Vansteenwegen, P. & Aghezzaf, E. H., 2014. An extension of the arc orienteering problem and its application to cycle trip planning. *Transportation Research Part E: Logistics and Transportation Review*, Volumen 68, pp. 64-78.

Wang, Q., Sun, X., Golden, B. L. & Jia, J., 1995. Using artificial neural networks to solve the orienteering problem. 61(1)(111-120).

Wang, X., Golden, B. L. & Wasil, E. A., 2008. Using a genetic algorithm to solve the generalized orienteering problem. In *The vehicle routing problem: latest advances and new challenges*. Issue 263-274.

Xu, W. y otros, 2020. *Approximation Algorithms for the Team Orienteering Problem*. Toronto, IEEE.

