© Revista de Matemática: Teoría y Aplicaciones 2020 **27**(2) : 305–332
cimpa – ucr issn: 1409-2433 (Print), 2215-3373 (Online)
doi: https://doi.org/10.15517/rmta.v27i2.37889

MATHEURÍSTICAS PARA RESOLVER EL PROBLEMA DE RUTEO DE VEHÍCULOS CON VENTANAS DE TIEMPO

MATHEURISTICS FOR SOLVING THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Edwin Montes-Orozco* Román A. Mora-Gutiérrez†
Bibiana Obregón-Quintana‡
Sergio G. de-los-Cobos-Silva§ Eric A. Rincón-García¶
Miguel A. Gutiérrez-Andrade|| Pedro Lara-Velázquez**

Received: 2/Jul/2019; Revised: 12/Nov/2019; Accepted: 6/Feb/2020

---

*Universidad Autónoma Metropolitana Iztapalapa, Posgrado en Ciencias y Tecnologías de la Información, Ciudad de México, México. E-Mail: emontes@xanum.uam.mx
†Universidad Autónoma Metropolitana Azcapotzalco, Departamento de Sistemas, Ciudad de México, México. E-Mail: mgra@azc.uam.mx
‡Universidad Nacional Autónoma de México, Departamento de Matemáticas, Facultad de Ciencias, Ciudad de México, México. E-Mail: bobregon@ciencias.unam.mx
§Universidad Autónoma Metropolitana Iztapalapa, Departamento de Ingeniería Eléctrica, Ciudad de México, México. E-Mail: cobos@xanum.uam.mx
¶Misma dirección que/*Same address as*: S. de los Cobos. E-Mail: rincon@xanum.uam.mx
||Misma dirección que/*Same address as*: S. de los Cobos. E-Mail: gamma@xanum.uam.mx
**Misma dirección que/*Same address as*: S. de los Cobos. E-Mail: plara@xanum.uam.mx

Abstract

En este trabajo, se presentan dos técnicas matheurísticas basadas en dos técnicas heurísticas: Sistema de hormigas (AS), método de composición musical (MMC) y dos métodos exactos: Algoritmo primal-dual (PDA) y algoritmo simplex dual (DSA). Estas técnicas se denotan como DS-AS-PDA y DS-MMC-AS y se caracterizan por aprovechar la información de la estructura y características del modelo matemático para el problema de ruteo de vehículos con ventanas de tiempo (VRP-TW). Con el objetivo de caracterizar el comportamiento de las técnicas propuestas en este trabajo, se utilizaron 29 instancias de prueba para el VRP-TW. Los resultados numéricos, muestran que DS-AS-PDA y DS-MMC-AS presentan un comportamiento robusto y son capaces de generar las mejores soluciones reportadas en la literatura con un número menor de llamadas a la función objetivo para diversos tamaños de instancias.

Palabras clave: heurísticas; optimización; algoritmos híbridos; logística.

Abstract

In this work, we present two matheuristic techniques based on two heuristic techniques: Ant system (AS), method of musical composition (MMC) and two exact methods: Primal-dual algorithm (PDA) and dual simplex algorithm (DSA). These techniques are denoted as DS-AS-PDA and DS-MMC-AS and are characterized by taking advantage of the information of the structure and characteristics of the mathematical model for the vehicle routing problem with time windows (VRP-TW). In order to characterize the behavior of the techniques proposed in this work, we use 29 test instances for the VRP-TW. The numerical results show that DS-AS-PDA and DS-MMC-AS exhibit robust behavior and are capable of generating the best solutions reported in the literature with a smaller number of calls to the objective function.

Keywords: heuristics; optimization; hybrid algorithms; logistic.

Mathematics Subject Classification: 90B20, 90C27.

1 Introducción

Las técnicas matheurísticas, se definen como técnicas de optimización que se basan en la interoperación de las metaheurísticas y métodos de programación matemática [3]. En otras palabras, una matheurística es un algoritmo híbrido que combina y entrelaza de manera sinérgica ideas, conceptos y procesos tomados de métodos exactos con procedimientos heurísticos. Una característica esencial de las matheurísticas, es la explotación tanto del aspecto como de la información derivada del modelo matemático del problema a resolver [5].

Por otro lado, los problemas de logística de transporte pueden ser tratados como problemas de interés económico, porque se busca generar soluciones que minimicen diversos aspectos tales como la longitud de las rutas, el número de vehículos, los tiempos generados para cada ruta, entre otros. Además, en la logística de transporte, los problemas se vuelven de gran interés para la ciencia que involucran una serie de disciplinas como la investigación de operaciones, inteligencia artificial, optimización, entre otras [11].

Cabe mencionar que, estos problemas tienen una gran cantidad de restricciones que involucran espacio y tiempo, y a lo largo de los años se plantearon problemas como el problema del agente viajero (Traveling Sallesman Problem, o TSP) y el problema de ruteo de vehículos (VRP), los cuales han sido ampliamente estudiados debido a su complejidad [22, 25, 27], surgiendo diversas variaciones agregando restricciones, que han dado lugar a nuevos problemas.

El VRP, pertenece a la clase de problemas $\mathcal{NP}$-hard [16] y se describe como un depósito central que cuenta con una flota de vehículos que debe satisfacer la demanda de cierto producto a un conjunto de clientes geográficamente distribuidos. El objetivo del VRP, es minimizar el costo de entrega de los bienes solicitados por el conjunto de clientes, creando rutas que se originan y terminan en el depósito.

Una forma de ampliar el VRP tradicional, es agregar la restricción de asociar una ventana de tiempo para cada cliente, definiendo así un intervalo de tiempo en el cual cada uno puede ser atendido. Esta ampliación resulta en el problema de ruteo de vehículos con ventanas de tiempo (VRP-TW) [7]. Es importante mencionar que, los problemas del tipo VRP cuentan con una gran cantidad de aplicaciones.

La Tabla 1 muestra las principales aplicaciones para el VRP-TW. Con base en la información mostrada en la Tabla 1 se puede ver que, existe una amplia variedad de campos de aplicación para este tipo de problemas, por lo que es de gran importancia encontrar buenas soluciones en un tiempo corto.

Por tanto, en este trabajo se presentan dos técnicas matheurísticas para resolver el VRP-TW, haciendo uso de la información del problema para realizar la menor cantidad de evaluaciones a la función objetivo, debido a que el VRP-TW tiene como dificultad particular el cumplimiento de las restricciones temporales (tiempo en el que se sirve a cada cliente), las cuales provocan que las soluciones factibles sean pocas y diversas entre sí y que la dificultad de encontrar las soluciones óptimas en corto tiempo para una instancia en particular sea aún mayor que en el VRP tradicional.

Área económica	Aplicación
Transporte de materiales	Combustible, gas natural, hormigón.
Salud	Reparto de medicamentos a farmacias.
Banca	Reparto y recolección de dinero en efectivo.
Sector público	Recolección de basura,reparto de correo.
Servicios	Reparación de electrodomésticos a domicilio.
Industria	Suministro de piezas o mercancías entre almacenes.
Educación	Rutas de autobuses escolares.
Defensa	Rutas de aviones espías, logística militar.
Transporte	Planificación de flotas vehículares.

Tabla 1: Aplicaciones del VRP y sus variantes.

El presente trabajo se divide de la siguiente forma: en la Sección 2, se describe el modelo matemático para el VRP-TW. En la Sección 3, se presentan las características y el funcionamiento de las técnicas utilizadas para la realización de las matheurísticas. La Sección 4, muestra las características y funcionamiento de las técnicas matheurísticas presentadas en este trabajo. En la Sección 5, se detalla la metodología para la realización de los experimentos, la Sección 6 muestra la comparativa de los resultados obtenidos en este trabajo con los presentados en la literatura específica y por último la Sección 7, presenta las conclusiones y trabajo futuro.

2 Modelo matemático

En esta sección, se describe el modelo de programación matemática para el VRP-TW, para lo cual se requiere de la siguiente notación [7]:

- $G = (Cl, A)$ se define como una gráfica completa no dirigida, con $Cl = \{cl_0, cl_1, cl_2, \ldots, cl_n\}$ como el conjunto de clientes (donde cl_0 es el depósito y n es el número total de clientes) y $A = \{(i, j) : i, j \epsilon V, i \neq j\}$ como el conjunto de caminos o aristas entre cualquier par de clientes.
- $C = (C_{ij})$ es el costo del viaje entre los clientes i y j.
- $d_i \,\epsilon\, \mathbb{Z}^+$ se define como la demanda del cliente i.
- $Q \,\epsilon\, \mathbb{Z}^+$ como la capacidad de cada vehículo $v = 1, 2, \ldots, k$, donde k es el número máximo de vehículos disponibles en la flota.

Además, es necesario definir las siguientes variables de decisión:

$$X_{ij}^{v} = \begin{cases} 1, & \text{si el vehículo } v \text{ viaja del cliente } i \text{ al cliente } j; \\ 0, & \text{en otro caso.} \end{cases}.$$

Las variables auxiliares Y_{iv} como la cantidad entregada por el vehículo v al cliente i y b_i^v como la hora de llegada del vehículo v al sitio del cliente i. En esta variante del problema, además de las capacidades, cada cliente $cl_i \in Cl\{cl_0\}$ tiene asociado una ventana de tiempo $[e_i, l_i]$, la cual establece el tiempo en el que un vehículo debe llegar a cada cliente para abastecerlo, así como un tiempo de transporte desde él hacia cualquier otro cliente (denotado como t_{ij}) y un tiempo de servicio (tiempo en el que se abastece al cliente) s_i.

Finalmente, el modelo matemático para el VRP-TW se puede ver de la siguiente forma:

$$Minimizar \sum_{i=0}^{n}, \sum_{j=0}^{n} \sum_{v=1}^{k} C_{ij} X_{ij}^{v}, \tag{1}$$

sujeto a:

$$\sum_{i=0}^{n} \sum_{v=1}^{k} X_{ij}^{v} = 1;\ \forall\, j = 1, \ldots, n; \tag{2}$$

$$\sum_{j=1}^{n} X_{0j}^{v} = 1;\ \forall\, v = 1, \ldots, k; \tag{3}$$

$$\sum_{i=1}^{n} X_{i0}^{v} = 1;\ \forall\, v = 1, \ldots, k; \tag{4}$$

$$\sum_{i=0}^{n} X_{ip}^{v} - \sum_{j=0}^{n} X_{pj}^{v} = 0;\ \forall\, p = 0, \ldots, n;\ v = 1, \ldots, k; \tag{5}$$

$$Y_{iv} = d_i \sum_{j=0}^{n} X_{ji}^{v};\ \forall\, i = 1, \ldots, n;\ v = 1, \ldots, k; \tag{6}$$

$$\sum_{i=0}^{n} Y_{iv} \leq Q;\ \forall\, v = 1, \ldots, k; \tag{7}$$

$$e_i \leq b_i^v \leq l_i\ \forall\, i, j = 0, \ldots, n;\ v = 1, \ldots, k. \tag{8}$$

$$X_{ij}^{v}(b_i^v + s_i + t_{ij} - b_j^v) \leq 0\ \forall\, i, j = 1, 2, \ldots, n, v = 1, 2, \ldots, k. \tag{9}$$

$$X_{ij}^{v}\ \epsilon\ \{0, 1\} \text{ y } Y_{iv} \geq 0;\ \forall\, i = 1, \ldots, n;\ v = 1, \ldots, k. \tag{10}$$

El conjunto de restricciones (2), garantiza que solo un vehículo debe visitar a cada cliente. Las restricciones (3), (4) y (5), ayudan a evitar la formación de sub-ciclos, ya que indican que los vehículos que salen del depósito deben volver a este. Las restricciones (6), implican que el vehículo v solo puede atender al cliente i si este se encuentra en su ruta. El conjunto de restricciones (7), implica que la cantidad entregada en cada ruta no excede la capacidad del vehículo. Las restricciones (8), aseguran que se cumplan los límites de las ventanas de tiempo para cada cliente. Las restricciones (9) aseguran que el vehículo v no pueda iniciar el servicio si la suma del tiempo de transporte entre los clientes i y j, la duración del servicio para el cliente i y el tiempo de llegada al cliente i no es mayor que la ventana de tiempo para el cliente j. Finalmente, la restricción (10) define el tipo de variables utilizadas.

3 Descripción de las técnicas utilizadas

En esta sección, se describen las técnicas base utilizadas en este trabajo. Aquí, se realizaron diversas modificaciones en las técnicas originales, con el objeto de aprovechar la estructura del problema y obtener mejores resultados. En particular, se eligieron las técnicas de Sistema de hormigas (AS) y Método de Composición Musical (MMC) debido a que son técnicas que trabajan como una población de agentes que se comunican e interactúan localmente entre ellos y con su medio ambiente (sociedad). En estas ténicas, gracias a que los agentes siguen reglas simples y a las interacciones locales entre estos, se logra la emergencia hacia un comportamiento global.

3.1 Sistema de hormigas

AS, es una metaheurística presentada por Dorigo *et al.* (para más información, ver [12]) y está inspirada en el medio de comunicación que las hormigas utilizan para encontrar los caminos más cortos desde su nido hacia las fuentes de alimento. En este trabajo, se utiliza una variante propia de la técnica, en la cual, la selección de probabilidad se adapta al introducir un elemento nuevo en la solución, ya que se considera la ventana de tiempo para cada cliente.

A continuación, se describe cada uno de los procesos de esta técnica.

3.1.1 Inicializacion de la feromona

Para la primera iteración de la técnica, el nivel de feromona se inicializa como se muestra en la ecuación (11):

$$\tau_{ij}^{a+1} = \frac{1}{l_j}, \text{ con } a = 0. \tag{11}$$

Donde, τ_{ij}^{1} es el nivel de feromona que tiene el camino entre el cliente *i* y el cliente *j* y l_j el último momento en que el cliente j puede ser atendido.

3.1.2 Actualización de las feromonas

Para las demás iteraciones, el nivel de la matriz de feromonas se actualiza utilizando la siguiente ecuación:

$$\tau_{ij}^{a+1} = \rho(\tau_{ij}^{a}) + \Delta, \tag{12}$$

donde: ρ es el coeficiente de evaporación de la feromona, τ_{ij}^{a} es el nivel de feromona que tiene el camino entre el cliente *i* y el cliente *j* para la iteración a, τ_{ij}^{a+1} es el nivel de la feromona que guiará la siguiente iteración de cada hormiga *h* y Δ es la cantidad de feromonas depositadas por la mejor hormiga de la iteración en los caminos que pertenecen a las rutas formadas en su solución. Entonces, Δ:

$$\Delta = \frac{sol[a] - sol[a-1]}{sol[a-1]}, \tag{13}$$

siendo $sol[a]$ el valor de la solución entregada por la mejor hormiga de la iteración *a*; y $sol[a-1]$ el valor de la mejor solución en la iteración *a-1*.

3.1.3 Probabilidad de selección

Cuando un cliente se agrega a la solución o se completa una iteración, la probabilidad de seleccionar el camino entre los clientes i y j, se actualiza mediante la siguiente ecuación [12]:

$$P_{ij}^{a+1} = \frac{(\tau_{ij}^{a+1})^{\alpha}(\eta_{ij})^{\beta}(\nu_{ij})^{\gamma}}{\sum(\tau_{ij}^{a+1})^{\alpha}(\eta_{ij})^{\beta}(\nu_{ij})^{\gamma}}. \tag{14}$$

Donde: τ_{ij} es el nivel de la feromona, $\eta_{ij} = 1/d_{ij}$ regula la conveniencia del estado de transición, $\nu_{ij} = 1/l_i$ indica que cuanto menor sea el valor del último momento en el que se puede visitar, la probabilidad de visitarlo al principio será mayor. Adicionalmente, α se utiliza para controlar la influencia del nivel de feromonas, β es un parámetro para controlar la influencia de la cercanía entre clientes y, γ controla la influencia de los valores l_i.

3.1.4 Construcción de la solución

Cada hormiga del sistema construye una solución partiendo del depósito, agregando a la ruta un cliente a la vez con base en la probabilidad de elección de cada uno (ecuación (14)) partiendo del cliente que fue agregado anteriormente a esta (Pseudocódigo 1) [18].

Pseudocódigo 1 Construcción de soluciones.

mientras no se satisfaga la capacidad del vehículo o no se pueda agregar un elemento más **hacer**
    Inicio
    Generar un número "Aleatorio" continuo entre 0 y 1
    si "Aleatorio" $\leq Pn_{i,j}$ y "Aleatorio" $> Pn_{i,j-1}$ **entonces**
        Agregar el cliente *j* a la ruta del vehículo *k*.
        Modificar capacidad del vehículo *k* y calcular tiempo de finalización de servicio para *j*.
        Actualizar la probabilidad de elección
    fin si
fin mientras
Agregar el nodo depósito al final de la ruta *k* (es necesario agregarlo la cantidad restante de elementos disponibles en el vector)

Finalmente, se tendrá una solución que se representa mediante el vector mostrado en la Figura 1.

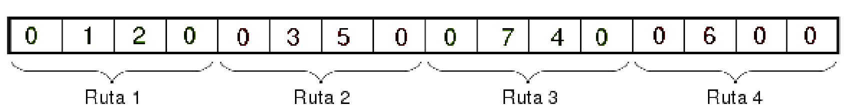

Figura 1: Vector solución.

De esta forma, cada ruta tiene asignado en el vector solución, un número *e* de espacios, el cual está dado por el mayor número de clientes en una ruta, más los espacios que representan el nodo depósito en salida y llegada.

3.1.5 Procedimiento general de la técnica

Con el objetivo de entender de mejor manera como trabaja la técnica AS, el Pseudocódigo 2, muestra el procedimiento general de esta.

Pseudocódigo 2 Sistema de hormigas.

Inicializar la matriz de feromonas usando la ecuación (11)
Actualizar la probabilidad de elección para la primera iteración de n hormigas
para *a=1* **a** *a=m* **hacer**
    para *l=1* **a** *l=h* **hacer**
        Construir la solución de la hormiga l
        Evaluar la función objetivo
    fin para
    Busca la mejor solución entre las h soluciones
    Guardar en memoria la mejor solución de la iteración a
    Actualizar la matriz de feromonas usando la mejor solución del ciclo a a través de la ecuación (12)
    Actualizar la probabilidad de elección utilizando la ecuación (14)
    Determinar la mejor solución entre las mejores de cada iteración
fin para
devolver La mejor solución encontrada

3.2 Método de composición musical

El método de composición musical *(MMC)*, fue propuesto por Mora-Gutiérrez *et al.* [19]; y es una técnica que se inspira en la analogía del proceso creativo de la composición musical en un entorno sociocultural (para más información, consulte [20]) y se ha utilizado en diferentes problemas de optimización, obteniendo buenos resultados.

En esta técnica, se crea una sociedad de compositores que define una red de interacción entre ellos. Inicialmente, se genera un conjunto aleatorio de soluciones para cada compositor y posteriormente, los compositores interactúan entre sí, analizando la información recibida.

En este trabajo, una melodía es representada como una solución del VRP-TW que se trabaja como una matriz de k rutas. Además, una sociedad se genera con $Nsol_c$ compositores, donde cada compositor posee $Nsol_s$ melodías; por tanto, el algoritmo crea $Nsol_c \times Nsol_s$ soluciones en cada iteración; es decir, se compondrán $Nsol_c$ nuevas melodías, una para cada compositor, y este decidirá si lo incluye o no en su repertorio.

El pseudocódigo 3 muestra el procedimiento general de MMC.

Pseudocódigo 3 Método de composición musical,

```
Crear una sociedad artificial con las reglas de interacción entre agentes (compositores)
para cada individuo de la sociedad hacer
    Inicializar un conjunto de soluciones para cada compositor
fin para
mientras el criterio de parada no se cumple hacer
    Intercambiar información entre agentes
    para cada individuo de la sociedad hacer
        Actualizar la matriz de conocimiento.
        Generar y evaluar una nueva improvisación sol_n
        si sol_n es mejor que la peor solución del compositor (sol_worst) entonces
            Reemplazar sol_worst por sol_n
        fin si
    fin para
    Construir el conjunto de soluciones
fin mientras
```

3.3 Relajaciones al modelo original

Los métodos exactos, se utilizan para resolver dos relajaciones del modelo matemático original presentado en la Sección 2. En la primera relajación, se eliminan las restricciones impuestas por las ventanas de tiempo (ecuaciones (8) y (9)), mientras que para la segunda, se eliminan las restricciones impuestas por las capacidades de los vehículos (ecuaciones (7)). Además, para poder resolverlo como un problema lineal, se eliminan las restricciones de integralidad (ecuación (10)).

3.4 Algoritmo primal-dual

En general, esta técnica se aplica a programas lineales y problemas de optimización que pueden ser resueltos en tiempo polinomial. Por tanto, considerando el programa lineal:

$$\min\ c^T x, \tag{15}$$

$$\text{sujeto a} \quad \begin{aligned} Ax &\geq b,\\ x &\geq 0, \end{aligned}$$

y su dual:

$$\max\ b^T y, \tag{16}$$

$$\text{sujeto a} \quad \begin{aligned} A^T y &\leq c,\\ y &\geq 0, \end{aligned}$$

donde $A \in Q^{m\times n}, c, x \in Q^n, b, y \in Q^m$, y T denota el vector transpuesto. Para facilitar la representación se supone que $c \geq 0$.

En el algoritmo primal-dual (PDA), se supone que el primal tiene una solución factible *y* al dual; inicialmente se puede establecer $y = 0$ [21]. En el método primal-dual, se puede encontrar una solución *x* al primal que cumpla las condiciones de holguras complementarias con respecto a y, demostrando así que tanto x como y son óptimas, o bien se puede encontrar una nueva solución factible dual con un mayor valor de la función objetivo [13].

Se debe considerar lo que significa que x sea complementario a y; sea A_i la i-ésima fila de A y A^j la j-ésima columna de A; para los programas lineal 15 y dual 16, existen dos tipos de condiciones para holguras complementarias. El primero correspondiente a las variables primales, es decir

$$x_j > 0 \Rightarrow A^j y = c_j, \tag{17}$$

siendo $J = \{j | A^j y = c_j\}$.

En segundo lugar, correspondientes a las variables duales, es decir:

$$y_i > 0 \Rightarrow A_i x = b_i, \tag{18}$$

siendo $I = \{i | y_i = 0\}$.

El conjunto de condiciones (17), (18) son satisfechas si x es factible en (15). Ahora, se debe suponer que se tiene una y factible en (16) si se pudiera encontrar de cualquier manera una x que satisfaga

$$x_j = 0.$$

Entonces, x y y serían óptimas. Entonces el algoritmo primal-dual se deriva de la idea de buscar una x, dada una y. Se busca esa x resolviendo un problema auxiliar llamado primal restringido (RP), determinado por el primal con el que se está trabajando. Si la búsqueda de x no tiene éxito, nunca se obtendrá la información del dual de *RP* denominado *DRP*, que dice cómo mejorar la solución y con la que se inició; de esta manera, se converge a la optimalidad en un número finito de pasos [21].

El PDA, comienza con una solución y factible en el programa dual 16. En el caso que $c \geq 0$, se puede tomar $y = 0$ inmediatamente como un punto inicial factible en el dual. Cuando c es no negativo, se puede encontrar una solución y fácilmente introduciendo la variable X_{n+1} al problema primal (15) y la restricción

$$x_1 + x_2 + \ldots + x_n + x_{n+1} = b_{m+1}, \tag{19}$$

donde b_{m+1} es mayor que la suma de cualquier valor de la posible solución $x_1, \ldots, x_n$ a (15) y el costo $C_{n+1} = 0$. Claramente la ecuación (19) no cambia la solución a (15). El dual del nuevo primal se obtiene agregando una nueva variable y_{n+1} como se muestra en[21]:

$$\max\ w = y'b + y_{m+1}b_{m+1},$$
$$y'A_j + y_{m+1} = C_j; j = 1, \ldots, n,$$
$$y_{m+1} \leq 0.$$

Una solución factible a este *LP* es simple:

$$y_i = 0; i = 1, \ldots, m;$$
$$y_{m+1} = \min\{c_j\} < 0$$
$$1 \leq j \leq m.$$

La última desigualdad se deduce de la suposición de que c es negativo. En cada iteración debe mantenerse la factibilidad de la solución y.

Ahora, se tiene una y factible en el programa dual (16), y algunas de las restricciones de desigualdad

$$y'A_j \leq c_j.$$

Se cumplirán estrictamente y otras no. Se define el conjunto de índices J por

$$J = \{j_y^{'} : \; A_j = c_j\}. \tag{20}$$

Sabemos de (17) y (18) que una x factible en P es óptima exactamente cuando

$$x_j = 0 \textit{ para todo } j \notin J. \tag{21}$$

Esto equivale a buscar una x que satisfaga

$$\begin{aligned} \sum_{j \in J} a_{ij} x_j = b_i; \; i = 1, \ldots, m; \\ x_j \geq 0, \; j \in J, \\ \sum_{j \in J} a_{ij} x_j = b_i; \; i = 1, \ldots, m. \end{aligned} \tag{22}$$

Este conjunto de igualdades utiliza sólo las columnas de A correspondientes a las igualdades en D, que a su vez corresponden al conjunto J; Por esta razón se denomina J al conjunto de columnas admisibles. Para buscar tal x, se crea un nuevo LP, llamado el primal restringido, como se muestra a continuación [21]:

$$\begin{aligned} \min \; \varepsilon = \sum_{i=1}^{m} x_i^a, \\ \sum_{j \in J} a_{ij} x_j + x_i^a = b_i; \; i = 1, \ldots, m; \\ x_j \geq 0, \; j \in J, \\ x_j = 0, \; j \in J, \\ x_i^a \geq 0. \end{aligned} \tag{23}$$

Es decir, se introducen variables artificiales, una para cada ecuación en P (ecuación (15)). El primal restringido (23) puede ser resuelto usando el algoritmo simplex ordinario. Si la solución óptima a RP es $\varepsilon_{opt} = 0$, entonces se tiene una solución a (22) y por lo tanto una solución óptima a P. Lo que es de interés ahora, es lo que sucede cuando $\varepsilon_{opt} = 0$ en RP, por lo que se considera el dual de RP (DRP) [21]

$$\max \; w = y'b, \tag{24}$$

$$y'A_j \leq 0; j \in J, \tag{25}$$

$$y_i \leq 1; i = 1, \ldots, m; \tag{26}$$

$$y_i \neq 0. \tag{27}$$

Se denota la solución óptima de DRP, obtenida cuando RP se resuelve, por y'. Ahora se ha tratado de encontrar una x factible utilizando sólo columnas admisibles, pero si $\varepsilon_{opt} > 0$ en RP, hay que elegir otra solución y. En la Figura 2 se muestra la forma en que trabaja el algoritmo primal-dual utilizando los diversos programas lineales.

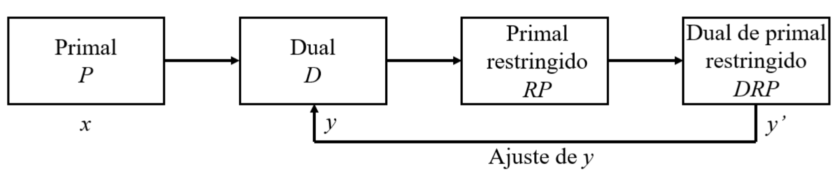

Figura 2: Procedimiento algoritmo primal-dual [21].

4 Métodos desarrollados

En esta sección, se muestra el desarrollo e integración de las técnicas metaheurísticas y métodos exactos para la formación de las técnicas matheurísticas propuestas en este trabajo.

4.1 Algoritmo híbrido dual-simplex-Método de composición musical-Sistema de hormigas (DS-MMC-AS)

Esta técnica, se basa en una técnica de programación matemática (Dual-Simplex *(DS)*, y dos heurísticas *(AS)* y *(MMC)*.

Para el desarrollo de esta técnica, se utilizó la herramienta *Gurobi* [14] para resolver las relajaciones del modelo matemático mostrado en la Sección 2 empleando el método dual-simplex. Partiendo de las soluciones regresadas por Gurobi, se toma esa información y se inicializa el nivel de feromona en AS y, una vez que finaliza cada iteración, se ejecuta MMC para encontrar una mejor solución con la población devuelta por AS. Con el objetivo de describir el procedimiento general de DS-MMC-AS, se muestra el Pseudocódigo 4.

4.2 Algoritmo híbrido dual simplex-Sistema de hormigas-Algoritmo primal dual (DS-AS-PDA)

Esta técnica, se basa en dos técnicas de programación matemática: DS [9] y Algoritmo Primal Dual (PDA) [10] y en la metaheurística AS.

Para el desarrollo de esta, se utilizó nuevamente la herramienta Gurobi para resolver dos relajaciones del modelo matemático descrito en la Sección 2 mediante el método dual-simplex. Una vez que Gurobi devuelve las dos soluciones, se toma esa información y se inicializa el nivel de *feromona* en *AS* y al terminar su ejecución, se ejecuta el algoritmo primal-dual (aplicado a ambas relajaciones) para verificar qué variables son candidatas para ingresar a la base y darles una mayor probabilidad de selección en *AS* para el siguiente ciclo. El Pseudocódigo 5 muestra, las características de esta técnica.

Pseudocódigo 4 Método dual-simplex-Composición musical-Sistema de hormigas (DS-MMC-AS).

Relajar el programa lineal original en términos de integralidad y restricciones de tiempo.
Relajar el programa lineal original en términos de integralidad y restricción de la capacidad vehicular.
Resolver los programas lineales relajados utilizando Gurobi
Con base en las dos soluciones entregadas por Gurobi, se inicializan dos matrices de feromonas (a_1 y a_2)
para *i=1* **a** *i=*n_c **hacer**
    Generar una matriz de feromonas para el compositor *i* basado en las matrices iniciales a_1 y a_2 de la siguiente forma:
    $A_i = a_1 + (a_2 - a_1) * R_1$, con $R_1 \in [0, 1]$
fin para
para *i=1* **a** *i=*n_c **hacer**
    para *j=1* **a** *j=*n_s **hacer**
        Construir la solución *j* para el compositor *i* usando *AS* (Pseudocódigo 2)
    fin para
    Actualizar la matriz de feromonas A_i con la mejor solución generada
    Guardar la matriz de feromonas de la peor solución generada (P_i).
fin para
para *l=1* **a** $l = \max\_iter$ **hacer**
    Intercambiar el conocimiento de los compositores de la siguiente manera.:
    para *i=1* **a** $i = n_s$ **hacer**
        para i'*=1* **a** $i' = n_s$ **hacer**
            si existe alguna relación entre i y i' **entonces**
                si $P_{i'}$ is best that P_i **entonces**
                    Actualizar A_i de la siguiente forma:
                    $A_i = (f(P_i) - f(P_{i'})) * A_{i'} + A_i$
                fin si
            fin si
        fin para
    fin para
    para *j=1* **to** *j=*n_s **hacer**
        Construir la solución *j* para el compositor *i* utilizando *AS* (Pseudocódigo 2)
        Actualizar la matriz de feromonas A_i con la mejor solución generada
        Guardar la matriz de feromonas de la peor solution generada. (P_i).
    fin para
fin para

Pseudocódigo 5 Dual simplex-Sistema de hormigas-Algoritmo primal-dual (DS-AS-PDA).

```
Crear los programas lineales relajados en capacidades y ventanas de tiempo para el modelo original del VRPTW
Resolver las dos relajaciones del modelo con Gurobi
Con las soluciones devueltas por Gurobi, inicializar el nivel de feromona
Actualizar la probabilidad de selección para el primer ciclo de n hormigas
Ejecutar AS
Después de un número a de ciclos de AS, ejecutar PDA
mientras no se cumpla el criterio de paro hacer
    Mediante PDA encontrar qué variables son candidatas para ingresar a la base y actualizar su probabilidad dentro de AS
    Ir a ejecutar AS
fin mientras
devolver la mejor solución encontrada
```

5 Metodología para la experimentación

En el presente trabajo, se utiliza un conjunto de instancias de VRP-TW para analizar y caracterizar el comportamiento de las 2 matheurísticas desarrolladas. Cada una de las técnicas, se ejecutó con 20 soluciones iniciales distintas sobre cada una de las instancias y, en cada una de las ejecuciones se registró el mejor valor de la función objetivo.

Cada técnica se programó en lenguaje C, y cada uno de los experimentos se realizó en una computadora portátil con procesador Intel Core i5-400U a 2.60 GHz, 1600 MHz y 3 MB de caché con 8 Gb de RAM.

5.1 Instancias de prueba

El conjunto de prueba se integró por 29 instancias del VRP-TW de diferentes tamaños, las cuales se tomaron del conjunto propuesto por Cordeau y Solomon [8, 27], debido a que este conjunto de prueba ha sido utilizado por varios autores (p.e.: [17, 24]).

A continuación, la Tabla 2 muestra el número máximo de vehículos (flota) y el número de clientes pertenecientes a cada instancia.

Tabla 2: Características de las instancias de prueba.

Instancia	No. de clientes	No. de vehículos	Instancia	No. de clientes	No. de vehículos
C101	100	10	C207	100	4
C102	100	10	C208	100	4
C103	100	10	R101	100	19
C104	100	10	R102	100	17
C105	100	10	R103	100	13
C106	100	10	R104	100	9
C107	100	10	R105	100	14
C108	100	10	R106	100	12
C109	100	10	R107	100	10
C201	100	4	R108	100	9
C202	100	4	R109	100	11
C203	100	4	R110	100	10
C204	100	4	R111	100	10
C205	100	4	R112	100	9
C206	100	4	–	–	

Además, los resultados obtenidos por las técnicas propuestas en este trabajo, fueron comparados por los resultados generados por los siguientes trabajos:

- Modified Artificial Bee Colony (ABC-M) [15].
- Variable neighborhood search with compound operators (VNS-C) [6].
- Three meta-heuristics designed for the VRP-TW (Heurísticas) [26].
- Tabu Search for Fleet Size and Mix Vehicle Routing Problem with Hard and Soft Time windows (Tabu) [2].
- A Hybrid Approach for the Capacitated Vehicle Routing Problem with Time Windows (Híbrido) [4].

Es importante tener en cuenta que, en la ejecución de DS-MMC-AS y DS-AS-PDA se emplearon 2,500 evaluaciones y 1,000 evaluaciones, respectivamente. Mientras que en ABC-M [1] se necesitaron alrededor de 20,000 evaluaciones de la función objetivo, en VNS-C [6] se reportaron más de 1,000,000 de evaluaciones, en el método desarrollado en [26] se realizaron alrededor de 10,000 evaluaciones, en el método [2] alrededor de 40,000 y en [4] alrededor de 3,000.

5.2 Ajuste de parámetros

El ajuste de parámetros es una actividad importante para lograr un comportamiento adecuado de las técnicas heurísticas y metaheurísticas. Por tanto, en este trabajo se realizó utilizando el algoritmo de Evolución Diferencial Autoadaptable *(SA-DE)* [23]. Esta técnica requiere dos parámetros de control (factor de cruza (Cr) y factor de mutación (F), ambos utilizados con valor = 0.5).

El algoritmo SA-DE, trabaja con poblaciones donde cada individuo es un vector que contiene los parámetros para DS-MMC-AS o DS-AS-PDA. Después de ejecutar SA-DE 30 veces, se obtuvieron los resultados para los parámetros que se muestran en la Tabla 3).

Tabla 3: Ajuste de parámetros.

Parámetro	DS-MMC-AS	DS-AS-PDA
Número de iteraciones	25	50-60
Tamaño de la población	10	14
α	1	1.14
Número de improvisaciones	10	–
β	1.5	1.49
γ	1.3	1.35
ρ	0.75	0.71
Llamadas a la función objetivo	2,500	1,000

6 Resultados

Con el objetivo de analizar el comportamiento de las técnicas propuestas en este trabajo, se utilizaron 29 instancias de 100 clientes y se compararon con los resultados descritos en los trabajos [4], [6], [15], [2] y [26] (Tablas 4 y 5).

Tabla 4: Resultados comparativos (1).

Inst.	DS-AS-PDA			DS-MMC-AS			**Heurísticas**
	Mejor	Prom.	T. (s)	Mejor	Prom.	T. (s)	Mejor
C101	828.94	828.94	54.7	828.94	828.94	56.5	828.94
C102	828.94	828.94	40.32	828.94	829.67	49.6	828.94
C103	828.06	829.49	52.63	828.06	828.86	52.9	828.06
C104	**824.78**	825.69	47.32	**824.78**	824.78	53.4	**824.78**
C105	828.94	831.01	48.23	828.94	828.94	54.7	828.94
C106	828.94	828.94	51.20	828.94	828.94	55.8	828.94
C107	828.94	828.94	57.31	828.94	829.30	63.9	828.94
C108	828.94	828.94	62.46	828.94	828.94	64.5	828.94
C109	828.94	828.94	69.82	828.94	829.24	68.3	828.94
C201	**589.1**	589.1	34.54	**589.1**	589.1	37.85	591.56
C202	**589.1**	589.1	32.64	**589.1**	589.1	36.48	591.56
C203	**588.71**	588.71	34.26	**588.71**	588.71	33.89	591.17
C204	**588.11**	588.11	28.72	**588.11**	588.11	29.63	590.60
C205	**586.43**	586.43	30.85	**586.43**	621.37	28.69	588.88
C206	**586**	586	27.38	**586**	586	29.42	588.49
C207	**585.81**	585.81	26.83	**585.81**	585.81	28.46	588.29
C208	**585.81**	585.81	25.94	**585.81**	585.81	27.53	588.32
R101	1637.7	1745.86	60.43	1637.7	1637.7	58.47	1643.79
R102	1466.6	1576.58	57.90	1466.6	1601.82	59.42	1474.28
R103	**1208.7**	1208.7	53.28	**1208.7**	1208.7	52.78	1215.59
R104	**971.5**	971.5	32.56	**971.5**	971.5	31.78	981.57
R105	**1355.33**	1476.3	56.90	**1355.33**	1476.3	57.48	1360.78
R106	**1234.6**	1347.6	58.47	**1234.6**	1347.6	56.34	1240.53
R107	**1064.6**	1064.6	60.42	**1064.6**	1064.6	56.73	1083.89
R108	964.38	964.38	58.43	964.38	964.38	55.23	946.74
R109	**1146.9**	1346.4	62.57	**1146.9**	1146.9	60.23	1154.73
R110	**1068**	1068	60.43	**1068**	1068	58.49	1105.98
R111	**1048.7**	1048.7	64.60	**1048.7**	1048.7	53.45	1054.23
R112	**957.04**	957.04	57.32	**957.04**	957.04	59.43	969.34

Tabla 5: Resultados comparativos (2).

Inst.	ABC-M			VNS-C		Tabu	Híbrido	
	Mejor	Prom.	T. (s)	Mejor	Prom.	Mejor	Mejor	Prom.
C101	828.94	828.94	700	828.94	828.94	828.94	828.94	831.02
C102	828.94	828.94	700	828.94	876.79	828.94	828.94	831.61
C103	828.94	840.66	700	828.94	832.65	832.07	828.06	844.84
C104	858.90	889.10	700	825.65	831.79	847.45	825.54	874.79
C105	828.94	828.94	700	828.94	852.33	828.94	828.94	834.68
C106	828.94	828.94	700	828.94	836.25	828.94	828.94	832.05
C107	828.94	828.94	700	828.94	853.9	828.94	828.94	840.29
C108	828.94	830.85	700	828.94	840.48	828.94	828.94	835.76
C109	828.94	836.97	700	828.94	828.94	828.94	828.94	842.94
C201	591.56	591.56	700	591.56	591.56	591.56	591.56	624.77
C202	591.56	601.78	700	591.56	613.94	591.56	591.56	654.16
C203	600.54	616.39	700	591.17	599.16	636.08	591.17	548.53
C204	610.01	648.57	700	590.6	609.81	648.07	592.13	659.12
C205	588.88	596.10	700	588.88	588.88	588.88	588.88	647.58
C206	588.88	601.49	700	588.49	588.49	588.49	588.49	648.99
C207	589.58	601.60	700	588.29	588.29	602.99	488.28	649.92
C208	591.65	613.47	700	588.32	588.32	588.49	588.32	663.74
R101	1643.18	1647.91	700	1643.34	1647.90	**1626.42**	1653.23	1679.79
R102	1480.73	1490.66	700	1476.06	1493.30	**1452.2**	1488.18	1519.51
R103	1240.87	1490.66	700	1219.89	1230.92	1232.13	1236.88	1263.68
R104	1047.06	1070.27	700	994.85	1009.9	979.63	1003.98	1028
R105	1369.52	1382.35	700	1360.78	1377.24	1362.88	1378.94	1404.88
R106	1271.13	1285.81	700	1234.72	1264.04	1246.08	1255.16	1294.08
R107	1129.99	1142.23	700	1077.24	1097.07	1075.48	1088.68	1116.52
R108	1004.11	1026.11	700	956.22	974.46	955.35	960.8	987.04
R109	1170.50	1211.12	700	1157.61	1181.99	1178.57	1158.29	1191.53
R110	1123.36	1145.30	700	1081.88	1106.02	1114.86	1095.43	1120.85
R111	1101.59	1129.55	700	1062.58	1080.1	1081.08	1061.61	1100.18
R112	1019.84	1026.25	700	958.7	979.52	991.46	965.87	1005.11

En las Tablas 4 y 5, se puede ver que las técnicas propuestas en este trabajo, obtienen en la mayoría de las instancias, los mejores valores reportados en distintos trabajos de la literatura específica (marcados en negrita). Además, es importante mencionar que presentan un comportamiento más robusto en comparación a las técnicas presentadas en [4], [6], [15], [2] y [26], ya que obtienen mejores valores promedio.

A pesar de que no se puede realizar una buena comparación de los tiempos de ejecución (debido a que todas las técnicas están implementadas en distintos lenguajes de programación y ejecutadas en computadoras con diferentes características), las técnicas propuestas en este trabajo, ejecutan menos evaluaciones a la función objetivo.

Por otro lado, para comparar de mejor manera el comportamiento de las técnicas presentadas en este trabajo, los resultados obtenidos en las Tablas 4 y 5, se normalizaron a través de la ecuación (28):

$$f(x^{norm-\alpha}) = \begin{cases} 1 & \text{si } f(x^{\text{worst in } \beta}) - f(x^\star) \neq 0 \\ f(x^{\text{worst in } \beta}) - f(x^{\text{method}-\alpha}) & \text{si } f(x^{\text{worst in } \beta}) - f(x^\star) = 0 \\ \frac{f(x^{\text{worst in } \beta}) - f(x^{\text{method}-\alpha}))}{f(x^{\text{worst in } \beta}) - f(x^\star)} & \text{si } f(x^{\text{worst in } \beta}) = f(x^\star), \end{cases} \tag{28}$$

donde, $f(x^\star)$ es el mejor valor de la función objetivo encontrada; $f(x^{method-\alpha})$ es el promedio del valor de la función objetivo encontrada por la metaheurística α; $f(x^{worsten\ \beta})$ es el peor valor encontrado por las metaheurísticas, para la instancia β y $f(x^{norm-\alpha})$ es el valor normalizado de la función objetivo encontrada por la metaheurística α. Entonces, el valor de $f(x^{norm-\alpha})$ varía de 0 a 1.

Si $f(x^{norm-\alpha})$ es cercano a 1, entonces el valor de $f(x^{Method-\alpha})$ está cercano al peor valor reportado. En contraste, si $f(x^{norm-\alpha})$ está cerca de 0, entonces el valor de $f(x^{method-\alpha})$ es cercano a $f(x^\star)$ (valor óptimo).

A continuación, las Figuras 3 y 4, muestran la comparación gráfica de los resultados normalizados de los métodos presentados en este trabajo y en la literatura específica para las 29 instancias de prueba.

En las Figuras 3 y 4, se muestran gráficamente los resultados obtenidos por cada una de las técnicas. Aquí, se puede constatar que DS-MMC-AS y DS-AS-PDA obtienen los mejores resultados, ya que su mediana y el rango intercuartil se encuentran muy cercanos al valor óptimo; mientras que para las técnicas mostradas en la literatura la mediana se encuentra a

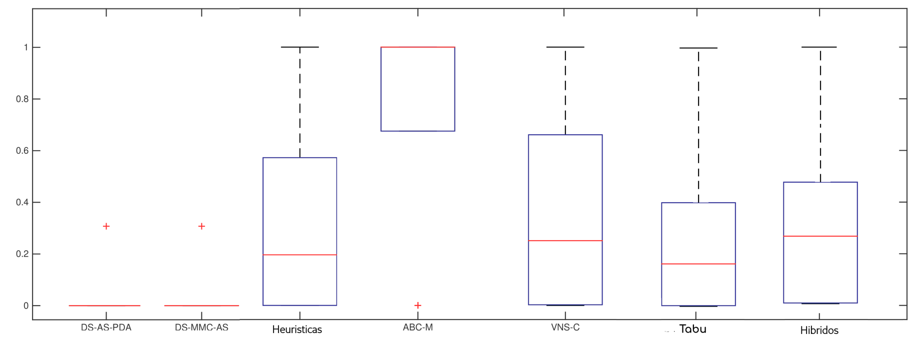

Figura 3: Diagrama de caja para los resultados normalizados (mejores).

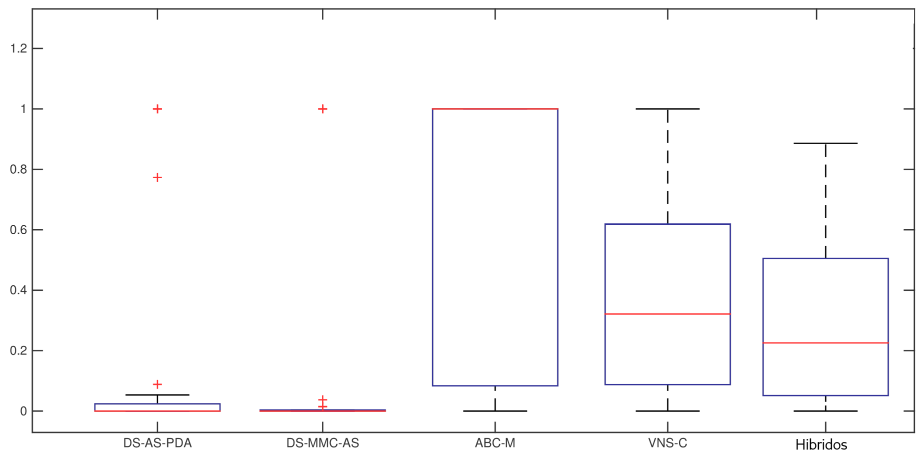

Figura 4: Diagrama de caja para los resultados normalizados (promedio).

20% de distancia y los rangos intercuartiles hasta un 60%. Por lo tanto, se puede constatar que DS-MMC-AS y DS-AS-PDA presentan un comportamiento más robusto en comparación con las demás técnicas.

Ahora, en la Tabla 6, se muestran los resultados obtenidos al aplicar la prueba de Wilcoxon a los mejores resultados obtenidos para cada técnica.

En función de los resultados mostrados en las Tablas 6 y 7 y en las Figuras 3 y 4, se puede decir que DS-MMC-AS y DS-AS-PDA son estadísticamente similares entre sí, y que son estadísticamente diferentes respecto a los demás métodos. Además, es importante mencionar que DS-MMC-AS y DS-AS-PDA generaron mejores resultados en 18 de 29 instancias.

Tabla 6: Prueba de Wilcoxon para los mejores valores (p-value).

Técnica	DS-MMC-AS	DS-AS-PDA	Heurísticas	ABC-M	VNS-C	Tabu	Híbrido
DS-MMC-AS	0	0	1	1	1	1	1
DS-AS-PDA	0	0	1	1	1	1	1
Heurísticas	1	1	0	1	0	1	1
ABC-M	1	1	1	0	1	1	1
VNS-C	1	1	0	1	1	1	1
Tabu	1	1	1	1	1	0	1
Híbrido	1	1	1	1	1	1	0

Tabla 7: Prueba de Wilcoxon para resultados promedio (h-value).

Técnica	DS-MMC-AS	DS-AS-PDA	Heurísticas	ABC-M	VNS-C	Tabu	Híbrido
DS-MMC-AS	0	0	-	1	1	-	1
DS-AS-PDA	0	0	-	1	1	-	1
Heurísticas	-	-	-	-	-	-	-
ABC-M	1	1	-	0	1	-	1
VNS-C	1	1	-	0	1	-	1
Tabu	-	-	-	-	-	-	-
Híbrido	1	1	-	1	1	-	1

7 Conclusiones y trabajo futuro

En este trabajo, se presentaron dos técnicas matheurísticas para resolver el Problema de Ruteo de Vehículos con Ventanas de Tiempo (VRP-TW), los cuales son identificados como: DS-MMC-AS y DS-AS -PDA. Estas técnicas, son capaces de generar mejores resultados que algunos métodos reportados en la literatura para el conjunto de instancias de prueba propuesto por Cordeau *et al.*

Es importante destacar que, ambos métodos pueden generar buenos resultados realizando pocas llamadas a la función objetivo y utilizando menos recursos computacionales. Las pruebas estadísticas mostradas, indican que las técnicas matheurísticas son más robustas, estables y estadísticamente diferentes a las técnicas presentadas en la literatura.

Como trabajo futuro, se pretende desarrollar versiones autoadaptativas o reactivas para cada matheurística con el fin de obtener un comportamiento más inteligente (establecer los parámetros de cada técnica, automáticamente) y reducir las llamadas a la función objetivo; además, de probar con otros tipos de problemas de optimización.

Fuente de financiamiento

La investigación está financiada por la Universidad Autónoma Metropolitana unidades Iztapalapa y Azcapotzalco, la Facultad de Ciencias de la Universidad Nacional Autónoma de México y el Consejo Nacional de Ciencia y Tecnología.

Referencias

[1] M. Alzaqebah, S. Abdullah, S. Jawarneh, *Modified artificial bee colony for the vehicle routing problems with time windows*, SpringerPlus **5**(2016), 1298. Doi: 10.1186/s40064-016-2940-8

[2] M. Batsyn, I. Bychkov, L. Komosko, A. Nikolaev, *Tabu search for fleet size and mix vehicle routing problem with hard and soft time windows*, in: V.A. Kalyagin, P.M. Pardalos, O. Prokopyev, I. Utkina (Eds.) Computational Aspects and Applications in Large-Scale Networks, Springer Proceedings in Mathematics & Statistics **247**, Springer, Cham, 2018. Doi: 10.1007/978-3-319-96247-4_1

[3] M.A. Boschetti, V. Maniezzo, M. Roffilli, A. Bolufé Röhler, *Matheuristics: Optimization, simulation and control*, in: M.J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, A. Schaerf (Eds.) Hybrid Metaheuristics, Lecture Notes in Computer Science 5818, Springer, Berlin Heidelberg, 2009, pp. 171–177. Doi: 10.1007/978-3-642-04918-7_13

[4] I. Bychkov, M. Batsyn, *A hybrid approach for the capacitated vehicle routing problem with time windows*, in: Optimization Problems and their Applications, Omsk, Russia, 2018, pp. 66–81. http://ceur-ws.org/Vol-2098/paper6.pdf

[5] M. Caserta, S. Voß, *Metaheuristics: Intelligent problem solving*, in: V. Maniezzo, T. Stützle, S. Voß (Eds.) Matheuristics. Hybridizing Metaheuristics and Mathematical Programming, Annals of Information Systems, Vol. 10, Springer, 2009, pp. 1–38. Doi: 10.1007/978-1-4419-1306-7_1

[6] B. Chen, R. Qu, R. Bai, H. Ishibuchi, *A variable neighbourhood search algorithm with compound neighbourhoods for VRPTW*, Proceedings of 5th the International Conference on Operations Research and Enterprise Systems, Vol. 1, Rome, Italy, 2016, pp. 25–35. Doi: `10.5220/0005661800250035`

[7] J.F. Cordeau, G. Desaulniers, J. Desrosiers, M.M. Solomon, F. Soumis, *The VRP with time windows*, Les Cahiers du GERAD, G-99-13, 2000, pp. 1–38 `http://www.bernabe.dorronsoro.es/vrp/data/articles/VRPTW.pdf`

[8] J.F. Cordeau, G. Laporte, *The dial-a-ride problem: models and algorithms*, Annals of Operations Research **153**(2007), no. 1, 29–46. Doi: `10.1007/s10479-007-0170-8`

[9] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, The Rand Corporation, Princeton NJ, 1963. Doi: `10.7249/R366`

[10] G.B. Dantzig, L.R. Ford, D.R. Fulkerson, *A primal-dual algorithm for linear programs*, in: H.W. Kuhn & A.W. Tucker (Eds.) Linear Inequalities and Related Systems, AM-38, Princeton University Press, Princeton NJ, 1956, pp. 171–181.

[11] S.G. De-Los-Cobos-Silva, R.A. Mora-Gutiérrez, M.A. Gutiérrez-Andrade, E.A. Rincón-García, A. Ponsich, P. Lara-Velázquez, *Development of seven hybrid methods based on collective intelligence for solving nonlinear constrained optimization problems*, Artificial Intelligence Review **49**(2018), no. 2, 245–279. Doi: `10.1007/s10462-016-9524-4`

[12] M. Dorigo, V. Maniezzo, A. Colorni, *Ant system: Optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **26**(1996), no. 1, 29–41. Doi: `10.1109/3477.484436`

[13] M.X. Goemans, D.P. Williamson, *The primal-dual method for approximation algorithms and its application to network design problems*, in: D.S. Hochbaum (Ed.) Approximation Algorithms for NP-hard Problems, PWS Publishing, New Orleans, 1996: 144–191. Doi: `10.5555/241938.241942`

[14] Gurobi Optimizer Inc., *Gurobi optimizer reference manual*, G. Optimization, Inc., 2019. https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/9.0/refman.pdf

[15] D. Karaboga, B. Basturk, *Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems*, in: P. Melin, O. Castillo, L.T. Aguilar, J. Kacptrzyk, W. Pedrycz (Eds.), Lecture Notes in Artificial Intelligence 4529 Springer, Berlin, 2007. pp. 789–798. Doi: 10.1007/978-3-540-72950-1_77

[16] J.K. Lenstra, A.H.G. Rinooy Kan, *Complexity of vehicle routing and scheduling problems*, Networks **11**(1981), no. 2, 221–227. Doi: 10.1002/net.3230110211

[17] T.Y. Ma, *A cross entropy multiagent learning algorithm for solving vehicle routing problems with time windows*, in: J.W. Böse, H. Hu, C. Jahn, X. Shi, R. Stahlbock & S. Voß (Eds) Computational Logistics, Lecture Notes in Computer Science 6971, Springer, Berlin, 2011, pp. 59–73. Doi: 10.1007/978-3-642-24264-9_5

[18] E. Montes-Orozco, *Metaheurísticas para el problema de ruteo de vehículos con ventanas de tiempo (VRP-TW)*, Master's thesis, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2017.

[19] R.A. Mora-Gutiérrez, J. Ramírez-Rodríguez, E.A. Rincón-García, *An optimization algorithm inspired by musical composition*, Artificial Intelligence Review **41**(2014), no. 3, 301–315. Doi: 10.1007/s10462-011-9309-8

[20] R.A. Mora-Gutiérrez, J. Ramírez-Rodríguez, E.A. Rincón-García, A. Ponsich, O. Herrera, P. Lara-Velázquez, *Adaptation of the musical composition method for solving constrained optimization problems*, Soft Computing **18**(2014), no. 10, 1931–1948. Doi: 10.1007/s00500-013-1177-5

[21] C.H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, New York, 1998.

[22] H.G.M. Pullen, M.H.J. Webb, *A computer application to a transport scheduling problem*, The Computer Journal **10**(1967), no. 1, 10–13. Doi: 10.1093/comjnl/10.1.10

[23] A.K. Qin, P.N. Suganthan, *Self-adaptive differential evolution algorithm for numerical optimization*, in: Evolutionary Computation, The 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, 2005, pp. 1785–1791. Doi: 10.1109/CEC.2005.1554904

[24] P.P. Repoussis, C.D. Tarantilis, G. Ioannou, *Arc-guided evolutionary algorithm for the vehicle routing problem with time windows*, IEEE Transactions on Evolutionary Computation, **13**(2009), no. 3, 624–647. Doi: 10.1109/TEVC.2008.2011740

[25] D.M. Ryan, C. Hjorring, F. Glover, *Extensions of the petal method for vehicle routing*, Journal of the Operational Research Society **44**(1993), no. 3, 289–296. http://www2.imm.dtu.dk/courses/02735/hjorring.pdf

[26] T. Saeheaw, N. Charoenchai, *Comparison of meta-heuristic algorithms for vehicle routing problem with time windows*, in: H.A Sulaiman et al. (Eds) Advanced Computer and Communication Engineering Technology, Lecture Notes in Electrical Engineering 362, Springer Switzerland, 2016, pp. 1263–1273. Doi: 10.1007/978-3-319-24584-3_108

[27] M.M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research **35**(1987), no. 2, 254–265. Doi: 10.1287/opre.35.2.254

[28] P. Toth, D. Vigo, *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, 2002. Doi: 10.1137/1.9780898718515