Un algoritmo híbrido GRASP+RVND para un Problema de Rutas de Vehículos Capacitados con Ventanas de Tiempo y Sincronización

Oswaldo J. Pérez Luis
Departamento de Ingeniería Informática y de Sistemas
Instituto Universitario de Desarrollo Regional
Universidad de La Laguna
38200, San Cristóbal de La Laguna
operezlu@ull.edu.es

Belén Melián Batista
Departamento de Ingeniería Informática y de Sistemas
Instituto Universitario de Desarrollo Regional
Universidad de La Laguna
38200, San Cristóbal de La Laguna
mbmelian@ull.edu.es

J. Marcos Moreno Vega
Departamento de Ingeniería Informática y de Sistemas
Instituto Universitario de Desarrollo Regional
Universidad de La Laguna
38200, San Cristóbal de La Laguna
jmmoreno@ull.edu.es

Abstract

En este trabajo estudiamos un Problema de Rutas de Vehículos Capacitados con Ventanas de Tiempo y Sincronización en la industria de la recogida de residuos, centrándonos específicamente en la coordinación de múltiples vehículos en las ubicaciones de los clientes. Este problema surge en contextos donde se requiere el servicio simultáneo de diferentes vehículos, como en la recogida especializada de residuos, en la que deben ser recogidos simultáneamente varios tipos de residuos. Proponemos una estrategia híbrida GRASP+RVND, en el que las soluciones construidas con un procedimiento GRASP son mejoradas con un método de Descenso por Entornos Variables Aleatorios (RVND). La experiencia computacional muestra la validez de nuestra propuesta. Se obtienen resultados que igualan o mejoran los obtenidos al resolver de manera exacta el modelo matemático.

1. Introducción

Uno de los mayores desafíos a los que nos enfrentamos es la gestión de los residuos generados por la actividad humana. Se trata de una tarea compleja que involucra a numerosos actores y requiere la movilización y coordinación de diversos recursos. En esta tarea se identifican algunas fases o procesos entre los que destacan la recogida, el transporte, el tratamiento y el desecho final de los residuos no reutilizados o transformados. Estas fases se relacionan con el ciclo de vida de los residuos

XVI XVI Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (maeb 2025).

que, desde una perspectiva de economía circular, persigue reciclar, revalorizar y reutilizar tantos residuos como sea posible.

Para facilitar el posterior tratamiento de los residuos, se fomenta la separación en origen, que traslada a la población y a las empresas parte de la responsabilidad del proceso de reciclaje. Ello obliga a localizar en determinadas ubicaciones contenedores especializados en cada tipo de residuo. La recogida se realiza por medio de vehículos que recorren estas ubicaciones para trasladar los residuos a las plantas de tratamiento.

La eficiencia en la gestión de la recogida y transporte de residuos se vuelve más relevante en entornos urbanos, donde es necesario equilibrar restricciones operativas y medioambientales. La complejidad de estos sistemas se ve agravada por varios factores. Entre ellos se encuentran: la incertidumbre en la cantidad de residuos generados, lo que introduce variabilidad en la planificación y afecta tanto al diseño de las rutas como a la utilización de la capacidad de los vehículos; la necesidad de respetar estrictamente los plazos, no solo para atender a los clientes, sino también para coordinar la llegada de los vehículos a los puntos de recogida, con el objetivo de minimizar los tiempos de espera y los costes asociados; o la necesidad de que el sistema sea capaz de adaptarse a diversas limitaciones operativas, como las horas de trabajo de los empleados de la flota de vehículos.

Los problemas de rutas de vehículos en la recolección de residuos han sido clave en la optimización de estas operaciones, con varias extensiones desarrolladas para abordar retos específicos del mundo real, como se discute en la revisión realizada en Hess et al. [2023]. En este trabajo, los autores distinguen los Problemas de Rutas de Vehículos (VRP), entre otros, como uno de los modelos más utilizados para tratar la problemática de la recogida de residuos.

Los sistemas de recogida de residuos suelen estructurarse en dos niveles. En el primero, vehículos de pequeño tamaño recogen los residuos de las zonas urbanas y los transfieren, en el segundo, a vehículos de mayor capacidad que los transportan a instalaciones de tratamiento o vertederos. Como señalan Ghiani et al. [2021], esta configuración requiere una sincronización precisa en las estaciones de transferencia de residuos para garantizar una operación eficiente y evitar cuellos de botella operativos. Estudios recientes destacan la importancia de integrar la sincronización temporal con la gestión del espacio en los Problemas de Rutas de Vehículos de Recogida de Residuos de Dos Niveles (Rahmanifar et al. [2024]), especialmente cuando los vehículos de gran capacidad funcionan como puntos de transbordo dinámicos, donde el almacenamiento prolongado de residuos no resulta práctico ni deseable. El desafío consiste en desarrollar soluciones que, además de minimizar costes y distancias, optimicen la sincronización de los vehículos para reducir tiempos de espera y mejorar la eficiencia operativa.

La introducción de restricciones de sincronización supone una evolución significativa con respecto a las formulaciones tradicionales de los problemas de rutas de vehículos, donde las rutas suelen planificarse de forma independiente. En el contexto de la recogida de residuos, la sincronización es particularmente relevante cuando varios vehículos deben coordinar sus operaciones para recogidas conjuntas o en puntos de transferencia dentro de una red, como discuten Soares et al. [2024] en su revisión sobre problemas de rutas con sincronización. Estas restricciones introducen interdependencias complejas entre las rutas, ya que la viabilidad de una influye directamente en las demás.

Shao et al. [2020] tratan un problema de recogida selectiva de residuos con restricciones de sincronización. El problema abordado se centra en la contratación de servicios de transporte de para cargas que no llenan la capacidad total de los vehículos de recogida en el sector empresarial. En este contexto, varias empresas solicitan la recogida selectiva de sus residuos al único transportista que presta el servicio. Este dispone de una flota de vehículos especializados en cada tipo de residuo con los que realiza la recogida y transporte. Las empresas presentan diferentes ventanas de tiempo para la recogida y el precio que están dispuestas a pagar por ser atendidas en ellas. Además, imponen la restricción de que todos los vehículos que la visitan lo hagan de forma coordinada, es decir, llegando todos en la misma ventana de tiempo. Por limitaciones de capacidad, sincronización y costes, el transportista debe seleccionar los clientes que va a atender y diseñar las rutas de recogida. En su estudio, Shao et al. [2020] proponen un mecanismo de sincronización basado en subastas, donde el transportista actúa como subastador y los clientes pujan por los servicios de recogida, incorporando ventanas de tiempo en sus ofertas. Para abordar este problema, los autores presentan un algoritmo VNTS de dos capas que combina la Búsqueda por Entornos Variables, para la asignación de las ofertas, y la Búsqueda Tabú, para la construcción de las rutas.

Sin embargo, el modelo de Shao et al. [2020] hace simplificaciones poco realistas para la validez de sus experimentos computacionales. En primer lugar, se considera que todas las empresas requieren la recogida de residuos de todos los tipos, lo que no es previsible que ocurra en la práctica. Asimismo, asumen que las empresas generan la misma cantidad de residuos, independientemente del tipo, lo que es muy poco probable.

En este trabajo realizamos una generalización más realista, donde los clientes pueden generar varios tipos de residuos en cantidades variables no negativas, evitando así la homogeneización en los datos. Un cliente puede, por ejemplo, generar residuos de tres tipos, pero otro solo generarlos de dos. Esto introduce complicaciones adicionales en el modelo, siendo la más significativa el hecho de que no existe un número fijo de vehículos que visiten a los clientes seleccionados para ser atendidos. En el modelo de Shao et al. [2020], a los clientes seleccionados llega el mismo número de vehículos de recogida. Además, en su modelo, las cantidades de residuos a recoger en un cliente son todas iguales. Estos hechos simplifican de tal modo el problema que, en realidad, para dar solución al mismo, basta con planificar las rutas para un residuo y replicarlas para el resto. En nuestro modelo, las rutas diseñadas adoptan formas más diversas.

Para resolver este problema, proponemos un algoritmo híbrido GRASP+RVND en el que, en un esquema multiarranque, se mejoran las soluciones construidas con un Procedimiento de Búsqueda Adaptativa Aleatoria Adaptativa Voraz (GRASP) (Resende and Ribeiro [2019]) con un método de Descenso por Entornos Variables Aleatorios (RVND) (Hansen et al. [2019]). Los resultados computacionales demuestran que GRASP+RVND obtiene soluciones de alta calidad que mejoran las obtenidas al resolver el modelo matemático con Gurobi en problemas de mayor tamaño.

El resto del artículo se estructura como sigue: en la Sección 2 se describe el problema y se presenta un ejemplo ilustrativo; en la Sección 3 se detalla el algoritmo GRASP+RVND; en la Sección 4 se analizan los resultados computacionales; y en la Sección 5 se presentan las conclusiones y futuras líneas de investigación.

2. Descripción del problema

Se considera una red de recogida de residuos compuesta por un único transportista y varios clientes que generan distintos tipos de residuos. El transportista dispone de una flota homogénea de vehículos especializados en cada tipo de residuo con los que presta el servicio de recogida. Si un cliente genera más de un tipo de residuo, varios vehículos deberán visitarlo. El tamaño de la flota es fijo y la cantidad de residuos generados por cada cliente es conocida de antemano. La cantidad generada es siempre inferior a la capacidad de los vehículos, lo que permite que un mismo vehículo pueda visitar a múltiples clientes hasta alcanzar su límite de capacidad. Para que la recogida interfiera lo menos posible en la operativa diaria de los clientes, todos los vehículos que los visiten deben llegar en la misma ventana de tiempo. Debido a las limitaciones de flota, capacidad total o rentabilidad, es posible que algunos clientes no sean atendidos, por lo que corresponde al transportista decidir qué clientes atenderá.

Los clientes expresan su demanda y preferencia horaria mediante pujas, pudiendo presentar varias. Cada puja indica una ventana de tiempo y el precio correspondiente que el cliente está dispuesto a pagar. El cliente asignará la puja más alta a la franja horaria que considere idónea para la recolección de sus residuos. Si su puja resulta ganadora, será atendido en la ventana de tiempo correspondiente por todos los tipos de vehículos necesarios. En este sistema, la empresa de transporte actúa como subastador, seleccionando las pujas ganadoras con el objetivo de maximizar su beneficio, definido como la diferencia entre el valor total de las pujas ganadoras y el coste de transporte asociado a la recolección de los residuos.

El objetivo es maximizar el beneficio de la empresa de transporte bajo cuatro restricciones naturales: la capacidad limitada de los vehículos, el tamaño de la flota disponible, las ventanas de tiempo establecidas en las pujas y la sincronización en la atención a cada cliente. Con este contexto, el problema se clasifica como un Problema de Rutas de Vehículos Capacitado con Ventanas de Tiempo y Sincronización; es decir, se trata de un problema NP-difícil. Esto se deduce del hecho de que el VRP puede reducirse a una formulación del Problema del Viajante de Comercio, uno de los problemas de optimización combinatoria más conocidos y que se sabe que es NP-difícil, lo que implica que no se conoce un algoritmo eficiente para resolver todas sus instancias de manera óptima.

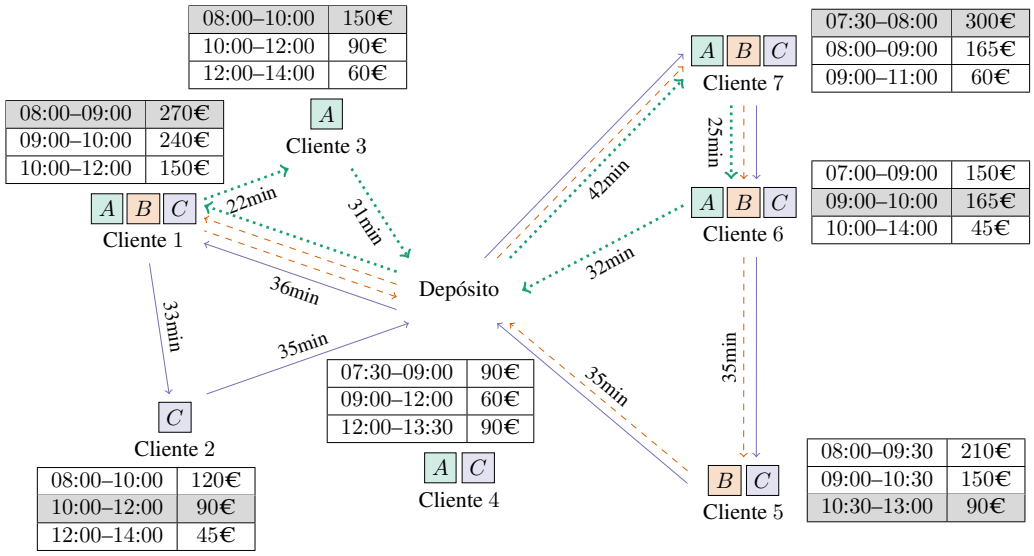

Figura 1: Ejemplo ilustrativo del problema de recogida de residuos.

La Figura 1 ilustra un ejemplo ilustrativo del problema de recogida de residuos con múltiples ventanas de tiempo y sincronización en los puntos de recogida (clientes). Supongamos que el transportista dispone de dos vehículos por tipo de residuo y que el tiempo de servicio en cada punto de recogida es homogéneo e igual a 90 minutos. La jornada laboral comienza a las 07:00 y finaliza a las 14:00 en el depósito, con un coste de transporte equivalente a 1€ por unidad de distancia recorrida. En el grafo de la Figura 1 se representa el tiempo de viaje entre el depósito y los clientes, así como entre estos entre sí. El tiempo de viaje entre el depósito y el cliente 4 es igual a 28 minutos. La sincronización obliga a iniciar el servicio antes de que finalice la ventana de tiempo seleccionada. Si un vehículo llega a un cliente antes del inicio de la franja horaria, debe esperar a su apertura, contemplándose así tiempos de espera.

En este ejemplo, la red consta de de siete clientes que requieren el servicio de recogida de residuos. Cada uno propone tres ventanas de tiempo, asociando una puja a cada una según su conveniencia y flexibilidad. El transportista solamente selecciona una por cliente. Así, de las ventanas propuestas por el cliente 1, el transportista elige la franja [08:00–09:00], lo que le genera unos ingresos de 270€. De los tres vehículos necesarios para atenderlo, el encargado de recoger el residuo de tipo B regresa al depósito tras la recogida, pues no hay otros clientes que requieran ese servicio sin generar costes de transporte superiores al posible beneficio. El resto de vehículos continúa operando antes de regresar al depósito, de tal manera que uno atiende al cliente 2 para recoger el residuo de tipo C, y el otro atiende al cliente 3 para el residuo de tipo A. Nótese que el cliente 3 se encuentra lo suficientemente cerca del cliente 1 como para ser atendido en su ventana de tiempo con mayor puja, mientras que para el cliente 2 se debe optar por la segunda mejor puja para mantener la factibilidad de la solución.

Por otro lado, nótese que el cliente 4 no es atendido pese a estar relativamente cerca del depósito, ya que los gastos de transporte superan su mayor puja. Para recoger sus residuos, sería necesario enviar dos vehículos directamente desde el depósito, uno por cada tipo de residuo, con un coste total de 112€, superior a su mayor puja de 90€. Las pujas seleccionadas por el transportista le reportan unas ganancias de 1065€, mientras que los costes de transporte ascienden a 638€. Con esta solución, el transportista obtiene unos beneficios de 448€.

Sean $C = \{1, 2, \ldots, |C|\}$ un conjunto de clientes, $W = \{1, 2, \ldots, |W|\}$ un conjunto de tipos de residuos y $V^w = \{1, 2, \ldots, |V^w|\}$ un conjunto de vehículos para el tipo de residuo $w \in W$. Cada cliente $i \in C$ tiene un conjunto de ventanas de tiempo $T_i = \{1, 2, \ldots, |T_i|\}$ determinadas por él mismo. Se denota por p_i^t la ganancia obtenida si el cliente $i \in C$ es atendido en la ventana de tiempo $t \in T_i$; es decir, existe una asignación factible para el cliente $i \in C$ en alguna ruta que satisface todas las restricciones del problema en su ventana $t \in T_i$. El transportista debe determinar un conjunto de rutas que atiendan al subconjunto de clientes factibles garantizando que se cumplan las restricciones de factibilidad y sincronización con el propósito de maximizar el beneficio. Para

ello, se debe balancear el valor de las ganancias obtenidas con el valor del coste de transporte que surge del servicio dado a los clientes. Nótese que, a diferencia de las ganancias, el coste de transporte depende de la cantidad de tipos de residuos que se deben recoger en la ubicación de cada cliente.

Sean τ_{ij} el coste de transporte entre los nodos $i,j \in N = C \cup \{0\}$, donde $\{0\}$ representa el depósito, y z_i^{vw} el tiempo en el que el vehículo $v \in V^w$ comienza a dar su servicio al cliente $i \in C$ para recoger el residuo de tipo $w \in W$. Sean $x_{ij}^{vw} \in \{0,1\}$ e $y_i^t \in \{0,1\}$ las variables de decisión que indican si el arco de ruta que une a los nodos $i,j \in N$ seleccionado en la solución para el itinerario de los vehículos asociados al residuo $w \in W$ y si el cliente $i \in C$ es atendido en su ventana de tiempo $t \in T_i$, respectivamente.

Para garantizar la factibilidad y coherencia del plan de rutas de recogida de residuos se imponen las siguientes restricciones: cada cliente puede tener una puja aceptada a lo sumo entre las disponibles (2); si se acepta una puja, deben recogerse todos los tipos d residuos solicitados, asegurando que cada tipo de residuo se recoja solo una vez (3); cada vehículo asignado a al menos un cliente debe iniciar y finalizar su recorrido en el deposito (4, 5); si un vehículo llega a un cliente, debe continuar su ruta hacia otro destino (6); la cantidad total de residuos transportados por un vehículo no puede superar su capacidad máxima (7); todos los vehículos comienzan su servicio al inicio del horizonte de planificación (8); la hora de llegada a cada cliente debe considerar tanto el tiempo de viaje como el de servicio (9); la llegada al depósito y a otros puntos también debe cumplir con relaciones temporales consistentes (10); cada vehículo debe respetar las ventanas de tiempo de las pujas aceptadas, asegurando que las llegadas se produzcan dentro del periodo especificado (11).

Por lo tanto, el problema de optimización resultante se formula como un modelo de programación entera, cuyo objetivo es maximizar el beneficio neto de la empresa de transporte sujeto a todas las restricciones anteriores (1), donde el primer término representa los ingresos totales obtenidos por atender a los clientes dentro de sus ventanas de tiempo seleccionadas, mientras que el segundo término corresponde a los costes de transporte asociados a todos los tipos de residuos y vehículos.

$$\text{máx} \sum_{i \in C} \sum_{t \in T_i} p_i^t y_i^t - \sum_{w \in W} \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} \tau_{ij} x_{ij}^{vw} \tag{1}$$

$s.a:$

$$\sum_{t \in T_i} y_i^t \leq 1, \quad \forall i \in C \tag{2}$$

$$\sum_{v \in V^w} \sum_{j \in N} x_{ij}^{vw} = \sum_{t \in T_i} y_i^t, \quad \forall i \in C,\, w \in W,\, d_i^w > 0 \tag{3}$$

$$\sum_{j \in N} x_{0j}^{vw} = 1, \quad \forall w \in W,\, v \in V^w \iff \exists i \in C \colon d_i^w > 0 \tag{4}$$

$$\sum_{i \in N} x_{i,|C|+1}^{vw} = 1, \quad \forall w \in W,\, v \in V^w \iff \exists i \in C \colon d_i^w > 0 \tag{5}$$

$$\sum_{i \in N} x_{ih}^{vw} = \sum_{j \in N} x_{hj}^{vw}, \quad \forall h \in C,\, w \in W,\, v \in V^w \tag{6}$$

$$\sum_{i \in C} d_i^w \cdot \sum_{j \in N} x_{ij}^{vw} \leq Q^w, \quad \forall w \in W,\, v \in V^w \tag{7}$$

$$z_0^{vw} = 0, \quad \forall w \in W,\, v \in V^w \tag{8}$$

$$z_i^{vw} + \tau_{ij} + s_i - M \cdot (1 - x_{ij}^{vw}) \leq z_j^{vw}, \quad \forall i,j \in C,\, i \neq j,\, w \in W,\, v \in V^w\, d_i^w > 0 \tag{9}$$

$$z_0^{vw} + \tau_{ij} - M \cdot (1 - x_{ij}^{vw}) \leq z_j^{vw}, \quad \forall j \in N,\, i \neq j,\, w \in W,\, v \in V^w\, d_i^w > 0 \tag{10}$$

$$L_i^t - M \cdot (1 - y_i^t) \leq z_i^{vw} \leq U_i^t + M \cdot (1 - y_i^t), \quad \forall i \in C,\, t \in T_i,\, w \in W,\, v \in V^w \tag{11}$$

$$x_{ij}^{vw} \in \{0,1\}, \quad \forall i,j \in N,\, w \in W,\, v \in V^w\, d_i^w > 0 \tag{12}$$

3. Método de solución

Para tratar el problema descrito, proponemos una heurística híbrida que construye soluciones siguiendo una estrategia GRASP que luego son mejoradas con RVND. La fase constructiva de GRASP garantiza que los clientes seleccionados sean atendidos para todos los residuos que generan. Sea

$R^w = \{r_1^w, r_2^w, \ldots, r_{|V^w|}^w\}$ el conjunto de rutas asociadas al residuo w, con r_v^w la ruta asociada al vehículo v. El proceso se inicia escogiendo al azar un tipo de residuo $w' \in W$ y considerando el conjunto de clientes $C^{w'}$ con demanda positiva para este. Para cada uno de estos clientes, se evalúa su factibilidad y beneficio en las rutas actuales de $R^{w'}$. Se genera una lista restringida $C_k^{w'}$ con los k clientes más prometedores. De esta lista, se selecciona aleatoriamente uno para su mejor inserción y se añade al conjunto C' de clientes a visitar; en caso contrario, se descarta. Este proceso se repite hasta agotar los candidatos.

Tras definir el subconjunto de clientes C', se construyen rutas para los tipos de residuos restantes. Para cada residuo $w \in W \setminus \{w'\}$, se filtran los clientes del conjunto común que tienen demanda positiva para ese residuo ($d_i^w > 0$). De esta manera, se garantiza que cada cliente sea atendido para todos los tipos de residuos que demanda, cumpliendo con la restricción principal del problema. Para estos clientes, se construyen rutas factibles $r_v^w \in R^w$ siguiendo los criterios de inserción descritos anteriormente.

Luego, una solución factible $s = \bigcup_{w \in W} R^w$ del problema está representada por un itinerario factible para cada tipo de residuo implicado; es decir, un conjunto de rutas en el que, si un cliente aparece en alguna ruta de un tipo de residuo, debe aparecer también en las rutas correspondientes a los demás residuos para los que genere demanda. El Algoritmo 1 describe este procedimiento. Nótese que en la heurística diseñada para generar una solución inicial no se permite la violación de ninguna restricción; es decir, al seleccionar un nuevo cliente para insertar en una ruta este debe de satisfacer tanto las restricciones de ventanas de tiempo como su demanda no sobrepasar la capacidad del vehículo asociado a la ruta actual.

Algoritmo 1: Marco de la fase constructiva del GRASP

Entrada: C, conjunto de clientes; W, conjunto de residuos; V^w, conjunto de vehículos v para el residuo w; R^w, conjunto de rutas para el residuo w
Salida: s, solución con rutas por tipo de residuo

Seleccionar $w' \in W$ un tipo de residuo al azar; Fijar $k \in \mathbb{N}$
$C' \leftarrow \emptyset$; $C^{w'} \leftarrow \{i \in C \mid d_i^{w'} > 0\}$; $R^{w'} = \emptyset$
mientras $C^{w'} \neq \emptyset \wedge \exists i \in C^{w'}$ *con inserción factible en alguna ruta* $r_v^{w'} \in R^{w'}$ **hacer**
    $C_k^{w'} \leftarrow k$ clientes de $C^{w'}$ de mayor beneficio que sean factibles
    Seleccionar al azar $i \in C_k^{w'}$ e insertar en $r_v^{w'} \in R^{w'}$
    $C' \leftarrow C' \cup \{i\}$
    $C^{w'} \leftarrow C^{w'} \setminus \{i\}$
para cada $w \in W \setminus \{w'\}$ **hacer**
    $C^w \leftarrow \{i \in C' \mid d_i^w > 0\}$
    Construir rutas $r_v^w \in R^w$ factibles para C^w usando vehículos $v \in V^w$
devolver $s = \bigcup_{w \in W} R^w$

Los métodos de búsqueda local mejoran iterativamente una solución explorando su entorno hasta alcanzar un óptimo local. En problemas de rutas de vehículos, las estructuras de entorno más comunes incluyen: intercambiar las posiciones de dos clientes en la misma ruta o en rutas diferentes, reubicar un cliente en su ruta o hacia otra, o aplicar movimientos 2-opt.

En el problema abordado en el presente trabajo, algunas demandas de clientes pueden no ser atendidas dentro del horizonte de planificación debido a las restricciones impuestas por la limitación en el número de vehículos disponibles y en la capacidad total de los mismos. En consecuencia, se implementan, además, estructuras de entorno que consideran la reinserción de clientes no presentes en la solución, la posibilidad de excluir clientes de la solución, y el intercambio entre clientes omitidos y clientes que forman parte de la solución. Además, dado un par de clientes, pueden darse diversas combinaciones en su asignación a las rutas; es decir, pueden compartir la misma ruta para un residuo, estar en rutas distintas para otros o, incluso, no ser incluidos en ciertas rutas si no generan demanda para determinados residuos. Este hecho complica el proceso de verificación de la factibilidad de los movimientos. En definitiva, los movimientos considerados en este trabajo son: reinserción intra-ruta, reinserción inter-rutas, reinserción de clientes excluidos, intercambio intra-ruta, intercambio inter-rutas, intercambio con clientes excluidos y movimiento 2-opt. A continuación se describen brevemente las mismas.

Un movimiento de reinserción intra-ruta consiste en reubicar un cliente dentro de su propia ruta. Para ello, se elimina de su posición original y se inserta en la mejor ubicación factible, garantizando que se respeten las restricciones de tiempo. Si resulta beneficioso para la solución, se permite ajustar la ventana horaria del cliente, siempre que las rutas a la que pertenezca el cliente permanezcan factibles. De manera similar, un movimiento de reinserción inter-ruta traslada un cliente a una ruta diferente, y un movimiento de reinserción de clientes excluidos reintroduce clientes previamente eliminados en la mejor ruta disponible. En ambos casos, además de las restricciones de tiempo, se debe garantizar que no se exceda la capacidad del vehículo.

Un movimiento de intercambio intra-ruta consiste en intercambiar las posiciones de dos clientes dentro de una misma ruta. Para ello, se elimina a un cliente de su posición original y se intercambia con otro en la mejor ubicación factible, garantizando que se respeten las restricciones de tiempo. A diferencia de los movimientos de reinserción, no se permite ajustar la ventana horaria del cliente. De manera análoga, un movimiento de intercambio inter-rutas afecta a clientes de rutas distintas, y un movimiento de intercambio con clientes excluidos introduce a un cliente previamente excluido en una ruta a cambio de otro que es retirado. En estos dos casos, también se debe respetar la restricción de capacidad del vehículo.

Por último, un movimiento 2-opt consiste en modificar la estructura de una ruta invirtiendo la secuencia de un subconjunto de clientes. Este procedimiento busca reducir la distancia recorrida eliminando cruces innecesarios en la trayectoria, respetando las restricciones de ventanas de tiempo, sin permitir cambiar la franja horaria de los clientes afectados.

Todas estas estructuras de entornos se integran en un esquema RVND, un algoritmo que se deriva del estándar del Descenso por Entornos Variables, que selecciona al azar una de las estructuras de entornos disponibles y aplica la búsqueda local con dicha estructura hasta alcanzar un óptimo local. Si este óptimo mejora la solución actual, se actualiza la mejor solución y se deshabilita la estructura de entornos empleada. En este trabajo se permite la reactivación de las otras estructuras de entornos que no mejoraron la solución en iteraciones previas. No se permite seleccionar consecutivamente la misma estructura de entornos dado que, al haberse alcanzado el óptimo local en dicho entorno, repetirla resultaría en la convergencia al mismo punto. Este proceso se describe en el Algoritmo 2 y termina cuando ninguna de las estructuras de entornos disponibles logra mejorar la solución actual. De esta manera, se obtiene un óptimo local con respecto a todas las estructuras de entornos disponibles, superior al construido con el método GRASP.

Algoritmo 2: Marco general del RVND

Entrada: s, solución inicial; $\mathcal{N}$, conjunto de estructuras de entornos
Salida: s^*, solución mejorada

```
s* ← s; N' ← N
mientras s* ≤ s' ∧ N' ≠ ∅ hacer
    Seleccionar aleatoriamente una estructura de entornos N ∈ N
    s' ← Aplicar búsqueda local de N sobre s*
    si f(s') > f(s*) entonces
        s* ← s'
        N' ← N \ {N}
    en otro caso
        N' ← N' \ {N}
devolver s*
```

4. Experiencia computacional

Para evaluar la validez de nuestra propuesta, comparamos los resultados de GRASP+RVND con los obtenidos al resolver el modelo matemático (1) con Gurobi 11.0.3 en Python para instancias pequeñas. Los experimentos computacionales se realizaron en un sistema Windows de 64 bits con un procesador Intel Core i5-12500T (2GHz) y 16GB de RAM.

Para facilitar una comparación detallada, se diseñó un conjunto de instancias basado en las de Solomon para el VRP (Solomon [1987]), seleccionando nueve instancias: R101, R102 y R103, con

clientes distribuidos aleatoriamente; C101, C102 y C103, con clientes agrupados; RC101, RC102 y RC103, que combinan ambos esquemas.

Las instancias de Solomon incluyen información sobre la ubicación del depósito y los clientes, su demanda, las ventanas de tiempo, la capacidad de los vehículos y el tamaño de la flota. Para adaptar estas instancias a nuestro problema de sincronización múltiple, se realizaron las siguientes modificaciones: se consideran hasta tres tipos de residuos simultáneamente, asignando demanda mediante una distribución uniforme discreta no negativa acotada superiormente por el valor original; se crean ventanas de tiempo de sincronización, duplicando y reduciendo a la mitad la duración original; cada cliente pueda presenta tres pujas distintas, generadas aleatoriamente entre 100 y 500, asignando la mayor puja a la ventana más corta y la menor a la más amplia; por último, se dispone de una flota de 20 vehículos con capacidad homogénea de 200 unidades por tipo de residuo.

Las instancias generadas se nombran siguiendo el formato "instancia base de Solomon"–"número de clientes"–"número de tipos de residuos". Por ejemplo, "C101–10–3" indica que se usa la instancia C101, con los diez primeros clientes y tres tipos de residuos.

Comparativa entre Gurobi y GRASP+RVND

La Tabla 1 muestra la comparación entre la resolución del modelo de programación entera con Gurobi Optimization, LLC [2024] y GRASP+RVND. La primera columna de esta tabla indica el nombre de la instancia. A continuación, se reportan: el valor objetivo obtenido con Gurobi, el *gap* con respecto a la cota superior y el tiempo de ejecución, donde se fijó un límite de 7200 segundos. Las tres últimas columnas indican el mejor valor objetivo alcanzado con GRASP+RVND en 5 ejecuciones de 1000 iteraciones cada una, la desviación con respecto a Gurobi y el tiempo de ejecución.

Tabla 1: Resultados de comparación de algoritmos

	Gurobi			**GRASP+RVND**		
Instancia	Objetivo	Gap (%)	Tiempo (s)	Objetivo	Desv. (%)	Tiempo (s)
C101–10–3	3238,36	0,00	8,69	3238,36	0,00	0,001
C102–10–3	3203,79	1,78	7200,00	3196,80	0,22	0,001
C103–10–3	3195,25	1,09	7200,00	3166,59	0,90	0,001
R101–10–3	2656,57	0,00	23,82	2652,61	0,15	0,001
R102–10–3	2721,69	0,00	1534,05	2709,18	0,46	0,001
R103–10–3	2742,10	0,00	506,86	2739,21	0,11	0,001
RC101–10–3	2929,04	0,00	570,10	2913,63	0,53	0,001
RC102–10–3	2911,45	5,14	7200,00	2899,83	0,40	0,001
RC103–10–3	2888,89	5,73	7200,00	2888,89	0,00	0,001
C101–15–3	5115,94	0,00	92,64	5115,94	0,00	0,001
C102–15–3	5041,35	3,74	7200,00	5006,18	0,70	0,001
C103–15–3	5058,19	4,00	7200,00	5044,00	0,28	0,001
R101–15–3	4485,46	0,00	1230,35	4427,20	1,30	0,001
R102–15–3	4612,27	1,43	7200,00	4604,24	0,17	0,001
R103–15–3	4673,21	4,72	7200,00	4670,85	0,05	0,001
RC101–15–3	4881,56	4,12	7200,00	4851,81	0,61	0,001
RC102–15–3	4948,15	7,29	7200,00	4924,44	0,48	0,001
RC103–15–3	4919,20	7,69	7200,00	4883,14	0,73	0,001
C101–25–3	9309,11	0,51	7021,99	9130,51	1,92	0,003
C102–25–3	9088,67	5,69	7200,00	9113,18	−0,27	0,003
C103–25–3	9104,25	5,53	7200,00	9113,18	−0,10	0,003
R101–25–3	8302,21	6,72	7200,00	8185,58	1,40	0,002
R102–25–3	8231,07	9,32	7200,00	8237,01	−0,07	0,005
R103–25–3	6003,63	52,02	7200,00	8571,04	−42,76	0,002
RC101–25–3	8543,47	11,72	7200,00	8631,20	−1,03	0,001
RC102–25–3	8754,20	9,51	7200,00	8777,37	−0,26	0,003
RC103–25–3	8612,74	11,30	7200,00	8744,58	−1,53	0,002

Se observa que Gurobi alcanza la optimalidad solamente en 6 de las 27 instancias, 4 de ellas con 10 clientes y otras 2 con 15 clientes, lo que evidencia la complejidad de este problema de sincronización

múltiple en la recogida de residuos. En contraste, nuestro algoritmo muestra robustez con bajas desviaciones respecto a las soluciones óptimas. Además, supera a Gurobi en casi todas las instancias con 25 clientes, destacando su desempeño en R103-25-3, donde Gurobi tiene dificultades para aproximarse al óptimo.

El tiempo de ejecución de GRASP+RVND se mantiene bajo a medida que aumenta el número de clientes, sin variaciones significativas según su distribución espacial. En cambio, la precisión de Gurobi depende de la distribución, obteniendo mejores soluciones en instancias con clientes agrupados, mientras que en distribuciones aleatorias la complejidad del problema aumenta.

Resultados para instancias de mayor tamaño

La Tabla 2 presenta los resultados de GRASP+RVND en instancias de 50 y 100 clientes, mostrando el mejor valor objetivo y el tiempo de ejecución. No se incluyen resultados de Gurobi ya que no logró encontrar soluciones en un tiempo razonable. Como en la comparativa anterior, el algoritmo GRASP+RVND se ejecuta 5 veces con un criterio de parada de 1000 iteraciones por ejecución.

Tabla 2: Resultados del algoritmo GRASP+RVND para instancias con 50 y 100 clientes

Instancias con 50 clientes			**Instancias con 100 clientes**		
Instancia	Objetivo	Tiempo (s)	Instancia	Objetivo	Tiempo (s)
C101–50–3	19022,1	0,321	C101–100–3	37041,5	6,354
C102–50–3	18809,6	0,380	C102–100–3	36859,1	6,403
C103–50–3	18783,6	0,340	C103–100–3	36786,7	6,660
R101–50–3	17317,3	0,318	R101–100–3	36315,4	6,877
R102–50–3	17578,3	0,350	R102–100–3	36454,0	6,372
R103–50–3	18030,5	0,409	R103–100–3	36656,4	6,399
RC101–50–3	17443,1	0,318	RC101–100–3	35991,2	6,185
RC102–50–3	17876,2	0,332	RC102–100–3	36106,7	6,283
RC103–50–3	17528,5	0,334	RC103–100–3	35899,9	6,399

Se puede apreciar que el tiempo de ejecución de nuestro algoritmo varía poco con la distribución de clientes, aunque crece significativamente con su número, manteniéndose dentro de límites razonables.

5. Conclusiones y líneas futuras

En este trabajo, proponemos un algoritmo híbrido GRASP+RVND para resolver un Problema de Rutas de Vehículos Capacitados con Ventanas de Tiempo y Sincronización en la recogida selectiva de residuos. Aunque nuestro enfoque muestra resultados prometedores, identificamos varias oportunidades de mejora.

Hemos observado que el hecho de que las estructuras de entornos actuales funcionen de forma independiente limita su eficacia cuando las modificaciones en la ruta de un tipo de residuo afectan a la viabilidad de las recogidas sincronizadas. Asimismo, las restricciones de ventanas de tiempo restringen significativamente el espacio de soluciones, excluyendo potencialmente soluciones de alta calidad que solo violan ligeramente estas restricciones. Además, en ocasiones, el algoritmo alcanza los óptimos locales antes de tiempo, lo que sugiere la necesidad de mecanismos de diversificación más sofisticados.

Basándonos en estas observaciones, proponemos que las futuras implementaciones consideren una búsqueda local con estructuras de entornos combinadas que gestionen simultáneamente múltiples tipos de residuos. Cuando un movimiento encuentra una nueva posición factible para un cliente con un tipo de residuo, pero altera su ventana de tiempo de servicio, los ajustes correspondientes deben propagarse a otros tipos de residuos para mantener la sincronización. Este enfoque integrado mejoraría la calidad de la solución al ampliar el espacio de soluciones.

Por otra parte, relajar las restricciones de ventana temporal permitiendo infracciones menores (llegadas tempranas o tardías) podría ampliar significativamente el espacio de soluciones. Esta modificación requeriría el desarrollo de funciones de penalización apropiadas que equilibren las expectativas de nivel de servicio. Este planteamiento reflejaría mejor la flexibilidad operativa del mundo real y podría identificar soluciones superiores.

Agradecimientos

Este trabajo ha sido financiado por la Agencia Estatal de Investigación (España; proyecto PID2019-104410RB-I00/AEI/10.13039/501100011033).

Referencias

G. Ghiani, A. Manni, E. Manni, and V. Moretto. Optimizing a waste collection system with solid waste transfer stations. *Computers & Industrial Engineering*, 161:107618, Nov. 2021. ISSN 0360-8352. doi: 10.1016/j.cie.2021.107618.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL `https://www.gurobi.com`.

P. Hansen, N. Mladenović, J. Brimberg, and J. A. Moreno Pérez. *Variable Neighborhood Search*, pages 57–97. Springer International Publishing, Cham, 2019. ISBN 978-3-319-91086-4. doi: 10.1007/978-3-319-91086-4_3. URL `https://doi.org/10.1007/978-3-319-91086-4_3`.

C. Hess, A. Dragomir, K. Doerner, and D. Vigo. Waste collection routing: a survey on problems and methods. *Central European Journal of Operations Research*, 32:1–36, 11 2023. doi: 10.1007/s10100-023-00892-y.

G. Rahmanifar, M. Mohammadi, M. Hajiaghaei-Keshteli, G. Fusco, and C. Colombaroni. An integrated temporal and spatial synchronization for two-echelon vehicle routing problem in waste collection system. *Journal of Industrial Information Integration*, 40:100611, July 2024. ISSN 2452-414X. doi: 10.1016/j.jii.2024.100611.

M. G. C. Resende and C. C. Ribeiro. *Greedy Randomized Adaptive Search Procedures: Advances and Extensions*, pages 169–220. Springer International Publishing, Cham, 2019. ISBN 978-3-319-91086-4. doi: 10.1007/978-3-319-91086-4_6. URL `https://doi.org/10.1007/978-3-319-91086-4_6`.

S. Shao, S. X. Xu, and G. Q. Huang. Variable neighborhood search and tabu search for auction-based waste collection synchronization. *Transportation Research Part B: Methodological*, 133:1–20, Mar. 2020. ISSN 0191-2615. doi: 10.1016/j.trb.2019.12.004.

R. Soares, A. Marques, P. Amorim, and S. Parragh. Synchronisation in vehicle routing: Classification schema, modelling framework and literature review. *European Journal of Operational Research*, 313:817–840, 03 2024. doi: 10.1016/j.ejor.2023.04.007.

M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987. doi: 10.1287/opre.35.2.254.