

Trabajo Fin de Máster
Máster Universitario en Ingeniería Industrial

Diseño y aplicación de un Algoritmo Memético
basado en Sistema de Colonias de Hormigas para la
resolución del problema VRP en Servicios de
Control Metrológico

Autor: Luis Moles Revert

Tutor: José Carlos Molina Gómez

Dpto. Organización Industrial y Gestión de Empresas I
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020



Trabajo Fin de Máster
Máster Universitario en Ingeniería Industrial

**Diseño y aplicación de un Algoritmo Memético
basado en Sistema de Colonias de Hormigas para la
resolución del problema VRP en Servicios de
Control Metrológico**

Autor:

Luis Moles Revert

Tutor:

José Carlos Molina Gómez

Profesor sustituto interino

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020

Trabajo Fin de Máster: Diseño y aplicación de un Algoritmo Memético basado en Sistema de Colonias de Hormigas para la resolución del problema VRP en Servicios de Control Metrológico

Autor: Luis Moles Revert

Tutor: José Carlos Molina Gómez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

En primer lugar, quiero expresar mi agradecimiento a mi tutor del trabajo José Carlos Molina por su tiempo, paciencia y predisposición en cada momento que he necesitado su ayuda.

En Segundo lugar, agradecer a la Universidad de Sevilla por la oportunidad que me ha dado de aprender y mejorar tanto en lo personal como en lo profesional durante esta etapa que finalizo.

Por ultimo, quiero dar las gracias a mi familia porque siempre han estado ahí para apoyarme, escucharme y darme consejos cuando más me ha hecho falta.

Luis Moles Revert

Sevilla, Noviembre 2020

Vivimos en un mundo globalizado, en el que acceder a cualquier tipo de producto o servicio resulta cada vez más sencillo para el consumidor. El abanico de posibilidades de las que dispone crece exponencialmente a medida que se expande el comercio tanto nacional como internacional, fomentado entre otras cosas por la mejora de las redes de transporte, la evolución de los medios de comunicación que dan visibilidad al mundo y la mejora continua en tecnología.

Debido a la feroz competencia entre empresas, hoy en día no basta con fabricar o realizar un buen producto o servicio. La gestión logística juega un papel decisivo en la diferenciación entre compañías ya que no solo puede incrementar su competitividad en el mercado al ser capaces de ofrecer productos con menor coste, sino que también pueden aumentar la satisfacción del cliente ofreciendo mayor calidad en el servicio.

El Problema de Ruteo de Vehículos o VRP (Vehicle Routing Problem), trata de un problema de optimización combinatoria que ha sido objeto de multitud de estudios en las últimas décadas y que juega un papel de gran importancia en entornos logísticos, ya que una adecuada planificación puede acarrear considerables ahorros.

El objetivo del trabajo es introducir y resolver el problema de enrutamiento de vehículos en empresas de control metrológico. Este problema de la vida real, combina múltiples atributos que se distinguen de los introducidos en los problemas tradicionales de enrutamiento de vehículos. Para resolver el problema se ha diseñado un Algoritmo de Colonia de Hormigas hibridado con una búsqueda local, al que llamaremos Algoritmo Memético ACS. A su vez, la búsqueda local es realizada por un algoritmo híbrido basado en un algoritmo de búsqueda en entornos variables en el que se utiliza un algoritmo de búsqueda tabú en el proceso de búsqueda local (VNTS). Finalmente, el algoritmo ha sido evaluado comparándolo con otros algoritmos del estado del arte de la literatura científica y ha sido aplicado con éxito al diseño y planificación de rutas de un caso real de una empresa que presta servicios de control metrológico.

Abstract

In today's globalized world, it is easier than ever before for consumers to have access to any product or service of their choosing. The range of possibilities available keeps growing exponentially, as both national and international trade expand, fostered by the improvement of transport networks, the expansion and reach of traditional and social media and the ongoing improvement in technology.

Due to fierce competition among companies, nowadays it is just not enough to deliver a quality product or service. Logistics management plays a decisive role in the differentiation of companies. It can not only increase their competitiveness by being able to offer products with lower cost, but it can also increase customer satisfaction by offering higher quality services.

The Vehicle Routing Problem or VRP deals with a problem of combinatorial optimization that has been the subject of many studies and plays a role of great importance in logistics environments, since proper planning can lead to considerable savings.

The focus of this master's thesis is to introduce and solve the VPR in metrological control companies. This real-life problem combines multiple attributes that are distinguished from those introduced in traditional vehicle routing problems. To achieve this, it is have been designed an Ant Colony Algorithm hybridized with a local search, which we will call the ACS Memetic Algorithm. In turn, the local search is performed by a hybrid algorithm based on a search algorithm in variable environments in which a tabo search algorithm is used in the local search process (VNTS). Finally, the algorithm has been evaluated comparing it with others from the state of the art in the scientific literature. It has then been successfully applied to the design and planning of routes of a real case of a company that provides metrological control services.

Agradecimientos	ix
Resumen	xi
Abstract	xii
Índice	xiii
Índice de Tablas	xv
Índice de Figuras	xvi
Notación	xviii
1 Introducción	1
1.1 <i>Introducción al Trabajo</i>	1
1.2 <i>Objetivos del trabajo y motivación</i>	3
1.3 <i>Estructura del Trabajo</i>	3
2 Revisión de la literatura	5
2.1 <i>Introducción</i>	5
2.2 <i>Problema VRP</i>	6
2.3 <i>VRP Enriquecido</i>	9
3 Descripción del problema de diseño de rutas en empresas que prestan servicios de control metrológico	13
3.1 <i>Problema MCSVRP</i>	13
3.1.1 <i>Operativa de una empresa que proporciona servicios de control metrológico</i>	14
3.1.2 <i>Atributos del MCSVRP</i>	15
3.2 <i>Modelado del problema MCSVRP</i>	17
4 Metodología de resolución	21
4.1 <i>Introducción a los métodos de resolución</i>	21
4.2 <i>Algoritmos Meméticos</i>	22
4.3 <i>Sistema de Colonia de Hormigas (ACS)</i>	24
4.3.1 <i>Construcción de Rutas</i>	25
4.3.2 <i>Inicialización y actualización de la feromona</i>	27
4.4 <i>VNTS</i>	30
4.4.1 <i>Búsqueda Tabú</i>	30
4.4.2 <i>Estructura de vecindad VNS</i>	32
5 Resultados Experimentales	37
5.1 <i>Introducción</i>	37
5.2 <i>Ajuste de parámetros del algoritmo</i>	38
5.3 <i>Análisis de los resultados obtenidos por el algoritmo</i>	41
5.3.1 <i>Comparación de los resultados en instancias de la literatura</i>	41
5.3.2 <i>Análisis comparativo entre algoritmos</i>	46
5.4 <i>Aplicación del Algoritmo a una planificación real</i>	48
5.4.1 <i>Introducción al problema</i>	48
5.4.2 <i>Descripción del problema en una empresa que da servicios de control metrológico</i>	48

6 Conclusiones

55

Referencias

57

ÍNDICE DE TABLAS

Tabla 1 Clasificación de las características de los RVRP (Lahyani et al., 2015)	11
Tabla 2 Parámetros a evaluar en la fase 1 de la experimentación	38
Tabla 3 Resultados de la FO y desviación obtenidas en la fase 1 de la experimentación	38
Tabla 4 Parámetros a evaluar en la fase 2 de la experimentación	39
Tabla 5 Resultados de la FO y desviación obtenidas en la fase 2 de la experimentación	39
Tabla 6 Parámetros a evaluar en la fase 3 de la experimentación	39
Tabla 7 Resultados de la FO y desviación obtenidas en la fase 3 de la experimentación	40
Tabla 8 Valores finales de los parámetros	40
Tabla 9 Valores de los parámetros alfa	40
Tabla 10 Resultados para los casos de HVRPTW en Paraskevopoulos et al. (2008) (Molina et al., 2020)	43
Tabla 11 Resultados para los casos de HVRPTW en Jiang et al (2014) (Molina et al., 2020)	44
Tabla 12 Resultado de los problemas HVRP de Taillard	45
Tabla 13 Resultados obtenidos por los algoritmos en el problema de Taillard	46
Tabla 14 Comparativa entre ACS, VNTS y Alg. Memético	47
Tabla 15 Comparativa entre VNTS y Alg. Memético	47
Tabla 16 Tipo de servicios y patrones necesarios	48
Tabla 17 Descripción de patrones y cantidades disponibles	49
Tabla 18 Cualificación de los verificadores	49
Tabla 19 Características de las órdenes recibidas	50
Tabla 20 Solución Obtenida por la Empresa	51
Tabla 21 Solución obtenida por el Algoritmo Memético	52
Tabla 22 Soluciones obtenidas para los distintos casos por el Algoritmo Memético ACS	53

ÍNDICE DE FIGURAS

Figura 1 Evolución de las medias anuales de PM10 (2001-2015) por tipo de estación y área	2
Figura 2 Representación gráfica del problema VRP (Tunga, H. et al., 2017)	6
Figura 3 Cronología del VRP (Lin, C. et al., 2014)	8
Figura 4 Comportamiento de una colonia de hormigas (Menéndez, E. 2005)	24
Figura 5 Variable Neighborhood Search: VRP (Hansen et al., 2008)	32
Figura 6 Primera estructura de vecindad	34
Figura 7 Segunda estructura de vecindad	34
Figura 8 Tercera estructura de vecindad	35

Notación

ACO	Ant Colony Optimization
ACS	Ant Colony System
BKS	Best Known Solution
BPP	Bin Packing Problem
CVRP	Capacited Vehicle Routing Problem
DVRP	Dynamic Vehicle Routing Problem
FSM	Fleet size and Mix Problem
FSMD	Fleet Size and Mix VRP with Vehicle Dependent Routing Costs
FSMF	Fleet Size and Mix Vehicle Routing Problem with Fixed Costs
FSMFD	Fleet Size and Mix VRP with Fixed Costs and Vehicle Dependent Routing Costs
GVRP	Green Vehicle Routing Problem
HEA	Hibrid Evolutionary Algorithm
HFFVRP	Heterogeneous Fixed Fleet Vehicle Routing Problem
HFVRP	Heterogeneous Fleet Vehicle Routing Problem
HL	Holding List
HVRPD	Heterogeneous Vehicle Routing Problem with Vehicle Dependent Routing Costs
HVRPFD	Heterogeneous VRP with Fixed Costs and Vehicle Dependent Routing Costs
LRP	Locatio Routing Problem
MA	Memetic Algorithm
MAVRP	Multi – Attribute Vehicle Routing Problem
MCSVRP	Metrological Control Service Vehicle Routing Problem
MCVRP	Multi Compartment Vehicle Routing Problem
MDVRP	Multiple Depot Vehicle Routing Problem
MEVRP	Multi-echelon Vehicle Routing Problem
m-TSP	Multiple Travelling Salesman Problem
PVRP	Periodic Vehicle Routing Problem
ReVNTS	Reactive Variable Neighborhood Tabu Search
RVRP	Rich Vehicle Routing Problem
SDVRP	Site – Dependent Vehicle Routing Problem
SDVRP	Split Delivery Vehicle Routing Problem
SVRP	Stochastic Vehicle Routing Problem
TDVRP	Time – Dependent Vehicle Routing Problem
TS	Tabu Search
TSP	Travelling Salesman Problem
VND	Variable Neighborhood Descent
VNS	Variable Neighborhood Search
VNTS	Variable Neighborhood Tabu Search
VRP	Vehicle Routing Problem

VRPB	Vehicle Routing Problem with Backhauls
VRPSPD	Vehicle Routing Problem with Simultaneous Pickups and Deliveries
VRPTW	Vehicle Routing Problem with Time Windows

1 INTRODUCCIÓN

1.1 Introducción al Trabajo

Vivimos en un mundo globalizado en el que acceder a cualquier tipo de producto o servicio resulta cada vez más sencillo para el consumidor. El abanico de posibilidades de las que disponemos crece exponencialmente a medida que se expande el comercio tanto nacional como internacional, todo esto fomentado entre otras cosas por la mejora de las redes de transporte, la evolución de los medios de comunicación que dan visibilidad al mundo y la mejora continua en tecnología.

Debido a la feroz competencia entre empresas hoy en día no basta con fabricar o realizar un buen producto o servicio, ya que la estrategia empresarial sería un fracaso si éstos no llegan a tiempo a su destino final. Es por esto que la gestión logística juega un papel decisivo en la diferenciación entre compañías, debido a que no solo puede incrementar su competitividad en el mercado siendo capaces de ofrecer productos con menor coste, sino que también pueden aumentar la satisfacción del cliente ofreciendo mayor calidad en el servicio.

La logística empresarial comprende la planificación, la organización y el control de todas las actividades relacionadas con la obtención, el traslado y el almacenamiento de materiales y productos, desde la adquisición hasta el consumo, a través de la organización y como un sistema integrado, incluyendo también todo lo referente a los flujos de información implicados. El objetivo perseguido es la satisfacción de las necesidades y los requerimientos de la demanda, de la manera más rápida y eficaz y con el mínimo coste posible. (Arbós, L. C., 2012)

Para cumplir al máximo los dos grandes objetivos de la logística empresarial, maximizar las ventas al menor coste posible, se deberá diseñar, planificar y controlar una red de distribución cuyo funcionamiento sea efectivo y eficiente. Para ello se deberá determinar una configuración de puntos de fabricación, almacenamiento, ventas y un buen sistema de transporte y tratamiento de la información que permitan cumplir los objetivos.

Dentro de lo que implica la logística empresarial, la logística en el transporte es un proceso complejo que se inicia con la llegada del pedido y finaliza con la entrega del producto o realización del servicio al cliente. Su importancia radica en conseguir que los productos y servicios adecuados estén en los lugares apropiados en el momento preciso y condiciones exigidas.

Los costes logísticos asociados al transporte y procesamiento de pedidos pueden alcanzar un peso importante en el precio final del producto. Se estima que los costos de transporte representan entre el 10% y 20% del coste final del producto (Toth & Vigo, 2001). Estos costes logísticos se asocian a la pérdida de ventas por falta de satisfacción de los clientes y están directamente relacionados con la duración de los envíos.

En España la movilidad interior de mercancías está estrechamente relacionada con el transporte por carretera, ya que representa cerca del 95% de la movilidad interior (OTLE, Informe anual 2018). Por este motivo es tan importante para las compañías de bienes y servicios el diseño de unos sistemas logísticos eficientes que les permitan optimizar las rutas de transporte, reduciendo así el consumo de recursos tanto humanos como materiales y mejorando la satisfacción de los clientes.

A continuación, se muestran unos datos proporcionados por el Ministerio para la Transición Ecológica y el Reto Demográfico del Gobierno de España en los que se refleja la evolución de las medias anuales hasta el año 2015 de emisiones de partículas en suspensión PM10 y el tipo de procedencia.

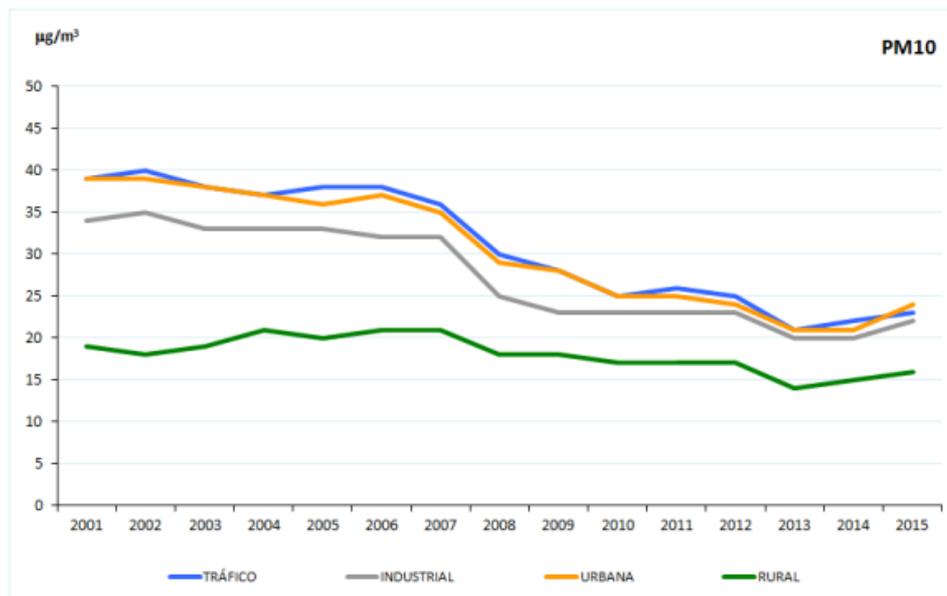


Figura 1 Evolución de las medias anuales de PM10 (2001-2015) por tipo de estación y área

Cabe señalar, aunque no serán temas de estudio a tratar directamente en este trabajo, que los efectos provocados por el transporte no son únicamente logísticos. El transporte por carretera es uno de los principales causantes del incremento en la concentración de CO₂ en la atmósfera, lo que provoca un incremento de la temperatura media de la tierra con sus correspondientes efectos en la naturaleza.

Además de los factores ambientales mencionados, se enumeran a continuación otros efectos causados por el tráfico rodado que afectan al conjunto de la sociedad: Contaminación acústica, siniestralidad en las carreteras, ocupación del territorio, congestión del tráfico y consumo de energía.

La movilidad sostenible, “es aquella capaz de satisfacer las necesidades de la sociedad de moverse libremente, acceder, comunicarse, comercializar o establecer relaciones sin sacrificar otros valores humanos o ecológicos básicos actuales o del futuro. Es decir, sin comprometer el bienestar de las generaciones futuras.” Definición del World Business Council for Sustainable Development (WBCSD).

1.2 Objetivos del trabajo y motivación

El objetivo del trabajo es introducir el problema de enrutamiento de vehículos en empresas de control metrológico (MCSVRP, *Metrological Control Services VRP*). El problema a tratar está incluido dentro de los problemas conocidos como problemas enriquecidos de enrutamiento de vehículos (RVRP, *Rich VRP*), ya que combina una serie de atributos presentes en entornos reales que no son normalmente abordados por los problemas tradicionales del VRP. Los atributos que consideramos incluyen flotas de vehículos fijas y heterogéneas, ventanas de tiempo para depósito y clientes, sincronización entre recursos, restricciones entre conductor-cliente y vehículo- cliente, prioridades entre clientes y demandas no servidas.

El problema es presentado mediante un modelo de programación lineal y resuelto mediante un algoritmo basado en el Sistema de Colonia de Hormigas (ACS, *Ant Colony System*), el cual ha sido ampliamente aplicado en problemas de optimización combinatoria. Además, para mejorar las soluciones obtenidas por el ACS en el problema MCSVRP, lo hibridizaremos con una búsqueda local. A esta unión entre el ACS con una búsqueda local lo conocemos como Algoritmo memético ACS. La búsqueda local de este algoritmo es realizada por un algoritmo híbrido basado en un algoritmo de búsqueda en entornos variables en el que se utiliza un algoritmo de búsqueda tabú en el proceso de búsqueda local (VNNTS, *Variable Neighborhood Tabu Search*). Finalmente, se comparan los resultados obtenidos con estudios anteriores realizados al mismo problema, valorando la eficacia del algoritmo.

La motivación a la hora de realizar este proyecto es una consecuencia del impacto que provoca el transporte por carretera en la economía de una empresa. La importancia que supone la gestión logística en una empresa es clave no solo para reducir el consumo y empleo de recursos humanos y materiales (combustible, vehículos, utillaje, ...), sino que una buena gestión logística también supone una diferenciación considerable respecto al resto de competidores en un mercado actual globalizado en el que la competencia entre compañías es feroz. Además, como se ha podido ver en los apartados anteriores, el transporte por carretera supone en torno al 95% del total de transportes de mercancías en España. Los vehículos motorizados son los causantes de aproximadamente el 25% de las emisiones de gases contaminantes a la atmósfera y dicha concentración está provocando el aumento de la temperatura media de la tierra, lo que puede derivar en consecuencias graves para los ecosistemas.

1.3 Estructura del Trabajo

El trabajo está constituido por seis capítulos. El primero de ellos muestra un breve resumen de la importancia que supone para las empresas la gestión logística, especialmente la asociada al transporte y procesamiento de pedidos. Se exponen además, los objetivos del trabajo y la motivación que implica su aplicación en la vida real.

El segundo capítulo comienza con una introducción a los problemas de ruteo de vehículos. Se realiza una revisión de la literatura comenzando por los inicios del problema clásico TSP y exponiendo las diferentes variantes al problema que han ido surgiendo con el paso de los años hasta llegar a los actuales problemas enriquecidos RVRP.

El capítulo 3 presenta el problema MCSVRP a abordar en este trabajo. Se muestran las diferentes características y atributos que definen al problema y se hace una breve revisión de la literatura de trabajos previos donde se exponen problemas que comparten cierto número de atributos. Finaliza con el modelo de programación lineal donde se muestran las múltiples variables empleadas y restricciones que comprenden el problema.

En el capítulo 4, se realiza una revisión de la literatura centrada en la resolución de los problemas RVRP a través de algoritmos meméticos. Además, la segunda parte de este capítulo presenta el algoritmo memético diseñado para este trabajo, definiendo con detalle cada una de sus partes.

En el Capítulo 5, se experimenta con el algoritmo memético. En primer lugar, se realiza un ajuste de los parámetros que son definidos en el algoritmo. Se realiza una comparación de los resultados obtenidos por el

algoritmo en baterías de problemas de referencia de la literatura científica, y se compara su efectividad con otros algoritmos. Por último, se aplica a un caso real de una empresa dedicada a dar servicios de control metrológico de equipos de medida en las ubicaciones de los clientes.

Finalmente, en el Capítulo 6, se extraen las diferentes conclusiones de la investigación realizada, analizando los resultados obtenidos en los capítulos anteriores y proponiendo líneas de investigación futuras.

2 REVISIÓN DE LA LITERATURA

2.1 Introducción

El problema de optimización de rutas es un problema que ha sido y sigue siendo ampliamente estudiado en la literatura debido al gran número de aplicaciones que tiene en la vida real. Para las empresas, la gestión logística supone una pieza clave en el buen funcionamiento de la cadena de suministro. Los precios de los productos y servicios en un mercado globalizado son cada vez más ajustados para las empresas, que deben luchar en un ambiente tremendamente competitivo donde sus márgenes de beneficio se ven reducidos considerablemente para poder sobrevivir. Llegados a este punto, no es suficiente con tener un producto o servicio de calidad, la logística juega un papel clave reduciendo tanto los costes como los tiempos en los que se satisface al cliente.

En este trabajo se trata el problema de planificación de rutas de vehículos que surge en empresas que dan un servicio de control metrológico de equipos de medida. Este tipo de empresas reciben las órdenes de trabajo y deben acudir a las instalaciones de los clientes para poder desempeñarlas. Este tipo de problemas tienen cierta complejidad, ya que en muchos casos el tiempo de planificación es corto, la demanda no es constante existiendo picos de trabajo en los que se debe dar prioridad a unas órdenes frente a otras, el tipo de tecnología de los equipos de medida puede ser variado por lo que es necesario personal cualificado en cada una de las materias, y se requiere de transporte de equipos para desempeñar los diferentes trabajos.

A continuación, se va a realizar un estudio del estado del arte de los problemas de optimización de rutas, desde el momento que se inician con el clásico TSP o problema del viajero hasta llegar a los conocidos como RVRP o problemas enriquecidos.

El propósito de esta revisión de la literatura es dar una visión general del problema de ruteo de vehículos, partiendo de los comienzos de los estudios en los que se desarrolla la versión más básica del problema, mostrando la evolución que ha seguido con el paso del tiempo y finalmente hacia donde se dirige.

2.2 Problema VRP

Aunque los inicios del problema datan entre los años treinta y cuarenta y se le atribuye a Hassler Whitney quien planteó el problema en 1934, el término Travelling Salesman Problem (TSP) aparece por primera vez en un artículo de Julia Robinson en 1949 en el que, dado un conjunto disperso de ciudades con un coste conocido en recorrer cada par de ellas, el problema del agente viajero busca diseñar la ruta más económica en recorrer todas las ciudades y volver al punto de partida. Desde entonces, el TSP ha sido aplicado en numerosas áreas como la logística, fabricación, telecomunicaciones, etc, siendo en Flood (1956) donde se introducen los primeros métodos computacionales aceptables para la resolución del problema.

Con Dantzing y Ramser en 1959, aparecen las primeras variaciones del problema. Este trabajo introduce el conocido como TSP generalizado o VRP, en el que una flota de camiones debe abastecer de combustible a diferentes estaciones de servicio partiendo de un depósito central. Desde entonces el interés por el VRP ha crecido entre matemáticos e investigadores, siendo aplicado en numerosas disciplinas.

El problema VRP trata la búsqueda de una solución óptima al problema de enrutamiento de vehículos teniendo en cuenta una serie de consideraciones tales como: el número de nodos (clientes) a servir, la demanda de los clientes, el número de vehículos y sus capacidades. Este tipo de problemas de optimización combinatoria pertenecen a la categoría denominada NP – Hard, ya que son problemas que no pueden resolverse en un tiempo polinomial y crecen exponencialmente a medida que se introducen nuevas variables o aumenta el número de clientes. Para su resolución se emplean mayoritariamente métodos heurísticos que proporcionan soluciones aproximadas en un espacio corto de tiempo. (Toth and Vigo, 2002).

El VRP clásico es aquel cuyo objetivo es minimizar la distancia total recorrida por un conjunto de vehículos que se encuentran ubicados en un depósito central. Las rutas se diseñan de forma que cada cliente es visitado una sola vez por un único vehículo. Todos los vehículos pueden realizar una sola ruta durante el periodo de planificación, partiendo del depósito central y terminando en el mismo punto (Figura 2).

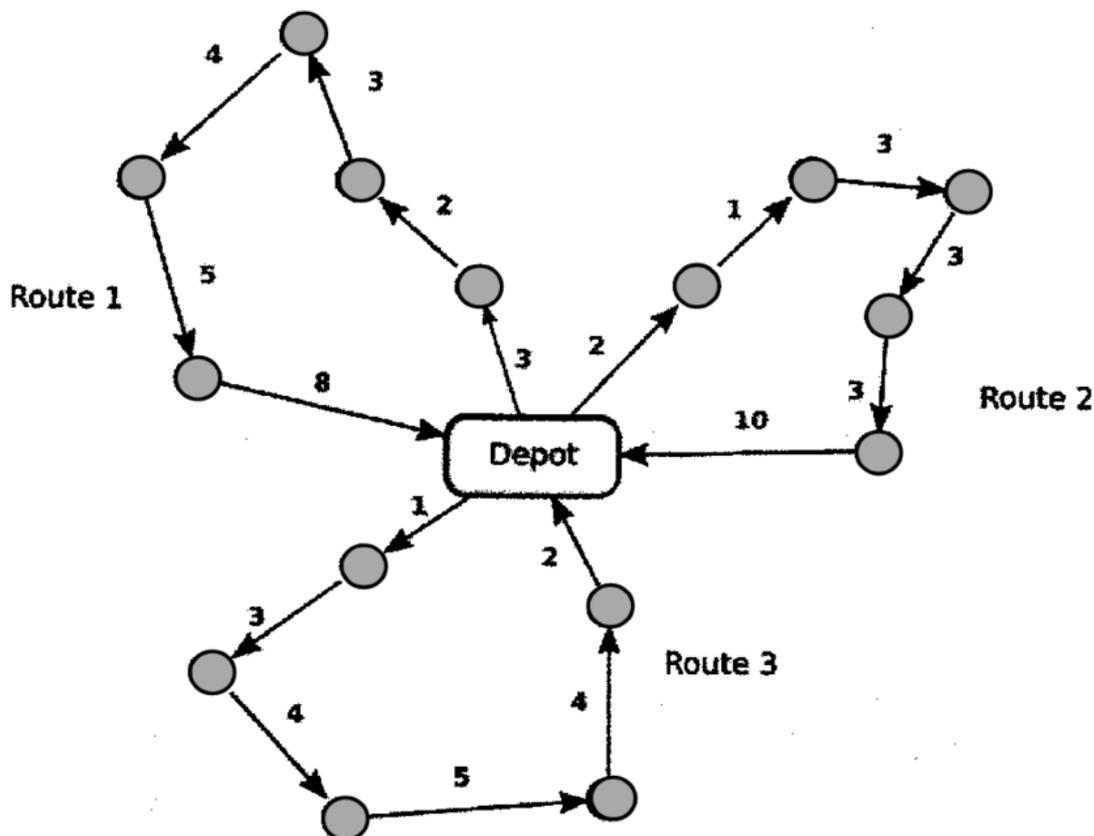


Figura 2 Representación gráfica del problema VRP (Tunga, H. et al., 2017)

El PVRP o VRP Periódico, es una generalización del problema clásico de enrutamiento de vehículos donde el diseño de las rutas realizadas por los vehículos se debe planificar para ser realizadas en un periodo de tiempo de M días. Durante cada día dentro del periodo de planificación, una flota de vehículos viaja a través de los nodos realizando las rutas que empiezan y terminan en el depósito. A cada vehículo se le asigna un conjunto de rutas que realiza durante el periodo de planificación. El PVRP fue introducido por Beltrami et al. (1974).

Un problema similar al VRP es el conocido como múltiple TSP (m-TSP) en el que, partiendo de un depósito central, el viajante debe realizar m rutas para visitar cada una de las n ciudades una sola vez, en cada ruta no puede visitar más de p ciudades. (Miller et al., 1960).

El CVRP surge como una evolución del m-TSP, y consiste en un problema de enrutamiento de vehículos en los que cada cliente tiene asociada una demanda d_i , todos los vehículos son iguales, tienen una capacidad de carga limitada C y todos parten del mismo depósito central. (Toth and Vigo, 2002).

Hasta este punto siempre se ha considerado una flota de vehículos homogénea, todos son iguales en cuanto a tamaño, consumos, capacidad, etc, por lo que realizar una ruta con uno u otro vehículo no es relevante en la resolución del problema. Sin embargo, a partir del CVRP en el que aparecen limitaciones de carga, surgen dos corrientes diferentes para la resolución de los problemas: VRP Homogéneo y el VRP Heterogéneo.

Si la flota de vehículos es heterogénea (HFVRP, *Heterogeneous Fleet VRP*), quiere decir que existe alguna diferencia entre vehículos que dé prioridad o limite el uso de unos sobre otros, dicha diferenciación puede ser debida al tamaño, capacidad, consumos, emisiones, equipamiento, etc. Este problema fue estructurado por primera vez en Golden et al. (1984).

La versión generalizada del HVRP consiste en el diseño de un conjunto válido de rutas con un coste total mínimo, donde cada cliente es visitado por una sola ruta y el número de rutas diseñadas por vehículos del tipo $k \in M$ no es mayor que m_k (Baldacci et al., 2008). En la literatura, a pesar de los múltiples estudios y variantes surgidas de este problema, se considera al HVRP como la variante del VRP con flota de vehículos heterogénea y limitada.

Un problema relacionado al anterior, que ha recibido numerosa atención en la literatura es el Site – Dependent VRP (SDVRP), en el que existe una flota heterogénea y limitada disponible para realizar el servicio, los costes de enrutamiento son independientes al vehículo y cada cliente debe incluir las restricciones del tipo de vehículo que lo debe servir. Se diferencia del HVRP general, donde el coste del vehículo k en recorrer el arco ij es infinito para aquellos vehículos que son incompatibles con el nodo j . (Nag et al., 1988).

El Fleet size and Mix Problem (FSM), engloba aquellos problemas en los que la flota es heterogénea y el número de vehículos es ilimitado. Es el caso, por ejemplo, de aquellas empresas que en lugar de tener los vehículos en propiedad, tienen la capacidad para alquilarlos en función de la carga de trabajo del periodo de tiempo estudiado. (Taillard, 1999).

En función de si el coste por emplear un vehículo u otro es fijo o variable, en los problemas HVRP y FSM añadimos la letra “F” cuando los costes son fijos y la letra “D” cuando los costes del ruteo dependen del vehículo utilizado (Baldacci et al., 2008). De esta forma obtenemos las siguientes combinaciones de problemas:

- Heterogeneous VRP with Fixed Costs and Vehicle Dependent Routing Costs (HVRPFD).
- Heterogeneous VRP with Vehicle Dependent Routing Costs (HVRPD).
- Fleet Size and Mix VRP with Fixed Costs and Vehicle Dependent Routing Costs (FSMFD).
- Fleet Size and Mix VRP with Vehicle Dependent Routing Costs (FSMD).
- Fleet Size and Mix VRP with Fixed Costs (FSMF).

El VRPTW (*Vehicle Routing Problem with Time Windows*) o VRP con ventanas de tiempo, es un problema con grandes aplicaciones en la vida real que ha sido ampliamente estudiado debido a la complejidad de su resolución y al impacto económico que supone para la empresa. Lo que diferencia a este problema del VRP convencional es la necesidad de atender a los clientes en un periodo de tiempo comprendido entre $[Ew_i,$

$LW_i]$, donde EW_i es el instante más temprano de tiempo en el que puede realizarse el servicio y LW_i el más tardío. Este problema se presenta por primera vez en Pullen et al. (1967).

Las ventanas de tiempo pueden ser especificadas en términos de una sola cara o doble cara (single – sided or double – sided). En el caso de una sola cara, los clientes especifican el instante límite en el que deben ser servidos, mientras que en el caso de doble cara, se impone tanto el momento más temprano como el más tardío en el que se debe servir a los nodos. Si un vehículo llega a su destino antes del comienzo de la ventana de tiempo, se le penalizará con un tiempo de espera, mientras que si llega más de su momento más tardío no podrá realizar el servicio. (Tan et al., 2006).

Si el VRP clásico se caracteriza porque cada cliente debe ser visitado por un solo vehículo, el SDVRP (*Split Vehicle Routing Problem*) permite la posibilidad de que un cliente sea servido en más de una ocasión. Es el caso, por ejemplo, de aquellos problemas en los que la necesidad del cliente sea mayor que la capacidad de los vehículos. El SDVRP fue introducido por Dror et al. (1989) donde demuestra un considerable ahorro en términos de distancia total recorrida y vehículos empleados utilizando esta metodología frente al VRP convencional.

El problema VRP con entregas y recogidas simultáneas en el mismo nodo, también conocido como VRPSPD (*VRP with simultaneous pickup and delivery*) fue tratado por primera vez por Min (1989). El objetivo es encontrar rutas para un conjunto de vehículos donde el coste sea mínimo, cumpliendo la restricción de capacidad del vehículo, y con la posibilidad de entregar y recoger productos de un nodo al mismo tiempo.

Un problema muy parecido al anterior es el VRPB (*VRP with Backhauls*) o VRP con retornos, que permite tanto la entrega como la recogida de productos del cliente. Los clientes se dividen en “*linehaul*”, que son aquellos que requieren de un bien o servicio del depósito, y los clientes “*backhaul*”, que envían mercancía al depósito. Estos problemas tienen dos principales características: la carga de bienes enviadas al cliente y las recibidas no pueden exceder la capacidad del vehículo, y el proceso de recogida no se puede iniciar hasta que se han finalizado todas las entregas, ya que los productos recogidos se cargan por la parte trasera del automóvil, lo que ralentizaría el hecho de tener que reorganizar la mercancía (Cuervo et al., 2014).

El VRP con múltiples depósitos o MDVRP (*Multi-Depot VRP*), es aquel problema en el que se dispone de más de un depósito. Los vehículos están asociados a cada uno de ellos desde los que deben comenzar y finalizar las rutas. Si los clientes se encuentran distribuidos alrededor de los depósitos, el problema puede ser modelado como un conjunto de VRPs independientes. El MDVRP fue desarrollado en el trabajo presentado por Wren and Holliday (1972). Cabe señalar también el caso del SVRP (*Stochastic VRP*) o VRP estocástico. Esta derivación del problema clásico permite que ciertas variables del problema sean aleatorias. Las variables a las que nos referimos son: el número de clientes a los que hay que servir, la demanda de cada cliente y los tiempos de transporte. El primer autor en proponer el algoritmo para resolver el problema SVRP fue Tillman (1969).

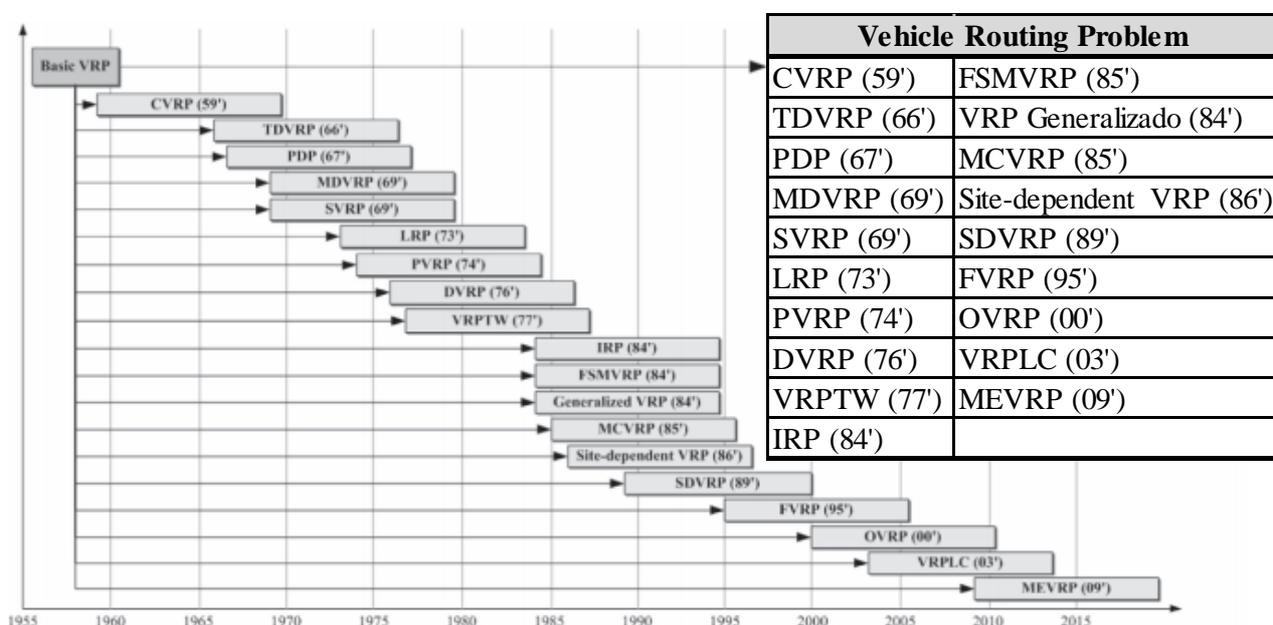


Figura 3 Cronología del VRP (Lin, C. et al., 2014)

Existen multitud de variantes y ramificaciones del problema VRP. En esta revisión de la literatura se ha pretendido mencionar aquellas que han adquirido mayor importancia en la literatura, pero no son las únicas. En la Figura 3 encontramos cronológicamente la evolución que ha seguido el problema, donde aparece el VRP con dependencia del tiempo (TDVRP) introducido por Cooke and Halsey (1966) en el que se asume que el tiempo de viaje entre dos clientes no es constante, sino que depende de factores externos (condiciones climáticas, horas punta, ...). El LRP (*Location Routing Problem*) consiste en dado uno o más depósitos situados en una localización definida y asignando un coste fijo de abrir cada depósito, diseña las rutas con el objetivo de minimizar dichos costes fijos (Watson-Gandy and Dohm, 1973). El VRP dinámico (DVRP, *Dinamic VRP*) incluye la incertidumbre del día a día en los problemas (tráfico, mecánica de vehículos, ...) saliendo del ambiente determinista y estático del problema convencional (Speidel, 1976). El VRP con múltiples compartimentos (MCVRP) donde los productos no son iguales. Cada cliente puede requerir diferentes productos, y la demanda de un producto por parte de un cliente no se puede dividir, aunque si es posible realizar varias visitas a un mismo cliente que requiere de más de un producto (Christophides, 1985). Por último, cabe mencionar el VRP con múltiples escalones (MEVRP) en el que el flujo de los materiales desde que salen del depósito inicial hasta que llegan al cliente final puede pasar por almacenes intermedios con el objetivo de minimizar el coste total de transporte. Es empleado, por ejemplo, en la distribución de prensa y en toda la industria actual de servicios de entregas a domicilio, servicios postales y e-commerce (Perboli et al., 2011).

2.3 VRP Enriquecido

La dificultad de los sistemas logísticos se ha visto acrecentada debido a la importancia que están adquiriendo en las empresas, promovidos por el rápido avance de la tecnología que ha logrado que todo el mundo esté conectado y que ha favorecido la creación de un mercado globalizado con complejas y exigentes redes de transporte (Osaba et al., 2017). Un ejemplo del importante papel que juega la logística lo podemos ver en el sector de la alimentación, en el que los costes de transporte pueden llegar a aumentar el precio del producto un 70% (Golden and Wasil, 1987).

El Problema de Ruteo de Vehículos (VRP) introducido por Dantzing y Ramser en 1959, ha sido un problema ampliamente estudiado en los últimos 50 años, tal y como se ha visto en el apartado anterior de este proyecto. El problema elemental es el VRP con restricciones de capacidad (CVRP), que tiene como objetivo minimizar costes diseñando rutas de reparto para servir a un conjunto de clientes distribuido en un espacio definido. Las rutas empiezan y terminan en el mismo depósito y se emplean para ello idénticos vehículos con límite de capacidad. Cada cliente debe ser servido por un único vehículo en el mismo periodo de tiempo.

Las investigaciones en torno al VRP han sido criticadas por estar demasiado centradas en modelos idealizados que no disponen de las restricciones suficientes para reflejar problemas reales. Sin embargo, la evolución tecnológica ha permitido resolver problemas relacionados con el VRP que antes eran considerados demasiado difíciles de tratar. Estos problemas son las variantes del VRP que son esenciales para el ruteo de vehículos en la vida real (Hartl et al., 2006).

La gama extremadamente grande de aplicaciones reales en las que se encuentran los problemas de enrutamiento conduce a la creación de VRPs con características y restricciones especiales, denominadas atributos, cuyo objetivo es obtener un mayor nivel de detalle del sistema que permitan tomar mejores decisiones en entornos reales. Estas restricciones complementan la formulación clásica del VRP y derivan en una nueva variedad de problemas de enrutamiento con múltiples atributos (MAVRP, *Multi – Attribute VRP*) o los problemas VRP enriquecidos (RVRP, *Rich VRP*) (Vidal et al. 2013). Estos atributos definen las cuatro grandes estructuras que caracterizan al problema:

- Estructuras del sistema: Número de depósitos, características de la flota de vehículos y tipos de producto.

- Requisitos del cliente: Periodo de planificación y ventanas de tiempo.
- Requisitos de operación del vehículo: Colocación de carga, restricciones de ruta en distancia o tiempo y reglas de trabajo del conductor.
- Contexto de decisión: Congestión del tráfico, planificación en periodos de tiempo prolongados.

El primer intento por definir el RVRP fue en Toth and Vigo (2002), donde los autores definen el potencial de extender la formulación del flujo de vehículos a unos modelos más flexibles, sin embargo, no existe una definición precisa que determine el significado o alcance del RVRP, ya que se trata de problemas que están en continuo cambio. Muchos investigadores aplican restricciones adicionales a los problemas básicos CVRP o añaden diferentes funciones objetivo para considerarlos enriquecidos (Rieck and Zimmermann, 2010).

Pellegrini (2007) sugiere que el término enriquecido está asociado con aquellos problemas que incluyen restricciones adicionales al VRP clásico para representar de manera más precisa casos aplicables en la vida real. Resuelve un problema RVRP aplicado a un caso real de una empresa de reparto de alimentos a restaurantes en Italia. El problema combina restricciones de ventanas de tiempo, flota de vehículos heterogénea, múltiples visitas a un mismo cliente y limitación de la duración de rutas. Es un problema multiobjetivo que busca completar todas las entregas con el menor número de vehículos posible, empleando para ello el menor tiempo como objetivo secundario.

Caceres-Cruz et al. (2014) concluyen que un RVRP refleja la mayoría de los atributos existentes en un sistema de distribución de rutas de vehículos en la vida real. Estos atributos incluyen dinamismo, estocasticidad, heterogeneidad, multiperiodos, integración con otras actividades relacionadas (Ej. gestión de inventarios, procesos de recogida), problemas contractuales y con el medio ambiente. De esta forma, un RVRP es una representación precisa de los sistemas de distribución en la vida real, pudiendo ser aplicadas directamente las soluciones obtenidas en escenarios reales.

Para ayudar a entender mejor estos problemas, Lahyani et al. (2015) crea una taxonomía con el objetivo, por un lado, de mostrar la literatura más relevante de RVRPs aplicados en la vida real, y por otro lado proporcionar una definición discriminatoria sobre el problema. Para desarrollar la taxonomía, enumera y clasifica los atributos en dos grandes grupos: según las características del escenario (*SCs, Scenario Characteristics*) y según las características físicas del problema (*PPCs, Problem Physical Characteristics*) (Tabla 1).

Los atributos a su vez pueden clasificarse en tres subniveles: estratégico, táctico y operacional. Las diferencias entre ellos dependerán del tipo de decisión que conllevan. Los niveles estratégicos y tácticos están asociados con las SCs ya que corresponden a estrategias del transporte que describen el sistema de distribución y diseño de sus principales componentes. El plan estratégico puede incluir decisiones relacionadas con la localización y el número de depósitos empleados, mientras que el nivel táctico define el tipo de orden y la frecuencia de visitas a los clientes en función del horizonte temporal establecido. El nivel operacional está asociado a las PPCs, en ellos se describe el plan de distribución incluyendo las restricciones de tiempo por parte del vehículo y conductor.

Para intentar establecer un consenso común del significado de estos problemas, Lahyani et al. (2015) llegó a la conclusión de que se considera un problema como RVRP a aquel con al menos cuatro atributos del tipo estratégico o táctico y que incluyera al menos seis restricciones relacionadas con características físicas del problema. Si el problema únicamente tiene atributos estratégicos o tácticos, debe tener al menos cinco para ser considerado RVRP. Si el problema está definido únicamente por restricciones físicas, al menos debe presentar nueve de ellas.

El objetivo de los problemas RVRP puede ser múltiple. Los más comunes son aquellos que minimizan el coste total de las rutas diseñadas reduciendo para ello: la distancia total recorrida, el tiempo empleado, el tamaño de la flota, etc. Pueden existir además otros objetivos, como son: maximizar la calidad del servicio o incrementar el beneficio máximo del servicio. Cuando existen múltiples objetivos en un mismo problema, se debe cuantificar la importancia de cada uno de ellos para evitar así conflictos entre objetivos.

Tabla 1 Clasificación de las características de los RVRP (Lahyani et al., 2015)

1.-Características del escenario (SC)	2.-Características físicas de los problemas (PPC)
1.1 Datos de entrada	2.1 Vehículos
1.1.1 Estático	2.1.1 Tipo
1.1.2 Dinámico	2.1.1.1 Homogénea
1.1.3 Determinista	2.1.1.2 Heterogénea
1.1.4 Estocástico	2.1.2 Número
1.2 Componente en la gestión de decisiones	2.1.2.1 Fija
1.2.1 Enrutado	2.1.2.2 Ilimitada
1.2.2 Inventario y enrutado	2.1.3 Estructura
1.2.3 Localización y enrutado	2.1.3.1 Compartimentada
1.2.4 Enrutado y planificación de conductores	2.1.3.2 No compartimentada
1.2.5 Planificación de la producción y distribución	2.1.4 Restricciones de capacidad
1.3 Número de depósitos	2.1.5 Política de carga en vehículo
1.3.1 Único	2.1.5.1 Orden cronológico
1.3.2 Múltiples	2.1.5.2 Sin criterio
1.4 Tipo de operación	2.1.6 Reglamentos en los conductores
1.4.1 Recogida o entrega	2.2 Restricciones de tiempo
1.4.2 Recogida y entrega	2.2.1 Restricciones en clientes
1.4.3 Backhauls	2.2.2 Restricciones en acceso a carreteras
1.4.4 Dial-a-ride	2.2.3 Restricciones en depósitos centrales
1.5 Restricciones en la división de la entrega de la carga	2.2.4 Tiempos de servicio
1.5.1 Permitidas	2.2.5 Tiempos de espera
1.5.2 No permitidas	2.3 Estructura de las ventanas de tiempo
1.6 Periodo de planificación	2.3.1 Únicas
1.6.1 Único	2.3.2 Múltiples
1.6.2 Múltiple	2.4 Restricciones de incompatibilidades
1.7 Uso múltiple de vehículos	2.5 Restricciones específicas de los problemas
1.7.1 Único viaje	2.6 Función objetivo
1.7.2 Múltiple	2.6.1 Única
	2.6.2 Múltiples

El VRP Generalizado (GVRP) aparece en Goel and Gruhn (2006), como un problema combinado entre aceptación de cargas y enrutamiento de vehículos y supone una unión entre problemas independientes. En este artículo se resuelve un caso real de un problema en el transporte de carga aérea en el que se introducen restricciones de ventanas de tiempo, flota de vehículos heterogénea con diferentes tiempos de viaje, costes de viaje y capacidad, restricciones de capacidad, restricciones de compatibilidad entre vehículo – orden, órdenes de entrega y recogida, restricciones de rutas para vehículos y diferente localización para los vehículos al comienzo y finalización de las rutas.

El término VRP dinámico o también conocido como VRP en tiempo real, puede ser considerado dentro del alcance de los problemas RVRP. Estos problemas incluyen incertidumbre en las variables (Número de clientes, tiempo de viaje y demandas) generando buenas soluciones ante cualquier cambio producido en las variables de una manera rápida (Pillac et al., 2013).

Oppen et al. (2010) presenta un problema de enrutamiento de vehículos con el objetivo de construir rutas para recoger animales de las granjas para su sacrificio en un matadero. Al problema clásico de ruteo de le añaden atributos como: flota heterogénea de vehículos, múltiples viajes por vehículo y restricciones relacionadas con el bienestar animal, como la duración de los viajes y división de los animales en categorías (edad, género y tamaño).

En Amorim et al. (2014) se resuelve un problema enriquecido de ruteo de vehículos tipo HFSDVRPMTW (*Heterogeneous Fleet Site-Dependent VRP with Multiple Time Windows*) en una empresa portuguesa de distribución diaria de comida. El problema cuenta con los atributos de flota heterogénea de vehículos tipo “*Site – Dependent*” y múltiples ventanas de tiempo. Se obtiene un algoritmo flexible que permite la introducción de nuevos clientes y productos, en el que comparándolo con otras compañías el tiempo de planificación de las rutas se ve reducido considerablemente.

En Sicilia et al. (2016) se presenta el problema enriquecido de ruteo de vehículos para la distribución de bienes en áreas urbanas, teniendo en cuenta atributos de la vida real como: ventanas de tiempo, restricciones de capacidad, compatibilidad entre órdenes y vehículos, maximización del número de órdenes por vehículo, órdenes de entrega y recogida y posibilidad de no retornar al depósito. El objetivo es reducir el coste minimizando las distancias y empleando el menor número de vehículos manteniendo la calidad del servicio.

Actualmente las investigaciones están dirigidas en la misma dirección de los problemas RVRP. Diseñando métodos de resolución de rutas de vehículos en problemas reales, incorporando para ello todos los factores posibles existentes en el entorno del problema y pudiendo ser aplicados directamente en la vida real.

3 DESCRIPCIÓN DEL PROBLEMA DE DISEÑO DE RUTAS EN EMPRESAS QUE PRESTAN SERVICIOS DE CONTROL METROLÓGICO

3.1 Problema MCSVRP

Introducimos el término MCSVRP (*Metrological Control Services Vehicle Routing Problem*) para referirnos al problema caracterizado por el diseño y planificación de rutas de vehículos en empresas que prestan un servicio de control metrológico de equipos de medida, en el que partiendo de un depósito central los vehículos deben desplazarse a las instalaciones del cliente para dar el servicio. Este tipo de problemas pertenecen al grupo de problemas RVRP que combinan una serie de atributos presentes en problemas de la vida real. Los atributos que van a definir el problema MCSVRP son los siguientes:

- Flota de vehículos fija y heterogénea
- Ventanas de tiempo para clientes y depósito
- Sincronización entre Recursos - Rutas
- Restricciones entre Conductor - Cliente
- Restricción entre Vehículo - Cliente
- Prioridades entre clientes
- Clientes no servidos

El MCSVRP supone una particularización del problema m-HVRPTW, variante del problema HVRPTW que introduce la posibilidad de que en la planificación de rutas queden clientes sin ser atendidos en las m rutas planificadas, a diferencia del HVRPTW en el que todos los clientes deben ser asignados a una de las rutas satisfaciendo las restricciones del problema. Para ello, los atributos mencionados anteriormente, que caracterizan a las empresas que prestan servicios de control metrológico, se incorporan al problema m-HVRPTW.

El propósito de este trabajo es poder definir el problema estándar de empresas que dan servicios de control metrológico para así poder aplicarlo en cualquier otro tipo de empresa que tenga similitudes con la estudiada.

3.1.1 Operativa de una empresa que proporciona servicios de control metrológico

Una empresa que presta un servicio de control metrológico se define como aquella que partiendo de un depósito o sede central, proporciona un servicio de control metrológico de equipos de medida a un conjunto de clientes esparcidos de forma aleatoria en un territorio considerando los atributos normalmente presentes en este tipo de servicios.

El flujo del proceso comienza cuando la empresa recibe órdenes de trabajo por parte de los clientes y finaliza una vez realizado el servicio y vuelto al laboratorio. Las órdenes de trabajo se clasificarán en función de una serie de factores que se definen a continuación y que condicionan el diseño de cada una de las rutas.

- Tipo de Servicio: Denominamos tipo de servicio a cada una de las tareas que la empresa puede desempeñar. El tipo de servicio en cada cliente variará, ya que se trabaja con clientes de diferentes sectores y tecnologías.
- Grupo de servicio: La forma de clasificar las diferentes tareas que llegan de los clientes es mediante tecnología / especialización de las mismas, en lo que llamamos Grupos de servicio. Cada grupo de servicio engloba una serie de tareas similares en cuanto a tecnología y conocimientos necesarios para desempeñarlas.
- Vehículos: La empresa posee una flota de vehículos heterogénea. Cada vehículo se asignará a las diferentes tareas en función de la maquinaria necesaria en cada una de ellas. Cabe la posibilidad de que haya tareas que no puedan ser desempeñadas por ciertos vehículos debido a que el vehículo no esté equipado con el material necesario para completar dichas tareas. También se pueden añadir restricciones que permitan priorizar ciertos vehículos sobre otros, evitando así utilizar vehículos de gran tonelaje en tareas que no lo requieran.
- Tiempo de servicio: Cada tipo de servicio tiene asociado un tiempo de ejecución. Este tiempo consideramos que es constante y no es modificado por factores externos, como problemas de tráfico o roturas de materiales.
- Verificadores: Llamamos verificadores a aquellas personas encargadas de realizar el servicio en los diferentes clientes. Cada verificador está especializado en un grupo de servicio, por lo que se deben diseñar rutas acordes a la especialización de cada verificador.
- Patrones: Cada tipo de servicio requiere de una serie de patrones que son necesarios para cumplimentar el servicio. El número de patrones es limitado por lo que se deben crear rutas compatibles en cantidad en un mismo periodo de tiempo. El tipo de patrón necesario en cada servicio limita el tipo de vehículo a utilizar.

La empresa recibe semanalmente órdenes de trabajo que debe clasificar en función de la tecnología y los recursos disponibles para planificar las rutas teniendo en cuenta todos los atributos mencionados. El objetivo principal es dar servicio al mayor número de clientes posibles recorriendo para ello el menor número de kilómetros.

3.1.2 Atributos del MCSVRP

Las características del problema MCSVRP mostradas en el apartado anterior, se deben tener en cuenta a la hora de diseñar el algoritmo que resuelve el problema mediante los siguientes atributos:

- Flota fija heterogénea: El VRP Heterogéneo con un número limitado de vehículos (HVRP) fue propuesto por Taillard (1999), consiste en optimizar un conjunto de rutas para una flota fija de vehículos con diferentes características de capacidad y costes y que sirven a los clientes partiendo desde un depósito central con una demanda conocida. Cada cliente es visitado una sola vez por un vehículo, y todos los vehículos comienzan y terminan sus rutas en el depósito central. Un ejemplo del uso de flotas Heterogéneas lo podemos encontrar en Penna et al. (2013).
- Ventanas de tiempo: Cuando se consideran ventanas de tiempo duras, se introducen nuevas restricciones en el problema que impiden que un vehículo pueda servir a un cliente fuera de un espacio de tiempo $[EW_i, LW_i]$, donde EW_i y LW_i comprenden el tiempo más temprano y el más tardío en el que un cliente i puede ser servido. Si un vehículo llega antes de EW_i , se debe tener en cuenta un tiempo de espera correspondiente a la apertura de esa ventana de tiempo, a su vez, si un vehículo llega más tardes de LW_i la solución no es factible. Un ejemplo de resolución de un problema HVRP con ventanas de tiempo (HVRPTW) se encuentr en Buhrkal et al. (2011). En el problema en estudio se va a considerar una restricción de tiempo en la que cada empleado no podrá realizar rutas con una duración mayor a 8 horas.
- Restricciones de patrones: Para realizar los diferentes servicios a los clientes, se emplean una serie de patrones o herramientas que dependerán en cantidad y tipo del servicio a desempeñar. Estos patrones son recursos limitados y por tanto será necesario diseñar rutas que satisfagan servicios que compartan los mismos patrones. Será necesaria una sincronización entre vehículos que impida superar la cantidad de patrones de la que se dispone en un periodo de tiempo. Puede existir incompatibilidad entre patrones que impiden que puedan ir juntos en un mismo vehículo. Drexl (2012) presenta un elaborado estudio sobre problemas de ruteo de vehículos con restricciones de sincronización.
- Restricciones Conductor – Orden: En el problema en estudio se va a considerar diferentes tipos de órdenes de trabajo, las cuales van a requerir de diferentes cualificaciones por parte de los conductores para poder realizarlas. No todos los conductores van a disponer de todas las cualificaciones. Dentro de esa certificación existirán diferentes tareas posibles. Por tanto, será necesario asignar órdenes a conductores en función del tipo de cualificación que éste posea. Un ejemplo de aplicación de este tipo de restricciones lo podemos encontrar en Rieck and Zimmermann (2010).
- Restricciones Vehículo – Orden: La flota de vehículos es heterogénea, la elección de uno u otro para la realización de las tareas de una ruta va a depender del tipo de patrón que se emplee en dicha ruta. No todos los vehículos pueden transportar todos los patrones, existe una serie de limitaciones de capacidad y preferencias en la elección del mismo en función del tamaño de los patrones. Un ejemplo de aplicación de este tipo de restricciones lo podemos encontrar en Oppen and Løkketangen (2008).
- Prioridad en las Órdenes: Al tratarse de una generalización de problemas de empresas destinadas a ofrecer un tipo de servicio, nos podemos encontrar en el caso en el que la carga de trabajo sea muy elevada y por tanto tengamos que priorizar ciertas órdenes sobre otras. La priorización de ciertas órdenes puede estar asociado a la importancia de un cliente específico que necesitamos que sea

satisfecho primero. Algunos estudios de la literatura incluyen prioridades en los clientes, por ejemplo, Bravo et al. (2009), enfocado en la gestión adecuada del despacho de mercancías.

- Órdenes no servidas: Las compañías de servicios tienen demandas variables en el tiempo, llegando a alcanzar picos de trabajo que no pueden ser satisfechos en el periodo de planificación actual y que por tanto deben ser pospuestos al periodo siguiente. Para tratar este tipo de problemas se crean las conocidas como listas de espera (Holding List) que contienen aquellas órdenes que no pueden ser servidas en un periodo de tiempo. Esta idea fue inicialmente introducida por Lau et al. (2003) y implementada por otros autores como Jiang et al. (2014).

Aunque es difícil encontrar en la literatura un problema que cumpla exactamente con las características del MCSVRP, sí encontramos estudios que combinan varios atributos que comparten con nuestro problema. Golden et al. (1984) desarrolla una heurística para resolver el problema de ruteo de una flota heterogénea e ilimitada de vehículos (FSMVRP). Posteriormente, en Liu y Shen (1999) se describe el problema FSMVRPTW que se caracteriza por tener ventanas de tiempo y una flota ilimitada y heterogénea de vehículos.

En Taillard (1999) se presenta una heurística en la que se aplica el método de generación de columnas para resolver el problema de ruteo con una flota fija y heterogénea de vehículos (HFFVRP), el método fue empleado también para resolver el problema FSMVRP.

La versión del problema anterior con flota limitada de vehículos homogéneos y ventanas de tiempo sería la conocida como m-VRPTW introducida por Lau et al. (2003) donde aplica un enfoque de búsqueda Tabú con lista de espera.

Existe multitud de estudios en la literatura que versan sobre problemas con ventanas de tiempo y flotas homogéneas y heterogéneas, sin embargo, es difícil encontrar en la literatura un problema que trate sobre el problema de ruteo con flota fija y heterogénea de vehículos junto con ventanas de tiempo (m-HVRPTW). En Baldacci et al. (2012) se presenta una revisión de las formulaciones matemáticas del problema VRPTW y sus variantes.

El problema de ruteo de vehículos en empresas de control metrológico ha sido previamente estudiado en Molina et al. (2018), donde se aplica en un caso real una metaheurística de vecindad descendente basado en una búsqueda tabú con lista de espera. El resultado de la metaheurística es comparado con los resultados que se obtendrían con una flota fija y heterogénea de vehículos. Los resultados obtenidos mejoran la solución implementada por la compañía.

3.2 Modelado del problema MCSVRP

El problema en estudio trata de resolver el problema de ruteo de vehículos para las empresas que prestan servicios de control metrológico (MCSVRP). Este tipo de problemas, como hemos definido en apartados anteriores, cuentan con atributos específicos que caracterizan a este tipo de compañías. Estos atributos son: Flota de vehículos heterogénea, ventanas de tiempo, multi periodo, lista de espera y sincronización de recursos.

Sabemos que los métodos exactos no son apropiados para resolver problemas con un gran número de nodos y restricciones como es el problema en cuestión, ya que los tiempos de computación en problemas con muchos atributos crece de manera exponencial conforme aumenta el número de variables. A pesar de ello, se muestra a continuación el modelo matemático de programación lineal, ya que resulta de utilidad para llegar a entender las variables empleadas y los diferentes tipos de restricciones.

El problema se define como:

- $G = \{\mathcal{N}, \mathcal{A}\}$: siendo $\mathcal{N} = \{0, 1, \dots, N\}$ un conjunto de nodos para el periodo de planificación en estudio, considerando $\mathcal{N} = 0$ como depósito central y \mathcal{A} representando al conjunto de arcos que une los diferentes nodos.
- $Z = \{1, \dots, K, K + 1\}$: representa a la flota heterogénea de vehículos K disponibles en el depósito central, siendo $K + 1$ un vehículo ficticio al que se le asignarán las tareas que no pueden ser satisfechas en un periodo de tiempo.
- $\mathcal{S} = \{1, \dots, S\}$: son los verificadores que han de asignarse a los vehículos en función del servicio a realizar.
- $\mathcal{P} = \{1, \dots, P\}$: conjunto de patrones necesarios para desempeñar un tipo de servicio.
- $\mathcal{T} = \{1, \dots, T\}$: represental el horizonte de planificación expresado en días.

La notación empleada para definir las diferentes variables es la siguiente:

- PR_i : prioridad en el nodo i .
- $[EW_i, LW_i]$: hora más temprana y tardía para el comienzo del servicio en el nodo i .
- ST_i : tiempo de servicio en el nodo i .
- TD_{ij} : distancia del nodo i al nodo j ($i \neq j$).
- TT_{ij} : tiempo de viaje entre los nodos i y j .
- $TMax_s$: hora máxima permitida para la llegada al depósito del verificador s .
- NJ_s : número máximo de jornadas laborales permitidas para el verificador s .
- SW_{is} : igual a 1 si el verificador s puede realizar el servicio en el nodo i , 0 en caso contrario.
- WJ_{st} : igual a 1 si el verificador s puede trabajar el día t , 0 en caso contrario.
- SV_{ik} : igual a 1 si el nodo i puede ser atendido con el vehículo k , 0 en caso contrario.
- NP_p : número de patrones p disponibles diariamente.
- SP_{ip} : igual a 1 si el nodo i requiere del patrón p para ser atendido, 0 en caso contrario.

Se emplean las siguientes variables de decisión:

- X_{ijkt} : variable binaria igual a 1 si el vehículo $k \in \{1, \dots, K+1\}$ viaja del nodo i al j el día t , siendo ($i \neq j$); $X_{ijK+1t} = 1$ representa la lista de órdenes i no atendidas en el horizonte de planificación especificado
- Y_{ikt} : tiempo de comienzo del servicio en el nodo $i \in \{0, 1, \dots, N\}$ realizado por el vehículo $k \in \{1, \dots, K\}$ el día t ; y_{0kt} es la hora de llegada al laboratorio central.

- Z_{skt} : variable binaria, igual a 1 si el verificador $s \in \{1, \dots, S\}$ se asigna al vehículo $k \in \{1, \dots, K\}$ el día t .
- W_{pkt} : variable binaria, igual a 1 si el patrón $p \in \{1, \dots, P\}$ se asigna al vehículo $k \in \{1, \dots, K\}$ el día t .

De acuerdo con las hipótesis establecidas anteriormente para el problema de empresas que prestan un servicio de control metrológico, las restricciones del modelo de programación lineal entera mixta son las siguientes:

$$\sum_{j=1}^N X_{0jkt} \leq 1 \quad \forall k \neq K+1, \forall t \quad (3.1)$$

$$\sum_{\substack{j=0 \\ j \neq i}}^N X_{jikt} - \sum_{\substack{j=0 \\ j \neq i}}^N X_{ijkt} = 0 \quad \forall i, \forall k \neq K+1, \forall t \quad (3.1)$$

$$\sum_{t=1}^T \sum_{k=1}^{K+1} \sum_{\substack{j=0 \\ j \neq i}}^N X_{ijkt} = 1 \quad \forall i \neq 0 \quad (3.2)$$

$$Y_{ikt} + ST_i + TT_{ij} \leq Y_{jkt} + \left(\sum_{s=1}^S TMax_s \right) \cdot (1 - X_{ijkt}) \quad \forall i \neq 0, \forall j; i \neq j, \forall k \neq K+1, \forall t \quad (3.3)$$

$$TT_{0j} \leq Y_{jkt} + \left(\sum_{s=1}^S TMax_s \right) \cdot (1 - X_{0jkt}) \quad \forall j \neq 0, \forall k \neq K+1, \forall t \quad (3.4)$$

$$EW_i \leq Y_{ikt} \leq LW_i \quad \forall i \neq 0, \forall k \neq K+1, \forall t \quad (3.5)$$

$$\sum_{j=1}^N X_{0jkt} = \sum_{s=1}^S Z_{skt} \quad \forall k \neq K+1, \forall t \quad (3.6)$$

$$Y_{0kt} \leq TMax_s + \left(\sum_{r=1}^S TMax_r \right) \cdot (1 - Z_{skt}) \quad \forall s, \forall k \neq K+1, \forall t \quad (3.7)$$

$$\sum_{t=1}^T \sum_{k=1}^K Z_{skt} \leq NJ_s \quad \forall s \quad (3.8)$$

$$\sum_{k=1}^K Z_{skt} \leq WJ_{st} \quad \forall s, \forall t \quad (3.9)$$

$$\sum_{t=1}^T \sum_{\substack{j=0 \\ j \neq i}}^N X_{jikt} \leq SV_{ik} \quad \forall i \neq 0, \forall k \neq K+1 \quad (3.10)$$

$$\sum_{\substack{j=0 \\ j \neq i}}^N X_{jikt} + 1 - SW_{is} \leq 1 + 1 - Z_{sk} \quad \forall i \neq 0, \forall k \neq K + 1, \forall s, \forall t \quad (3.11)$$

$$\sum_{\substack{j=0 \\ j \neq i}}^N X_{jikt} + SP_{ip} \leq 1 + W_{pkt} \quad \forall i \neq 0, \forall k \neq K + 1, \forall p, \forall t \quad (3.12)$$

$$\sum_{k=1}^K W_{pkt} \leq NP_p \quad \forall p, \forall t \quad (3.13)$$

$$\sum_{p=1}^P W_{pkt} \leq P \cdot \sum_{j=1}^N X_{0jkt} \quad \forall k \neq K + 1, \forall t \quad (3.14)$$

$$\sum_{i=0}^N Y_{ikt} \leq N \cdot \left(\sum_{r=1}^S TMax_r \right) \cdot \sum_{j=1}^N X_{0jkt} \quad \forall k \neq K + 1, \forall t \quad (3.15)$$

A continuación, se explican detalladamente el significado de cada una de las restricciones empleadas:

[3.1] Cada vehículo K realiza como máximo una ruta al día, por lo que sale del depósito una sola vez.

[3.2] Si un vehículo K entra en el nodo j, debe salir de él en el mismo día t.

[3.3] Cada cliente es visitado una sola vez por el vehículo K en el periodo de planificación.

[3.4] y [3.5] Restricciones de tiempo que evitan las sub-rutas y calculan el instante de llegada de un vehículo a un nodo. Impide el comienzo del servicio en un nodo si la suma entre el acumulado más el tiempo de servicio y tiempo de viaje entre nodos sobrepasa el total establecido para esa ruta, siendo y_{0kt} la hora de llegada al laboratorio central.

[3.6] Restricción de ventanas de tiempo. El tiempo de comienzo del servicio en el nodo i debe estar comprendido entre la hora más temprana y tardía para comenzar [EW_i, LW_i].

[3.7] El número de vehículos que sale del depósito central coincide con el número de verificadores a los que se les asigna una ruta en el día t.

[3.8] Limitación del tiempo máximo de llegada al depósito para cada verificador.

[3.9] Al tratarse de un problema con multiples periodos de tiempo, la restricción 9 restringe el número máximo de días trabajados por el verificador s en un periodo de planificación.

- [3.10] Cada verificador puede ser asignado a un solo vehículo en un día de trabajo permitido.
- [3.11] Únicamente se incluye en una ruta realizada por el vehículo k al nodo i si dicho nodo puede ser atendido por el vehículo K .
- [3.12] Solo se asigna una ruta a un verificador si éste puede desempeñar el servicio.
- [3.13] El vehículo K debe llevar el el patrón P para realizar una tarea que lo requiera.
- [3.14] Limita la cantidad de patrones disponibles para poder ser empleados en un día t .
- [3.15] Evita la asignación de patrones a aquellos vehículos que no tienen planificada una ruta en un día.
- [3.16] Evita realizar un cálculo del tiempo de inicio del servicio a aquellos vehículos que no tienen planificada una ruta.

El problema a considerar pertenece a la categoría de problemas multiobjetivo, con una función objetivo jerarquizada, donde el objetivo principal es maximizar el número de clientes servidos en un periodo de planificación y el objetivo secundario minimizar la distancia total recorrida. Se introduce en la función objetivo (3.17) el término ρ al que se le da un valor positivo de pequeña magnitud, con el fin de escoger la solución de menor distancia total recorrida ante soluciones con el mismo número de clientes servidos.

$$MAX \sum_{t=1}^T \sum_{k=1}^K \sum_{\substack{j=1 \\ i \neq j}}^N \sum_{i=0}^N PR_j X_{ijkt} - \rho \cdot \sum_{t=1}^T \sum_{k=1}^K \sum_{\substack{j=0 \\ i \neq j}}^N \sum_{i=0}^N TD_{ij} X_{ijkt} \quad (3.16)$$

4 METODOLOGÍA DE RESOLUCIÓN

4.1 Introducción a los métodos de resolución

El problema de enrutamiento de vehículos forma parte de la categoría de problemas conocida como NP-Hard (Garey y Johnson, 1979), cuya principal característica es la dificultad para encontrar soluciones óptimas en tiempos computacionales aceptables, ya que se deberían evaluar todas las rutas posibles para hallar la solución óptima. El VRP combina dos tipos de problemas de optimización combinatoria, el Bin Packing Problem (BPP) que asigna carga a los vehículos, y el Travelling Salesman Problem (TSP) que optimiza las rutas de los vehículos.

La versión clásica del problema busca minimizar la distancia total recorrida por vehículos con capacidad limitada de carga, cada destino debe ser visitado una sola vez en cada periodo de planificación. Sin embargo, como ya hemos visto en apartados anteriores, existen multitud de versiones adicionales que incluyen diferentes atributos como las ventanas de tiempo, recogida y entrega de mercancías, vehículos heterogéneos, ... incluso modificaciones en la función objetivo. En Jaramillo (2011) se introduce el término Green Vehicle Routing Problem (GVRP) donde se busca minimizar las emisiones de CO₂, para ello se incluye una función objetivo que minimiza la relación entre peso-distancia recorrida por los vehículos. Una revisión detallada de las diferentes versiones de enrutamiento que podemos encontrar se muestra en Marinakis y Migdalas (2007).

Los métodos de resolución que se emplean en la literatura para resolver problemas de ruteo de vehículos dependen principalmente del tamaño y las restricciones que tengan. Existen tres grandes categorías en las que se pueden agrupar estos métodos: Métodos exactos, heurísticas y metaheurísticas.

Un problema VRP clásico con hasta 50 clientes y restricciones simples puede ser resuelto mediante métodos exactos. Uno de los métodos exactos más conocido es el Branch and Bound, aplicado en Christofides et al. (1981) para resolver un problema de ruteo de vehículos con restricciones de distancia. Si se añaden restricciones adicionales o crece el número de clientes, será necesario aplicar heurísticas o metaheurísticas para obtener soluciones aproximadas.

Las heurísticas y metaheurísticas se caracterizan principalmente por tres aspectos: proporcionan soluciones óptimas en un periodo corto de tiempo, son capaces de resolver problemas de gran complejidad y obtienen algoritmos robustos. Los métodos heurísticos proporcionan una buena solución en un corto periodo de tiempo, aunque dicha solución no tiene por qué ser la óptima del problema ya que la exploración del espacio de búsqueda es limitada. Clarke and Wright propusieron en 1964 el primer algoritmo efectivo para resolver el problema del VRP.

Los algoritmos metaheurísticos resuelven problemas tipo NP-Hard mediante algoritmos de aproximación. Estos algoritmos permiten encontrar buenas soluciones en tiempos de computación aceptables. Algunos de los algoritmos utilizados en la literatura para resolver problemas de ruteo con capacidad limitada son: Recocido Simulado, Búsqueda Tabú, Colonia de Hormigas y Algoritmos Genéticos.

En este trabajo nos centraremos en los conocidos como Algoritmos Meméticos introducidos por Moscato (1989). Estos algoritmos combinan la búsqueda global responsable de explorar el espacio, con una búsqueda local que intensifica la explotación de aquellas buenas soluciones encontradas. Este tipo de metaheurísticas proporcionan soluciones más precisas que en los métodos anteriores.

4.2 Algoritmos Meméticos

Los Algoritmos Meméticos (MAs) son técnicas de optimización que constituyen una clase de metaheurísticas que combinan una búsqueda global basada en la población, responsable de realizar una exploración del espacio de búsqueda, con técnicas de búsqueda local destinadas a una mayor exploración (explotación) de aquellas regiones consideradas prometedoras, lo que es útil para lograr soluciones más precisas (Salmeron et al., 2017). Esta combinación permite algoritmos más versátiles y eficaces.

Aunque una gran proporción de MAs empleados en la literatura son una combinación entre algoritmos evolutivos, como los algoritmos genéticos (Goldberg, 1989) y estrategias evolutivas (Beyer & Schwefel, 2002) combinados con técnicas de búsqueda local, existen numerosos MAs en la literatura que derivan de otras metaheurísticas basadas en poblaciones de individuos como el ACO (Dorigo & Birattari, 2010), ACS (Chen et al., 2013), optimización de enjambres de partículas (Kennedy, 2006) y colonia de abejas (Karaboga, 2005).

Los orígenes de los MAs se remontan a finales de los años ochenta con la idea de combinar conceptos y estrategias de diferentes metaheurísticas para aunar las ventajas de las mismas (Moscato 1989). La denominación “memético”, surge del término inglés “meme”, acuñado por R. Dawkins como el análogo del gen en el contexto de la evolución cultural (Moscato et al., 2003).

Partiendo de un problema de optimización, se crea un cierto número de individuos, ya sea de forma aleatoria o de acuerdo a un determinado procedimiento de inicialización, aplicando una heurística para inicializar la población. A partir de esto, cada individuo realiza una búsqueda local. El objetivo de la búsqueda local puede ser alcanzar el óptimo local o mejorar la Función Objetivo hasta unos niveles predefinidos. Una vez que cada individuo ha alcanzado un cierto desarrollo, interactúa con el resto de los miembros de la población.

La interacción puede ser competitiva o cooperativa. La competencia selecciona a los mejores individuos, mientras que la cooperación resulta del apareamiento o intercambio de información entre miembros. El comportamiento cooperativo se puede entender como los mecanismos de cruce en los Algoritmos Genéticos que dan como resultado la creación de nuevos individuos. La búsqueda local y la posterior cooperación o competencia se repiten hasta que se cumpla un criterio de detención establecido (Moscato 1989).

En este trabajo se propone, para resolver el problema MCSVRP anteriormente descrito, una solución que consiste en hibridar un Sistema de Colonia de Hormigas (ACS) con una búsqueda local, llamando a esta combinación Algoritmo Memético ACS. El algoritmo memético ACS busca mejorar la calidad de las soluciones mediante el uso de un algoritmo híbrido de búsqueda de entornos variables y una búsqueda tabú (VNTS) para la exploración de la vecindad con lista de espera. Este algoritmo combina dos metaheurísticas (ACS y VNTS) que han sido aplicadas con éxito por separado en una gran variedad de VRPs. La ventaja de utilizar el ACS es la generación de buenas soluciones iniciales, por tanto la posterior búsqueda local requerirá de un menor número de iteraciones para alcanzar el óptimo. De esta manera, el tiempo de ejecución del algoritmo será mucho menor que si partiéramos de una solución inicial aleatoria.

El problema MCSVRP no ha sido definido previamente como tal en la literatura, a excepción del artículo Molina et al. (2018) que estudia el mismo problema aplicando una metodología de resolución diferente pero que comparte los mismos atributos. Únicamente se tiene conocimiento de artículos previos a los que se les haya aplicado el Algoritmo memético ACS combinado con un VNTS en Molina et al. (2020) que resuelve el problema HVRPTW-LR con dicho algoritmo. Por estos motivos, el estudio de la literatura realizada a continuación pretende mostrar ejemplos de problemas que comparten ciertos atributos y que han sido resueltos con algoritmos meméticos.

Posada (2013), aplica un algoritmo memético para resolver el problema de enrutamiento de vehículos con capacidad limitada. El algoritmo empleado integra un algoritmo evolutivo inspirado en la mutación viral con los principios de la búsqueda tabú.

De la Cruz et al. (2013) propone un algoritmo híbrido entre un sistema de colonia de hormigas (ACS) con una búsqueda Tabú para resolver el problema VRP con una flota heterogénea de vehículos con ventanas de tiempo y múltiples productos (HVRPTWMP). El objetivo del problema es minimizar la distancia total recorrida a la vez que se reduce el tamaño total de la flota de vehículos empleados en las rutas.

Abdulkader et al. (2015) resuelve un problema de ruteo de vehículos con múltiples compartimentos aplicando un algoritmo de colonia de hormigas hibridizado con una búsqueda local. Los resultados obtenidos los compara con el mismo problema sin aplicar la búsqueda local, observándose mejores resultados en el nuevo algoritmo.

En de Armas (2015) se trata un problema enriquecido de ruteo de vehículos con atributos similares a los estudiados en este proyecto. Los atributos que tiene en consideración son: Flota fija y heterogénea de vehículos, ventanas de tiempo, prioridad entre clientes, posibilidad de clientes no servidos y restricciones de compatibilidad entre vehículo y cliente. El método empleado para resolver el problema es un algoritmo VNS (Variable Neighborhood Search) el cual es aplicado a un problema real de una compañía en las Islas Canarias. El algoritmo VNS había sido aplicado previamente en otros trabajos pero ninguno de ellos trataba un problema de ruteo enriquecido con ventanas de tiempo.

Sicilia et al. (2016) presenta un algoritmo de optimización basado en un VNS al que se le añade una Búsqueda Tabú con el objetivo de alcanzar el óptimo local de la solución, para resolver un problema del tipo RVRP. Las características del problema son: Ventanas de tiempo, restricciones de capacidad, compatibilidad entre órdenes y vehículos, maximizar el número de órdenes por vehículo, órdenes con entrega y recogida y la posibilidad de no volver al depósito. El objetivo del problema es reducir el coste minimizando el número de vehículos empleados manteniendo la calidad en el servicio.

En Molina et al. (2018) implementa, para el problema de ruteo de vehículos en empresas que dan un servicio de control metrológico a sus clientes, un algoritmo de vecindad descendente (VND) junto con una búsqueda tabú con listas de espera.

4.3 Sistema de Colonia de Hormigas (ACS)

El ACS (Ant Colony System) es un algoritmo que fue presentado por primera vez por Dorigo y Gambardella (1997). El ACS es una evolución de la metaheurística *Ant Colony Optimization* (ACO) presentada por primera vez por Marco Dorigo en 1992 en su tesis doctoral y que aplicó a un problema TSP. Respecto al algoritmo original, el ACS introduce tres principales cambios: modificación de la función de selección (exploración-explotación), actualización global y actualización local de la feromona. Estas funciones se verán en los siguientes puntos.

Los sistemas basados en colonia de hormigas son metaheurísticas diseñadas para resolver problemas de optimización combinatoria, inspirados en los rastros de feromonas que las hormigas utilizan para comunicarse entre ellas. Este tipo de algoritmos están basados en el comportamiento de enjambres, donde una población de agentes (hormigas) interactúan localmente entre ellos y el medio ambiente, alcanzando un comportamiento global complejo sin tener una estructura de control centralizado.

Las hormigas son capaces de encontrar el camino más corto desde el nido hasta la comida sin necesidad de señales visuales, para ello se guían mediante un rastro (feromona) que van dejando a su paso. La feromona es volátil, por lo que si una ruta no es tomada por las hormigas en un cierto periodo de tiempo desaparece el rastro. Por el contrario, cuantas más hormigas recorran una ruta, más feromonas segregan y por tanto el estímulo provocado por ese camino será mayor.

Como se muestra en la Figura 4, inicialmente cada hormiga toma una ruta de manera aleatoria (b). El camino inferior es más corto, por lo que las hormigas tardan menos tiempo en llegar al extremo contrario dejando el rastro de feromonas a su paso (c). El siguiente grupo de hormigas que inician la ruta se decantarán por el camino inferior, ya que el rastro de feromonas es mayor (d). Finalmente, la totalidad de la colonia se decantará por el camino inferior, más corto y rápido, y el superior perderá el rastro de feromonas.

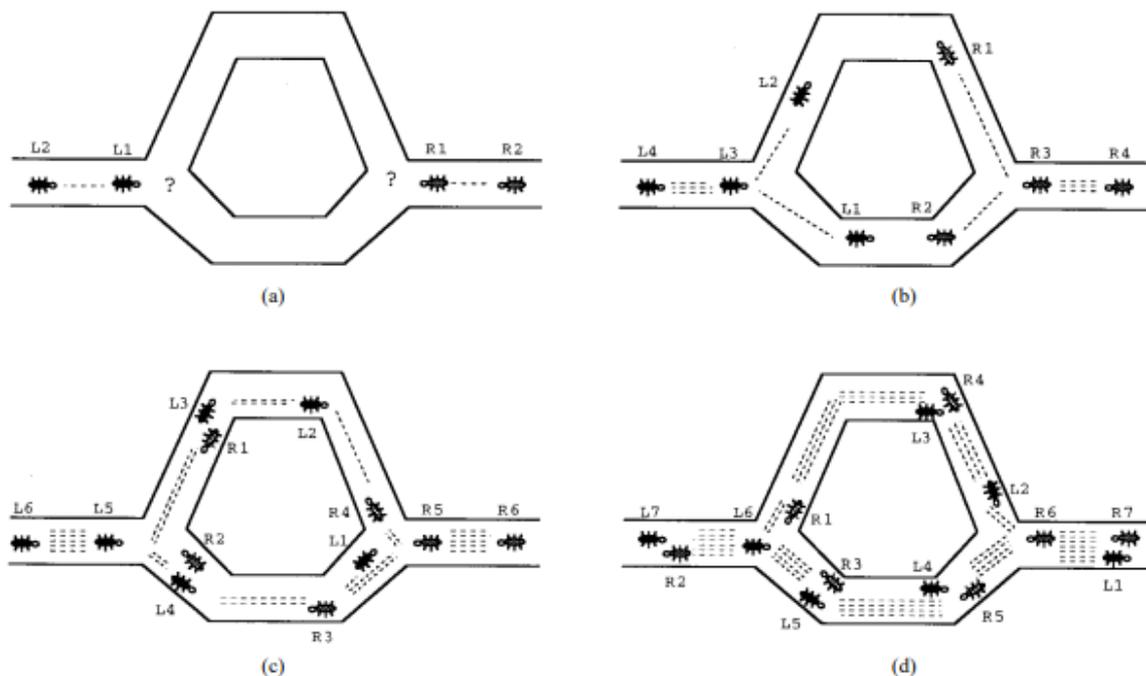


Figura 4 Comportamiento de una colonia de hormigas (Menéndez, E. 2005)

El comportamiento real de las hormigas ha inspirado la creación del Sistema de Colonia de Hormigas, donde un conjunto de hormigas artificiales cooperan intercambiando la información de las feromonas para llegar a la mejor solución del problema. El Sistema de Colonia de Hormigas ha sido aplicado en numerosos problemas de optimización combinatoria como es el TSP. Algunos ejemplos los podemos ver en Sarhadi et al. (2010) se presenta el ACS para resolver el problema de optimización de rutas TSP con ventanas de tiempo. En Favaretto et al. (2007) aplica el ACS en un problema VRPTW con múltiples visitas, o en Schyns (2015) donde se aplica el algoritmo para resolver un problema con flota de vehículos heterogénea, ventanas de tiempo y entregas divididas.

4.3.1 Construcción de Rutas

Para tratar el problema MCSVRP aplicando el ACS, contamos con que las hormigas simulan una flota de vehículos, donde cada ruta es diseñada de forma secuencial una a una. Partiendo del listado de vehículos heterogéneos disponibles ($V_k, k=1...K$) y de los clientes que no han sido asignados a ninguna ruta ($C_s, s=1...N$) es necesario implementar un algoritmo que diseñe cada una de las rutas de forma que se satisfaga las necesidades de todos los clientes. Para aquellos clientes que no pueden ser servidos en un periodo de tiempo debido a la limitada capacidad de los vehículos, se incluirán en una lista de espera (HL, *Holding List*) que será asignada a un vehículo fantasma $K+1$.

El diseño de las rutas comienza cuando una hormiga selecciona un vehículo de la flota V_k de manera aleatoria e inicia una ruta en la que selecciona a aquel cliente más alejado del conjunto de clientes C_s (cliente semilla). A continuación, se va insertando en la ruta nuevos clientes que no han sido servidos anteriormente mediante la regla pseudo-aleatoria definida en la ecuación (4.1).

Si un número aleatorio comprendido entre $[0,1]$ al que denominamos q es menor que q_0 , aplicamos la ecuación (4.1), en caso contrario, el cliente es seleccionado mediante el método conocido como "*Roulette Wheel Selection*", de acuerdo a la distribución de probabilidad dada en la ecuación 1. Por lo tanto, con probabilidad q_0 se aplica el mejor movimiento descrito, mientras que con probabilidad $(1 - q_0)$ se exploran nuevos arcos. De esta forma, ajustando el valor del parámetro q_0 se puede lograr concentrar la búsqueda alrededor de la mejor solución encontrada hasta el momento con valores de q_0 cercanos a 1 y si por el contrario se quiere realizar una mayor exploración del espacio de soluciones dispersando la búsqueda, el valor de q_0 debe ser menor.

Los clientes se insertan de manera iterativa en cualquier posición de la ruta hasta que la capacidad del vehículo sea superada o no se cumpla con las restricciones de ventanas de tiempo. Es entonces cuando se añade la ruta al conjunto de posibles rutas finales y se eliminan los clientes ya servidos y vehículos empleados de los listados C_s y V_k respectivamente.

El proceso se repite de forma iterativa hasta que todos los clientes son asignados a alguna ruta. En el caso de que todos los vehículos hayan sido asignados a rutas y siga habiendo clientes sin servir, se añadirán a una lista de espera que se incluirá en la búsqueda local posterior.

A continuación, se presentan las ecuaciones empleadas y una breve descripción de las mismas:

- τ_{iu} : Representa el nivel de feromonas existente en el arco (i,u) .
- η_{ijk} : Representa la atractividad de insertar un cliente u entre otros dos clientes i y j .
- $J(i)$: Representa el vecindario disponible para el nodo i , el cual estará formado por todos aquellos clientes no servidos (C_s) y que al insertarlos en cualquier posición de la ruta formen soluciones factibles.
- P_{ijk} : Probabilidad de elegir al cliente u para ser insertado entre los clientes i y j en la ruta realizada por el vehículo k .

$$P_{ijk} = \begin{cases} \frac{(\tau_{iu})^\alpha \cdot (\eta_{ijk})^\beta}{\sum_{w \in J(i)} (\tau_{iw})^\alpha \cdot (\eta_{iwk})^\beta}, & \text{if } u \in J(i) \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

Para medir la atractividad (η), utilizaremos la función “greedy” definida en la ecuación (4.2). Ésta es una variante de la utilizada en Paraskevopoulos et al. (2008) para resolver el problema HFVRPTW. Está formada por métricas que evalúan el coste de añadir a un cliente no asignado en las rutas parcialmente construidas. Es una función que considera aspectos tanto temporales como de carga de los clientes, evaluándolos y seleccionando a aquel con menor valor en la función “greedy” (menor coste de inserción). Como se busca que el valor de la atractividad sea alto, ésta la definimos como la inversa de dicha función.

$$\eta_{ijk}^{-1} = \Phi_{ijk} = \alpha_1 \cdot C_{iju}^1 + \alpha_2 \cdot C_{iju}^2 + \alpha_3 \cdot C_u^3 \quad (4.2)$$

- Φ_{ijk} : Función “greedy” que mide el coste de insertar al cliente sin asignar u entre otros dos clientes consecutivos i y j pertenecientes a una ruta parcialmente construida y realizada por el vehículo k .
- α : Pesos no negativos que se les da a cada una de las métricas. La suma total debe ser igual a 1 ($\alpha_1 + \alpha_2 + \alpha_3 = 1$).

$$C_{ij,u}^1 = TT_{iu} + TT_{uj} - TT_{ij} \quad (4.3a)$$

$$C_{ij,u}^2 = TMax_s - \sum_{i \in R} ST_i - \sum_{(i,j) \in R} TT_{ij} - ST_u \quad (4.3b)$$

$$C_u^3 = TT_{0G} - TT_{0u} \quad (4.3c)$$

La primera métrica (Ecuación 4.3a) mide el aumento del tiempo de la ruta al insertar una orden u entre dos órdenes consecutivos (i, j). La métrica definida en la Ecuación (4.3b) se introduce para dar prioridad a la inserción de órdenes en la ruta cuyo tiempo de servicio sea mayor, de esta forma se busca maximizar el tiempo de trabajo de los verificadores en cada ruta. La tercera métrica (Ecuación 4.3c) da prioridad a la inserción en cada ruta de aquellas órdenes que se encuentren más alejadas del depósito, donde 0 representa el depósito central y G la orden no servida con mayor distancia en tiempo al depósito.

4.3.2 Inicialización y actualización de la feromona

Al comenzar a aplicar el Algoritmo Memético ACS, inicialmente obtenemos una primera solución a partir de la cual calculamos el valor inicial de la feromona. El valor inicial de la feromona lo definimos como τ_0 en la ecuación (4.4), donde N es el número total de clientes que son servidos, y L la distancia total realizada por los vehículos en la primera solución obtenida.

$$\tau_0 = \frac{1}{N \cdot L} \quad (4.4)$$

El ACS emplea dos tipos de actualización de feromonas: local y global.

- Actualización local: Cada vez que una hormiga construye una nueva solución, se procede a la actualización local del nivel de feromonas del arco (i, j) de la solución obtenida (S) mediante la aplicación de la ecuación (4.5), donde el parámetro de evaporación ρ regula la reducción de la intensidad de los rastros de feromonas en los arcos. Con esta reducción del valor de la feromona se pretende que las hormigas abandonen de forma progresiva las soluciones pasadas y busquen nuevas, evitando que varias hormigas elijan el mismo camino. Cuanto mayor sea el valor de ρ , más se atenuará el valor de la feromona en cada iteración:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad \text{if } (i, j) \in S \quad (4.5)$$

- Actualización global: Una vez que todas las hormigas de una población han completado una solución, la actualización global es realizada únicamente por aquella hormiga que ha alcanzado la mejor solución hasta el momento (S^*), depositando la feromona en dicha ruta. El rastro de feromonas se actualiza con las siguientes ecuaciones:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij}^{bs} \quad \forall (i, j) \in S^* \quad (4.6a)$$

$$\Delta\tau_{ij}^{bs} = (N \cdot L_{best})^{-1}. \quad (4.6b)$$

Algoritmo Memético de sistema de colonia de hormigas

```

1  Inicializacion de parametros ( $\alpha_1, \alpha_2, \alpha_3, iter\_max, M, \rho, \alpha, \beta, q_0, iter\_tabu\_max, TT_{min}, TT_{max}$ );
2   $\tau_{(N,N)} \leftarrow Matriz\_Feromona()$ ;
3   $\tau_0 \leftarrow Calcular\_Valor\_Feromona()$ ;  $\Delta\tau^{bs} \leftarrow \tau_0$ ;
4   $\tau_{(N,N)} \leftarrow Inicializar\_Matriz\_feromonas(\tau_0)$ ;
5   $iteracion=1$ ;
6  While ( $iteracion \leq iter\_max$ ) AND ( $CPU\ tiempo\_consumido \leq \gamma$ ) do:
7     $hormiga=1$ ;
8    While ( $hormiga \leq M$ ) AND ( $CPU\ tiempo\_consumido \leq \gamma$ ) do:
9       $S' \leftarrow Construcion\_Ruta_(\alpha, \beta, q_0, \alpha_1, \alpha_2, \alpha_3)$ ;
10      $\tau_{(N,N)} \leftarrow Descarga\_Local\_Matriz\_Feromonas(S', \rho, \tau_0)$ ;
11      $S'' \leftarrow VNTS(S', iter\_tabu\_max, TT_{min}, TT_{max})$ ;
12     If  $f(S'')$  mejora  $f(S)$  then
13        $S \leftarrow S''$ ;  $\Delta\tau^{bs} \leftarrow Calcular\_valor\_Feromona(S'')$ ;
14      $hormiga = hormiga + 1$ ;
15   EndWhile
16    $\tau_{(N,N)} \leftarrow Descarga\_Global\_Matriz\_Feromona(S, \rho, \Delta\tau^{bs})$ ;
17    $iteracion=iteracion+1$ ;
18 EndWhile

```

Algoritmo: Sistema de Colonia de Hormigas construcción de rutas ($\alpha, \beta, q_0, \alpha_1, \alpha_2, \alpha_3$)

```

1  Inicializamos la lista de vehículos disponibles  $V_k, k=1,2,\dots,K$ ;
2  Inicializamos la lista de clientes  $C_s, s=1,2,\dots,N$ ;
3  While ( $C_s \neq 0$ ) AND ( $V_k \neq 0$ ) do:
4     $k \leftarrow \text{SeleccionAleatoria}(V_k), V_k \leftarrow \text{EliminarVehiculo}(k)$ ;
5     $\text{semilla} \leftarrow \text{EncontrarClienteSemilla}(C_s, k)$ ;
6    If ( $\text{semilla} \neq 0$ ):
7       $r_k \leftarrow \text{InicializarRuta}(k), r_k \leftarrow \text{InsertarClienteSemilla}(\text{semilla}), C_s \leftarrow \text{EliminarCliente}(\text{semilla})$ ,
Select_cust  $\leftarrow$  True;
8      While ( $C_s \neq 0$ ) AND (Select_cust) do:
9        Select_cust  $\leftarrow$  False;
10        $q \leftarrow \text{ElegirNumeroAleatorio}()$ ;
11       For todos los clientes  $u$  de  $C_s$  do:
12         For todas las inserciones entre las posiciones  $i, j$  de  $r_k$  do:
13            $\eta_{i,j,u,k} \leftarrow \text{FuncionGreedy}(i, j, u, k, \alpha_1, \alpha_2, \alpha_3)$ ;
14            $P_{i,j,u,k} \leftarrow \text{Calcular\_Probabilidades}(\eta_{i,j,u,k}, \tau_{(N,N)})$ ;
15         EndFor
16       EndFor
17       If ( $P_{i,j,u,k} \neq 0$ ):
18         If ( $q \leq q_0$ ):
19            $n \leftarrow \text{Valor\_Maximo}(P_{i,j,u,k})$ ;
20         Else:
21            $n \leftarrow \text{Seleccion\_Roulette\_Wheel}(P_{i,j,u,k})$ ;
22         EndIf
23        $r_k \leftarrow \text{InsertAT}(i, j, n, k); C_s \leftarrow \text{EliminarCliente}(n)$ 
24       Select_cust  $\leftarrow$  True
25     EndIf
26   EndWhile
27 EndIf
28 EndWhile
29 If ( $C_s \neq 0$ ) do:
30    $r_k \leftarrow \text{Insertar\_AT\_ListadeEspera}(C_s)$ ;
31 EndIf

```

4.4 VNTS

El VNTS utiliza una Búsqueda Tabú (TS, *Tabu Search*) dentro de un algoritmo de búsqueda en entornos variables (VNS, *Variable Neighborhood Search*), el cual se presenta para resolver el problema MCSVRP. El VNTS lo forman un conjunto de vecindades N_λ ($\lambda = 1 \dots \lambda_{\max}$). El proceso iterativo comienza desde que se obtiene una solución inicial s , en nuestro caso por medio del ACS, es entonces cuando la TS determina una nueva solución s' en N_λ . Si la nueva solución $f(s')$ es mejor que la solución $f(s)$, entonces s es reemplazado por s' y la búsqueda comienza de nuevo en la vecindad N_λ . En el caso de que la nueva solución no mejore a la anterior, se explora la siguiente vecindad $N_{\lambda+1}$. Este método es repetido hasta que todas las vecindades han sido examinadas ($\lambda = \lambda_{\max}$), saltando de una solución a otra cuando se produce alguna mejora. El VNTS termina cuando se alcanza la solución mínima local con respecto a todas las estructuras de vecindades.

La combinación entre el VNS y la TS permiten alcanzar soluciones mucho más eficientes y de mayor calidad que si empleáramos alguno de los métodos por separado. Por un lado, el VNS impide estancarse en soluciones que son óptimos locales de poca calidad al introducirlas en otras vecindades en las que dicha solución puede no ser la óptima. Por otro lado, la TS permite dentro de cada estructura de vecindad, intensificar la búsqueda al permitir movimientos que no mejoren la solución y evitando volver a soluciones recientes ya visitadas.

Paraskevopoulos et al. (2008) resuelve el problema VRP con flota fija heterogénea de vehículos y ventanas de tiempo (HFVRPTW), para ello aplica una metaheurística semi paralela para construir varias soluciones iniciales y a continuación aplica un algoritmo VNTS. Molina et al. (2018) aplica a un problema de ruteo de vehículos en empresas que dan un servicio de control metrológico, en el que los atributos empleados para definir el problema son similares al nuestro. El algoritmo empleado para resolver el problema es un sistema de vecindarios variables descendentes con búsqueda tabú y lista de espera (VNDTS-HL). El algoritmo se inicia con una solución inicial proporcionada por una heurística de construcción semi paralela.

4.4.1 Búsqueda Tabú

La TS fue propuesta por Glover (1989) y se ha convertido en uno de las mejores y más extendidas metodologías de búsqueda local para resolver problemas de optimización combinatoria. Es una metaheurística en la que partiendo de una solución inicial s se realizan diferentes iteraciones para mejorar la solución en esa vecindad. El empleo de una memoria o lista tabú (TL, *Tabu List*) permite almacenar cierta información sobre rutas ya exploradas con el objetivo de que el algoritmo no se atasque en óptimos locales, sino que pueda escapar de ellos y buscar en zonas más amplias.

La TS permite el empeoramiento de la solución con el fin de encontrar una mejor solución final, para ello explora el espacio de búsqueda aplicando mecanismos de memoria flexible junto a restricciones o niveles de aspiración. Las principales características diferenciadoras de la TS respecto a otras metaheurísticas se definen a continuación, aunque cabe decir que en cada problema se definirán unos requisitos diferentes pudiendo no incluir todas las características:

- Entorno reducido: al tratarse de un método que limita la posibilidad de realizar ciertos movimientos una vez que los hemos incluido en la TL, el entorno de posibles soluciones vecinas se ve reducido y como consecuencia también el tiempo de resolución. Al definir la TL es necesario fijar el tamaño de la misma para conocer el número de iteraciones que deben pasar hasta que los atributos puedan salir de ella y vuelvan a ser empleados al considerarse que se ha alejado suficiente de la solución anterior.
- Nivel de aspiración: son las condiciones que se deben cumplir para poder alcanzar una solución, aunque dicho movimiento tenga el atributo tabú. En estos casos se puede ignorar la lista tabú para que la solución o atributo salga de ella antes de que se cumpla su plazo de iteraciones. Un nivel de aspiración puede ser el encontrar una solución que mejore todas las obtenidas anteriormente, de esta forma aunque el movimiento estuviera prohibido por la TL se podría considerar admisible.
- Memoria a corto plazo: almacena movimientos o atributos de las soluciones recientes obtenidas para evitar así repetir dichas soluciones y poder explorar el espacio desconocido. Los movimientos o

atributos introducidos en la TL permanecerán en ella un cierto número de iteraciones definido en el problema hasta que se considere que la búsqueda se encuentra en una región distinta y por tanto se pueden liberar de la lista.

- Memoria a largo plazo: intensifica y diversifica la búsqueda almacenando para ello la frecuencia de ocurrencia de características en las soluciones. Por ejemplo, puede identificar si un atributo es atractivo o no si pertenece a una secuencia de soluciones de alta o baja calidad.

El algoritmo TS empleado en este trabajo ha sido adoptado previamente por otros autores en la literatura (Molina et al., 2019). Proporciona un balance entre la diversificación y la intensificación de la búsqueda. El uso de TL de reducido tamaño permite intensificar la búsqueda permitiendo los ciclos de pequeños periodos, mientras que el empleo de TL de tamaños mayores permitirá diversificar la búsqueda en el espacio de soluciones, de esta manera se consigue escapar de soluciones que son óptimos locales (Paraskevopoulos et al., 2008).

En este trabajo se emplea un mecanismo de control de la longitud de la TL de forma que su tamaño pueda variar para encontrar un equilibrio entre intensificación y diversificación. Inicialmente, el tamaño de la lista adquiere un valor mínimo t_{\min} . Mediante un proceso de diversificación, la lista irá aumentando una unidad en cada una de las iteraciones hasta llegar a un valor máximo t_{\max} siempre y cuando no se haya mejorado la solución. En el caso de que se haya alcanzado una mejor solución en alguna de las iteraciones, se procede a la intensificación de la búsqueda en dicha solución, para ello se inicia la TL a su valor mínimo t_{\min} eliminando las soluciones más antiguas y añadiendo a la lista esta nueva solución. La condición de finalización del número de iteraciones en la búsqueda viene definido por el valor de de un número máximo de iteraciones (*MaxIteraciones*) sin observar una mejora de la función objetivo.

Algoritmo de Búsqueda Tabú

- 1 Dada una solución s y una estructura de vecindad k ;
- 2 Inicializar la lista tabú TL_z de longitud t_{\min} ;
- 3 $elite \leftarrow s$, $contador \leftarrow 0$, $tsize \leftarrow t_{\min}$, CondiciónAspiración(s);
- 4 While ($contador \leq MaxIteraciones$) do:
- 5 Encontrar $s' \in N_k(s) \mid s$ sujeto a condiciones tabú y de aspiración
- 6 ConjuntoPermitido(s) $\leftarrow s'$;
- 7 $s \leftarrow ElegirPrimeroMejora(ConjuntoPermitido(s))$;
- 8 ActualizarListaTabu();
- 9 If $f(s)$ mejora $f(elite)$ then
- 10 $elite \leftarrow s$; $contador \leftarrow 0$, $tsize \leftarrow t_{\min}$, CondiciónAspiración ($elite$);
- 11 Else
- 12 $contador \leftarrow contador + 1$;
- 13 If ($tsize < t_{\max}$) then
- 14 $tsize \leftarrow tsize + 1$;
- 15 EndIf
- 16 EndIf
- 17 EndWhile
- 18 $s \leftarrow elite$

Algoritmo del Procedimiento VNTS (S' , $iter_tabu_max$, TT_{min} , TT_{max})

```

1  Definir un conjunto de estructuras de vecindades  $N_\lambda$ ,  $\lambda=1, 2, \dots, \lambda_{max}$ ;
2   $S \leftarrow S'$ ;
3   $\lambda=1$ ;
4  While ( $\lambda \leq \lambda_{max}$ ) do:
5     $S'' \leftarrow$  Búsqueda_Tabu ( $S$ ,  $\lambda$ ,  $iter\_tabu\_max$ ,  $TT_{min}$ ,  $TT_{max}$ );
6    If  $f(S'')$  mejora  $f(S)$  then
7       $S \leftarrow S''$ ;  $\lambda \leftarrow 1$ ;
8    Else
9       $\lambda = \lambda + 1$ ;
10   EndIf
11 EndWhile
12 Return ( $S$ )

```

4.4.2 Estructura de vecindad VNS

El VNS es una metaheurística propuesta para resolver problemas de optimización combinatoria. Su idea básica es el cambio sistemático de vecindades dentro de la búsqueda local. Aunque el VNS trata una sola solución en cada iteración, la trayectoria seguida durante la búsqueda es discontinua (Figura 5). Explora vecindades cada vez más alejados de una solución inicial moviéndose de forma discontinua desde una solución a otra cada vez que se produce una mejora. Las características favorables de una solución se mantendrán y se utilizarán para obtener soluciones prometedoras en otras vecindades (Hansen y Mladenovic, 2001). La ventaja de utilizar diferentes vecindades reside en que un óptimo local en una estructura de vecindad determinada no tiene por qué serlo en otras vecindades, por lo tanto la búsqueda debe continuar hasta que se alcance un mínimo local en todas las vecindades.

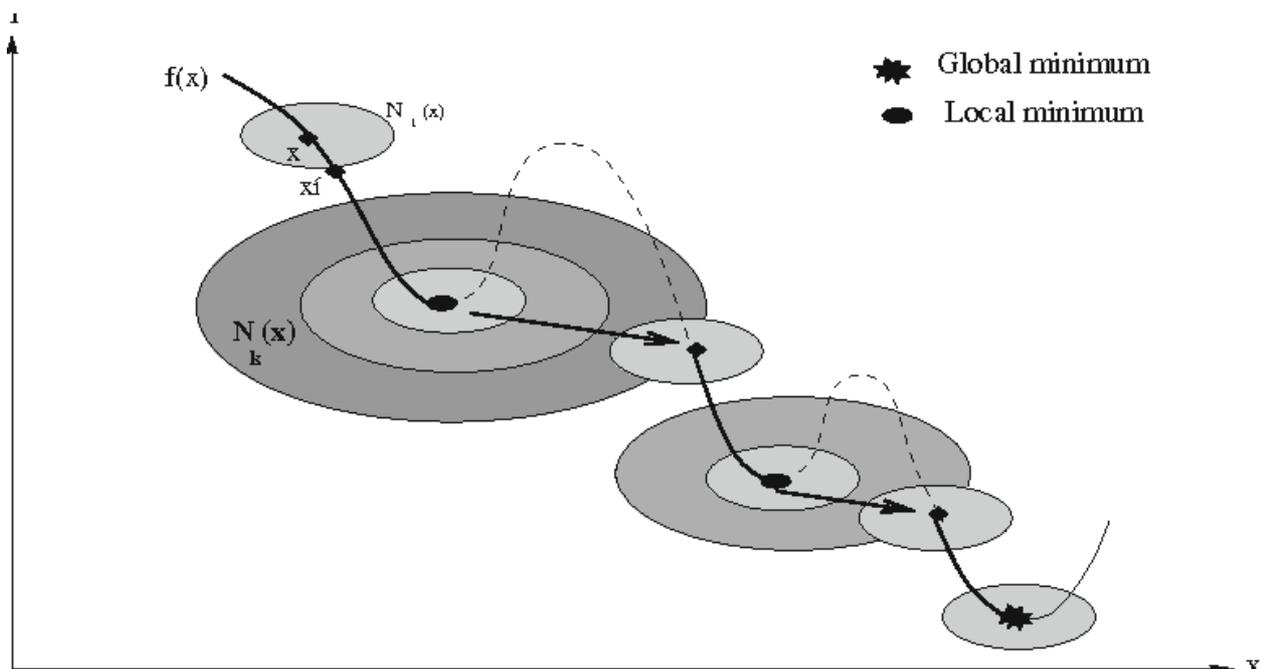


Figura 5 Variable Neighborhood Search: VRP (Hansen et al., 2008)

El VNS engloba las principales propiedades que se desean en una metaheurística, las cuales garantizan su interés tanto teórico como práctico:

- Simplicidad: una metaheurística debe basarse en principios simples y claros que puedan ser ampliamente aplicables.
- Precisión: los pasos de la metaheurística se deben formular en términos matemáticos precisos evitando descripciones vagas o sin sentido.
- Coherencia: todos los pasos de la heurística en cada problema deben seguir los principios de la metaheurística.
- Eficiencia: las heurísticas deben proporcionar soluciones óptimas o cercanas a ellas.
- Efectividad: el tiempo de computación para hallar las soluciones óptimas en las heurísticas debe ser moderado.
- Robustez: la actuación de las heurísticas debe responder a diferentes variaciones del problema.
- Facilidad de uso: las heurísticas deben estar expresadas de forma clara, que sean fáciles de entender y de utilizar.
- Innovación: los principios de las heurísticas deben poder ser aplicada a nuevos tipos de problemas.

De todas las características mostradas, quizá la que más caracteriza al VNS es la simplicidad, que le permite obtener mejores resultados que muchas otras metaheurísticas de forma mucho más sencilla (Hansen y Mladenovic, 2003).

En la Tesis doctoral Molina et al. (2016) emplea una serie de operadores que pueden ser únicamente aplicables a VRP con flotas heterogéneas. El objetivo de cada uno de los operadores es escapar de cada uno de los óptimos locales de las diferentes vecindades evitando así quedar estancado y poder acceder a mejores resultados del espacio de soluciones. Los operadores utilizados son: Inserción (inter-ruta), Intercambio (inter-ruta), GENI, 2-Opt, 3-Opt, Inserción (intra-ruta), Intercambio (intra-ruta), Doble inserción, CROSS-exchange y GENI-exchange.

Debido a que las rutas diseñadas en el MCSVRP tienen un horizonte de planificación corto y para evitar aumentar en exceso el tiempo de computación de la fase de la búsqueda local, se ha empleado únicamente tres estructuras de vecindades ($\lambda_{\max}=3$). La primera vecindad se define como la unión de las vecindades representadas por los operadores Inserción (inter-ruta) e Intercambio (inter-ruta). La segunda vecindad se define por el operador 2-Opt y la tercera vecindad por el operador CROSS-exchange.

El operador Inserción (inter-ruta) es una estructura que genera una nueva solución eliminando a un cliente de una posición de una ruta e insertándola en una nueva posición de otra ruta.

El operador Intercambio (inter-ruta) es una estructura que genera una nueva solución intercambiando simultáneamente la posición entre dos clientes entre dos rutas diferentes.

El operador 2-Opt es una estructura de vecindad únicamente aplicable en rutas individuales (intra-rutas) con el objetivo de generar nuevas soluciones eliminando dos clientes consecutivos de una solución e insertándolos en otro segmento de la solución añadiendo dos nuevos arcos. La idea principal es crear nuevas soluciones reordenando los clientes y manteniendo la estructura de la ruta.

El operador CROSS-Exchange es una estructura de vecindad que se aplica a pares de rutas únicamente. Consiste en el intercambio de manera simultánea de varios segmentos de clientes entre dos rutas. El número de clientes que se intercambian se debe definir previamente, en el caso en estudio se ha limitado a dos. El intercambio de clientes entre rutas se realiza en el siguiente orden: 1-2 y 2-2, manteniendo siempre la orientación de los arcos de las rutas. Estos movimientos permiten obtener soluciones de mayor calidad y evitan

quedar atrapados en mínimos locales.

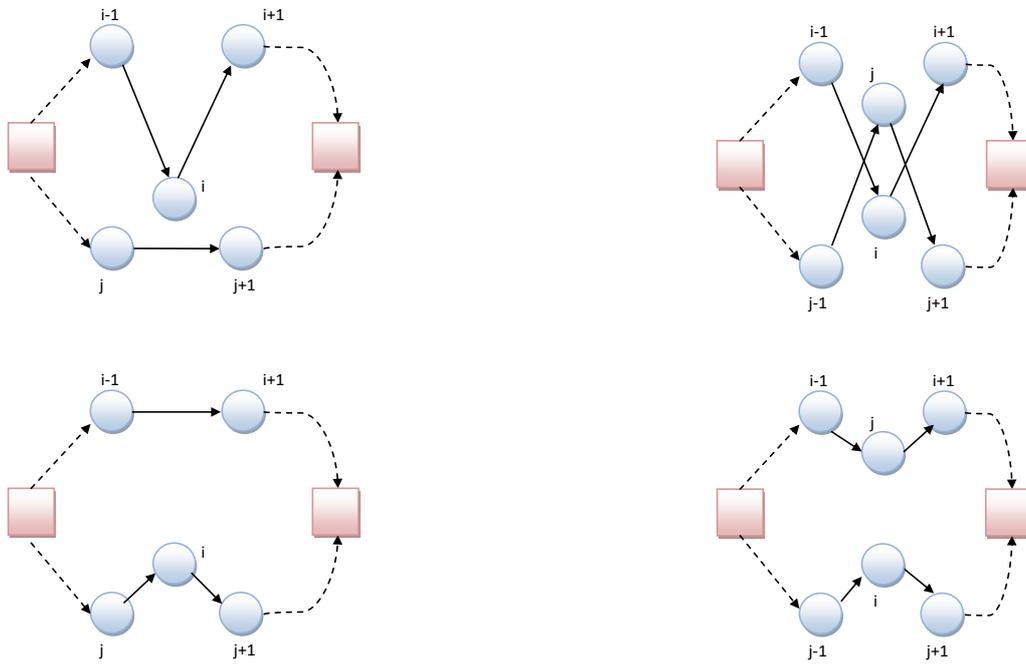


Figura 6 Primera estructura de vecindad

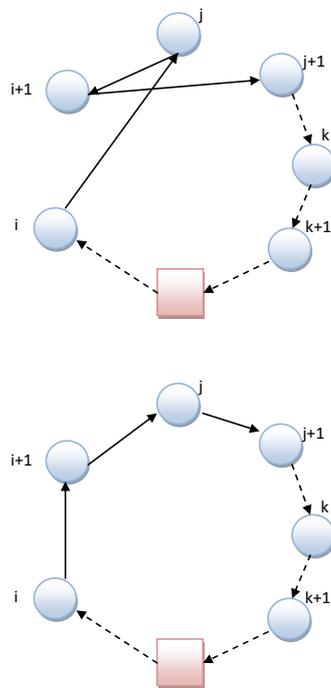


Figura 7 Segunda estructura de vecindad

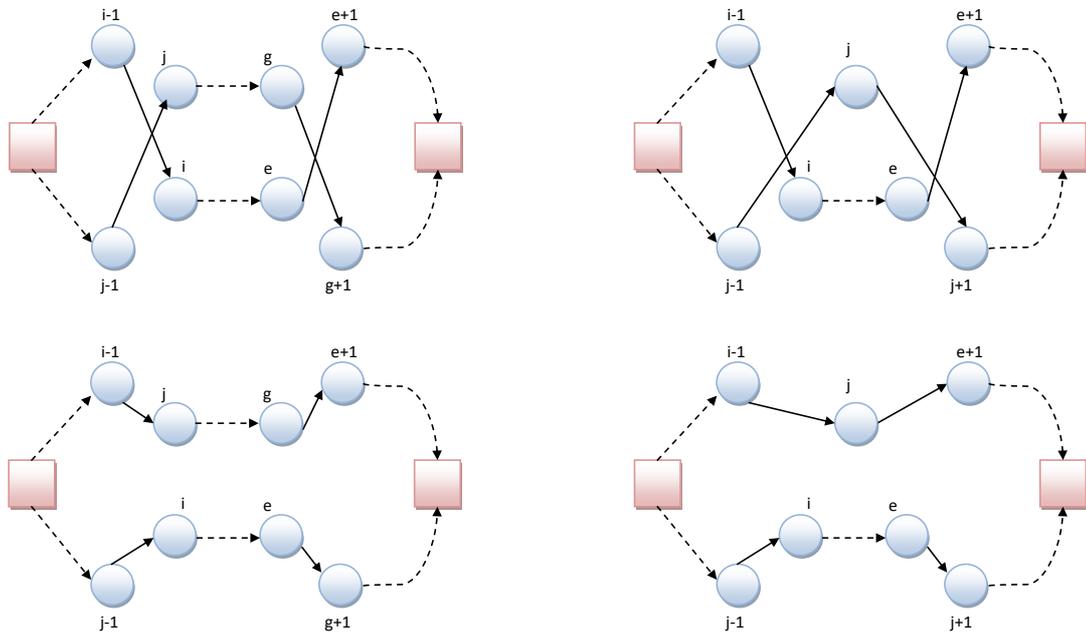


Figura 88 Tercera estructura de vecindad

Al aplicar inicialmente el algoritmo de colonia de hormigas (ACS), se obtienen una serie de soluciones con las que partimos a la hora de iniciar la búsqueda local. Entre esas soluciones iniciales se encuentra una lista de espera (Holding List) donde se almacenan todos aquellos clientes que no pueden ser servidos en el periodo planificado. Esta HL actúa como una ruta fantasma que participará en la búsqueda local, ampliando de esta forma el espacio de búsqueda y evitando quedar atrapados en mínimos locales. Por ello, se añaden movimientos adicionales en cada iteración y estructuras de vecindades del VNTS. Estos nuevos movimientos posibles son:

- Insertar desde la lista de espera: Consiste en transferir un cliente desde la lista de espera a una ruta existente. Es un movimiento que será examinado en cada iteración de la búsqueda local en cada una de las vecindades N_i .
- Insertar a la lista de espera: Consiste en transferir un cliente desde una ruta existente a la lista de espera. Se emplea cuando todos los movimientos de inserción posibles entre rutas son tabú.
- Intercambio con la lista de espera: Intercambio entre un cliente perteneciente a una ruta existente con otro de la lista de espera. Este tipo de movimientos solo se realiza cuando se aplican operadores de intercambio entre rutas.

5 RESULTADOS EXPERIMENTALES

5.1 Introducción

En esta sección del trabajo se lleva a cabo la experimentación con el fin de comprobar la eficacia del algoritmo desarrollado para resolver el problema MCSVRP. El algoritmo ha sido desarrollado en C++ y se ejecuta en una CPU con 2,53 GHz Intel Core (TM) i3-M380.

En primer lugar, en el apartado 5.2, se describe el proceso de obtención de parámetros que se van a emplear en el algoritmo. La búsqueda de unos valores adecuados en los parámetros nos va a permitir obtener soluciones de mayor calidad. Posteriormente, en el apartado 5.3, se comprueba la eficacia del Algoritmo memético realizando una comparación con otros algoritmos del estado del arte de la literatura científica en baterías de problemas de referencia. Además, se realiza una comparativa entre los resultados obtenidos empleando el Algoritmo memético frente a los resultados obtenidos empleando el VNTS y el ACS por separado. Por último, en el apartado 5.4, se aplica el Algoritmo Memético en un problema real y se comparará con resultados obtenidos en estudios anteriores y por la propia empresa.

5.2 Ajuste de parámetros del algoritmo

En este apartado, se seleccionan los valores de los parámetros que componen el Algoritmo Memético propuesto en el trabajo. El Algoritmo Memético emplea una serie de parámetros cuyo valor influye en los resultados finales obtenidos, y por tanto cobra un papel fundamental la obtención de aquellos valores que mejoren la calidad de las soluciones finales.

El problema empleado para la realización del ajuste de parámetros ha sido elegido de forma aleatoria entre los problemas de Taillard, habiendo sido seleccionado el problema número trece.

El proceso de simulación para la obtención de los parámetros se ha dividido en tres fases. En cada fase se han fijado ciertos parámetros y simulado el problema variando los valores de otros parámetros, quedándonos con los mejores resultados obtenidos en la función objetivo.

La función objetivo (FO) empleada para la simulación, da prioridad al número de nodos visitados frente a la distancia total recorrida.:

$$FO = (\text{Número Nodos Totales} - \text{Número de Nodos Servidos}) + (\xi * \text{Distancia Total Recorrida})$$

Siendo $\xi = 0,0001$

A continuación, se muestran cada una de las fases de la experimentación:

Fase 1:

En la primera fase de la experimentación los parámetros a evaluar han sido: $\beta \in \{1, 1.5, 2, 2.5, 3\}$ y $q_0 \in \{0.7, 0.8\}$ manteniendo fijos el resto de parámetros como se muestran en la siguiente tabla.

Tabla 2 Parámetros a evaluar en la fase 1 de la experimentación

Parámetros							
Iter. Max	Pob. Hormigas M	ρ	q_0	α	β	Iter. Tabú	Lista Tabú
20 % N	20 % N	0,2		1		10	2...5

La tabla 3 muestra los mejores valores de la FO obtenidos y el porcentaje de la desviación con respecto al mejor de ellos. Como se observa, los valores adoptados corresponden a $\beta=2$ y $q_0=0,7$:

Tabla 3 Resultados de la FO y desviación obtenidas en la fase 1 de la experimentación

β	q_0			
	0,7		0,8	
	FO	(%)	FO	(%)
1	3,102	19,0	3,102	19,0
1,5	3,103	19,0	3,102	19,0
2	2,607	0,0	3,102	19,0
2,5	3,101	19,0	2,855	9,5
3	3,103	19,0	3,103	19,0

Fase 2:

Manteniendo los valores anteriores de los parámetros y actualizando la tabla con los obtenidos en la Fase 1 para los parámetros β y q_0 , procedemos con la evaluación de los siguientes parámetros: Número de iteraciones máximas en el ACS (Iter_Max) $\in \{10\% N, 20\% N, 30\% N\}$ y Población de hormigas (M) $\in \{10\% N, 20\% N, 40\% N, 60\% N\}$ siendo N el número de nodos que comprende el problema.

Tabla 4 Parámetros a evaluar en la fase 2 de la experimentación

Parámetros							
Iter. Max	Pob. Hormigas M	ρ	q_0	α	β	Iter. Tabú	Lista Tabú
		0,2	0,7	1	2	10	2...5

La tabla 5 muestra los mejores valores de la FO obtenidos y el porcentaje de desviación con respecto al mejor de ellos. Los valores adoptados corresponden a $M=40\%N$ y $Iter_Max=30\%N$.

Tabla 5 Resultados de la FO y desviación obtenidas en la fase 2 de la experimentación

M	Iter Max					
	10 % N		20 % N		30 % N	
	FO	(%)	FO	(%)	FO	(%)
10 % N	3,105	31,618	3,104	31,578	3,103	31,497
20 % N	3,104	31,544	2,854	20,973	3,102	31,486
40 % N	3,102	31,456	2,855	20,991	2,359	0,000
60 % N	2,855	21,005	3,101	31,425	2,855	21,003

Fase 3:

Por último, fijamos los valores obtenidos en las fases anteriores y nos centramos en los parámetros correspondientes al VNTS: Número máximo de iteraciones de TS (Iter_Tabú) $\in \{20\% N, 25\% N, 30\% N\}$ y tamaño máximo de la lista Tabú (Lista_Tabú) $\in \{10\% N, 15\% N, 20\% N\}$.

Tabla 6 Parámetros a evaluar en la fase 3 de la experimentación

Parámetros							
Iter. Max	Pob. Hormigas M	ρ	q_0	α	β	Iter. Tabú	Lista Tabú
30 % N	40 % N	0,2	0,7	1	2		

La siguiente tabla muestra los mejores valores de la FO obtenidos y el porcentaje de desviación con respecto al mejor de ellos. Los valores adoptados corresponden a $\text{Iter_tabú}=30\%N$ y $\text{Lista_Tabú}=20\%N$.

Tabla 7 Resultados de la FO y desviación obtenidas en la fase 3 de la experimentación

Lista Tabú (2...x %N)	Iter Tabú					
	20 % N		25 % N		30 % N	
	FO	(%)	FO	(%)	FO	(%)
10 % N	3,102	19,053	2,607	0,051	3,102	19,050
15 % N	3,101	19,035	2,853	9,516	2,854	9,535
20 % N	3,102	19,042	3,101	19,031	2,606	0,000

El listado final de los valores obtenidos por los parámetros es el siguiente:

Tabla 8 Valores finales de los parámetros

		Parámetros ACS				Parámetros VNTS	
Iter. Max	Pob. Hormigas M	ρ	q0	α	β	Iter. Tabú	Lista Tabú
30 % N	40 % N	0,2	0,7	1	2	30 % N	2...20 % N

Los valores de los parámetros alfa pertenecientes a la función “greedy” se han obtenido mediante una variación de los mismos dentro del rango comprendido entre los valores mostrados en la tabla. Aplicando incrementos de 0.1 en la búsqueda, se han seleccionado aquellos que han derivado en mejores soluciones, obteniendo los siguientes resultados:

Tabla 9 Valores de los parámetros alfa

Parámetros ACS		
α_1	α_2	α_3
0.3-0.5	0.2-0.4	0.2-0.4
0.3	0.3	0.4

5.3 Análisis de los resultados obtenidos por el algoritmo

En los siguientes sub-apartados, se analizan los resultados obtenidos con la aplicación del Algoritmo Memético ACS. En primer lugar se estudian los casos en la literatura en los que se ha aplicado el algoritmo en problemas similares al MCSVRP. Posteriormente se realiza un análisis comparativo entre algoritmos para comprobar la eficacia del mismo.

5.3.1 Comparación de los resultados en instancias de la literatura

Analizando la literatura, no encontramos ningún estudio en el que se haya aplicado el Algoritmo Memético ACS en un problema que comparta los mismos atributos que el MCSVRP expuesto en este trabajo, sin embargo, si encontramos en Molina et al. (2020) la aplicación del algoritmo a un problema similar como es el problema HVRPTW. Con este problema compartimos numerosas similitudes como son, la flota fija y heterogénea de vehículos, restricciones de capacidad en los vehículos, restricciones de ventanas de tiempo y restricciones de tiempo en la duración de las rutas.

Molina et al. (2020) presenta el algoritmo memético ACS hibridizado con una búsqueda local VNTS para resolver el problema de ruteo de vehículos con ventanas de tiempo (HVRPTW). El problema está compuesto por una flota limitada de vehículos caracterizada por tener diferentes capacidades y costes fijos y variables, que deben servir a un conjunto de clientes dentro de unas ventanas de tiempo predefinidas. El objetivo del problema es el diseño de rutas que minimicen los costes fijos totales y costes de distribución satisfaciendo las restricciones del problema.

Para evaluar la efectividad del algoritmo memético, Molina et al. (2020) toma como referencia los datos obtenidos por Paraskevopoulos et al. (2008) donde expone 24 casos y Jiang et al. (2014) 56 casos, ambos agrupados en 6 tipos de datos. Clientes distribuidos de forma aleatoria en los casos tipo R, agrupados en los casos tipo C y semi agrupados en tipo RC. El dígito 1 o 2 indica si las ventanas de tiempo y la capacidad de los vehículos es pequeña o grande respectivamente. El coste total de la ruta se obtiene de la suma de los costes fijos de los vehículos y el tiempo total de cada ruta.

En la tabla 10 se muestran los mejores resultados obtenidos por el Algoritmo Memético ACS en los problemas expuestos en Paraskevopoulos et al. (2008), comparándolos con los mejores resultados (BKS) obtenidos hasta el momento para el problema HVRPTW. Los algoritmos referenciados son ReVNTS empleado en Paraskevopoulos et al. (2008) y HEA en Koc et al. (2015). Los resultados obtenidos muestran que el Algoritmo Memético ACS mejora diez de los BKS expuestos, e iguala las soluciones en otros cinco problemas. Para los otros nueve casos, la desviación de los resultados en siete de ellos es menor al 1% y solo en dos casos supera ese valor.

Por otro lado, en la tabla 11 se comparan los mejores resultados obtenidos por el Algoritmo Memético ACS con los BKS obtenidos en Jiang et al. (2014). La primera columna de la tabla refleja los costes fijos (FC), la segunda los costes variables (VC) y por último los costes totales (TC). El Algoritmo Memético mejora 55 de las 56 BKS e iguala en la solución del caso restante.

Los resultados obtenidos muestran que el Algoritmo Memético ACS proporciona soluciones de buena calidad para todos los tipos de HVRPTW propuestos. Es relevante su utilización cuando quedan clientes sin servir por causa de la limitación del número de vehículos, debido a que en esas situaciones el algoritmo maximiza el uso de los vehículos minimizando el número de clientes no servidos y por tanto reduciendo el coste total.

Con el fin de analizar la eficacia del algoritmo en diferentes escenarios, se ha realizado una experimentación del comportamiento del Algoritmo memético en problemas HVRP, es decir, sin considerar ventanas de tiempo. Para ello se ha ejecutado el algoritmo en los problemas de referencia de Taillard (1999). Éstos se componen de 8 problemas numerados del 13 al 20, con una flota fija y un número de nodos que oscila entre 50 y 100. Los resultados obtenidos igualan los BKS en cuatro de los ocho problemas (Tabla 12).

Tabla 10 Resultados para los casos de HVRPTW en Paraskevopoulos et al. (2008) (Molina et al., 2020)

Instance	Fleet	BKS in the literature			Mem-ACS			BKS	
		Mix	TC	Reference	Mix	TC	Gap (%)	=	<
R101A	A ¹ B ¹¹ C ¹¹ D ¹	B ¹⁰ C ¹¹ D ¹	4583.99	ReVNTS	B ¹⁰ C ¹¹ D ¹	4641.79	1.26	0	0
R102A	A ¹ B ⁴ C ¹⁴ D ²	A ¹ B ⁴ C ¹³ D ²	4376.54	HEA	A ¹ B ² C ¹⁴ D ²	4422.93	1.06	0	0
R103A	B ⁷ C ¹⁵	B ⁶ C ¹⁵	4195.05	ReVNTS	B ⁶ C ¹⁵	4198.80	0.09	0	0
R104A	B ⁹ C ¹⁴	B ⁹ C ¹³	4027.69	HEA	B ⁷ C ¹⁴	4004.51*	-0.58	0	1
C101A	A ¹ B ¹⁰	B ¹⁰	8828.93	ReVNTS, HEA	B ¹⁰	8828.93	0.00	1	0
C102A	A ¹⁹	A ¹⁹	7137.79	ReVNTS	A ¹⁹	7119.35*	-0.26	0	1
C103A	A ¹⁹	A ¹⁹	7122.57	HEA	A ¹⁹	7105.39*	-0.24	0	1
C104A	A ¹⁹	A ¹⁹	7083.74	HEA	A ¹⁹	7081.51*	-0.03	0	1
RC101A	A ⁷ B ⁷ C ⁷	A ⁴ B ⁷ C ⁷	5266.36	HEA	A ⁴ B ⁷ C ⁷	5257.67*	-0.16	0	1
RC102A	A ⁵ B ⁶ C ⁸	A ⁴ B ⁵ C ⁸	5099.55	HEA	A ² B ⁶ C ⁸	5072.33*	-0.53	0	1
RC103A	A ¹¹ B ² C ⁸	A ¹⁰ B ² C ⁸	4991.29	HEA	A ¹⁰ B ² C ⁸	4990.94*	-0.01	0	1
RC104A	A ² B ¹³ C ³ D ¹	A ² B ¹³ C ³ D ¹	5016.97	HEA	A ² B ¹³ C ³ D ¹	5003.94*	-0.26	0	1
R201A	A ⁵	A ⁵	3779.12	ReVNTS	A ⁵	3789.55	0.28	0	0
R202A	A ⁵	A ⁵	3578.91	ReVNTS	A ⁵	3585.31	0.18	0	0
R203A	A ⁴ B ¹	A ⁴ B ¹	3553.92	HEA	A ⁴ B ¹	3573.78	0.56	0	0
R204A	A ⁵	A ⁵	3081.80	HEA	A ⁵	3080.44*	-0.04	0	1
C201A	A ⁴ B ¹	A ⁴ B ¹	6140.64	ReVNTS, HEA	A ⁴ B ¹	6140.64	0.00	1	0
C202A	A ¹ C ³	A ¹ C ³	7623.96	HEA	A ¹ C ³	7623.96	0.00	1	0
C203A	C ² D ¹	C ² D ¹	7303.37	ReVNTS, HEA	C ² D ¹	7303.37	0.00	1	0
C204A	A ⁵	A ⁵	5680.46	HEA	A ⁵	5680.46	0.00	1	0
RC201A	C ¹ E ³	C ¹ E ³	5523.15	ReVNTS	C ¹ E ³	5550.88	0.50	0	0
RC202A	A ¹ C ¹ D ¹ E ²	A ¹ C ¹ D ¹ E ²	5132.08	ReVNTS	A ¹ C ¹ D ¹ E ²	5148.90	0.33	0	0
RC203A	A ¹ B ¹ C ⁵	A ¹ B ¹ C ⁵	4471.92	HEA	A ¹ B ¹ C ⁵	4473.13	0.03	0	0
RC204A	A ¹⁴ B ²	A ¹⁴ B ²	4241.83	HEA	A ¹⁴ B ²	4234.46*	-0.17	0	1
N° BKS								5	10
Average							0.08		
Max							1.26		
Min							-0.58		
Runs							10		
Processor							Core i5 3.3 GHz		
Avg Time (minutes)							12.24		

Tabla 11 Resultados para los casos de HVRPTW en Jiang et al (2014) (Molina et al., 2020)

Instance	BKS in the literature				Mem-ACS			Instance	BKS in the literature				Mem-ACS		
	F.C	V.C	T.C.	%Dev.	F.C	V.C	T.C.		F.C	V.C	T.C.	%Dev.	F.C	V.C	T.C.
HC101	670	1215.33	1885.33	0.00%	670	1215.33	1885.33	HR112	2540	1712.71	4252.71	-0.43%	2540	1694.62	*4234.62
HC102	670	1220.66	1890.66	-3.49%	670	1154.59	*1824.59	HR201	255	1510.74	1765.74	-7.05%	250	1391.31	*1641.31
HC103	650	1258.04	1908.04	-2.96%	650	1201.65	*1851.65	HR202	255	1281.81	1536.81	-5.35%	270	1184.66	*1454.66
HC104	670	1139.78	1809.78	-1.06%	670	1120.54	*1790.54	HR203	255	1082.39	1337.39	-1.07%	270	1053.08	*1323.08
HC105	670	1184.73	1854.73	-0.22%	670	1180.70	*1850.70	HR204	275	839.94	1114.94	-0.55%	250	858.79	*1108.79
HC106	670	1210.64	1880.64	-0.86%	670	1194.49	*1864.49	HR205	270	993.91	1263.91	-0.33%	270	989.74	*1259.74
HC107	670	1169.52	1839.52	-0.00%	670	1169.45	*1839.45	HR206	270	910.44	1180.44	-0.19%	225	953.20	*1178.20
HC108	670	1156.49	1826.49	-0.31%	670	1150.74	*1820.74	HR207	225	877.06	1102.06	-0.89%	225	867.23	*1092.23
HC109	670	1129.22	1799.22	-0.42%	670	1121.71	*1791.71	HR208	225	782.00	1007.00	-0.94%	225	772.52	*997.52
HC201	520	793.28	1313.28	-0.00%	520	793.25	*1313.25	HR209	225	894.04	1119.04	-0.18%	225	891.98	*1116.98
HC202	540	743.58	1283.58	-1.93%	500	758.83	*1258.83	HR210	275	1032.53	1307.53	-2.66%	225	1047.79	*1272.79
HC203	540	719.97	1259.97	-0.21%	540	717.36	*1257.36	HR211	225	785.22	1010.22	-0.74%	225	777.71	*1002.71
HC204	540	716.09	1256.09	-0.32%	500	752.08	*1252.08	HRC101	3330	2373.97	5703.97	-1.60%	3270	2342.57	*5612.57
HC205	520	805.84	1325.84	-0.45%	540	779.88	*1319.88	HRC102	3420	2136.02	5556.02	-0.12%	3360	2189.20	*5549.20
HC206	540	723.63	1263.63	-0.10%	540	722.39	*1262.39	HRC103	3510	1928.89	5438.89	-0.27%	3390	2034.35	*5424.35
HC207	520	787.35	1307.35	-3.52%	540	721.39	*1261.39	HRC104	3450	1881.41	5331.41	-0.47%	3390	1916.52	*5306.52
HC208	500	690.81	1190.81	-0.35%	500	686.67	*1186.67	HRC105	3570	2135.79	5705.79	-1.16%	3390	2249.81	*5639.81
HR101	3030	2095.52	5125.52	-3.04%	2760	2209.52	*4969.52	HRC106	3510	2018.42	5528.42	-1.65%	3390	2047.39	*5437.39
HR102	3040	1942.39	4982.39	-1.37%	3040	1874.35	*4914.35	HRC107	3450	2001.31	5451.31	-1.65%	3390	1971.36	*5361.36
HR103	2880	1781.22	4661.22	-0.29%	2790	1857.81	*4647.81	HRC108	3510	1812.31	5322.31	-0.78%	3450	1831.05	*5281.05
HR104	2940	1590.55	4530.55	-3.10%	2740	1650.16	*4390.16	HRC201	2900	1601.65	4501.65	-0.02%	2900	1600.72	*4500.72
HR105	2700	1870.39	4570.39	-1.01%	2700	1824.06	*4524.06	HRC202	2950	1458.53	4408.53	-0.86%	2950	1420.70	*4370.70
HR106	2610	1821.31	4431.31	-1.02%	2590	1796.13	*4386.13	HRC203	2900	1421.87	4321.87	-0.22%	2900	1412.26	*4312.26
HR107	2700	1691.14	4391.14	-0.20%	2680	1702.37	*4382.37	HRC204	2900	1406.65	4306.65	-1.12%	2950	1308.25	*4258.25
HR108	2580	1700.05	4280.05	-0.72%	2540	1709.14	*4249.14	HRC205	2900	1552.88	4452.88	-0.16%	2900	1545.61	*4445.61
HR109	2540	1799.86	4339.86	-0.68%	2540	1770.30	*4310.30	HRC206	2900	1519.09	4419.09	-0.68%	2900	1488.86	*4388.86
HR110	2540	1732.25	4272.25	-4.91%	2540	1522.31	*4062.31	HRC207	2950	1393.55	4343.55	-0.62%	2900	1416.65	*4316.65
HR111	2670	1696.32	4366.32	-0.87%	2650	1678.41	*4328.41	HRC208	2900	1376.15	4276.15	-0.34%	2900	1361.43	*4261.43
Min				-7.05%											
Average				-1.17%											
Max				0.00%											
Av. FC		1728.66		-1.57%		1701.61									
Av. VC		1376.92		-0.59%		1368.84									
Runs		1				10									
Processor		Core2 Duo 2.4 GHz				Core i5 3.3 GHz									

Tabla 12 Resultado de los problemas HVRP de Taillard

	Nodos	Alg. Memético	Mejor Sol.	Desv.
13	50	1517.84	1517.84	0.00
14	50	607.53	607.53	0.00
15	50	1015.29	1015.29	0.00
16	50	1144.94	1144.94	0.00
17	75	1066.40	1061.96	0.42
18	75	1843.26	1823.58	1.08
19	100	1144.47	1117.51	2.41
20	100	1557.15	1534.17	1.50

5.3.2 Análisis comparativo entre algoritmos

Para conocer la eficacia del Algoritmo Memético utilizado en el trabajo, en este apartado se analizan los resultados obtenidos al aplicar el ACS y el VNTS por separado y se comparan los resultados con los obtenidos al aplicar el Algoritmo Memético (ACS + VNTS). Para ello, se emplea el problema de Taillard nº 13 con el que anteriormente hemos definido los valores de los parámetros. Se han aumentado a 50 el valor obtenido en el apartado anterior del número de iteraciones en el caso del ACS y la lista tabú en el VNTS, debido a que al aplicar los algoritmos por separado queremos ampliar el espacio de búsqueda lo máximo posible. En el caso del algoritmo memético mantenemos el valor de los parámetros obtenidos en el apartado anterior. Los algoritmos se han resuelto con un número de órdenes a servir $N = 50$ y la distancia entre las diferentes órdenes ha sido definida en todos los problemas por igual, para obtener una mayor fiabilidad de los resultados.

Analizando los valores de la tabla, se observa que los mejores resultados obtenidos en el problema de Taillard los alcanzamos al aplicar el Algoritmo Memético ya que se sirven 48 de los 50 nodos que conforman el problema. Estos buenos resultados se deben a la combinación entre el ACS, que proporciona una buena solución de partida, para posteriormente ejecutar el VNTS que permite una mayor explotación de las soluciones.

Aplicando únicamente el ACS se visitan 41 de las 50 órdenes, esto puede deberse a la tipología del algoritmo, que gracias a su componente aleatoria en la obtención de soluciones permite una exploración del espacio de búsqueda mucho más amplio que otros algoritmos, pero que sin embargo, no realiza una intensificación en la búsqueda de dichos valores con las que se podrían alcanzar mejores resultados.

Por otro lado, el VNTS obtiene muy buenos resultados al visitar 47 de los 50 nodos del problema, quedándose a un solo nodo del mejor valor obtenido por el Algoritmo Memético. Esto se debe a que el VNTS combina las estructuras centradas en una intensificación de la búsqueda con una exploración de zonas que no han sido visitadas aún (VNS + TS).

Analizando el tiempo empleado en la resolución de los algoritmos, se podría pensar que el VNTS puede ser una mejor opción frente al Algoritmo Memético cuya complejidad influye en su mayor tiempo de resolución, sin embargo, se debe tener en cuenta que los resultados mostrados en la tabla son los mejores resultados obtenidos después de ejecutar cada uno de los algoritmos en diez ocasiones. Como se puede observar en la tabla 15 la variabilidad en las soluciones obtenidas por el VNTS es muy elevada, debido a la gran dependencia a una solución inicial aleatoria que le influye en gran medida en sus resultados finales. No es el caso del Algoritmo Memético, en el que al aplicar el ACS permite siempre al VNTS partir de una buena solución, provocando que la variación en los resultados sea mucho menor.

Tabla 13 Resultados obtenidos por los algoritmos en el problema de Taillard

ACS			VNTS			ACS + VNTS		
Órdenes servidas	Dist. (Km)	Nº Rutas	Órdenes servidas	Dist. (Km)	Nº Rutas	Órdenes servidas	Dist. (Km)	Nº Rutas
41	1120,419	20	47	1044,875	19	48	1126,524	19

Tabla 14 Comparativa entre ACS, VNTS y Alg. Memético

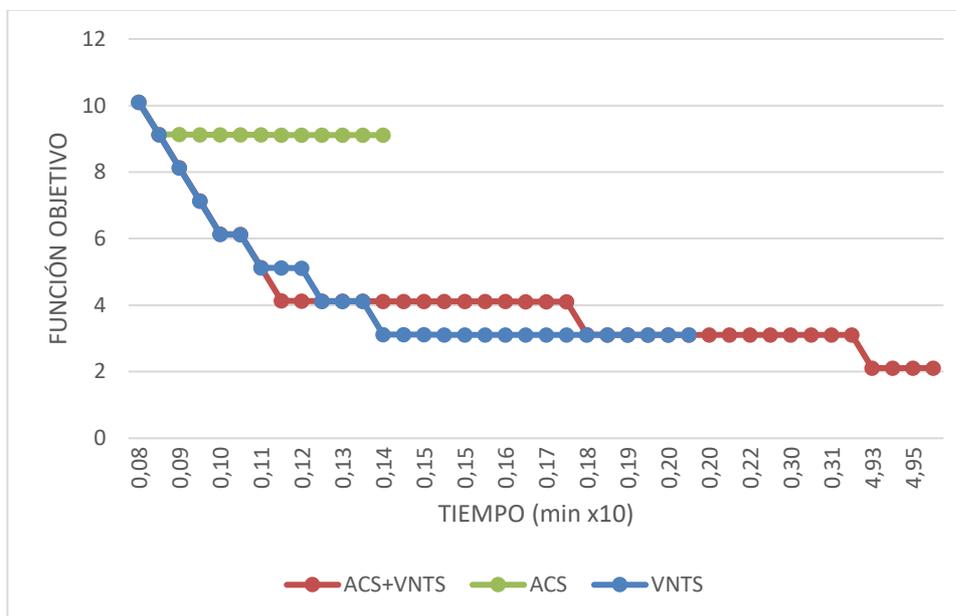
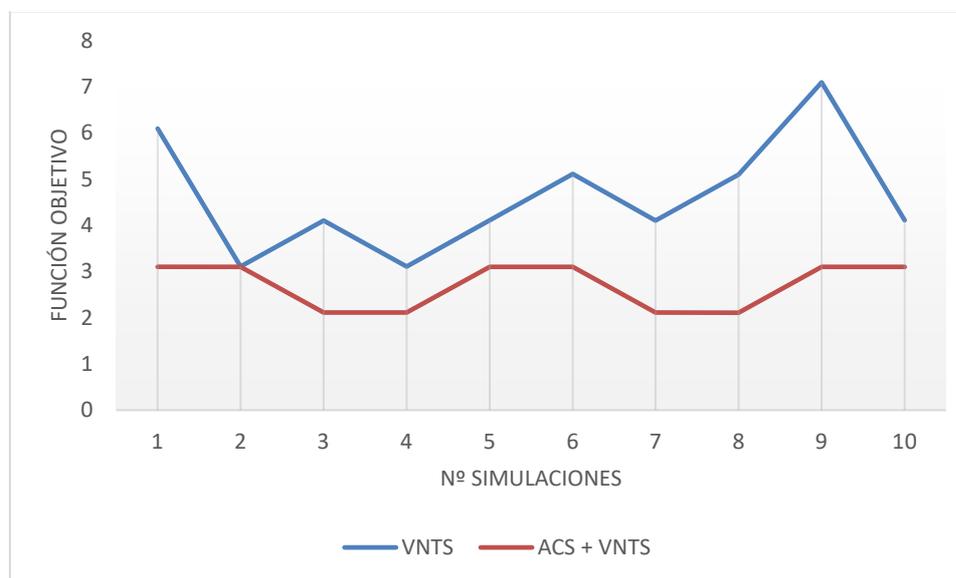


Tabla 15 Comparativa entre VNTS y Alg. Memético



5.4 Aplicación del Algoritmo a una planificación real

5.4.1 Introducción al problema

Este capítulo se centra en la aplicación del algoritmo a un problema real que aparece en una empresa que proporciona un servicio de verificaciones industriales y que ha sido previamente estudiado en otros trabajos académicos (González M. (2014); Hernandez M. (2015); Molina (2016)). Todos estos trabajos tienen en común la aplicación de ciertas metodologías de diseño de rutas a un caso real en una empresa de control metrológico.

Con el diseño y aplicación de este algoritmo al problema real de una empresa que da un servicio de control metrológico, buscamos comparar los resultados obtenidos por el Algoritmo memético ACS con los ya obtenidos por los propios planificadores de la empresa y con otros procedimientos u algoritmos de optimización aplicados a este problema.

Al tratarse de un problema estudiado previamente en los trabajos anteriores, no se va a hacer hincapié en aquellos datos de la empresa que no sean relevantes para la comprensión y resolución de nuestro problema, dejando las referencias de los anteriores trabajos disponibles para más información sobre la misma.

5.4.2 Descripción del problema en una empresa que da servicios de control metrológico

El caso real que vamos a analizar en este trabajo trata de una empresa encargada del Control Metrológico de equipos de medida y de las Inspecciones Técnicas de Vehículos (ITV) en la región de Andalucía.

Este trabajo se centra en la línea de negocio de Control Metrológico que la empresa ejerce desde 1992, la cual realiza trabajos en ocho laboratorios provinciales y cuyo laboratorio central se encuentra en Sevilla. El objetivo del trabajo es la creación de una metodología que pueda ser aplicada en cualquier problema con características similares a este, proporcionando una solución eficiente de diseño de rutas que permita satisfacer la demanda de los clientes.

Las siguientes tablas describen los tipos de servicio que se realizan en la empresa junto con los recursos que requieren cada uno de ellos (Tabla 16), los diferentes patrones necesarios para el desempeño de las mismas y la cantidad disponible de ellos en los laboratorios (Tabla 17).

Tabla 16 Tipo de servicios y patrones necesarios

Tipo de servicio	Descripción	Grupo de servicio	Vehículo	Tiempo de servicio (min)	Patrones necesarios
T1	Verificación de surtidores	Hidrocarburos (H)	Furgoneta	40	3,16,17,18,19,20,28
T2	Verificación de mangueras	Hidrocarburos (H)	Furgoneta	20	3,16,17,18,19,20,28
T3	Pesaje hasta 60 Kg	Masa (M)	Furgoneta	40	5,20
T4	Pesaje hasta 500 Kg	Masa (M)	Furgoneta	130	10,20
T5	Pesaje mayor de 500 Kg	Masa (M)	Camión	130	9,11,20
T6	Verificación de manómetros	Presión (P)	Furgoneta	30	2,14,20,25
T7	Calibración de opacímetros	Gases (G)	Furgoneta	30	3,20,22,23,26
T8	Verificación de opacímetros	Gases (G)	Furgoneta	45	3,20,22,23,26
T9	Verificación de contadores volumétricos	Volumen (V)	Furgoneta	260	3,13,20,21,24,27
T10	Verificación de termómetros	Temperatura (T)	Furgoneta	60	1,3,20,26,28
T11	Calibración de analizadores de gases de escape	Gases (G)	Furgoneta especial	30	20
T12	Verificación de analizadores de gases de escape	Gases (G)	Furgoneta especial	45	20

Tabla 17 Descripción de patrones y cantidades disponibles

Código del patrón	Descripción	Cantidad disponible	Código del patrón	Descripción	Cantidad disponible
1	Amortiguador térmico	4	15	Masa de 150 gr	1
2	Botella de nitrógeno líquido	6	16	Matraz de 10 litros de cristal	7
3	Cronómetro	6	17	Matraz de 2 litros de cristal	7
4	Juego de filtros ópticos	6	18	Matraz de 20 litros de cristal	7
5	Juego de pesas de camiones	8	19	Matraz de 5 litros de cristal	7
6	Juego de pesas de 1 gr a 10 Kg	1	20	Medidor condiciones ambientales	14
7	Juego de pesas de 1 mg a 10 Kg	3	21	Medidor volumétrico de gasolinas y gasóleos	1
8	Juego de pesas de 1 mg a 500 mg	5	22	Motor generador de humos para opacímetros	1
9	Juego de pesas de 100 Kg a 200 Kg	1	23	Opacímetro patrón	1
10	Juego de pesas de 100 mg a 10 Kg	4	24	Patrón inalámbrico de temperatura	4
11	Juego de pesas de 20 Kg	1	25	Patrón para manómetros neumáticos	7
12	Juego de pesas pales	1	26	PC portátil y software	1
13	Juego matraces metálicos de 2,5,10 y 20 litros	1	27	Sistema de medición inalámbrica de temperatura	2
14	Manómetro neumático de referencia	1	28	Termómetro con sonda de temperatura	9

Semanalmente cada laboratorio analiza la demanda recibida y planifica las rutas para la siguiente semana. La demanda semanal no es constante, por lo que cabe la posibilidad de que el horizonte de planificación de una semana no sea capaz de cubrir la totalidad de las órdenes recibidas. Para esta casuística, se diseña una ruta adicional ficticia con todas aquellas órdenes que no pudieron ser satisfechas y que se incluirán en el siguiente horizonte temporal.

Para evitar impactar en el trabajo diario de los clientes y mantener la fidelización de los mismos, hay ciertas órdenes a las que se les da prioridad sobre otras, o se planifican para ser realizadas en las primeras horas del día. El diseño de las rutas está limitado por el número de horas de trabajo de cada verificador, que se estima que es de 8 h/día. Cada verificador trabaja cuatro días a la semana dando servicio en las instalaciones de los clientes, el quinto día está reservado para realizar otras tareas administrativas necesarias dentro de la empresa.

La empresa dispone de cinco verificadores y cada uno de ellos está especializado en un tipo de servicio, como se muestra en la tabla 18. El laboratorio cuenta además con una flota propia de vehículos que consiste en cinco furgonetas estándar y una furgoneta especial preparada para los servicios de Gases.

Tabla 18 Cualificación de los verificadores

Verificador	Cualificaciones de los tipos de servicio
1	T1-T2-T9-T10
2	T1-T2-T3-T4-T5-T9-T10
3	T1-T2-T3-T4-T5-T10
4	T1-T2-T3-T4-T5-T10
5	T1-T2-T6-T7-T8-T10-T11-T12

Para realizar la comparativa de los resultados obtenidos en un periodo de planificación entre la metodología empleada por la empresa y el algoritmo memético, se han considerado cincuenta tipos de órdenes recibidas, cada una de ellas definida según el tipo de servicio, la tecnología empleada y el tiempo medio total requeridos en realizar dichas órdenes. En la tabla 19 se clasifican cada una de las 50 órdenes recibidas en el periodo de planificación estudiado.

Tabla 19 Características de las órdenes recibidas

Orden N°	Tipos de servicio (cantidad)	Grupo	Total T.S.(min)	TW (min)	Orden N°	Tipos de servicio (cantidad)	Grupo	Total T.S.(min)	TW (min)
1	T1(2)+T2(16)	H	400		26	T3(1)	M	40	[0-60]
2	T1(2)+T2(6)	H	200		27	T3(1)	M	40	[0-60]
3	T1(2)+T2(18)	H	440		28	T6(1)	P	30	
4	T1(2)+T2(8)	H	240		29	T6(2)	P	60	
5	T1(4)+T2(7)	H	300		30	T6(1)	P	30	
6	T1(4)+T2(5)	H	260		31	T9(1)	V	260	
7	T1(1)+T2(8)	H	200	[0-60]	32	T9(1)	V	260	
8	T1(1)+T2(1)	H	60	[0-60]	33	T9(1)	V	260	
9	T1(3)+T2(5)	H	220		34	T9(1)	V	260	
10	T1(3)+T2(14)	H	400		35	T12(1)	G	45	
11	T1(2)+T2(3)	H	140		36	T12(1)	G	45	
12	T1(1)+T2(2)	H	80	[0-60]	37	T12(1)	G	45	
13	T1(3)+T2(10)	H	320		38	T1(3)+T2(6)	H	240	
14	T1(3)+T2(14)	H	400		39	T3(4)	M	160	
15	T1(1)+T2(7)	H	180	[0-60]	40	T3(3)	M	120	
16	T1(2)+T2(6)	H	200		41	T3(1)	M	40	
17	T1(2)+T2(9)	H	260		42	T3(1)	M	40	
18	T1(3)+T2(6)	H	240		43	T3(1)	M	40	
19	T1(2)+T2(7)	H	220		44	T3(1)	M	40	
20	T1(2)+T2(18)	H	440		45	T3(1)	M	40	
21	T1(1)	H	40		46	T3(3)	M	120	
22	T3(10)	M	400		47	T3(1)	M	40	
23	T3(2)	M	80		48	T6(1)	P	30	
24	T3(2)	M	80		49	T6(1)	P	30	
25	T3(1)	M	40	[0-60]	50	T9(1)	V	260	

La metodología empleada por la empresa para realizar la planificación semanal consiste en que los jefes de los laboratorios, gracias a su experiencia, diseñan rutas acordes a la tecnología a emplear en las tareas y la cercanía de las mismas entre ellas. Además, son conocedores de aquellas pequeñas empresas con escasos recursos en las que no dar un servicio inminente supone grandes pérdidas para su negocio, por tanto, se les da cierta prioridad para atenderlos a primera hora de la jornada laboral, cuando su clientela es más reducida.

A pesar de la experiencia de los jefes de cada laboratorio, el método empleado por la empresa no es eficiente. La media semanal de tiempo que emplean en la planificación de las rutas es superior a las cinco horas. Tampoco utilizan herramienta que permitan optimizar las distancias entre los diferentes destinos, por lo que se crean rutas con tamaños excesivos que provocan un aumento en el gasto de combustible, rutas más largas en el tiempo y mayor número de órdenes no servidas. La tabla 20 muestra los resultados proporcionados por la empresa. Cada ruta realiza un único tipo de servicio para evitar incompatibilidad de patrones y cualificación de los verificadores.

La planificación proporcionada por la empresa sirve a 44 de las 50 órdenes y recorre una distancia total de 1460.83 km, mientras que con la aplicación del algoritmo memético obtenemos una solución (tabla 21) donde se sirven 47 de las 50 órdenes recibidas con una distancia total recorrida de 1309.98 km. La aplicación del Algoritmo Memético supone una disminución del 10.33% de la distancia total recorrida con respecto a la planificación obtenida por el laboratorio, además de un aumento del 6,82% del número de órdenes visitadas.

Tabla 20 Solución Obtenida por la Empresa

Día	Ruta	Grupo	Vehículo	Verificador	Tiempo ruta (min)	Dist. (Km)	Patrones utilizados
1	0-31-0	V	1	1	265.78	4.17	3-13-20-21-24-27
	0-10-0	H	2	2	413.72	11.53	3-16-17-18-19-20-28
	0-23-39-43-47-42-0	M	3	3	479.92	176.00	5-20
	0-7-21-0	H	4	4	334.28	130.48	3-16-17-18-19-20-28
	0-30-49-29-28-48-0	P	5	5	326.77	176.65	2-14-20-25
2	0-32-0	V	1	1	265.78	4.17	3-13-20-21-24-27
	0-20-0	H	2	2	459.23	17.79	3-16-17-18-19-20-28
	0-27-26-24-46-0	M	3	3	361.99	102.28	5-20
	0-15-11-0	H	4	4	467.14	220.67	3-16-17-18-19-20-28
	0-36-35-37-0	G	6	5	255.76	154.04	20
3	0-3-0	H	1	1	458.68	17.62	3-16-17-18-19-20-28
	0-33-0	V	2	2	265.78	4.17	3-13-20-21-24-27
	0-19-9-0	H	3	3	473.63	36.09	3-16-17-18-19-20-28
	0-25-41-45-44-40-0	M	4	4	354.70	83.90	5-20
	0-18-16-0	H	5	5	456.31	13.35	3-16-17-18-19-20-28
4	0-34-0	V	1	1	265.78	4.17	3-13-20-21-24-27
	0-22-0	M	2	2	419.33	14.92	5-20
	0-12-17-0	H	3	3	441.53	140.32	3-16-17-18-19-20-28
	0-8-5-0	H	4	4	458.31	130.28	3-16-17-18-19-20-28
	0-4-2-0	H	5	5	459.46	18.24	3-16-17-18-19-20-28
-	VEHÍCULO FANTASMA				1-6-13-14-38-50		

Con el propósito de analizar más a fondo el algoritmo memético se han añadido tres casuísticas adicionales que pueden aparecer con bastante frecuencia en este tipo de empresas. La primera casuística es añadir prioridades a aquellas órdenes que no han sido servidas en un periodo de planificación, a las cuales se le asignará una prioridad doble. En la solución real proporcionada por el algoritmo memético correspondería a las órdenes 50, 15 y 5. La segunda casuística resuelve el problema cuando el número de vehículos se ve reducido en una unidad, en nuestro caso de estudio una furgoneta. La tercera casuística es aquella en la que se dispone de un verificador menos en un periodo de planificación. Los mejores resultados obtenidos después de diez iteraciones por el algoritmo memético los comparamos con los obtenidos mediante el procedimiento actual de la empresa realizado por el jefe del laboratorio, además comparamos todas las casuísticas con los valores del algoritmo VNDTS aplicado en Molina et al. (2018).

Analizando los resultados obtenidos (tabla 22), se observa una clara mejoría de los mismos aplicando el Algoritmo memético frente a la planificación semanal realizada por los jefes de los laboratorios. Para el caso de estudio (b) aumentamos un 6,82% el número de órdenes servidas disminuyendo la distancia total recorrida. En el caso (c) aumentamos un 7,31% el número de órdenes servidas con solo un incremento de 86,57 km (6,13%) en la distancia total recorrida. Por último en el caso (d) aumentamos un 5% el número de clientes servidos disminuyendo un 36,5% la distancia recorrida.

No solamente se abastece a un mayor número de clientes, que en nuestra función objetivo es la prioridad principal, sino que también supone, en la mayoría de los casos estudiados, una importante disminución en la distancia total recorrida, lo que conlleva para la empresa un ahorro adicional de combustible. Adicionalmente, el tiempo de computación empleado por el algoritmo memético es de 60 min aproximadamente, mientras que el tiempo de planificación efectuado por los expertos supone para la empresa una media de cinco horas semanales.

Al comparar los resultados con los obtenidos en Molina et al. (2018) por el VNDTS observamos una mejora de los resultados obtenidos por el algoritmo memético en el caso (c) donde aumentamos el número de órdenes servidas un 2,3% e igualamos el número de órdenes pero reducimos un 4,6% la distancia total recorrida en el caso (d). En los casos (a) y (b) aunque no mejoran los resultados se obtienen valores muy cercanos. Cabe recordar los valores mostrados en la tabla 15 en los que tras diez iteraciones el algoritmo memético proporciona siempre resultados muy cercanos a los mejores valores conocidos, a diferencia del VNDTS donde la desviación entre soluciones es mucho mayor.

Tabla 21 Solución obtenida por el Algoritmo Memético

Día	Ruta	Grupo	Vehículo	Verificador	Tiempo ruta (min)	Dist. (Km)	Patrones utilizados
1	0-31-16-0	V+H	1	1	473.94	11.91	3-13-16-17-18-19-20-21-24-27-28
	0-46-38-0	M+H	2	2	474.98	175.05	3-5-16-17-18-19-20-28
	0-7-40-44-0	M+H	3	3	473.42	140.97	3-5-16-17-18-19-20-28
	0-43-39-47-42-23-0	M	4	4	479.91	174.53	5-20
	0-36-35-37-0	G	6	5	255.76	154.04	20
2	0-14-0	H	1	1	420.35	18.99	3-16-17-18-19-20-28
	0-25-41-45-34-0	M+V	2	2	443.36	73.54	3-5-13-20-21-24-27
	0-3-0	H	3	3	458.68	17.62	3-16-17-18-19-20-28
	0-20-0	H	4	4	459.23	17.79	3-16-17-18-19-20-28
	0-49-17-0	H+P	5	5	390.85	136.79	2-3-14-16-17-18-19-20-25-28
3	0-21-4-11-0	H	1	1	453.84	32.40	3-16-17-18-19-20-28
	0-27-8-24-32-0	M+H+V	2	2	465.50	22.48	3-5-13-16-17-18-19-20-21-24-27-28
	0-19-9-0	H	3	3	473.63	36.09	3-16-17-18-19-20-28
	0-26-1-0	M+H	4	4	478.36	40.23	3-5-16-17-18-19-20-28
	0-6-48-28-29-0	H+P	5	5	464.47	87.43	2-3-14-16-17-18-19-20-25-28
4	0-12-33-0	H+V	1	1	350.27	8.24	3-13-16-17-18-19-20-21-24-27-28
	0-10-0	H	2	2	413.71	11.52	3-16-17-18-19-20-28
	0-22-0	M	3	3	419.33	14.91	5-20
	0-2-18-0	H	4	4	457.07	15.42	3-16-17-18-19-20-28
	0-30-13-0	H+P	5	5	440.90	119.99	2-3-14-16-17-18-19-20-25-28
-	VEHÍCULO FANTASMA				50-15-5		

Tabla 22 Soluciones obtenidas para los distintos casos por el Algoritmo Memético ACS

Caso de estudio		Jefe de laboratorio		VNDTS		Alg. Memético ACS	
		Órdenes servidas	Dist. (Km)	Órdenes servidas	Dist. (Km)	Órdenes servidas	Dist. (Km)
a)	Planificación Real	44	1460.83	47	1242.99	47	1309.98
b)	Con prioridades en las órdenes	44	1664.32	47	1328.75	47	1426.20
c)	1 Vehículo menos	41	1410.50	43	999.67	44	1497.07
d)	1 Verificador menos	40	1398.97	42	931.21	42	888.16

6 CONCLUSIONES

Los problemas enriquecidos de enrutamiento de vehículos presentan una variedad de atributos que no son considerados en los VRP tradicionales. En este trabajo, se presenta una nueva variante de problemas enriquecidos a la que hemos denominado MCSVRP, caracterizado por tener una flota fija y heterogénea de vehículos, restricciones vehículo – orden, conductor – orden, sincronización de recursos debido a la presencia de un número limitado de patrones disponibles y restricciones de ventanas de tiempo. Incluye también la posibilidad de no servir a la totalidad de los clientes, por tanto pueden quedar servicios sin cubrir en un periodo de planificación. La función objetivo considera como prioritario maximizar el número de clientes servidos a la vez que se minimiza la distancia total recorrida. Es un problema pensado para empresas destinadas a dar servicios de control metrológico de equipos de medida en la ubicación del cliente, sin embargo, puede ser utilizado para resolver cualquier otro problema que cumpla con los atributos definidos.

Para resolver el problema MCSVRP se ha empleado un Algoritmo Memético ACS que consiste en la hibridación de un algoritmo ACS con una búsqueda local, que en nuestro caso es realizada a su vez por un algoritmo que combina una estructura VNS con una búsqueda tabú (TS). Esta decisión se basó en el éxito del algoritmo ACS ya que es un algoritmo evolutivo basado en poblaciones que obtiene nuevas soluciones aportando una mayor carga de feromonas a aquellas soluciones de mayor calidad y proporciona una buena solución de partida a la posterior búsqueda local. En la posterior fase de búsqueda local, el algoritmo VNTS nos permite recorrer el espacio de soluciones saltando entre diferentes vecindades y de esta manera evitar quedarnos estancados en un óptimo local. Por otro lado, la búsqueda tabú intensifica la búsqueda en cada una de las vecindades evitando repetir movimientos guardados en la lista tabú favoreciendo así recorrer zonas que no han sido exploradas previamente.

Para evaluar la eficacia del algoritmo memético, se ha comparado con los resultados obtenidos en el problema de Taillard por los algoritmos ACS y VNTS por separado, dando unos resultados muy favorables para el algoritmo memético que demuestra la calidad de las soluciones obtenidas en cada una de las simulaciones realizadas, alcanzando el mejor resultado en cada una de ellas. Otros artículos de la literatura demuestran la robustez del algoritmo mostrando su efectividad en un amplio rango de problemas HVRPTW, que comparten la mayoría de atributos con el MCSVRP, donde mejora e iguala los mejores resultados en la mayoría de ellos, especialmente en aquellos casos en los que las ventanas de tiempo y la capacidad de los vehículos son reducidas. El algoritmo también se ha probado en los problemas de referencia HVRP, sin considerar ventanas de tiempo, obteniendo los mejores valores en 4 de los 8 problemas resueltos. Por último, se ha aplicado el algoritmo a un caso real de una empresa dedicada a dar servicios de control metrológico de equipos de medida. El resultado de aplicar el algoritmo memético muestran una gran mejoría respecto a los valores obtenidos por la empresa, ya que como se ha visto en el apartado anterior, supone una disminución del 10.33% de la distancia total recorrida con respecto a la planificación obtenida por el laboratorio, además de un aumento del 6,82% del número de órdenes servidas. El algoritmo memético ha sido validado en tres casos de estudio adicionales que suelen aparecer en este tipo de empresas.

Como conclusión, la experimentación computacional realizada en este trabajo valida la eficacia del enfoque propuesto. Se ha desarrollado una nueva herramienta para alcanzar una gestión más eficiente de los recursos de la empresa en estudio. Esta metodología ayudará al jefe del laboratorio en sus decisiones, identificando la alternativa que prioriza la maximización del número de órdenes servidas reduciendo también distancias, costos y tiempo empleado en el proceso de planificación.

REFERENCIAS

- Abdulkader, M. M., Gajpal, Y., & ElMekkawy, T. Y. (2015). Hybridized ant colony algorithm for the multi compartment vehicle routing problem. *Applied Soft Computing*, 37, 196-203.
- Alvarez, A., Cordeau, J. F., Jans, R., Munari, P., & Morabito, R. (2020). Formulations, branch-and-cut and a hybrid heuristic algorithm for an inventory routing problem with perishable products. *European Journal of Operational Research*, 283(2), 511-529.
- Amorim, P., Parragh, S. N., Sperandio, F., & Almada-Lobo, B. (2014). A rich vehicle routing problem dealing with perishable food: a case study. *Top*, 22(2), 489-508.
- Arbós, L. C. (2012). *Logística. Gestión de la cadena de suministros: Organización de la producción y dirección de operaciones*. Ediciones Díaz de Santos.
- Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In *The vehicle routing problem: latest advances and new challenges* (pp. 3-27). Springer, Boston, MA.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., & Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3), 316-329.
- Belmecheri, F., Prins, C., Yalaoui, F., & Amodéo, L. (2009). An ant colony optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. *IFAC Proceedings Volumes*, 42(4), 1550-1555.
- Beltrami, E. J., & Bodin, L. D. (1974). Networks and vehicle routing for municipal waste collection. *Networks*, 4(1), 65-94.
- Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies—A comprehensive introduction. *Natural computing*, 1(1), 3-52.
- Bravo, J. J., Orejuela, J. P., & Osorio, J. C. (2007). Administración de recursos de distribución: Indicadores para la priorización en transporte. *Estudios Gerenciales*, 23(102), 101-118.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., & Lusby, R. (2011). Models for the discrete berth allocation problem: A computational comparison. *Transportation Research Part E: Logistics and Transportation Review*, 47(4), 461-473.
- Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., & Juan, A. A. (2014). Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, 47(2), 1-28.
- Christofides, N. (1985). *Vehicle routing. The traveling salesman problem*.
- Christofides, N., Mingozzi, A., & Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical programming*, 20(1), 255-282.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery

points. *Operations research*, 12(4), 568-581.

Cooke, K. L., & Halsey, E. (1966). The shortest route through a network with time-dependent internodal transit times. *Journal of mathematical analysis and applications*, 14(3), 493-498.

Cuervo, D. P., Goos, P., Sörensen, K., & Arráiz, E. (2014). An iterated local search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, 237(2), 454-464.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91.

De Armas, J., & Melián-Batista, B. (2015). Variable neighborhood search for a dynamic rich vehicle routing problem with time windows. *Computers & Industrial Engineering*, 85, 120-131.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1), 53-66.

Dror, M., & Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, 23(2), 141-145.

Favaretto, D., Moretti, E., & Pellegrini, P. (2007). Ant colony system for a VRP with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics*, 10(2), 263-284.

Flood, M. M. (1956). The traveling-salesman problem. *Operations research*, 4(1), 61-75.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability* (Vol. 174). San Francisco: freeman.

Glover, F. (1989). Tabu search-part I. *ORSA Journal on computing*, 1(3), 190-206.

Goel, A., & Gruhn, V. (2008). A general vehicle routing problem. *European Journal of Operational Research*, 191(3), 650-660.

Goldberg, D. E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex systems*, 3(5), 493-530.

Golden, B. L., & Wasil, E. A. (1987). OR Practice—Computerized Vehicle Routing in the Soft Drink Industry. *Operations research*, 35(1), 6-17.

Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1), 49-66.

Golden, B. L., Raghavan, S., & Wasil, E. A. (Eds.). (2008). *The vehicle routing problem: latest advances and new challenges* (Vol. 43). Springer Science & Business Media.

González, M. (2014). Desarrollo de una herramienta de optimización de rutas para la empresa Verificaciones Industriales de Andalucía (VEIASA). Proyecto Fin de Carrera. Universidad de Sevilla.

Hansen, P., & Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3), 449-467.

Hansen, P., & Mladenović, N. (2003). Variable neighborhood search. In *Handbook of metaheuristics* (pp. 145-184). Springer, Boston, MA.

- Hansen, P., Mladenović, N., & Pérez, J. A. M. (2008). Variable neighbourhood search: methods and applications. *4OR*, 6(4), 319-360.
- Hartl, R. F., Hasle, G., & Janssens, G. K. (2006). Special issue on rich vehicle routing problems. *Central European Journal of Operations Research*, 14(2), 103.
- Hernandez M. (2015). Diseño de rutas de vehículos para la prestación de servicios de una empresa de control metrológico. Trabajo final de Máster. Universidad de Sevilla.
- Jair, J., Paternina-Arboleda, C. D., Cantillo, V., & Montoya-Torres, J. R. (2013). A two-pheromone trail ant colony system—tabu search approach for the heterogeneous vehicle routing problem with time windows and multiple products. *Journal of Heuristics*, 19(2), 233-252.
- Jaramillo, J. R. (2011). The Green Vehicle Routing Problem. *Southeastern INFORMS Proceedings*, 470-477.
- Jiang, J., Ng, K. M., Poh, K. L., & Teo, K. M. (2014). Vehicle routing problem with a heterogeneous fleet and time windows. *Expert Systems with Applications*, 41(8), 3748-3760.
- Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing* (pp. 187-219). Springer, Boston, MA.
- Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2015). A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Computers & Operations Research*, 64, 11-27.
- Lahyani, R., Khemakhem, M., & Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, 241(1), 1-14.
- Lau, H. C., Sim, M., & Teo, K. M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European journal of operational research*, 148(3), 559-569.
- Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem: past and future trends. *Expert systems with applications*, 41(4), 1118-1138.
- Liu, F. H., & Shen, S. Y. (1999). The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research society*, 50(7), 721-732.
- M. Dorigo, «Optimization, Learning and Natural Algorithms (in Italian),» Milan, 1992.
- Marinakis, Y., & Migdalas, A. (2007). Annotated bibliography in vehicle routing. *Operational Research*, 7(1), 27-46.
- Menéndez, E. (2005). Metaheurística de Optimización Mediante Colonia de Hormigas y Aplicaciones.
- Mín, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5), 377-386.
- Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling

- salesman problems. *Journal of the ACM (JACM)*, 7(4), 326-329.
- Molina, J. C., Eguia, I., & Racero, J. (2018). An optimization approach for designing routes in metrological control services: a case study. *Flexible Services and Manufacturing Journal*, 30(4), 924-952.
- Molina J.C. (2016). Diseño y aplicación de una herramienta para la optimización de rutas de vehículos con aspectos medioambientales. Tesis doctoral. Universidad de Sevilla.
- Molina, J. C., Eguia, I., & Racero, J. (2019). Reducing pollutant emissions in a waste collection vehicle routing problem using a variable neighborhood tabu search algorithm: a case study. *Top*, 27(2), 253-287.
- Molina, J. C., Salmeron, J. L., & Eguia, I. (2020). An ACS-based Memetic algorithm for the Heterogeneous Vehicle Routing Problem with time windows. *Expert Systems with Applications*, 113379.
- Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826, 1989.
- Moscato, P., & Cotta, C. (2003). Una introducción a los algoritmos meméticos. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 7(19), 0.
- Nag, B., Golden, B. L., & Assad, A. (1988). Vehicle routing with site dependencies. *Vehicle routing: methods and studies*, 149-159.
- Oppen, J., Løkketangen, A., & Desrosiers, J. (2010). Solving a rich vehicle routing and inventory problem using column generation. *Computers & Operations Research*, 37(7), 1308-1317.
- Osaba, E., Yang, X. S., Diaz, F., Onieva, E., Masegosa, A. D., & Perallos, A. (2017). A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Soft Computing*, 21(18), 5295-5308.
- Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C. D., Ioannou, G., & Prastacos, G. P. (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, 14(5), 425-455.
- Penna, P. H. V., Subramanian, A., & Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2), 201-232.
- Pellegrini, P., Favaretto, D., & Moretti, E. (2007, September). Multiple ant colony optimization for a rich vehicle routing problem: a case study. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 627-634). Springer, Berlin, Heidelberg.
- Perboli, G., Tadei, R., & Vigo, D. (2011). The two-echelon capacitated vehicle routing problem: models and math-based heuristics. *Transportation Science*, 45(3), 364-380.
- Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1), 1-11.
- Posada, J. R. J. (2013). Algoritmo memético para resolver el problema de enrutamiento de vehiculos con capacidad limitada. *Revista EIA*, 10(20), 13-22.
- Pullen, H. G. M., & Webb, M. H. J. (1967). A computer application to a transport scheduling problem. *The computer journal*, 10(1), 10-13.

- Rieck, J., & Zimmermann, J. (2010). A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Annals of Operations Research*, 181(1), 337-358.
- Robinson, J. (1949). On the Hamiltonian game (a traveling salesman problem) (No. RAND/RM-303). RAND PROJECT AIR FORCE ARLINGTON VA.
- Salmeron, J. L., Ruiz-Celma, A., & Mena, A. (2017). Learning FCMs with multi-local and balanced memetic algorithms for forecasting industrial drying processes. *Neurocomputing*, 232, 52-57.
- Sarhadi, H., & Ghoseiri, K. (2010). An ant colony system approach for fuzzy traveling salesman problem with time windows. *The International Journal of Advanced Manufacturing Technology*, 50(9-12), 1203-1215.
- Schyns, M. (2015). An ant colony system for responsive dynamic vehicle routing. *European Journal of Operational Research*, 245(3), 704-718.
- Sicilia, J. A., Quemada, C., Royo, B., & Escuín, D. (2016). An optimization algorithm for solving the rich vehicle routing problem based on Variable Neighborhood Search and Tabu Search metaheuristics. *Journal of Computational and Applied Mathematics*, 291, 468-477.
- Speidel, V. (1976). EDP-assisted fleet scheduling in tramp and coastal shipping. In *Proceedings of the 2nd International Ship Operation Automation Symposium*, Washington, DC, August 30-September 2, 1976. Proceedings expected to be available about December 1976. (Vol. 5, No. Proceeding).
- Taillard, É. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO-Operations Research*, 33(1), 1-14.
- Tan, K. C., Khor, E. F., & Lee, T. H. (2006). *Multiobjective evolutionary algorithms and applications*. Springer Science & Business Media.
- Ting, C. J., & Chen, C. H. (2013). A multiple ant colony optimization algorithm for the capacitated location routing problem. *International Journal of Production Economics*, 141(1), 34-44.
- Tillman, F. A. (1969). The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3(3), 192-204.
- Toth, P. & Vigo, D. (2002). *The vehicle routing problem*. SIAM Monographs on Discrete Mathematics and Applications, SIAM, 195-224.
- Tunga, H., Bhaumik, A., & Kar, S. (2017). A method for solving bi-objective green vehicle routing problem (G-VRP) through genetic algorithm. *Journal of the Association of Engineers*, 1(2), 33-48.
- Twomey, C., Stützle, T., Dorigo, M., Manfrin, M., & Birattari, M. (2010). An analysis of communication policies for homogeneous multi-colony ACO algorithms. *Information Sciences*, 180(12), 2390-2404.
- Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1), 1-21.
- Watson-Gandy, C. D. T., & Dohrn, P. J. (1973). Depot location with van salesmen—a practical approach. *Omega*, 1(3), 321-329.

Wren, A., & Holliday, A. (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Journal of the Operational Research Society*, 23(3), 333-344.