



Revista UIS ingenierías

ISSN: 1657-4583

ISSN: 2145-8456

Universidad Industrial de Santander

Tarazona-Jiménez, Jeamy; Jimenez-Romero, Jhon; Aguilar-Imitola, Karin; Lamos-Díaz, Henry

Un algoritmo ALNS para el VRPD en la distribución de última milla

Revista UIS ingenierías, vol. 21, núm. 3, 2022, Julio-Septiembre, pp. 153-176

Universidad Industrial de Santander

DOI: <https://doi.org/10.18273/revuin.v21n3-2022012>

Disponible en: <https://www.redalyc.org/articulo.oa?id=553773616012>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica Redalyc

Red de Revistas Científicas de América Latina y el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Un algoritmo ALNS para el VRPD en la distribución de última milla

An ALNS algorithm for VRPD in last mile distribution

Jeamy Tarazona-Jiménez ^{1a}, Jhon Jimenez-Romero ^{1b}, Karin Aguilar-Imitola ^{2a}
Henry Lamos-Díaz ^{2b}

¹ Escuela de Estudios Industriales y Empresariales, Universidad Industrial de Santander, Colombia. Correos electrónicos: jeamy.tarazona@correo.uis.edu.co ^a, jhonalex1031@gmail.com ^b. Orcid: 0000-0002-4912-1685 ^a, 0000-0002-2866-5130 ^b

² Grupo de optimización de sistemas productivos, administrativos y logísticos (OPALO), Escuela de Estudios Industriales y Empresariales, Universidad Industrial de Santander, Colombia. Correos electrónicos: aguilarimitola.karin@gmail.com ^a, hldiaz@uis.edu.co ^b. Orcid: 0000-0001-9776-4258 ^a, 0000-0003-1778-9768 ^b

Recibido: 13 abril, 2022. Aceptado: 10 julio, 2022. Versión final: 15 septiembre, 2022.

Resumen

Los vehículos aéreos no tripulados más conocidos como drones han despertado gran interés en los últimos años, teniendo aplicaciones en operaciones militares y civiles, recientemente se ha investigado acerca de las ventajas de su uso en la distribución de paquetes. En esta investigación se aborda un Problema de Ruteo de Vehículos con Drones (VRPD) enfocado a la distribución de última milla, en el cual los drones y camiones pueden trabajar de manera simultánea, considerando un límite de capacidad para el camión y el dron, así como restricciones asociadas al tiempo de la ruta. Dentro de la formulación matemática se propone una ecuación de velocidad de vuelo del dron y se restringen las rutas a un porcentaje límite de batería disponible para evitar mal funcionamiento en el aire. Para resolver esta formulación se usa el algoritmo *Adaptive Large Neighborhood Search* (ALNS), que es validado usando instancias propuestas en la literatura. Además, se verifica cómo varía la función objetivo de la solución inicial mediante la utilización de las heurísticas de destrucción y reparación, finalmente se realiza un análisis de sensibilidad para algunos parámetros del algoritmo y características de los drones; realizando conclusiones de los resultados arrojados y efectuando recomendaciones para futuras investigaciones.

Palabras clave: Problema de Ruteo de Vehículos con Drones; Adaptive Large Neighborhood Search; UAV; drones; vehículos; Remoción Aleatoria; Peor Remoción; Cluster Removal; Inserción Codiciosa con Función de Ruido; Regret-Insertion; Closest Insertion.

Abstract

Unmanned aerial vehicles, better known as drones, have aroused great interest in recent years, having applications in military and civil operations. Research has recently been carried out on the advantages of their use in parcel distribution. This research addresses a Vehicle Routing Problem with Drones (VRPD) focused on last mile distribution, in which drones and trucks can work simultaneously, considering a capacity limit for the truck and the drone, as well as restrictions associated with the time of the route. Within the mathematical formulation, a drone flight speed equation is proposed, and the routes are restricted to a limit percentage of available battery to avoid malfunctions in the air. To

ISSN impreso: 1657 - 4583. ISSN en línea: 2145 – 8456.

Este es un artículo de acceso abierto bajo la licencia **CC BY-ND 4.0**



Como citar: J. Tarazona-Jiménez, J. Jimenez-Romero, K. Aguilar-Imitola, H. Lamos-Díaz, “Un algoritmo ALNS para el VRPD en la distribución de última milla,” *Rev. UIS Ing.*, vol. 21, no. 3, pp. 153-176, 2022, doi: <https://doi.org/10.18273/revuin.v21n3-2022012>.

solve this formulation, the Adaptive Large Neighborhood Search (ALNS) algorithm is used, which is validated using instances proposed in the literature. In addition, it is verified how the objective function of the initial solution varies by using the destruction and repair heuristics, finally a sensitivity analysis is performed for some parameters of the algorithm and characteristics of the drones; making conclusions from the results obtained and making recommendations for future research.

Keywords: Vehicle Route Problem with Drones; Adaptive Large Neighborhood Search; UAV; drones; vehicles; Random removal; Worst Removal; Cluster Removal; Greedy Insertion with Noise Function; Regret-N Insertion; Closest Insertion.

1. Introducción

En los últimos años, la complejidad en la distribución de la última milla ha incrementado significativamente, debido al aumento en las ventas de productos que se entregan puerta a puerta; se estima que los envíos generalmente oscilan de 50 a 150 paradas por día [1]. Adicionalmente, según datos del Estudio del *E-Commerce* 2020 tuvo un balance positivo. Por ejemplo, según una encuesta de Ernst & Young, 63% de las tiendas online calcularon que su crecimiento sería mayor al 10% lo que resulta fundamental elegir un comercio electrónico y que el plazo de entrega propuesto sea rápido. Así mismo, entre los aspectos que denotan un crecimiento, destaca la intención de internacionalización, según el observatorio *e-commerce*, el 51% de los comercios encuestados prevén internacionalizar su negocio [2]. Dicha actividad requiere un conjunto de acciones propias o ejecutadas por la distribución de última milla, operación que implica el movimiento de la mercancía o producto desde el último punto de distribución hasta el punto de entrega del usuario o destinatario final. En este sentido, se comprende que los inconvenientes no solo tienen que ver con el tráfico, dificultad en zonas peatonales, la reglamentación y regulación de dichas descargas, sino que también se despachan productos de tamaño pequeño a clientes individuales; por ello, este proceso tiene un costo considerable y una eficiencia baja. Además, desencadena disputas corporativas, causando temas relacionados con la contaminación ambiental, la conciencia ecológica, circunstancias laborales en los repartidores, etcétera. Por lo anterior, las empresas y los investigadores han centrado su atención en determinar rutas con el mínimo costo posible que permitan satisfacer a la totalidad de sus clientes y asignar los mejores recursos para maximizar los niveles de servicio y minimizar los costos en el segmento de transporte final [3]. Encontrado, por ejemplo, que los costos de transacción y transporte juntos suman entre 20% (para los agricultores menos remotos) y más del 50% (para los agricultores más remotos) a los precios de los fertilizantes cobrados en el centro de distribución de insumos [4].

El uso de Vehículos Aéreos No Tripulados o más conocidos como drones, ha ido incrementando considerablemente debido a los avances tecnológicos que han permitido mejorar sus características técnicas además de tener ventajas sobre los medios de transporte terrestres, una de ellas es el bajo impacto negativo que tiene sobre el medio ambiente, la eliminación de atascos de tráfico, entre otras; todo esto ha motivado a usarlos en una gran variedad de aplicaciones. Entre las primeras aplicaciones se tiene las operaciones militares, donde se han usado en labores de vigilancia, desminado de territorios y bombardeos; otra aplicación ha sido en operaciones de logística humanitaria, entre las que se destacan el mapeo de zonas de desastre, apoyo en operaciones de rescate y la distribución de insumos médicos [5]. Viendo estos avances y las ventajas que podría generar, el CEO de Amazon, anunció en 2013 el Prime air, un servicio que usa drones multirrotor para realizar la entrega comercial de paquetes [6]. Así mismo, DHL también inició su proyecto Parcelcopter en 2013, este ha transportado medicamentos a la isla de Juist en el mar del norte [7]. En 2014 Google reveló el proyecto “Wing”, en el que se producen drones con mayor capacidad de carga que los anteriormente mencionados [8].

A partir de estas motivaciones, surgen el FSTSP (*Flying Sidekick Traveling Salesman Problem*) [9] y posteriormente el VRPD (*Vehicle Routing Problem with Drones*) como ramas de investigación para este problema. Por estos motivos, es fundamental plantear la posibilidad de usar flotas mixtas, en este caso una de camiones y Vehículos Aéreos No Tripulados, también conocidos como UAVs. Esta investigación pretende contribuir en el estudio de los problemas de ruteo de vehículos con drones, enfocada en la distribución de última milla, un tema relativamente reciente y que cuenta con gran interés por muchas empresas como Amazon [10], DHL [7] y Google [8]. Las contribuciones importantes del presente documento son: la consideración de la incidencia del peso del paquete que lleva el dron, la repercusión que tiene la velocidad del viento en la velocidad del dron; una restricción relevante, la cual es que la batería del dron no exceda el 80% de su capacidad, dado que, si lo hace, este puede tender a

caídas. Además, este trabajo evalúa y compara dos soluciones iniciales para el algoritmo ALNS. Estas consideraciones son tomadas en cuenta para lograr presentar una formulación matemática que se aproxime más a la realidad y al mismo tiempo definir el desempeño de los métodos de soluciones iniciales.

Este documento está organizado de la siguiente forma: revisión de la literatura relacionada con VRP y sus variantes (incluido el VRPD) secc. 2.1. En la secc. 2.2, se indica la formulación del modelo matemático, descripción del problema, y la vinculación entre el modelo VRPD y distribución de última milla. Secc. 2.3, diseño del algoritmo de solución, incluyendo dos soluciones iniciales (CWA y CWA Ext.) y algoritmos de destrucción e inserción. Luego, se presenta en la Secc. 3, la experimentación y resultados computacionales, con experimentación de instancia y GAPS del ALNS y resultados característicos para la resistencia de vuelo del UAV y las mejores soluciones para el problema. Finalmente, las conclusiones y recomendaciones para trabajos futuros se proporcionan en las secciones 4 y 5.

2. Metodología

2.1. Revisión de la literatura

La primera investigación referente al VRP se dio con “*The Truck Dispatching Problem*”, donde se realizó una formulación del problema para la distribución de combustible. En dicho artículo, Dantzig y Ramser [11] obtuvieron una ruta óptima entre una terminal granelera y las terminales de gasolina que atiende esta, mediante el uso de una flota homogénea de camiones repartidores de gasolina. Se estableció la demanda de los productos a entregar y se asignaron las estaciones de abastecimiento de los camiones repartidores, de manera que fuera satisfecha la demanda y que, al mismo tiempo, el recorrido transitado por la flota de vehículos fuera el mínimo posible.

Con el paso de los años se han estudiado distintas variantes a este problema, con el fin de aproximarlos más a la realidad, una de ellas es el uso de una flota heterogénea denominado HVRP [12], donde se analiza el comportamiento de la respuesta al tener vehículos de diferentes capacidades. En este el problema, *The Fleet Size and Mix Vehicle Routing Problem* [13], realizó una comparación entre distintos algoritmos y finalmente se estableció la combinación óptima en la que se minimizaran los costos totales. En los últimos años, y motivado en gran manera por los cambios climáticos, se han incorporado a la flota de camiones, vehículos con combustión alterna como biodiésel, gas natural e híbridos [14], esto con el propósito de investigar las posibilidades

de contribuir a mejorar el medio ambiente y reducir la huella de carbono de las rutas actuales. Luego de esta contribución surge el Problema de Ruteo de Vehículos Eléctricos (EVRP) [15], donde se consideran restricciones como el tiempo de carga de la batería del vehículo y la disponibilidad de la infraestructura de carga a lo largo de la ruta. Otra de las posibilidades para reducir la huella de carbono y además minimizar los tiempos de ruta, es el uso de camiones en combinación con drones para la entrega de paquetes en la distribución de última milla, este problema es también una variante del VRP; se denomina Problema de Ruteo de Vehículos con Drones (VRPD). Murray y Chu en 2015 realizaron un primer acercamiento al problema abordando el denominado FSTSP, el problema consistía en determinar las asignaciones óptimas de clientes para un UAV que trabaja en conjunto con un único camión de reparto; el objetivo del modelo es reducir el tiempo de duración de la ruta de los camiones y drones para dar servicio a todos los clientes y devolver ambos vehículos al depósito. En este problema se contaba con un dron que podía desplazarse sin necesidad de que el camión esperara, el problema fue resuelto mediante un modelo de Programación Lineal Entera Mixta (MILP) y probado con instancias de la literatura [9].

En 2017 [16], presentaron un problema multiobjetivo de ruteo de vehículos con flota heterogénea de UAV, donde se consideró la autonomía limitada de los vehículos, múltiples estaciones de carga y se respetaron los requisitos operativos, se formuló usando un modelo MILP, resuelto mediante un algoritmo de búsqueda dinámica y validado mediante un caso de estudio de dos capas.

A mediados de 2018, los autores [17] estudiaron sobre el uso de los drones para la distribución de última milla por considerarlos una idea prometedora para reducir el tráfico excesivo por carretera. Con el fin de superar las dificultades causadas por la corta capacidad relativa de vuelo de los drones, estos vehículos aéreos no tripulados propusieron un concepto innovador al usar los camiones como plataformas móviles de despegue y aterrizaje, partiendo de que la ruta de los camiones ya estuviese dada, el objetivo es optimizar el cronograma de los drones lanzados desde el camión hasta el préstamo del servicio a un conjunto determinado de clientes, resuelto con dos modelos de programación entero y mixto, probado con la generación de tres tipos de instancias. Durante 2019 los autores [18] consideraron la utilización de los drones con camiones para la distribución de paquetes, teniendo dentro de su modelo restricciones de capacidad, carga y tiempo de vuelo de los drones, además los clientes de este modelo se agrupaban. El objetivo de esta investigación fue minimizar el costo de las rutas,

para lograrlo emplearon el algoritmo ALNS (Adaptive Large Neighborhood Search) del cual se obtuvo buenas soluciones, que posteriormente fueron comparadas y evaluadas mediante 112 instancias de [19]. Es por esto por lo que, a partir de las combinaciones y de las restricciones que se han establecido en el área de investigación, se ha logrado abarcar un problema de ruteo de vehículos con drones donde se consideró que la flota de drones es homogénea, la resistencia de la batería dron, su velocidad, y el tiempo de vuelo de este el cual se relaciona directamente con los factores anteriormente mencionados.

2.2. Formulación del modelo matemático

Considerando los hallazgos de la revisión de la literatura específica del área, se plantea un modelo matemático adaptado de aquel formulado en [18], el modelo del presente documento tiene en consideración aspectos importantes del dron como lo son las implicaciones del peso del paquete, la incidencia del viento, la limitación del consumo de batería.

2.2.1. Descripción del problema

El problema de distribución propuesto consta de dos subproblemas a resolver sugerido por [9]. El primero consistió en la necesidad de trazar una ruta que optimizara el uso de una flota heterogénea que llevara a cabo la repartición de bienes y servicios encaminados a la distribución de última milla; el segundo, en determinar cuál ruta reduce los costos operacionales y cumple con restricciones como el tiempo máximo de la ruta.

Para el primer caso, en el trazado de la ruta, es necesario considerar las particularidades de una flota que está compuesta por un conjunto de vehículos y UAVs, entre ellas se tiene que el camión de reparto funciona bajo el mando de un conductor y debe estar equipado por un dron, ambos con la misma misión de entregar paquetes a los clientes, los cuales son servidos una sola vez, ya sea por un camión o por un dron, además, cada camión transporta su correspondiente dron y este debe partir y regresar a un único depósito, el retorno puede ser en tándem o de manera independiente. En los momentos en los que el dron no realiza ninguna entrega, permanecerá en el camión para ahorrar energía de la batería, en cambio, cuando el dron está en operación es lanzado por el vehículo desde un nodo i y recogido por este en varios nodos k , este proceso se realiza por cada uno de los arcos de la ruta trazada. En el modelo se considera que el dron únicamente puede atender un cliente a la vez, debido a que la capacidad de carga útil del dron no es ilimitada y que el peso de determinados paquetes solamente puede

ser transportado por los vehículos, puesto que los drones no cuentan con la fuerza idónea para llevar productos que excedan su capacidad de elevación. Así mismo, se tiene un tiempo asociado al lanzamiento y recuperación del dron; igualmente, se considera un tiempo de servicio para entregar los paquetes a los clientes. Por otra parte, los vehículos tienen una restricción vinculada, a fin de que no se exceda un límite de tiempo de operación por día.

2.2.2. Modelo matemático

Índices

i = Nodo de partida, $i = 1, 2, 3 \dots, n$

j = Nodo de llegada, $j = 1, 2, 3 \dots, n$

k = Nodo de recuperación, $k = 1, 2, 3 \dots, n$

n = Número de nodos

v = Número de vehículos

d = Número de drones

s = Número de tuplas $\langle i, j, k \rangle$

Listado de conjuntos

C = Conjunto de clientes $\{1, 2, 3 \dots, c\}$

$C' \subseteq C$ = Subconjunto de clientes visitados por drones

V = Conjunto de vehículos $\{1, 2, 3 \dots, v\}$

D = Conjunto de drones UAV $\{1, 2, 3 \dots, d\}$

N = Conjunto de todos los nodos incluyendo el

depósito cuando sea $0 \{0, 1, 2, \dots, c, (c + 1)\}$

N_0 = Conjunto de nodos desde el cual un vehículo puede partir $\{1, 2, 3 \dots, c\}$

N_i^* = Conjunto de nodos al cual un vehículo puede llegar $\{1, 2, 3 \dots, c, (c + 1)\}$

$\Delta^+(i)$ = El conjunto de nodos que pueden ser alcanzados desde el nodo $i \in N_0 = N_i^* \{i\}$

$\Delta^-(i)$ = El conjunto de nodos que pueden ser usados para alcanzar el nodo $i \in N_i^* = N_0 \{i\}$

A = Conjunto de todos los arcos $\{(i, j): i \in N_0, j \in \Delta^+(i)\}$

Parámetros determinísticos

q = Capacidad del vehículo $[Kg]$

d_i = Demanda del cliente $i \in C$ $[Kg]$

W = Capacidad de carga del dron $[Kg]$

e = Capacidad de la batería dada en tiempo de vuelo del dron $[H]$

A_{ij} = Distancia requerida para ir desde el nodo $i \in N_0$ hasta el nodo $j \in N_i^*$ $[Km]$

t_{ij}^v = Tiempo requerido por un vehículo para viajar desde $i \in N_0$ a $j \in N_i^*$ $[H]$

t_{ij}^d = Tiempo requerido por un UAV para viajar desde $i \in N_0$ a $j \in N_i^*$ $[H]$

C_{ij}^v = Costo de operar un vehículo para viajar desde $i \in N_0$ a $j \in N_i^*$ $[\$]$

C_{ij}^d = Costo de operar un UAV para viajar desde $i \in N_0$ a $j \in N_i^*$ $[\$]$

M = Un número suficientemente grande

Se_i^v = Tiempo de servicio del vehículo al cliente $i \in C$ [H]

Se_i^d = Tiempo de servicio para el UAV al cliente $i \in C'$ [H]

SL = Tiempo requerido para lanzar el UAV [H]

SR = Tiempo requerido para recuperar el UAV [H]

T_{max} = Tiempo máximo de duración de una ruta [H]

T_{max}^d = Tiempo máximo de duración de una ruta para el dron [H]

El C_{ij}^v se definió como el producto entre la distancia recorrida, el consumo y el precio de combustible. El costo del dron se estableció como un factor α del costo del vehículo. Adicionalmente, fue necesaria una nomenclatura complementaria para identificar los posibles puntos de partida o salida de un dron; una salida del dron se compone por tres nodos de la ruta desde los cuales se puede operar. Siendo P el conjunto de posibles salidas, representadas por las tuplas $\langle i, j, k \rangle$. En donde el primer nodo (i) representa la posición de lanzamiento del UAV, el segundo nodo (j) representa al cliente que es visitado por el UAV y, finalmente, el tercer nodo (k) representa la posición de recuperación del UAV por parte del vehículo. Un elemento $\langle i, j, k \rangle$, pertenece al conjunto P si se obedecen las siguientes condiciones:

- Posición del nodo de partida del dron o lanzamiento $i \in N_0$ de una tupla: es la ubicación desde la cual se puede lanzar un UAV y puede salir un camión.
- Posición del nodo llegada del dron/entrega $j \in \{C': j \neq i\}$ de una tupla: es el conjunto de clientes que pueden ser atendidos por el UAV diferente de la posición de lanzamiento i .
- Posición del nodo de encuentro camión-dron o de recuperación $k \in \{N_i^* : k \neq i, k \neq j, SL + SR + \tau_{ij}^D + \tau_{jk}^D + Se_j^D \leq e\}$ de una tupla: es la ubicación en la que el camión puede recuperar el UAV respetando la duración de la batería.

Conjuntamente, las tuplas $\langle 0, i, (c + 1) \rangle$ se encuentran excluidas del conjunto P para todo $i \in C'$. Dichas tuplas hacen referencia a las entregas que deben realizar los drones, donde sus posiciones de lanzamiento y recuperación son en el depósito. Para una salida $s = \langle i, j, k \rangle$, se define el costo como $C_s^D = C_{ij}^D + C_{jk}^D$. Adicionalmente, se establece a P_i^+ como todas las salidas

de P lanzadas desde el nodo $i \in N_0$, P_k^- como todas las salidas de P con un nodo de recuperación $k \in N_i^*$ y P_j como todas las salidas de P que sirven al cliente $j \in C'$.

Variables

p_{ij}^v : Indica si un cliente $j \in C$ se visita después de la ubicación $i \in N_0$ en la ruta del camión $v \in V$ [0 o 1]

x_{ij}^v : 1, si el vehículo $v \in V$ viaja desde i hasta j . 0, en otros casos

y_s^v : 1, si la salida $s \in P \langle i, j, k \rangle$ es usada en la ruta del vehículo $v \in V$. 0, en otros casos

z_{ij}^v : 1, si la salida $j \in C$ es visitada después de la ubicación $i \in N_0$ en la ruta del vehículo $v \in V$. 0, en otros casos

u_i^v : Indica la posición de la visita $i \in N$ en la ruta del vehículo $v \in V$ [(x, y)km]

τ_i^v : Indica el tiempo de la ruta del vehículo $v \in V$ desde un nodo n llegando a la ubicación $i \in N$ [min]

τ_i^D : Indica el tiempo de un dron $d \in D$ desde el vehículo $v \in V$ llegando a la ubicación $i \in N$ [min]

Donde:

$\tau_i^v = \frac{V_v}{d_v}$ en [min], siendo V_v la velocidad del vehículo y d_v la distancia que recorre el vehículo.

$\tau_i^D = \frac{V_{dc}/V_{dd}}{d_d}$ en [min], siendo V_{dc} la velocidad del dron cuando lleva un paquete o carga, V_{dd} la velocidad del dron cuando no lleva un paquete o carga y d_d la distancia que recorre el dron.

Función objetivo

a. Costo operacional de los vehículos

$$\left\{ \sum C_{ij}^v x_{ij}^v \right\} \quad (1)$$

b. Costo operacional de los drones UAV

$$\left\{ \sum C_s^D = \sum C_{ij}^D + \sum C_{jk}^D \right\} \quad (2)$$

c. F.O. = Min {Costo operacional de los vehículos + Costo operacional de los drones}

$$\text{Min } C = \sum_{v \in V} \left(\sum_{(i,j) \in A} C_{ij}^v x_{ij}^v + \sum_{s \in P} C_s^D y_s^v \right) \quad (3)$$

La función objetivo (3) minimiza el costo operativo de la visita a los clientes, esta se compone de dos partes: el costo de operación de la flota de vehículos (1) y el costo de operación de la flota de drones (2), utilizados para la distribución de los recursos. Para el caso de los drones se consideran dos costos: el primero, es el costo de operación desde la ubicación i , en que se puede lanzar un UAV hasta la ubicación de entrega de un cliente j . El segundo, es el costo de operación desde la ubicación del cliente j hasta el nodo de recolección k del UAV. La suma de los dos costos anteriores representa el costo de operación total de los drones y, a su vez, con la suma de los costos (1) y (2) se obtiene el costo total del recorrido de vehículos y drones trazado en la ruta.

Es importante destacar que, a diferencia de lo presentado en el modelo propuesto por [18] donde la velocidad del dron es un valor fijo, en la presente investigación se propone una ecuación para la velocidad, la cual considera el factor de carga del dron y la velocidad promedio del viento. Así mismo, se consultó hasta que porcentaje es recomendable usar la batería del dron, esto con el fin de no forzar el margen de uso de esta. La cual no debe estar por debajo del 30%, ya que puede ocasionar caídas repentinas, deterioro en el funcionamiento del dron o de su batería [20]. Por esta razón se incorpora una restricción sobre la carga energética del dron, que limita su funcionamiento si la capacidad de su batería (dada en tiempo de vuelo), se encuentra por debajo del 30% de la capacidad total. A continuación, se presentan las ecuaciones de velocidad del dron cuando lleva un paquete y cuando no lo está haciendo.

Ecuaciones de velocidad del dron:

$$V_{dc} = c_m(V_{fd}) - \bar{v}_v \quad (4)$$

$$V_{dd} = (V_{fd}) - \bar{v}_v \quad (5)$$

Donde:

V_{dc} = Velocidad del dron cuando lleva un paquete o carga $[m/s]$

V_{dd} = Velocidad del dron cuando no lleva un paquete o carga $[m/s]$

c_m = Factor de carga, $c_m = \frac{m_t + m_b}{m_t + m_b + m_l} < 1$

m_t = Masa del dron sin batería $[Kg]$

m_b = Masa de la batería del dron $[Kg]$

m_l = Masa de la carga o paquete en el dron $[Kg]$

V_{fd} = Velocidad de la ficha del dron $[m/s]$

$\bar{v}_v$ = Velocidad promedio del viento

$$\bar{v}_v = \frac{\sum V_v}{\# \text{velocidades}} \quad [m/s]$$

Las restricciones a las que se encuentra sujetas la función objetivo del problema son las siguientes:

$$\sum_{v \in V} \sum_{i \in \Delta^-(j)} x_{ij}^v + \sum_{v \in V} \sum_{s \in P_j} y_s^v = 1 \quad \forall j \in C \quad (6)$$

$$\sum_{j \in N_*} x_{0,j}^v \leq 1 \quad \forall v \in V \quad (7)$$

$$\sum_{i \in N_0} x_{i,c+1}^v \leq 1 \quad \forall v \in V \quad (8)$$

$$x_{0,c+1}^v = 0 \quad \forall v \in V \quad (9)$$

$$\sum_{i \in \Delta^-(j)} x_{ij}^v = \sum_{k \in \Delta^+(j)} x_{jk}^v \quad \forall v \in V, j \in C \quad (10)$$

$$u_i^v + 1 \leq u_j^v + M(1 - x_{ij}^v) \quad \forall v \in V, (i, j) \in A \quad (11)$$

$$u_j^v \leq M \left(\sum_{i \in \Delta^-(j)} x_{ij}^v \right) \quad \forall v \in V, j \in N_* \quad (12)$$

$$\sum_{j \in C} \left(\sum_{k \in \Delta^+(j)} d_j x_{jk}^v + \sum_{s \in P_j} d_j y_s^v \right) \leq q \quad \forall v \in V \quad (13)$$

$$\sum_{s \in P_i^+} y_s^v \leq 1 \quad \forall v \in V, i \in N_0 \quad (14)$$

$$\sum_{s \in P_k^-} y_s^v \leq 1 \quad \forall v \in V, k \in N_* \quad (15)$$

$$2y_s^v \leq \sum_{h \in \Delta^+(i)} x_{ih}^v + \sum_{l \in \Delta^-(k)} x_{lk}^v \quad \forall v \in V, s \quad (16)$$

$$\langle i, j, k \rangle \in P$$

$$\tau_i^v = 0 \quad \forall v \in V \quad (17)$$

$$\tau_i^D = 0 \quad \forall v \in V \quad (18)$$

$$\tau_{c+1}^v \leq T_{max} \sum_{i \in N_0} x_{i,c+1}^v \quad \forall v \in V \quad (19)$$

$$\tau_{c+1}^D \leq T_{max} \sum_{s \in P_{c+1}} y_s^v \quad \forall v \in V \quad (20)$$

$$\tau_h^v + t_{hk}^v + Se_h^T + SL \sum_{s \in P^+(h)} y_s^v + SR \sum_{s \in P^-(k)} y_s^v \leq \tau_k^v + T_{max}(1 - x_{hk}^v) \quad \forall v \in V, (h, k) \in A \quad (21)$$

$$\tau_i^v + t_{ij}^d + SL - T_{max} \left(1 - \sum_{s \in P_i^+ \cap P_j} y_s^v \right) \leq t_j^d \quad \forall v \in V, (i, j) \in A \quad (22)$$

$$\tau_i^d + t_{ij}^d + Se_j^d + SR - T_{max} \left(1 - \sum_{s \in P_j \cap P_k^-} y_s^v \right) \leq t_k^d \quad \forall v \in V, j \in C', k \in \Delta^+(j) \quad (23)$$

$$\tau_i^v - T_{max} \left(1 - \sum_{s \in P_i^+} y_s^v \right) \leq \tau_i^D \quad \forall v \in V, i \in N_0 \quad (24)$$

$$\tau_i^v + T_{max} \left(1 - \sum_{s \in P_i^+} y_s^v \right) \geq \tau_i^D \quad \forall v \in V, i \in N_0 \quad (25)$$

$$\tau_k^v - T_{max} \left(1 - \sum_{s \in P_k^-} y_s^v \right) \leq \tau_k^D \quad \forall v \in V, k \in C \quad (26)$$

$$\tau_k^v + T_{max} \left(1 - \sum_{s \in P_k^-} y_s^v \right) \geq \tau_k^D \quad \forall v \in V, k \in C \quad (27)$$

$$T_{max}^d \leq 0,7 \quad (e) \quad (28)$$

$$+T_{max} \left(1 - \sum_{s \in P_i^+ \cap P_k^-} y_s^v \right) \geq \tau_k^D - \tau_i^D \quad (29)$$

$$\forall v \in V, i \in N_0, k \in N_* \quad (u_j^v - u_i^v) \leq M(p_{ij}^v) \quad (30)$$

$$\forall v \in V, i \in N_0, j \in C \setminus \{i\} \quad (u_j^v - u_i^v) \geq M(p_{ij}^v - 1) + 1 \quad (31)$$

$$\tau_k^D - T_{max} \left(3 - \sum_{s \in (P_i^+ \cap P_k^-)} y_s^v - \sum_{s \in P_b^+} y_s^v - p_{ib}^v \right) \leq \tau_b^D \quad \forall v \in V, i \in N_0, k \in N_*, b \in C \setminus \{b\} \quad (32)$$

$$x_{ij}^v \in \{0,1\} \quad \forall v \in V, (i,j) \in A \quad (33)$$

$$y_s^v \in \{0,1\} \quad \forall v \in V, s \in P \quad (34)$$

$$u_i^v, \tau_i^v, \tau_i^D \geq 0 \quad \forall v \in V, i \in N \quad (35)$$

$$p_{ij}^v \in \{0,1\} \quad \forall v \in V, i \in N_0, j \in C \setminus \{i\} \quad (36)$$

La restricción (6) asegura que cada cliente sea visitado exactamente una vez, ya sea por un vehículo o por un avión no tripulado. La restricción (7) asegura que todos

los camiones deben salir del depósito como máximo una vez. De manera similar, en (8) se garantiza que todos los camiones deben regresar al depósito como máximo una vez. Además, está prohibido viajar entre depósitos según lo dado por (9). Las restricciones de conservación de flujo para el vehículo se definen en (10). Las restricciones (11) y (12), se refieren a la eliminación de recorridos secundarios para el vehículo en relación con la posición de un cliente visitado en la ruta del vehículo. También, la restricción de capacidad de los vehículos y drones se da en (13). Igualmente, el UAV puede ser lanzado y recuperado como máximo una vez desde cada nodo, como se indica en las restricciones (14) y (15). La restricción (16) asegura que si el UAV se lanza y recupera en la ubicación $i \in N_0$ y $k \in N_*$, el vehículo visita las mismas ubicaciones, respectivamente. La inicialización de los tiempos para el vehículo y el UAV al comienzo de cada ruta se dan en (17) y (18). Así mismo, la duración máxima de una ruta se establece imponiendo un límite al tiempo de regreso al depósito por los vehículos, como se indica en (19) y (20).

Las limitaciones de tiempo para el movimiento del vehículo (21), definen el momento en que el vehículo llega a la ubicación con respecto a las acciones correspondientes que pueden ocurrir en el medio. Las limitaciones de tiempo para el movimiento del dron establecen el tiempo en que un UAV visita un cliente según la posición del vehículo, estas se definen en (22) y (23). La restricción (22) asegura que, si se lanza un UAV desde la ubicación $i \in N_0$ al cliente $j \in C'$, entonces el tiempo de llegada del UAV al cliente j debe ser mayor que el tiempo de llegada del vehículo a la ubicación i más el tiempo de viaje del UAV entre la ubicación y el tiempo de lanzamiento del UAV. De manera similar para las operaciones de recuperación, la restricción (23) asegura que si un UAV termina de atender a un cliente $j \in C'$ y vuela de regreso al vehículo en la ubicación $k \in N_*$, la suma de los tiempos de llegada del UAV a la ubicación j , el tiempo de viaje entre las ubicaciones $j - k$, el de servicio del UAV en el nodo cliente j y el tiempo necesario en la recuperación de este; debe ser menor que el tiempo de llegada del UAV a la ubicación k .

Las restricciones de sincronización de tiempo para el vehículo y el UAV se definen en (24)-(27). Estas restricciones imponen que el lanzamiento y las operaciones de recuperación estén sincronizadas en el tiempo. Teniendo en cuenta que la sincronización no es necesaria cuando el vehículo y el UAV están regresando por separado al depósito, por ende, las restricciones (26) y (28) se definen para el conjunto de clientes C en lugar de N_* . La restricción de resistencia para la batería del UAV viene dada por las variables de lanzamiento y

recuperación. El tiempo máximo de vuelo de un dron en una ruta, no puede estar por debajo del 30% de la capacidad de su batería, esto se asegura con la restricción (28). La restricción (29) asegura que si se lanza un UAV desde la posición $i \in N_0$ al cliente $j \in C$ para recuperarlo en la posición $k \in N_*$, entonces la diferencia de tiempo entre las operaciones debe satisfacer el tiempo de la duración de la batería del dron.

La variable binaria p_{ij}^v define el orden en que el vehículo realiza las visitas, estableciendo que un cliente es visitado antes que otro en la ruta. El valor de esta variable se asigna de acuerdo con la diferencia de posiciones entre la ubicación del cliente y el punto de partida del dron, como se indica en (30) y (31). Suponiendo la ubicación de salida $i \in N_0$ y el cliente $j \in C$, si el vehículo visita al cliente j después de estar en la ubicación i , entonces la diferencia entre las variables de posición u_i^v será positiva, imponiendo el valor uno a p_{ij}^v . Por el contrario, si la diferencia es negativa, p_{ij}^v se impone el valor de cero. Cuando un vehículo no visita a un cliente en la ruta, establecido en las restricciones (11) y (12) se define la posición de la visita como cero. Por lo tanto, en la restricción (31) se agrega un 1 en el lado derecho para garantizar que p_{ij}^v permanezca en cero cuando tanto u_i^v como u_j^v tomen el valor de cero.

La restricción (32) fue definida para evitar que ocurrieran nuevos lanzamientos mientras el UAV estaba volando en la ruta. La hora de llegada de un UAV a otra ubicación del cliente $b \in C$ desde el cual el UAV puede ser lanzado nuevamente tiene que ser mayor que la hora de llegada a la ubicación $k \in N_*$, sí y solo si un dron se recupera en la ubicación k . Esto se tiene en cuenta en el orden en que el vehículo visita los lugares i y b (p_{ij}^v), si se produce un lanzamiento desde la posición b y si un avión aéreo no tripulado se recupera en la ubicación k . Finalmente, el dominio de las variables se define en las restricciones (33)-(36).

2.3. Diseño del ALNS

El modelo matemático formulado se resolverá mediante el uso del algoritmo *Adaptive Large Neighborhood Search*, cuyo procedimiento de solución se presenta en la figura 1. Este algoritmo consiste en mejorar progresivamente una solución inicial, destruyendo y reparando reiterativamente la solución parcial [21].

2.3.1. Algoritmo de generación de la solución inicial

En esta investigación, la solución inicial se dividió en dos pasos, el primero enfocado a la creación de las rutas para

los vehículos haciendo uso de la heurística de Clarke y Wright en su forma básica, y de la extensión propuesta por [23] en la que se incluyen los parámetros de λ y μ ; adicionalmente, a esta última se le aplica un operador por desplazamiento (*Shift* mutación) con el fin de intentar mejorar la solución inicial. En el segundo paso, para la construcción de la solución inicial, se realiza la inserción de los drones a las rutas creadas mediante el algoritmo de inserción de drones.

El algoritmo de ahorros de Clarke y Wright (CWA) [24] se preocupa por el enrutamiento inicial de suministros, los cuales se llevan a cabo desde un depósito central a una gran cantidad de puntos de entrega que, a través de una búsqueda en un espacio delimitado, logra encontrar los ahorros más representativos entre dos puntos. El proceso que sigue el algoritmo se basa en hallar los ahorros de distancia entre cada par de clientes, seguidamente se procede a ordenarlos de mayor a menor, posteriormente se va trazando la ruta entre dos puntos en el sistema, respetando siempre que la suma de las cargas asignadas sea por lo menos igual a la capacidad del vehículo transportador de mercancía.

Se evalúan dos rutas diferentes $(0, \dots, i, 0)$ y $(0, j, \dots, 0)$ que pueden ser combinadas formando una nueva ruta $(0, \dots, i, j, \dots, 0)$, la nueva ruta se crea sí y solo si el ahorro (en distancia) obtenido por dicha unión es mayor que cero. El ahorro para cada conjunto de rutas r es calculado mediante la siguiente expresión:

$$S_{ij} = d_{i0} + d_{0j} - d_{ij} \quad (37)$$

El algoritmo de ahorros tiene una versión paralela en la que se trabaja sobre todas las rutas simultáneamente, y otra secuencial que construye las rutas una a la vez. En este trabajo de investigación solo se hizo énfasis en la segunda versión.

Como se mencionó previamente, la extensión del algoritmo de ahorros presenta los parámetros de peso λ y μ . El parámetro λ es usado para evitar que se generen rutas circulares, impidiendo así dar lugar a la unión de puntos lejanos. [25] y [26], formularon este parámetro entre $0,1 \leq \lambda \leq 2$. Adicionalmente, se añadió el parámetro μ el cual tiene en cuenta la asimetría en términos de distancia desde el depósito hasta los nodos que se están evaluando, por ello [23] añadió a la expresión (37) un nuevo termino que estaba entre $0 \leq \mu \leq 2$. Debido a esto, la ecuación de ahorros sufre unas leves modificaciones, las cuales se presentan a continuación.

$$S_{ij} = d_{i0} + d_{0j} - \lambda d_{ij} - \mu |d_{0i} - d_{0j}| \quad (38)$$

Luego de la creación de la ruta inicial donde solo se consideran los vehículos, se procede a aplicar el algoritmo de inserción para incluir a los drones en las rutas creadas. Este algoritmo tiene en cuenta las restricciones de capacidad de vuelo y de carga del dron; el proceso inicia identificando los clientes que pueden ser atendidos por los drones, luego de esto se evalúa en la ruta del vehículo las posibles tuplas que se pueden generar para atender a dichos clientes trazando la que permita reducir los costos totales de la ruta.

2.3.2. Algoritmos de destrucción

El algoritmo sigue su procedimiento aplicando un algoritmo de destrucción a la solución inicial s , el cual consiste en remover un número de nodos de la solución. La cantidad de nodos que se quitan depende de la tasa de destrucción y esta es definida previamente al momento de seleccionar la heurística que va a ser aplicada en cada fase. La tasa es un número entero determinado aleatoriamente entre un intervalo, dicho intervalo se define con un valor mínimo (d_{min}) y uno máximo (d_{max}), en función del número de clientes totales. La tasa de destrucción debe estar por debajo del 0,5 de n y se sugiere que esta sea entre el 10% y 40% cuando se trate de instancias pequeñas (50 a 100 clientes) y entre 30 y 60 clientes cuando el número de nodos exceda dicho intervalo [27]. Las normas por las cuales se rige la elección de los clientes dependen de las heurísticas empleadas; las heurísticas usadas en la remoción de los clientes fueron, *Random Removal* (figura 2), *Worst Removal* (figura 3) y *Cluster Removal* (figura 4); a continuación, se hace una explicación del funcionamiento de cada una de ellas.

La eliminación aleatoria o *Random Removal* se encarga de retirar de manera uniforme y aleatoria γ clientes de la solución inicial, dando lugar a rutas parciales.

La heurística de destrucción *Worst Removal*, *Critical Removal* o peor eliminación, basa su mecanismo en ordenar los clientes servidos de una solución s de manera decreciente de acuerdo con sus costos, estos son guardados en una lista $\mathcal{L}$; aplicando como criterio la diferencia de costos $\Delta C_{-i}(s)$ de cada cliente en la ruta, esta se calcula a partir de la diferencia entre las funciones $[C(s) - C_{-i}(s)]$, la cual se compone de los valores de la solución s incluyendo y excluyendo al cliente i . El propósito central de esta heurística es eliminar al cliente de mayor costo. La eliminación en cada iteración se hace mediante el operador $\lfloor \vartheta^m \rfloor$ [28], con este se obtiene la posición del cliente a ser retirado de las lista. La forma

en la que se realiza esto es eligiendo uno a uno los γ clientes, generando así un número aleatorio $\vartheta \in [0, 1]$ y un factor $m \geq 1$ con el que se logra la diversificación de la heurística y controla su aleatoriedad, cuando $m = 1$ la remoción de los clientes en la solución s es hecha totalmente aleatoria.

Finalmente, la heurística de eliminación *Cluster Removal* inspira sus mecanismos en los resultados observados a partir del comportamiento de la remoción orientada a la distancia [29], para hacerlo divide el conjunto de nodos de una ruta en k subconjuntos disjuntos, siendo $k \geq 2$, agrupando los nodos que se encuentren de alguna manera “ceranos entre sí”. El algoritmo *Cluster Removal* comienza su procedimiento con la tasa de destrucción que es un valor aleatorio entre 10% y 30% del total de clientes, luego se procede a determinar el número de k subconjuntos a conformar, este se calcula aproximando hacia abajo la división de los clientes totales y el número de clientes a remover establecidos por la tasa. Los k subconjuntos se generan mediante el método de *K-means*, que tiene en cuenta los centroides con el fin de calcular los clientes que se encuentre más cercanos entre sí y así formar los clústeres, este proceso se repite iterativamente hasta lograr obtener clústeres de forma uniforme en la solución.

Se tienen dos casos: el primero es en el que el número de clientes a remover es mayor que el número de clientes pertenecientes al clúster, para este escenario se remueven todos los clientes que lo componen, la selección de este se hace al azar, luego los clientes faltantes para cumplir la tasa de destrucción son elegidos al azar de otro subconjunto de k clústeres creados. Por otro lado, tenemos el caso en que la cantidad de nodos a remover es menor al número de clientes del subconjunto seleccionado, estos son removidos en su totalidad. Estos procesos se repiten hasta que se completa con un número de clientes definidos por la tasa de destrucción. Una ventaja de este método es que contempla distintos clientes de distintas rutas y logra agrupar nodos de instancias de gran tamaño, debido a que el recorrido empieza y finaliza en la posición central y visita varios grupos, estos subconjuntos se implementan para mejorar notablemente la distancia d entre las rutas creadas y de esta manera mejorar los costos de solución.

La heurística de destrucción trata de eliminar un grupo completo y asignarlo a un recorrido distinto, por ello, lo que se busca es evitar una larga distancia entre la unión de ellos [30].

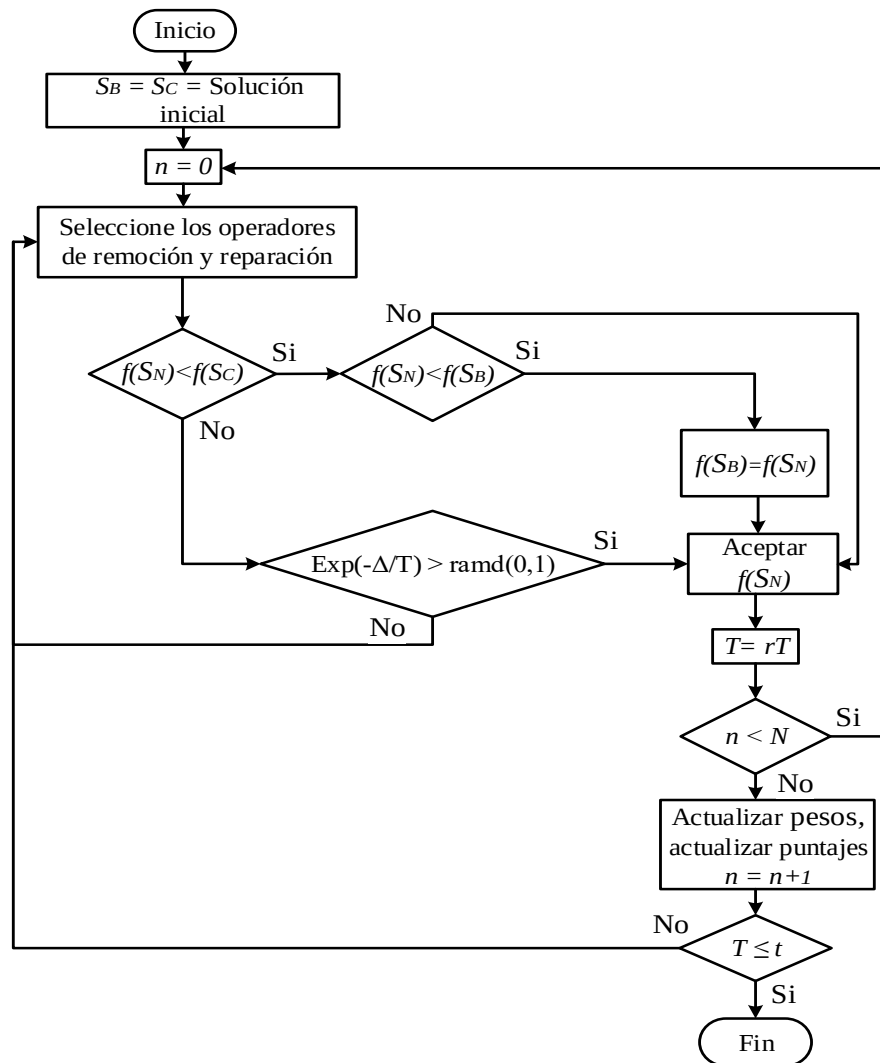


Figura 1. Diagrama de flujo del algoritmo ALNS. Fuente: Adaptado de [22].

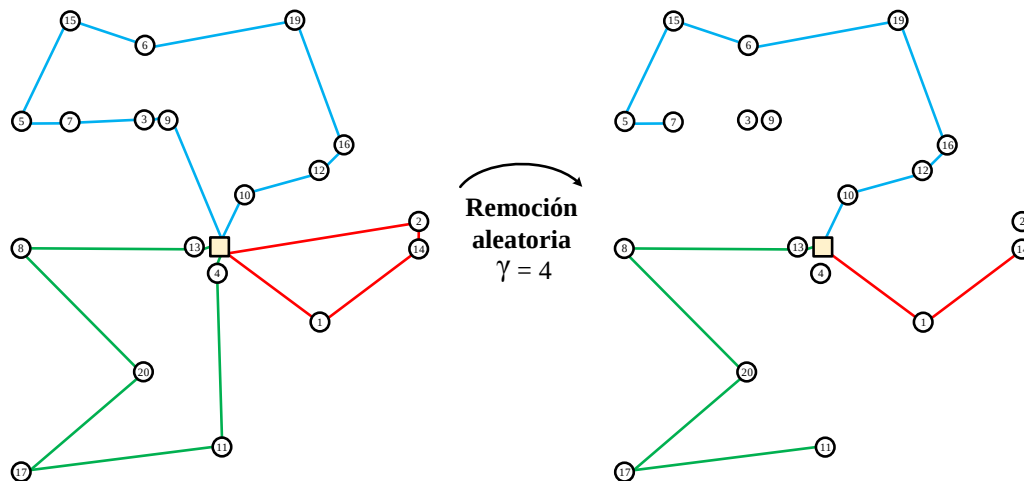
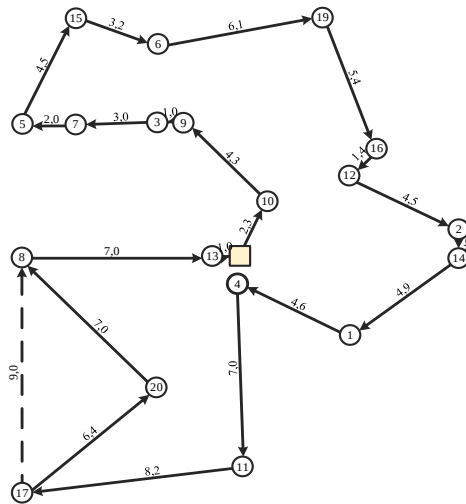


Figura 2. Ejemplo de la heurística *Random Removal* eliminando cuatro de veinte clientes. Fuente: elaboración propia.



Posición en L	Nodo	Costo
1	N17	10.2
2	N4	5.2
3	N20	4.4
4	N11	4
5	N8	3.9
6	N19	2.5
7	N5	2.5
8	N15	1.9
9	N12	1.6
10	N1	1.3
11	N10	1.1
12	N16	0.7
13	N2	0.5
14	N6	0.4
15	N14	0.3
16	N9	0.3
17	N7	0
18	N3	0
19	N18	0
20	N13	-1

$$C_{-20}(s) = 84,4 - 7 - 6,4 + 9 = 80,4$$

$$\Delta C_{-20}(s) = C(s) - C_{-20}(s) = 84,4 - 80,4 = 4,4$$

Figura 3. Ejemplo del algoritmo *Worst Removal* para una ruta compuesta por 20 clientes. Fuente: elaboración propia.

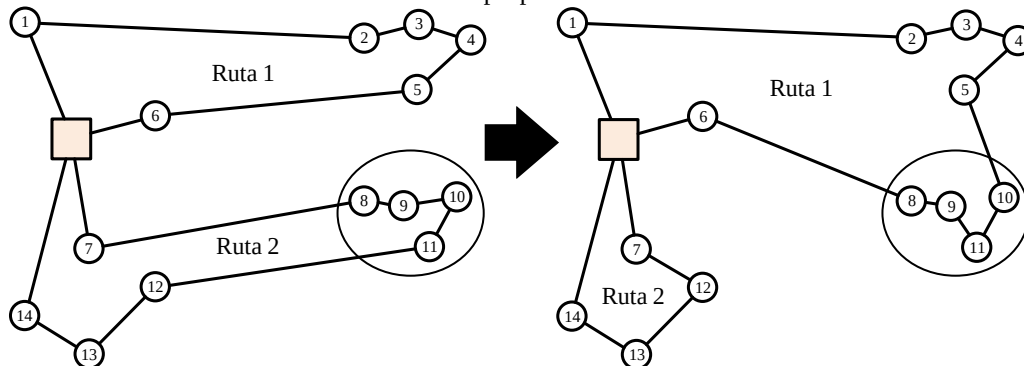


Figura 4. Ejemplo de la heurística *Cluster Removal* para cuatro clientes removidos. Fuente: Adaptado de [31].

2.3.3. Algoritmos de inserción

Para la fase de reparación de las rutas, se proponen 3 algoritmos, de los cuales dos son de tipo voraz, *Greedy Insertion with Noise function* (GIN) y *Regret-n Insertion* (nRI) debido a que reparan la solución parcial tomando a los clientes removidos e insertándolos mediante el cálculo del costo de inserción de los clientes que aportan una ganancia a la solución parcial. La tercera heurística usada se denomina *Repair Close Insertion* la cual contempla las distancias para realizar la inserción toma al nodo más cercano que pertenece a la solución parcial actual.

La primera heurística se conoce como inserción codiciosa con función de ruido o *Greedy Insertion with Noise function* (GIN) (figura 5), esta es una versión extendida del *Greedy Insertion* (GI), la cual se diferencia de esta última porque contempla un grado de libertad a la hora de elegir el mejor lugar en el cual se va a posicionar un nodo, con el que se obtiene un nuevo costo, este se

utiliza para localizar la ubicación en la cual se va a reinsertar el nodo. Esta extensión incorpora $\bar{d}$, α y ε al costo actual de la ruta, en donde $\bar{d}$ equivale a la distancia más larga que recorre el camión entre los nodos, si considera insertar a un nodo entrega del camión. De la misma manera, si se contempla la opción de insertar un cliente como nodo de dron, $\bar{d}$ será la distancia máxima que recorre un dron. Asimismo α representa un parámetro de ruido que es empleado para variar la selección de los nodos a escoger, este toma el valor de 0,1[32]. Adicionalmente, se tiene ε el cual es un número aleatorio entre $[-1,1]$.

El costo actual se compone por los costos de los camiones y drones. Siendo así como el GIN halla la posición y logra reinsertar un nodo con un mínimo costo, el cálculo mediante el cual se obtiene dicho costo es el siguiente:

$$\text{NuevoCosto} = \text{CostoActual} + (\bar{d} * \alpha * \varepsilon) \quad (39)$$

La segunda heurística usada es *Regret-n Insertion* (nRI) (figura 6), al igual que la anterior basa su mecanismo en los costos; sin embargo, en esta heurística no solo se tiene en cuenta la mejor posición de inserción, sino que se evalúan las n mejores posiciones para cada cliente removido buscando con esto evitar los problemas presentados en las heurísticas netamente voraces, uno de ellos es que del conjunto de clientes removidos comienzan a ser insertados en la ruta los clientes con mayores ahorros, pero al evaluar los últimos nodos pueden resultar difíciles de insertar llegando a generar soluciones no factibles. La ecuación (40) empleada en esta heurística calcula la diferencia de valor entre la n -ésima y la mejor posición del cliente, gracias a esto en las primeras iteraciones se puede identificar aquellos clientes críticos y los va insertando en las rutas.

$$\arg \max_{r \in \mathcal{L}} \left\{ \sum_{i=2}^n [c(s_i(r)) - c(s_1(r))] \right\} \quad (40)$$

La metodología de la tercera heurística de reparación *Repair Closest Insertion* (RCI), trata de insertar a los nodos removidos mediante el cálculo de distancias, seleccionando el nodo más cercano que pertenece a la solución parcial actual. El procedimiento comienza seleccionando aleatoriamente a un cliente de la lista $\mathcal{L}$ de clientes removidos, luego calcula la mejor inserción de este mediante el criterio del nodo más cercano que pertenece a la ruta parcial, siempre y cuando se cumplan

las restricciones de capacidad del vehículo y duración de la ruta, además que el nodo de la ruta parcial no tenga más de dos vecinos en la ruta. Dentro de la evaluación también determina si es menos costosa la inserción de un dron, el nodo insertado es removido de la lista $\mathcal{L}$ y se continúa con la selección aleatoria de otro nodo. En el caso de que el nodo seleccionado de la lista $\mathcal{L}$ no pueda ser insertado, este nodo es agregado al conjunto D_N de clientes faltantes por insertar, los cuales se agregan a la solución parcial mediante el algoritmo de *Greedy Insertion with Noise Function* (GIN).

2.3.4. Parametros del Algoritmos

Para el adecuado funcionamiento del algoritmo ALNS fue necesario definir cinco parámetros importantes, los cuales son: el periodo de actualización de los pesos, los pesos a asignarse a cada heurística tanto de destrucción como de reparación, basados en el éxito de cada heurística con relación a la función objetivo, el intervalo de la tasa de destrucción usado en estas heurísticas, el criterio de aceptación, el cual admite una solución factible peor que la actual y finalmente un factor que disminuye la temperatura en cada iteración y funciona como criterio de parada en el algoritmo.

El primer parámetro es el periodo de actualización de los pesos, según [28] la actualización de los pesos en las heurísticas debe realizarse cada cierto número de iteraciones p_u , esto se conoce como periodo de actualización o segmento u .

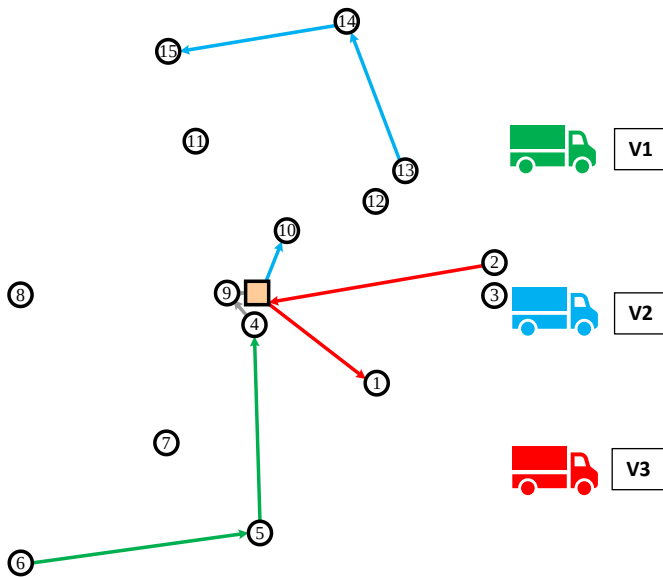


Figura 5. Ejemplo del algoritmo GIN para 15 clientes con la inserción del nodo 9 entre los nodos 4 y 0. Fuente: elaboración propia.

$\mathcal{L} = \{N_3, N_7, N_8, N_9, N_{11}, N_{12}\}$								
Vehículo	i	j	k	Costo	$\bar{d}$	α	ϵ	Nuevo costo
V1	N3	N4	N0	16,242	26,426	0,1	0,000	16,242
V1	N3	N0	N6	26,426	26,426	0,1	-0,458	25,216
V2	N3	N10	N13	12,507	26,426	0,1	0,000	12,507
V2	N3	N15	N0	21,789	26,426	0,1	0,435	22,933
V3	N3	N1	N2	6,049	26,426	0,1	0,335	6,935
V1	N7	N4	N0	10,826	20,059	0,1	0,000	10,826
V1	N7	N0	N6	12,238	20,059	0,1	0,000	12,238
V2	N7	N15	N0	18,953	20,059	0,1	0,919	20,796
V2	N7	N4	N0	10,826	20,059	0,1	0,022	10,871
V3	N7	N1	N2	20,059	20,059	0,1	0,530	21,123
V1	N8	N0	N6	16,980	28,546	0,1	-0,706	14,963
V1	N8	N4	N0	16,022	28,546	0,1	0,825	18,377
V2	N8	N14	N13	22,980	28,546	0,1	0,000	22,980
V2	N8	N15	N0	17,575	28,546	0,1	-0,669	15,664
V3	N8	N1	N2	28,546	28,546	0,1	0,738	30,654
V1	N9	N4	N0	2,423	15,029	0,1	-0,632	1,479
V1	N9	N0	N6	12,430	15,029	0,1	-0,144	12,213
V2	N9	N15	N0	9,336	15,029	0,1	0,000	9,336
V2	N9	N10	N13	10,209	15,029	0,1	0,000	10,209
V3	N9	N1	N2	15,029	15,029	0,1	-0,236	14,674
V1	N11	N0	N6	20,765	21,063	0,1	0,899	22,658
V1	N11	N4	N0	11,926	21,063	0,1	0,995	14,023
V2	N11	N15	N0	8,623	21,063	0,1	-0,538	7,488
V2	N11	N10	N13	11,480	21,063	0,1	0,614	12,773
V3	N11	N1	N2	21,063	21,063	0,1	-0,240	20,557
V1	N12	N0	N6	22,102	22,102	0,1	0,185	22,512
V1	N12	N4	N0	10,871	22,102	0,1	0,950	12,970
V2	N12	N10	N13	4,600	22,102	0,1	0,189	5,019
V2	N12	N15	N0	13,690	22,102	0,1	0,000	13,690
V3	N12	N1	N2	10,588	22,102	0,1	-0,460	9,571

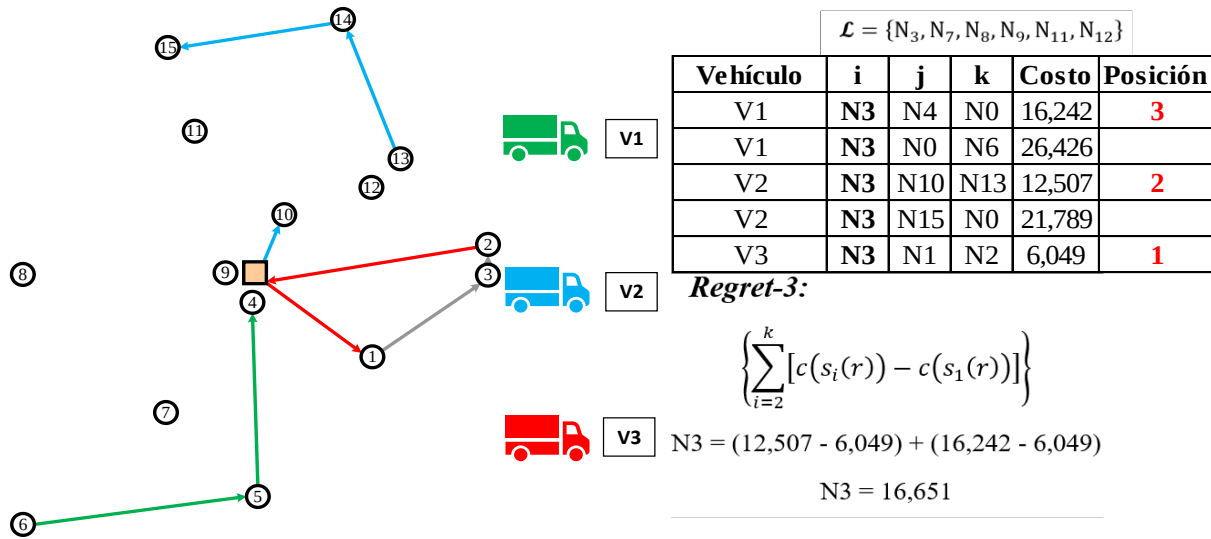


Figura 6. Ejemplo del algoritmo nRI para quince clientes realizando la inserción del nodo tres. Fuente: elaboración propia.

Los resultados del trabajo investigativo recomiendan ajustar razonablemente dicho periodo con base en la cantidad de heurísticas a usar en el ALNS, ya que si este es un valor pequeño, puede suceder que algunas de las heurísticas usadas no logren ejecutarse desde el principio, esto ocasiona que sus pesos se reduzcan, dicha reducción puede deberse a la falta de pruebas de la heurística y no a un mal desempeño de esta; con el fin de evitar estos problemas y además reducir el esfuerzo computacional que resultaría cuando se produce una modificación, Ropke y Pisinger [31] sugieren que el reajuste de los pesos suceda cada 100 iteraciones cuando se tengan más de diez heurísticas en total [28].

El segundo parámetro son los pesos adaptativos de las heurísticas, que varían según la calidad de la solución alternativa generada, en cada iteración los pesos son modificados [33]. Como se mencionó en el párrafo anterior, los pesos de las heurísticas son actualizados en un periodo p_u de iteraciones, estos pesos fueron definidos como $w(\theta_d)$ y $w(\theta_r)$, donde $\theta_d \{d = 1, 2, 3\}$ es el conjunto de heurísticas de destrucción y $\theta_r \{r = 1, 2, 3\}$ el conjunto de heurísticas de reparación. La elección de la heurística se realiza usando *The roulette wheel selection principle*, donde en cada iteración la probabilidad de seleccionar una heurística se define como:

$$P(\theta_d) = \frac{w(\theta_d)}{\sum_{l=1}^3 w(\theta_l)} \quad (41)$$

$$P(\theta_r) = \frac{w(\theta_r)}{\sum_{l=1}^3 w(\theta_l)} \quad (42)$$

Inicialmente, la probabilidad de selección es igual para todas las heurísticas, esta se va ajustando en cada periodo de actualización mediante un factor de reacción $0 \leq \rho \leq 1$, la puntuación Ψ que depende únicamente del tipo de solución obtenida con las heurísticas usadas, al final de cada iteración esta se va sumando en π_d para cada heurística de destrucción y en π_r en las de reparación, las cuales inicialmente tienen valor cero. Las puntuaciones de Ψ a asignar se definen en la tabla 1, y finalmente θ_d y θ_r cuentan el número de veces que ha sido usada la heurística. Para hacer la actualización de los pesos se usa para las heurísticas de destrucción y reparación respectivamente:

$$w(\theta_d) = \begin{cases} w(\theta_d)(1 - \rho) + \rho \frac{\pi_d}{\theta_d}, & \text{si } \theta_d > 0 \\ w(\theta_d)(1 - \rho), & \text{si } \theta_d = 0 \end{cases} \quad (43)$$

$$w(\theta_r) = \begin{cases} w(\theta_r)(1 - \rho) + \rho \frac{\pi_r}{\theta_r}, & \text{si } \theta_r > 0 \\ w(\theta_r)(1 - \rho), & \text{si } \theta_r = 0 \end{cases} \quad (44)$$

$$= \frac{w(\theta_r)}{\sum_{l=1}^3 w(\theta_l)}$$

El tercer parámetro del algoritmo es el intervalo de la tasa de destrucción de estas heurísticas, esta se establece que debe estar por debajo del 0,5 de n , se sugiere que esta sea entre el 10% y 40% cuando se trate de instancias pequeñas (50 a 100 clientes) y entre 30 y 60 clientes cuando el número de nodos exceda el intervalo de clientes para instancias pequeñas [34], esto con la finalidad de ampliar el espacio de búsqueda. Sin

embargo, si se fija un valor demasiado alto se pueden presentar inconvenientes en la fase de reparación, debido a que las heurísticas de inserción no están diseñadas para reparar completamente una solución.

Tabla 1. Puntuaciones que toma el parámetro Ψ , según el tipo de solución generada

Parámetro Ψ	Descripción
σ_1	La nueva solución es una mejor solución global.
σ_2	La nueva solución es aceptada y su costo es mejor que la actual.
σ_3	La nueva solución es aceptada, pero con un costo peor que el actual.
σ_4	La nueva solución es rechazada.

Fuente: Adaptado de: [18].

El cuarto parámetro es el criterio de aceptación, la función de este parámetro es controlar y aceptar las soluciones creadas en cada iteración del algoritmo; para definirlo se usó la heurística de Recocido Simulado [35], [36], en la que se utiliza un parámetro de temperatura T que regula la probabilidad de aceptación. Si la solución S_N generada por las fases de destrucción y reparación da como resultado una solución con menor valor objetivo que la solución actual S_C , entonces se acepta. Sin embargo, si la solución S_N tiene un valor objetivo más alto, se hace uso del algoritmo de recocido simulado, con el que se calcula una probabilidad de aceptación $Prob(S_N)$, la cual es comparada con un número generado aleatoriamente entre cero y uno; la solución únicamente es aceptada si el valor de dicha probabilidad es mayor al número aleatorio $Random(0,1) < Prob(S_N)$. El cálculo de la probabilidad se realiza con la diferencia entre el costo de las dos soluciones a razón de una temperatura que disminuye dependiendo del tiempo máximo y el tiempo empleado iterativamente, a continuación, se presenta la fórmula para el cálculo de la probabilidad:

$$Prob(S_N) = e^{\frac{f(S_C) - f(S_N)}{T}} \quad (45)$$

Donde $f(S_C)$ y $f(S_N)$ son los valores de la función objetivo evaluados con S_C y S_N respectivamente, además es necesario definir una temperatura inicial T_{in} para T que se reduzca linealmente hasta hacerse cero; esto con la finalidad de que también se establezca como un criterio de parada del algoritmo, para esto se controla la temperatura usando el tiempo transcurrido t^{pass} desde el inicio del algoritmo y se toma t^{max} como el límite de tiempo impuesto en el algoritmo. Para calcular la nueva temperatura se emplea la siguiente fórmula:

$$T = T_{in} * \left(1 - \frac{t^{pass}}{t^{max}}\right) \quad (46)$$

3. Experimentación y resultados

La experimentación y el análisis de los resultados se presentan en esta sección, se estudiará el rendimiento del algoritmo para el VRPD empleado en esta investigación.

3.1. Validación del algoritmo

Varias de las instancias presentadas en distintos artículos no pueden ser usadas en el problema de ruteo de vehículos con drones, debido a que este contempla parámetros como la capacidad de carga del dron, los tiempos asociados al dron, la capacidad de las rutas, el tiempo de vuelo de la batería del dron, entre otros, esto le da un trato especial a este problema. Por lo tanto, evaluar dicho problema con instancias que no contemplen este tipo de parámetros lleva a un escenario que no refleja correctamente las propiedades reales del problema.

Esta investigación evalúa el desempeño del algoritmo ALNS planteado mediante las instancias propuestas por [37]. Estas instancias presentan una distribución uniforme $U(-d, d)$, las instancias son generadas al azar y cuentan con una acomodación en la que solo se contempla un depósito central ubicado en las coordenadas (0,0). Los clientes se ubican en una cuadrícula de dimensiones $2d$ por $2d$ alrededor del depósito. Cada instancia es denominada $n.m.t$, donde n es el número de clientes para el escenario, m es dimensión de la cuadrícula y t es el nombre genérico del escenario. Cada instancia se encuentra compuesta por tres columnas: coordenada en X, coordenada en Y, y demanda de cada cliente. Por último, se tiene un archivo denominado *RouteData*, que proporciona la información de los parámetros del modelo usados para la ejecución del algoritmo.

Las instancias planteadas por los autores tienen ocho grandes grupos, los cuales son de distintos tamaños 6, 10, 12, 20, 50, 100, 150 y 200 clientes respectivamente. cada uno de estos sets contiene tres subgrupos de instancias con configuración diferente, teniendo así un total de 112 instancias donde se plantean diferentes escenarios para el problema de ruteo de vehículos con drones. La dimensión de las cuadrículas se establece en gran medida por los drones, dado que estos pueden operar un recorrido de ida y vuelta con un radio de alrededor de 10 millas desde el centro de distribución [6]; por esta razón disponer de un tamaño por encima de ese podría causar implicaciones y escenarios en los que el dron no pueda funcionar de

manera independiente, por ende, se obliga a que el camión interactúe con el dron.

Las instancias usadas tienen seis tamaños de cuadrícula, 5, 10, 20, 30 y 40 millas respectivamente; se distribuyen 6,10,12 y 20 clientes en los tamaños de cuadrícula 5,10 y 20, denominadas instancias pequeñas, se tienen cuatro configuraciones para cada número de clientes y cuadrícula, siendo 48 problemas en total. Así mismo, se cuentan con 50, 100, 150 y 200 clientes distribuidos en cuadrículas de 10, 20, 30 y 40 con cuadros de diferentes configuraciones, siendo 64, consideradas instancias grandes; obteniendo así un total de 112 instancias para realizar la validación del algoritmo. Inicialmente, se hicieron pruebas con instancias de 100 y 200 clientes, encontrando que el tiempo computacional era alto; por lo tanto, por temas de eficiencia computacional, se decide tomar una instancia de cada configuración, es decir, 5, 10 y 20 para instancias pequeñas, 10, 20, 30 y 40 para instancias medianas y grandes, probando el algoritmo en 28 instancias en total. Por último, en el archivo *RouteData*, se proporciona la información de los parámetros del modelo, sus respectivos valores se encuentran en la [tabla 2](#). Cada instancia se corrió 10 veces, el algoritmo fue realizado y ejecutado en un equipo con procesador AMD A9 – 9410 RADEON R5, 5 COMPUTE CORES 2C + 3G y 8 GB de RAM instalada.

Tabla 1. Valores de los parámetros utilizados en la configuración del problema

Notación	Parámetro	Valor
SL	Tiempo requerido para lanzar el UAV	1 [min]
SR	Tiempo requerido para recuperar el UAV	1 [min]
V^v	Velocidad del vehículo	56.35 [Km/h]
V^d	Velocidad del dron	80.5 [Km/h]
e	Capacidad de vuelo del dron	30 [min]
q	Capacidad del vehículo	1300 [Kg]
W	Capacidad de carga del dron	5 [Kg]
P_c	Precio combustible	1.13 [€/l]
C_c	Consumo combustible	0.07 [l/Km]
α	Factor de costo del dron	10%
T_{max}	Tiempo máximo de duración de la ruta	480 [min]

Fuente: Adaptado de [\[18\]](#).

3.2. Resultados validación del algoritmo ALNS

Para la validación del algoritmo, además de los parámetros del modelo y de las instancias, fue necesario establecer los parámetros propios del ALNS, para esto se emplearon los presentados por [\[18\]](#), con excepción del valor de σ_4 el cual tiene un valor de 0, pero en esta investigación se propone usar un valor de 1 con el fin de favorecer la diversificación en la búsqueda de nuevas soluciones.

En la [tabla 3](#) se representan los símbolos, parámetros y valores usados para correr las soluciones del recocido simulado, heurísticas de destrucción e iteraciones que hacen parte del algoritmo ALNS, los valores de dichos parámetros son los recomendados por autores como D. Sacramento [\[18\]](#), Ropke y Pisinger [\[31\]](#) que en sus investigaciones encontraron valores que funcionan para encontrar soluciones aceptables, además de reducir el tiempo de ejecución.

Tabla 2. Valores de los parámetros del algoritmo ALNS

Símbolo	Parámetro	Valor
σ_1	Puntaje de la mejor solución	33
σ_2	Puntaje de mejorar la solución actual	9
σ_3	Puntaje de aceptar una solución peor	13
σ_4	Puntaje por crear una solución no factible	1
T_{in}	Temperatura inicial del algoritmo Recocido Simulado	42
T_f	Temperatura final del algoritmo Recocido Simulado	0,004
ω	Tasa de enfriamiento del algoritmo Recocido Simulado	0,9919
ρ	Factor de reacción	0,9
m	Factor de diversificación del algoritmo <i>Worst Removal</i>	3
p_u	Número de iteraciones para actualizar pesos y probabilidades	100
d_{min}	Tasa mínima de destrucción	10%
d_{max}	Tasa máxima de destrucción	30%
i_{max}	Numero de iteraciones máximo sin mejorar solución para instancias pequeñas	1000

Fuente: Adaptado de [\[18\]](#).

3.3 Experimentación de Instancias

El criterio de evaluación para las instancias será el mejor costo global para hacer la distribución a los clientes. En la [tabla 4](#) se presentan los resultados de la experimentación para 28 instancias, en esta se incluyen los mejores valores de $Min(C)$ (función objetivo) obtenidos por el ALNS, las soluciones iniciales obtenidas mediante el Clarke y Wright (CWA), CWA extendido, la mejora porcentual del ALNS se calcula respecto a las soluciones iniciales dadas por el CWA y el CWA extendido. El CWA genera soluciones más aptas que su versión extendida, de manera que el ALNS no consigue obtener mejoras significativas sobre estas soluciones iniciales, esto observa en los porcentajes de mejoras alcanzados. En instancias pequeñas el porcentaje

máximo se aproxima al 22% mientras que instancias grandes ronda el 12% para las soluciones del CWA.

En la [figura 7](#) se utiliza el número de veces en que las soluciones aceptadas proporcionaron una nueva mejor solución global, y las veces que se aceptaron las distintas heurísticas de reparación durante la ejecución del algoritmo.

El análisis del criterio de aceptación se hace mediante el recocido simulado, por esto se grafican las soluciones que son aceptadas en una prueba con 1000 iteraciones para las instancias 20.5.1 ([figura 7](#)) utilizando los dos métodos de generación de la solución inicial, el CWA y el CWA extendido.

Tabla 3. Resultados de las mejores soluciones conocidas para el ALNS, CWA y CWA extendido

Instancia	Mejor Min (C) conocida	CWA	Mejora 1 (%)	Mejor Min (C) conocida	CWA Extendido	Mejora 2 (%)
6.5.1	1,209	1,312	7,831%	1,128	1,128	0,000%
6.10.2	1,894	1,894	0,000%	1,894	1,894	0,000%
6.20.3	4,904	5,722	14,291%	5,163	5,163	0,000%
10.5.2	1,384	1,476	6,202%	1,384	1,462	5,323%
10.10.1	2,681	2,973	9,824%	2,681	2,790	3,904%
10.20.4	5,986	6,363	5,937%	5,986	6,802	12,005%
12.5.4	1,511	1,598	5,420%	1,511	1,718	12,039%
12.10.3	2,473	3,035	18,525%	2,473	3,083	19,799%
12.20.1	5,777	6,436	10,237%	5,777	6,655	13,199%
20.5.1	1,705	2,176	21,620%	1,655	2,208	25,053%
20.10.2	3,576	4,301	16,848%	3,462	4,085	15,259%
20.20.4	6,951	6,951	0,000%	7,270	9,895	26,530%
50.10.1	6,079	6,316	3,761%	7,350	9,702	24,244%
50.20.2	11,223	11,223	0,000%	13,891	15,000	7,390%
50.30.3	20,183	20,183	0,000%	22,837	22,950	0,490%
50.40.4	24,474	25,624	4,488%	29,033	30,952	6,199%
100.10.4	8,256	8,823	6,430%	12,983	14,246	8,868%
100.20.1	17,070	17,070	0,000%	24,010	26,034	7,773%
100.30.2	24,296	25,382	4,280%	36,796	38,711	4,947%
100.40.4	35,408	35,408	0,000%	46,165	49,330	6,416%
150.10.1	10,948	11,576	5,420%	15,196	17,151	11,394%
150.20.2	23,089	23,089	0,000%	30,646	35,395	13,418%
150.30.1	30,783	33,357	7,717%	45,981	50,738	9,376%
150.40.3	43,039	48,816	11,834%	69,664	72,756	4,250%
200.10.1	11,692	12,969	9,844%	17,886	20,086	10,955%
200.20.4	24,353	24,353	0,000%	40,842	40,842	0,000%
200.30.3	41,398	41,398	0,000%	60,507	61,167	1,078%
200.40.4	50,524	50,524	0,000%	73,358	73,358	0,000%

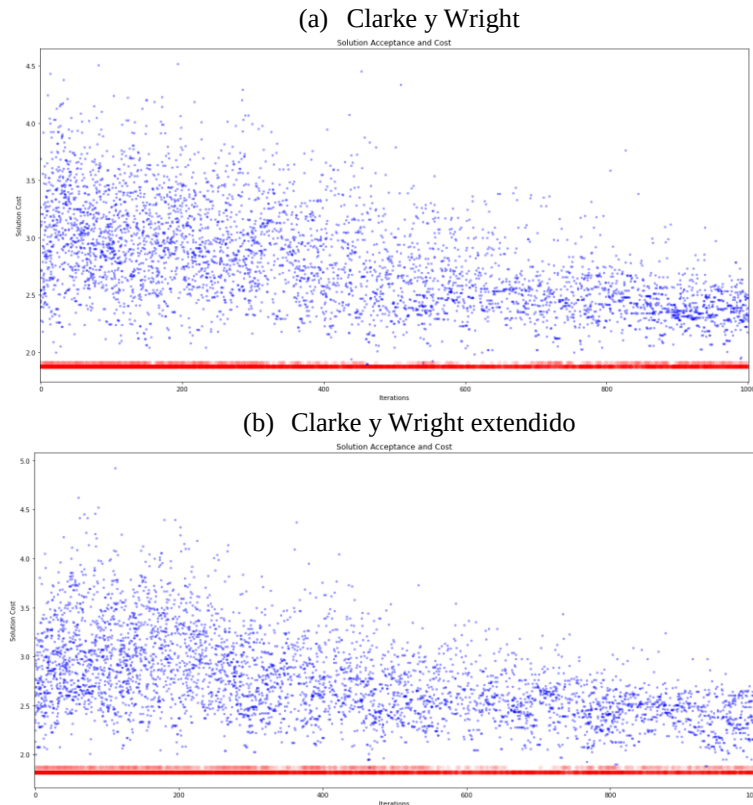


Figura 7. Resultados de la instancia de 20 clientes (a) para Clarke y Wright. (b) para Clarke y Wright Ext. Fuente: elaboración propia.

$$Mejora\ 1\ (\%) = \frac{CWA - Mejor\ Min(c)}{CWA} \quad (47)$$

$$Mejora\ 2\ (\%) = \frac{CWA\ Ext - Mejor\ Min(c)}{CWA} \quad (48)$$

La solución óptima se encuentra en las primeras iteraciones del algoritmo, en donde, la curva roja representa la mejor solución encontrada dado un número de iteraciones, presentado así un comportamiento plano. Los puntos azules que se registran en el espacio de solución son aquellas soluciones que fueron aceptadas de acuerdo con el criterio del recocido simulado, las cuales tienen costos peores que las mejores actuales y tiene un cambio agresivo, a media que transcurren las iteraciones estos son menos notorios. De acuerdo con la extensión del problema, el mecanismo del recocido simulado acepta propuestas de costos poco favorables al inicio de la búsqueda, con respecto a la mejor solución conocida y a medida que disminuye la temperatura, las variaciones van desapareciendo moderadamente y se llega a un punto en el que solo se aceptan aquellas soluciones ligeramente menores.

3.4 Resultados de los GAP del ALNS

La [figura 8](#), presenta los ahorros de la solución obtenida por la función objetivo y dada a partir del ALNS para el VRPD respecto a la solución inicial generada por Clarke y Wright y su versión extendida para las instancias de 20 clientes. En la mayoría de los casos, se nota cómo la metaheurística ALNS mejora la solución inicial en un 5-25%, demostrando así la efectividad del algoritmo en relación con que no permanece en los óptimos locales, permitiendo obtener mejores soluciones.

Para instancias pequeñas, el gráfico muestra dispersión respecto al ahorro en la solución inicial. Esto podría darse porque la solución inicial algunas veces halla soluciones casi óptimas, presentando una brecha pequeña en cuanto a la mejor solución generada por el ALNS. Los ahorros bajo el enfoque VRP, es decir, solo camiones, suelen estar entre el 50-70% y, a veces, incluso mayor, esto demuestra que hay una ventaja del uso combinado de camiones y drones, respecto al del VRP.

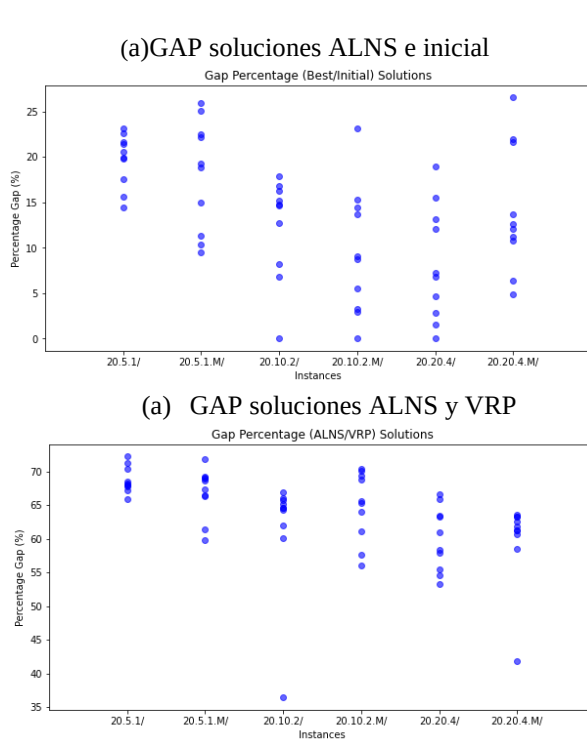


Figura 8. GAP del ALNS respecto a la solución inicial y VRP para instancias de 20 clientes.

Los GAP con respecto a la solución inicial (CWA) y bajo el enfoque de solo camiones (VRP) se calculan a partir de la siguiente expresión:

$$SI = 1 - \frac{Z^{ALNS(VRPD)}}{Z^{Inicial(CWA)}}, \quad (49)$$

$$SVRP = 1 - \frac{Z^{ALNS(VRPD)}}{Z^{VRP}}$$

3.5. Resultados característicos

3.5.1. Resultados de vuelo

La figura 9 presenta el porcentaje promedio del uso de la batería del dron, la cual tiende a aumentar a medida que incrementa la dimensión de la cuadrícula que contiene a los clientes, se debe agregar que este aumento está presente en todos los escenarios.

Usualmente, el consumo promedio de la batería del dron no supera el 45%, además se puede observar que para las instancias de 20 clientes no se tienen variaciones significativas entre la solución inicial generada por el Clarke y Wright y el CWA extendido.

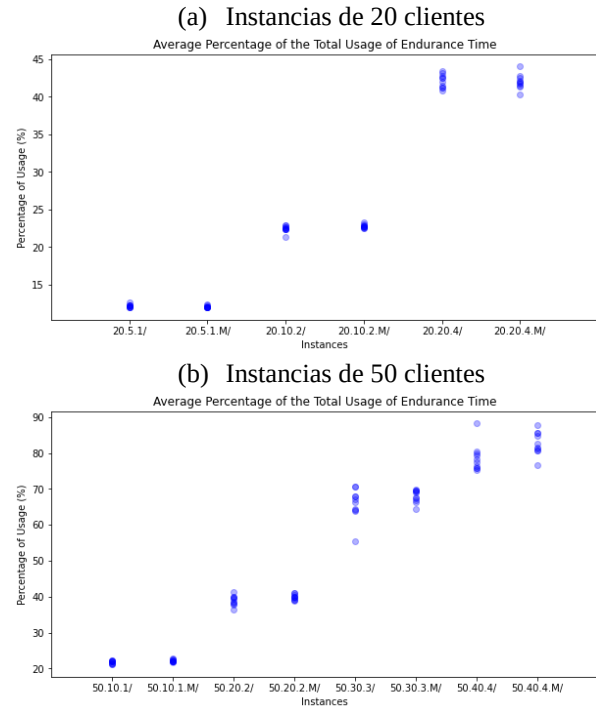


Figura 9. Porcentaje promedio del tiempo de la batería del dron. (a) utilizando instancias de 20 clientes y (b) instancias de 50 clientes. Fuente: elaboración propia.

3.5.2. Mejores soluciones

En la figura 10 se representan gráficamente las mejores soluciones encontradas por la metaheurística ALNS para el VRPD y VRP en un escenario con 20 y 100 clientes respectivamente. Cabe destacar cómo el escenario final para los camiones se ve afectado por la adición de drones en el problema, logrando así que las rutas de los drones tiendan a acortar significativamente los recorridos del camión, enviando al dron a aquellos clientes difíciles de alcanzar.

3.6. Resultados característicos

En el proceso de entrega otro factor que puede incidir es el tiempo, este se calcula mediante la velocidad del dron y la distancia recorrida por este. El valor que se estableció para la velocidad del dron inicialmente fue de 50 mph y es interesante estudiar qué sucedería si el dron operara a velocidades más bajas a causa de la carga que va a llevar este y la influencia que el aire impone sobre él.

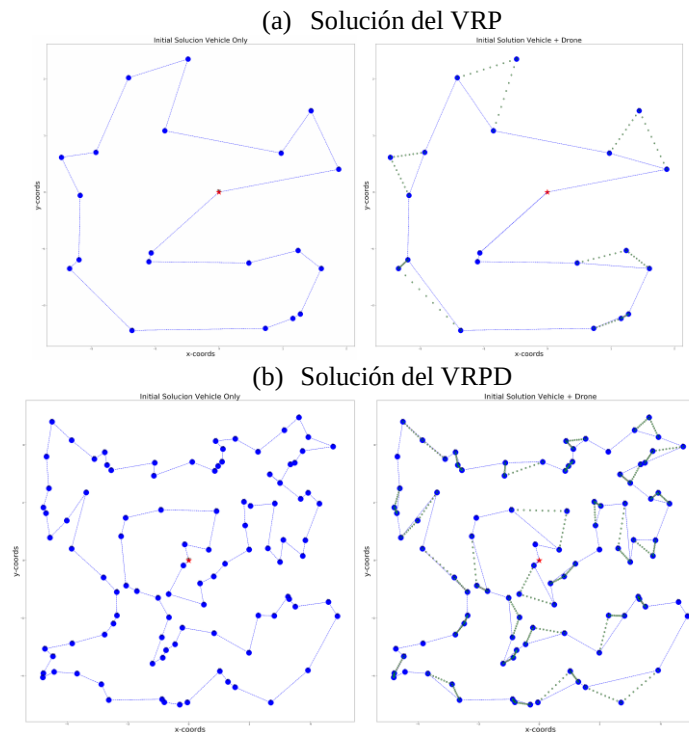


Figura 10. Mejor solución conocida para el VRP y VRPD, para instancias conformadas por 20 y 100 clientes, respectivamente. (a) soluciones con el VRP y (b) la solución con VRPD. Fuente: elaboración propia

Al disminuir la velocidad del dron, se amplía el tiempo que le va a tomar a este visitar los nodos, considerando que no puede violar las restricciones establecidas para la capacidad de la batería dada en tiempo de vuelo, es posible que no pueda alcanzar a todos los clientes que atiende en las soluciones actuales que abordan el parámetro fijo de velocidad. Por esta razón se decide evaluar la ecuación de velocidad propuesta en esta investigación en la sección 2.2. En la [tabla 5](#) y [tabla 6](#) se tienen las rutas que hace el dron dentro de cada instancia. Siendo $m_t + m_b$ la masa del dron (incluyendo batería cargada), m_l la masa del paquete, c_m el factor de carga, $\bar{v}$ la velocidad promedio del viento, V_{fd} la velocidad de la ficha del dron, d la distancia empleada en la ruta, t el tiempo sin incluir la influencia del viento y el factor de carga y por último el tiempo contemplando estos dos últimos factores.

4. Conclusiones

La influencia que tiene el estudio de los drones en los procesos de distribución de última milla tiene un efecto sumamente positivo. Las nuevas aplicaciones tecnológicas logran desplegar un sin número de intereses en las compañías que a su vez despiertan la atención de los investigadores en plantear tanto cuantiosos modelos matemáticos como estrategias que guíen a la resolución

de estos, y, por ende, aportar un amplio beneficio que proviene de la sinergia del VRPD, esto se puede conseguir gracias a la incorporación de los vehículos aéreos no tripulados en las entregas de mercancías, ya que las rutas que se conforman ahorran tiempo, costos y contribuyen a una programación más eficiente.

Es posible evidenciar que la corriente de investigación respecto al uso de drones en la distribución de última milla tiene tres vertientes, la primera consiste en que los camiones lanzan y recuperan a los drones en un mismo nodo; la segunda los camiones lanzan a los drones en un nodo cliente y este es recuperado en otro nodo cliente de la ruta; la tercera es que los drones pueden ser lanzados y recuperados en cualquier punto de la ruta de los vehículos, lo que ofrece una oportunidad para investigar el uso adecuado del dron dentro del ruteo de vehículos. Debido a que en las primeras fases la metaheurística ALNS cuenta con un algoritmo de ahorros (CWA) el cual logra encontrar buenas soluciones iniciales, esto influye en gran medida a la aplicación de las heurísticas de destrucción y reparación en instancias pequeñas.

Tabla 4. Resultados tiempos de vuelo del dron, al variar su velocidad con y sin la incidencia del viento, probado para 4 instancias de tamaños 6, 10, 12 y 20 clientes cuadrícula de 5x5

Cuadrícula de 5x5											
Ruta	Instancia	$m_t + m_b$	m_l	C_m	$\bar{v}_v$	V_{fd}	V_{dc}	d	t_v	t	T_{max}
1-3-2	6.5.1	15	0,8688	0,9453	6,211	50	41,0514	7,504	10,9682	9,005	21
5-6-4	6.5.1	15	1,3078	0,9198	3,106	50	42,8846	5,747	8,0406	6,896	21
9-7-8	10.5.2	15	0,8788	0,9447	4,348	50	42,8849	7,744	10,8352	9,293	21
10-2-3	10.5.2	15	1,3377	0,9181	4,969	50	40,9371	6,024	8,8289	7,229	21
1-6-5	10.5.2	15	1,5825	0,9046	3,106	50	42,1229	7,150	10,1847	8,580	21
3-11-10	12.5.4	15	1,38	0,916	3,7	50	42,073	5,0241354	7,165	6,0289624	21
5-12-1	12.5.4	15	1,38	0,916	4,3	50	41,444	9,7218163	14,075	11,66618	21
9-7-8	12.5.4	15	0,01	1,000	4,3	50	45,634	5,3976888	7,097	6,4772266	21
6-2-4	12.5.4	15	0,77	0,951	6,2	50	41,353	3,304111	4,794	3,9649332	21
7-15-4	20.5.1	15	2,31	0,867	1,9	50	41,476	6,6261941	9,586	7,951433	21
2-8-17	20.5.1	15	1,21	0,925	1,9	50	44,391	3,2692179	4,419	3,9230614	21
6-16-20	20.5.1	15	1,75	0,896	1,9	50	42,924	6,3797807	8,918	7,6557368	21
13-3-14	20.5.1	15	1,50	0,909	1,9	50	43,601	8,5076031	11,707	10,209124	21
1-9-10	20.5.1	15	0,63	0,960	1,9	50	46,113	1,070047	1,392	1,2840564	21
18-5-11	20.5.1	15	0,58	0,963	1,9	50	46,274	8,5722374	11,115	10,286685	21
12-19-0	20.5.1	15	2,23	0,870	1,9	50	41,654	4,896361	7,053	5,8756332	21

Fuente: elaboración propia.

Tabla 5. Resultados tiempos de vuelo del dron, al variar su velocidad con y sin la incidencia del viento, probado para 4 instancias de tamaños 6, 10, 12 y 20 clientes cuadrícula de 20x20

Cuadrícula 20x20											
Ruta	Instancia	$m_t + m_b$	m_l	C_m	$\bar{v}_v$	V_{fd}	V_{dc}	d	t_v	t	T_{max}
8-11-3	12.20.1	15	2,1173	0,8763	1,863	50	41,9519	21,209	30,3335	25,451	21
6-12-2	12.20.1	15	0,8808	0,9445	1,863	50	45,3635	34,329	45,4050	41,195	21
7-9-1	12.20.1	15	1,0471	0,9347	1,863	50	44,8741	29,494	39,4353	35,392	21
5-4-0	12.20.1	15	2,0596	0,8793	1,863	50	42,1002	15,715	22,3972	18,859	21
10-6-14	20.20.4	15	0,43	0,972	1,9	50	46,739	22,299556	28,627	26,759467	21
2-1-12	20.20.4	15	1,78	0,894	1,9	50	42,829	12,524339	17,545	15,029207	21
16-19-9	20.20.4	15	0,93	0,941	1,9	50	45,207	12,667636	16,813	15,201163	21
3-15-7	20.20.4	15	1,83	0,891	1,9	50	42,700	9,0906719	12,774	10,908806	21
8-5-17	20.20.4	15	2,15	0,875	1,9	50	41,865	20,27599	29,059	24,331189	21
20-11-4	20.20.4	15	0,29	0,981	1,9	50	47,195	12,466589	15,849	14,959907	21
18-13-0	20.20.4	15	0,68	0,957	1,9	50	45,966	1,4595915	1,905	1,7515098	21

Fuente: elaboración propia.

De modo que es difícil llegar a mejores soluciones en pocas iteraciones, originando que durante un número significativo de corridas no se encontraran soluciones que fueran aceptadas por tener un costo mayor al generado inicialmente, ocasionando que la temperatura inicial no disminuyera, lo cual generaba que el algoritmo no avanzara pues no se cumplía el criterio de parada de la temperatura final; por esto las instancias pequeñas del problema presentaron serias dificultades para llegar a la temperatura final del recocido simulado y por ende a mejores soluciones.

Se encontró que el CWA extendido encuentra peores soluciones iniciales que su versión original, el algoritmo ALNS ofrece menos soluciones aptas en CWA (independientemente del tamaño de la instancia) esto se

debe, en gran media, a que las soluciones iniciales que presenta son como máximo igual de buenas a las del ALNS. Por consiguiente, en pequeñas instancias el máximo porcentaje se acerca al 19,01%, mientras que para instancias grandes se aproxima al 11,4%.

Es posible afirmar que la combinación de las heurísticas de destrucción y reparación en la construcción del algoritmo ALNS favorece a que se explore sin límites el espacio de búsqueda, al contar con un componente de aleatoriedad y la adaptabilidad de estas.

5. Recomendaciones

Abordar en las futuras investigaciones el comportamiento del ALNS en problemas de ruteo de

vehículos con drones que impliquen ventanas de tiempo, combinando tanto el servicio orientado al tiempo como el enfoque orientado a la distancia. Utilizando así dentro del algoritmo, heurísticas de remoción, que permitan combinar estos dos enfoques, de tal manera que las heurísticas de reparación logren generar rutas que mantengan un equilibrio entre la cercanía de los clientes y sus periodos de atención.

Investigar sobre posibles algoritmos de búsqueda local simple y operadores de mutación que comparen todas las combinaciones válidas posibles de los mecanismos de intercambio en las rutas, generando así un aporte de mayor rendimiento a las rutas que se le ofrecen al algoritmo ALNS.

Para futuras investigaciones se podría explorar las otras configuraciones que existen entre camiones y drones, planteando además problemas y escenarios más cercanos a la realidad con la consideración de entregas múltiples, entregas parciales y problemas con una flota mixta de drones, identificando si existen cambios al evaluar la función objetivo; además hacer una experimentación de los parámetros del problema, identificando las posibles incidencias de estos en encontrar soluciones factibles y comprobar cómo influyen en el costo.

Realizar un diseño de experimentos para evaluar el efecto que tiene la variación de algunos parámetros del algoritmo como la tasa de destrucción, el factor de reacción y los puntajes asignados cuando una solución es aceptada.

Referencias

[1] Brown Eric, “E-commerce spurs innovation in last-mile logistics”, MIT News, 2019, Consultado: jul. 07, 2022. [En línea]. Disponible en: <https://news.mit.edu/2018/mit-e-commerce-spurs-innovations-last-mile-logistics-0904>

[2] “Ecommerce: Evolución y perspectivas 2020 | COMUNICAWEB”. [En línea]. Disponible en: <https://comunica-web.com/blog/marketing-digital/ecommerce-datos-evolucion/>

[3] B. Balcik, B. M. Beamon, K. Smilowitz, “Last mile distribution in humanitarian relief”, *J. Intell. Transp. Syst. Technol. Planning, Oper.*, vol. 12, núm. 2, pp. 51–63, abr. 2008, doi: <https://doi.org/10.1080/15472450802023329>

[4] B. Minten, B. Koru, D. Stifel, “The last mile(s) in modern input distribution: Pricing, profitability, and adoption”, *Agric. Econ.*, vol. 44, núm. 6, pp. 629–646, nov. 2013, doi: <https://doi.org/10.1111/AGEC.12078>

[5] “El nuevo mapa de rescates con drones de DJI - ITSitio”. <https://www.itsitio.com/co/nuevo-mapa-rescates-drones-dji/>

[6] C. Rose, “Jeff Bezos de Amazon mira hacia el futuro - CBS News”, 2013. <https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future>

[7] Bonn, “DHL Press Release English”, 2014. <https://www.reutersevents.com/supplychain/3pl/dhl-parcelcopter-launches-initial-operations-research-purposes>

[8] “Google tests drone deliveries in Project Wing trials - BBC News”. <https://www.bbc.com/news/technology-28964260>

[9] C. C. Murray, A. G. Chu, “The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery”, *Transp. Res. Part C Emerg. Technol.*, vol. 54, pp. 86–109, 2015, doi: <https://doi.org/10.1016/J.TRC.2015.03.005>

[10] I. or its affiliates Amazon.com, “Amazon.com: Prime Air”, 2016. <https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries>

[11] G. B. Dantzig, J. H. Ramser, “The Truck Dispatching Problem”, *Source Manag. Sci.*, vol. 6, núm. 1, pp. 80–91, 1959, doi: <https://doi.org/10.1287/mnsc.6.1.80>

[12] S. Erdoğan, E. Miller-Hooks, “A Green Vehicle Routing Problem”, *Transp. Res. Part E Logist. Transp. Rev.*, vol. 48, núm. 1, pp. 100–114, 2012, doi: <https://doi.org/10.1016/J.TRE.2011.08.001>

[13] B. Golden, A. Assad, L. Levy, F. Gheysens, “The fleet size and mix vehicle routing problem”, *Comput. Oper. Res.*, vol. 11, núm. 1, pp. 49–66, 1984, doi: [https://doi.org/10.1016/0305-0548\(84\)90007-8](https://doi.org/10.1016/0305-0548(84)90007-8)

[14] S. Erdoğan, E. Miller-Hooks, “A Green Vehicle Routing Problem”, *Transp. Res. Part E Logist. Transp. Rev.*, vol. 48(1), pp. 100–114, 2012, doi: <https://doi.org/10.1016/J.TRE.2011.08.001>

- [15] J. Lin, W. Zhou, O. Wolfson, "Electric Vehicle Routing Problem", *Transp. Res. Procedia*, vol. 12, pp. 508–521, ene. 2016, doi: <https://doi.org/10.1016/J.TRPRO.2016.02.007>
- [16] B. N. Coelho et al., "A multi-objective green UAV routing problem", *Comput. Oper. Res.*, vol. 88, pp. 306–315, dic. 2017, doi: <https://doi.org/10.1016/J.COR.2017.04.011>
- [17] N. Boysen, D. Briskorn, S. Fedtke, S. Schwerdfeger, "Drone delivery from trucks: Drone scheduling for given truck routes", *Networks*, 2018, doi: <https://doi.org/10.1002/net.21847>
- [18] D. Sacramento, D. Pisinger, S. Ropke, "An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones", *Transp. Res. Part C Emerg. Technol.*, vol. 102, pp. 289–315, 2019, doi: <https://doi.org/10.1016/J.TRC.2019.02.018>
- [19] "Existencias: Vehicle Routing Problem with Drones Instances". <https://zenodo.ups.edu.ec/Record/zen-1568853>
- [20] "Uso y cuidado de baterías para drones". <https://elvuelodeldrone.com/blog-de-drones/uso-cuidado-baterias-para-drones/>
- [21] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems", *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1520, pp. 417–431, 1998, doi: https://doi.org/10.1007/3-540-49481-2_30
- [22] W. Li, Y. Wu, P. N. R. Kumar, y K. Li, "Engineering Optimization Multi-trip vehicle routing problem with order release time Multi-trip vehicle routing problem with order release time", *Optim. Ing.*, 2019, doi: <https://doi.org/10.1080/0305215X.2019.1642880>
- [23] H. Paessens, "The savings algorithm for the vehicle routing problem", *Eur. J. Oper. Res.*, vol. 34, núm. 3, pp. 336–344, 1988, doi: [https://doi.org/10.1016/0377-2217\(88\)90154-3](https://doi.org/10.1016/0377-2217(88)90154-3)
- [24] T. Pichpibul, R. Kawtummachai, "A heuristic approach based on Clarke-Wright algorithm for open vehicle routing problem", *Sci. World J.*, vol. 2013, 2013, doi: <https://doi.org/10.1155/2013/874349>
- [25] T. J. Gaskell, "Bases for Vehicle Fleet Scheduling", *J. Oper. Res. Soc.*, vol. 18, núm. 3, pp. 281–295, sep. 1967, doi: <https://doi.org/10.1057/jors.1967.44>
- [26] P. C. Yellow, "A Computational Modification to the Savings Method of Vehicle Scheduling", *J. Oper. Res. Soc.*, vol. 21, núm. 2, pp. 281–283, 1970, doi: <https://doi.org/10.1057/JORS.1970.52>
- [27] D. Pisinger, S. Ropke, "A general heuristic for vehicle routing problems", *Comput. Oper. Res.*, vol. 34, núm. 8, pp. 2403–2435, 2007, doi: <https://doi.org/10.1016/J.COR.2005.09.012>
- [28] S. Ropke, D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows", *Transp. Sci.*, vol. 40, núm. 4, pp. 455–472, 2006, doi: <https://doi.org/10.1287/trsc.1050.0135>
- [29] R. Lutz, "Adaptive Large Neighborhood Search", 2014, [En línea]. Disponible en: <http://dx.doi.org/10.18725/OPARU-3237>
- [30] S. Ropke, "Heuristic and exact algorithms for vehicle routing problems", 2006. [En línea]. Disponible en: https://backend.orbit.dtu.dk/ws/portalfiles/portal/3155278/Heuristic+and+exact+algorithms+for+vehicle+routing+problems_Ropke.pdf
- [31] S. Ropke, D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows", *Transp. Sci.*, vol. 40, núm. 4, pp. 455–472, 2006, doi: <https://doi.org/10.1287/TRSC.1050.0135>
- [32] A. T. Phan, T. D. Nguyen, Q. D. Pham, "Traveling salesman problem with multiple drones", *ACM Int. Conf. Proceeding Ser.*, pp. 46–53, 2018, doi: <https://doi.org/10.1145/3287921.3287932>
- [33] M. Keskin, B. Catay, "Partial recharge strategies for the electric vehicle routing problem with time windows", *Transp. Res. Part C Emerg. Technol.*, vol. 65, pp. 111–127, abr. 2016, doi: <https://doi.org/10.1016/j.trc.2016.01.013>
- [34] D. Pisinger, S. Ropke, "A general heuristic for vehicle routing problems", *Comput. Oper. Res.*, vol. 34, núm. 8, pp. 2403–2435, 2007, doi: <https://doi.org/10.1016/j.cor.2005.09.012>

[35]S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, “Optimization by simulated annealing”, *Science*, vol. 220, núm. 4598, pp. 671–680, may 1983, doi: <https://doi.org/10.1126/science.220.4598.671>

[36]V. Černý, “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm”, *J. Optim. Theory Appl.*, vol. 45, núm. 1, pp. 41–51, 1985, doi: <https://doi.org/10.1007/BF00940812>

[37]D. Sacramento, “Vehicle Routing Problem with Drones Instances”, *Transportation Research Part C: Emerging Technologies*, ago. 2018, doi: <https://doi.org/10.5281/zenodo.2572764>