Problema de Rutas de Vehículos con Capacidad y Sincronización en Instalaciones Intermedias

Oswaldo J. Pérez-Luis[1 [0009−0009−6643−2870]], Belén Melián-Batista[✉1 [0000−0001−9577−6247]], J. Marcos Moreno-Vega[1 [0000−0002−7675−8577]], e Irma Delia García-Calvillo[2 [0000−0003−0609−4223]]

[1] Dpto. de Ingeniería Informática y de Sistemas, Instituto Universitario de Desarrollo Regional, Universidad de La Laguna, San Cristóbal de La Laguna, Santa Cruz de Tenerife, 38200, España
{operezlu,mbmelian,jmmoreno}@ull.edu.es

[2] Centro de Investigación en Matemáticas Aplicadas, Universidad Autónoma de Coahuila, Unidad Camporredondo, Saltillo, Coahuila, 25280, México
irma.garcia@uadec.edu.mx

Abstract. Este trabajo estudia una variante de dos niveles del Problema de Rutas de Vehículos con Capacidad e Instalaciones Intermedias aplicada a la gestión de residuos sólidos urbanos, en la que vehículos de recogida y vehículos de transporte de mayor capacidad deben coordinarse para dar servicio de forma eficiente. La condición de sincronización exacta en estaciones de transferencia sin capacidad de almacenamiento intermedio, combinada con múltiples viajes y visitas dentro del horizonte operativo, configura un problema *NP-hard* escasamente estudiado en la literatura. Se propone un algoritmo híbrido que integra *Reactive Greedy Randomized Adaptive Search Procedure*, *Variable Neighborhood Descent* y un *Adaptive Large Neighborhood Search* extendido con selección de operadores por *Q-learning* y memoria en línea que integra ambos niveles en la toma de decisiones del algoritmo. Los experimentos se realizan sobre 90 instancias generadas a partir de datos georreferenciados reales de la ciudad de Saltillo, México y los resultados muestran que Gurobi obtiene soluciones de alta calidad en instancias pequeñas, mientras que el algoritmo inteligente lo supera en calidad y eficiencia computacional en instancias de mayor tamaño.

Palabras claves: Recolección de residuos · 2E-VRP-IF · GRASP · ALNS

1. Introducción

La gestión de los residuos sólidos urbanos ha dado lugar a una amplia variedad de modelos de optimización de rutas que abarcan problemas sobre nodos, arcos y formulaciones de localización-ruta. En este ámbito, recientemente se presentó una taxonomía actualizada [7] clasifica la literatura del problema de rutas de vehículos (VRP) por tipo de problema e identifica características prácticas relevantes como

la demanda incierta, los sistemas de recogida alternativos y el uso generalizado de instalaciones intermedias (IF).

Una extensión fundamental surge cuando los vehículos descargan en instalaciones intermedias antes de continuar su ruta [12]. En el contexto de gestión de residuos, esta estructura fue formalizada inicialmente en [1] y estudiada mediante formulaciones exactas e híbridas. Cuando se incorpora una segunda flota de mayor capacidad entre el depósito y dichas instalaciones, el problema adopta una estructura de dos niveles [14], y la sincronización temporal en nodos intermedios, como las estaciones de transferencia de residuos, adquiere especial relevancia.

Una clase compleja de estos problemas surge cuando la recogida y el transporte a larga distancia se realizan mediante dos tipos distintos de vehículos que operan de forma sincronizada. En estos sistemas, vehículos de recogida (CV) descargan los residuos en IF, donde se consolidan y se cargan en vehículos de transporte de mayor capacidad (TV) para su traslado a instalaciones de eliminación. La planificación conjunta de ambas flotas con sincronización explícita en las IF se sitúa en la intersección del problema de ruta de vehículos con capacidad (CVRP) y sus variantes, en particular el VRP con instalaciones intermedias (VRP-IF) y el VRP de dos niveles (2E-VRP). No obstante, esta formulación integrada ha recibido escasa atención en la literatura. Las taxonomías sobre el VRP-IF y el 2E-VRP [12,14] identifican la coordinación de vehículos en IF como un reto de investigación abierto, mientras que las taxonomías de sincronización [2,15] formalizan los mecanismos de coordinación presentes en estos sistemas, incluyendo la sincronización de operaciones y de carga. Entre los pocos trabajos que consideran simultáneamente dos tipos de vehículos y sincronización en IF, [4] constituye un primer paso relevante al representar las rutas de los CV mediante una secuencia de tramos pre-enumerados entre depósitos e IF. A diferencia de este enfoque, el presente trabajo adopta una representación basada en arcos, en la que las rutas de los CV se describen directamente mediante variables binarias sobre la red completa, eliminando la necesidad de pre-enumerar tramos y favoreciendo la escalabilidad a instancias de mayor tamaño.

La complejidad de este problema ha motivado el desarrollo de enfoques exactos, heurísticos y mateheurísticos. La sincronización en nodos intermedios dentro del 2E-VRP con múltiples viajes fue estudiada en [5], y estudios posteriores han confirmado que la programación conjunta de ambas flotas resulta determinante para la eficiencia global del sistema [10]. En cuanto a los métodos de resolución, la metaheurística *Adaptive Large Neighborhood Search* (ALNS) ha mostrado un elevado rendimiento tanto en el VRP-IF como en el 2E-VRP [5], y las mateheurísticas han permitido mejorar significativamente la calidad de las soluciones en variantes complejas [14].

Las aportaciones de este trabajo son las que siguen. Se desarrolla un método híbrido, denominado «ALNS inteligente (S-ALNS)», que combina *Greedy Randomized Adaptive Search Procedure* reactivo (R-GRASP) con mejora local por *Variable Neighborhood Descent* (VND) y una fase de intensificación y diversificación mediante un ALNS guiado por un agente de *Q-learning* tabular con memoria de frecuencia en los operadores de inserción. Se presenta además un

estudio computacional sobre instancias de referencia generadas a partir de datos operativos reales de la ciudad de Saltillo, México, puestas a disposición pública en [8] para facilitar la reproducibilidad de los experimentos y la comparación futura de métodos, cuya validación incluye un análisis de ablación que cuantifica la contribución individual de cada componente del S-ALNS frente a sus variantes de referencia.

2. Descripción del problema

El problema se define sobre un conjunto de nodos de recogida, cada uno con una demanda de residuos conocida que debe satisfacerse dentro de un único horizonte de planificación, con una infraestructura que opera en dos niveles. El primer nivel comprende una flota de CV que parten de un depósito central, visitan los nodos y descargan los residuos en instalaciones intermedias antes de continuar o regresar al depósito, pudiendo realizar múltiples visitas a una IF dentro del horizonte. El segundo nivel está formado por una flota de TV que parten desde un vertedero remoto, se desplazan hasta las IF, consolidan la carga de varios CV y retornan al vertedero. El objetivo consiste en minimizar la distancia total recorrida por el conjunto de ambas flotas.

La principal restricción operativa es que, al carecer las IF de capacidad de almacenamiento intermedio, la primera llegada de un CV a una IF exige que el TV asignado se encuentre físicamente presente para recibir los residuos de forma inmediata; en los términos de [2], esta condición constituye una sincronización exacta de operaciones. Una vez iniciada la visita del TV, este permanece en la IF para atender todos los eventos de descarga sucesivos sin que generen un requisito adicional de sincronización, dado que el TV ya se encuentra posicionado; el TV no abandona la IF mientras su capacidad remanente pueda absorber la carga de al menos un CV adicional, de modo que el instante de partida queda determinado por el último evento de descarga que puede ser atendido sin exceder el límite de capacidad del TV.

El problema requiere, por tanto, la determinación simultánea de: la ruta de cada CV, incluyendo la secuencia de puntos de recogida visitados y las IF en las que el vehículo descarga; la ruta de cada TV, incluyendo la secuencia de IF visitadas y la programación temporal de dichas visitas; y la asignación de los eventos de descarga de los CV a TV concretos en cada IF, junto con una planificación factible que garantice que la condición de sincronización sin almacenamiento se cumple en cada punto de transferencia.

La Figura 1 ilustra un ejemplo con ocho nodos de recogida, dos IF, un depósito y un vertedero, atendidos por tres CV y dos TV, donde la restricción operativa determinante es la condición de ausencia de almacenamiento en ambas IF; es decir, en IF_1, el TV debe haber llegado antes o en el mismo instante en que el primer CV deposita su carga, y no puede partir hasta que el último CV haya completado también su descarga, dado que ambos CV están asignados al mismo TV en esta instalación.

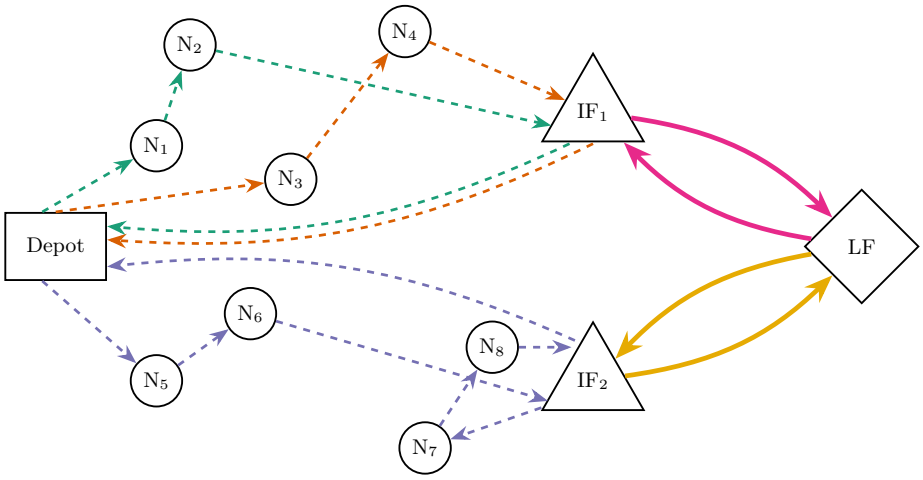


Figura 1: Ejemplo ilustrativo, donde ⇢ representa CV y → representa TV.

En el primer nivel, los tres CV parten simultáneamente del depósito al inicio del horizonte de planificación. Un CV visita N_1 y N_2 en secuencia, recogiendo residuos a lo largo del recorrido; una vez que su carga se aproxima a la capacidad máxima, se dirige directamente a IF_1 para descargar y regresar al depósito. Otro CV visita N_3 y N_4 antes de descargar en IF_1 y retornar al depósito. Ambos vehículos comparten, por tanto, IF_1 como punto de descarga, lo que implica que un TV debe estar presente en dicha instalación antes de que llegue el primer CV y permanecer en ella para recibir la carga de ambos antes de partir hacia el vertedero. Un último CV opera en la zona inferior, en la que visita N_5 y N_6, descarga en IF_2, continúa visitando N_7 y N_8, realiza una segunda descarga en IF_2 y regresa finalmente al depósito. Al igual que antes, cada uno de los dos eventos de descarga en IF_2 debe ir emparejado con la presencia de un TV.

En el segundo nivel, correspondiente a las rutas de los vehículos de transporte, un TV parte del vertedero y se desplaza hasta IF_1, donde aguarda para recibir los residuos consolidados depositados por dos CV. Una vez que todos los CV asignados han completado su descarga, condición impuesta por las restricciones de sincronización sin almacenamiento, el TV abandona IF_1 y regresa directamente al vertedero, donde su carga es descargada. Dada la capacidad de los vehículos de transporte, se asume que se pueden acomodar en un único viaje las cargas combinadas de ambos CV, por lo que un retorno directo desde IF_1 al vertedero resulta suficiente. Por otro lado, otro TV realiza una operación análoga en la parte inferior de la red; es decir, parte del vertedero, se desplaza hasta IF_2, recoge los residuos depositados por un CV en sus dos visitas y regresa al vertedero, siendo igualmente su capacidad suficiente para consolidar ambas cargas en un único viaje.

3. Método de resolución

Se propone el S-ALNS, un algoritmo híbrido que combina un R-GRASP [9] con mejora local por VND [3] y una fase de intensificación y diversificación mediante ALNS guiado.

El esquema general es el siguiente: en una primera fase, el R-GRASP genera un conjunto de soluciones iniciales de calidad mediante un mecanismo de construcción aleatorizado que es mejorado sistemáticamente por el VND, de modo que la mejor solución obtenida en esta fase sirve como punto de partida para el S-ALNS, que explora el espacio de soluciones de forma más agresiva mediante operadores de destrucción y reparación. A diferencia de un ALNS clásico, que operaría de forma enteramente desacoplada, concentrando la búsqueda en el primer nivel e ignorando el segundo durante la exploración, el S-ALNS integra activamente la información del segundo nivel a lo largo de toda la búsqueda; es decir, la planificación de TV se calcula bajo demanda en cada iteración y se utiliza para clasificar el estado de búsqueda, orientar los operadores de destrucción y reparación, y evaluar las soluciones candidatas sobre el coste combinado de ambos niveles. Esta integración mantiene el espacio de decisión circunscrito a las rutas del primer nivel, lo que acota el coste computacional, pero hace que la exploración e intensificación estén permanentemente informadas por el segundo nivel. Este esquema se describe en el Algoritmo 1.

La mejora local se realiza mediante un VND con seis entornos de primera mejora dedicados para el primer nivel en una secuencia fija y cíclica, ordenados de menor a mayor perturbación estructural, de modo que los operadores de menor coste computacional se emplean antes de comprometer la solución con movimientos de mayor perturbación. Los tres primeros entornos se basan en la reubicación de un único nodo: la reinserción dentro del mismo trayecto —recuérdese que una ruta de CV se compone de uno o varios trayectos a las IF—, *IntraTrip Reinsert*, que desplaza un nodo a otra posición dentro del mismo trayecto sin alterar el número de visitas a las instalaciones intermedias, entendiendo por trayecto aquella secuencia de nodos a los que visita un CV entre cada viaje a un IF; la reinserción entre trayectos de la misma ruta, *InterTrip Reinsert*, que transfiere un nodo a un trayecto distinto perteneciente al mismo CV, lo que puede implicar una reasignación de la instalación intermedia de los trayectos afectados; y la reinserción entre rutas de distintos CV, *InterRoute Reinsert*, que modifica simultáneamente la carga y la asignación de instalaciones de dos rutas diferentes. El cuarto entorno, *Swap*, generaliza la reubicación al considerar el intercambio simultáneo de dos nodos cualesquiera, abarcando los tres ámbitos anteriores en una única clase. El quinto entorno, *SplitShift*, es específico del VRP-IF con capacidades, en donde se desplaza el límite entre dos trayectos consecutivos de una misma ruta, reasignando el último nodo del trayecto anterior al inicio del siguiente o viceversa, lo que altera el punto de descarga en la instalación intermedia. El sexto entorno, *IntraTrip 2-opt*, invierte el orden de un subsegmento contiguo de nodos dentro de un único trayecto, permitiendo mejorar cadenas de coste de arco asimétricas que ninguna secuencia de movimientos de nodo individual encontraría.

3.1. GRASP reactivo

La fase constructiva se basa en un GRASP reactivo, una extensión del GRASP clásico que elimina la necesidad de fijar el parámetro de aleatoriedad α *a priori*; esto es, en lugar de un valor único, se mantiene un conjunto discreto de valores $\alpha_i \in$ A distribuidos uniformemente en el intervalo $[0, 1]$, a los que se les asocia una probabilidad de selección que se actualiza periódicamente en función de la calidad de las soluciones que ha producido. Concretamente, tras cada período de δ iteraciones, la probabilidad de selección del valor α_i se actualiza de forma proporcional al cociente de calidad $c^*/\bar{c}_i$, donde c^* es el coste de la mejor solución encontrada hasta el momento y $\bar{c}_i$ es el coste medio de las soluciones construidas con α_i. De este modo, los valores de A que generan soluciones de mayor calidad acumulan progresivamente mayor probabilidad de ser seleccionados, mientras que los que producen soluciones de peor calidad decaen hacia probabilidad nula.

En cada iteración se construye una solución completa mediante un procedimiento de inserción secuencial aleatorizado; es decir, en cada paso se evalúan todos los nodos aún no asignados y todas las posiciones factibles disponibles dentro de trayectos existentes, como nuevo trayecto en una ruta existente o como nueva ruta, y se selecciona aleatoriamente un par nodo-posición de entre aquellos cuyo coste de inserción no supera en más de un factor α el coste mínimo observado. Nótese que la selección de las instalaciones intermedias se realiza de forma integrada durante la construcción, eligiendo en cada paso la instalación que minimiza la suma de las distancias de acceso y salida del trayecto. Una vez construida la solución, esta se mejora mediante el VND descrito anteriormente. Finalmente, la mejor solución obtenida a lo largo de todas las iteraciones del R-GRASP se toma como solución inicial para la fase del S-ALNS.

3.2. ALNS inteligente

La fase de intensificación y diversificación se lleva a cabo mediante una variante extendida del ALNS [11] cuyo espacio de decisión son las rutas del primer nivel, aunque, como se describe a continuación, las decisiones de búsqueda están permanentemente informadas por el segundo nivel. El algoritmo alterna iterativamente entre la destrucción parcial de la solución actual y su reparación, aceptando o rechazando la solución candidata según un criterio de *simulated annealing* con enfriamiento geométrico. Esto es, la temperatura T se inicializa en T_0 y decrece hasta T_f a lo largo de las iteraciones, de modo que una solución candidata de coste superior se acepta con probabilidad $e^{-\Delta c/T}$, donde Δc es la diferencia de coste entre ambas soluciones.

Se emplean cuatro operadores de destrucción: *Random Removal* elimina un subconjunto de n nodos elegidos uniformemente al azar; *Worst Removal* selecciona los n nodos cuya eliminación produce la mayor reducción en el coste; *Related Removal*, inspirada en la eliminación de Shaw [13], extrae n nodos geográficamente próximos entre sí; y *TV Destroy*, exclusivo del S-ALNS, identifica el viaje TV de mayor ineficiencia, medida por el cociente entre el tiempo de desplazamiento y la carga transportada, y extrae los nodos de las rutas de CV que depositan carga en

las instalaciones visitadas por ese viaje, explotando la estructura de dos niveles para orientar la destrucción hacia las regiones de mayor potencial de mejora conjunta. En todos los casos, n se muestrea uniformemente en $[n_{\text{mín}}, n_{\text{máx}}]$.

Por otro lado, para la reparación se dispone de cuatro operadores, dos clásicos y sus variantes con memoria: *Greedy Insertion* reinserta cada nodo en la posición factible de menor coste en orden aleatorio; mientras que *Regret-2 Insertion* prioriza en cada paso el nodo con mayor diferencia entre su mejor y su segunda mejor posición de inserción. Sus variantes con memoria, *Memory Greedy Insertion* y *Memory Regret-2 Insertion*, modifican el criterio de selección de IF explotando el hecho de que las soluciones de élite tienden a compartir patrones de asignación CV–IF estables. En el operador estándar, se elige la IF f que minimiza el coste de arco $c(f)$, definido como la suma de las distancias desde el último nodo del viaje hasta f y desde f hasta el nodo siguiente. Estas variantes sustituyen este coste por uno ajustado que descuenta un bonus de familiaridad:

$$\tilde{c}(f, \mathcal{N}_t) = c(f) - \frac{\beta}{\sqrt{|\mathcal{N}_t|}} \sum_{n \in \mathcal{N}_t} F_{nf},$$

donde $\mathcal{N}_t$ es el conjunto de nodos del viaje en curso, F_{nf} contabiliza cuántas veces el nodo n ha sido servido desde la instalación f en soluciones que establecieron una nueva mejor solución global y $\beta \geq 0$ controla la intensidad del sesgo. Con $\beta = 0$ se recupera la selección por coste puro, mientras que la normalización por $\sqrt{|\mathcal{N}_t|}$ evita que viajes con muchos nodos acumulen una ventaja desproporcionada sobre los de menor tamaño.

Además, en lugar de la selección adaptativa mediante ruleta del ALNS clásico, cuya asignación de pesos es independiente del estado de la búsqueda, el S-ALNS centraliza la estrategia a escala macroscópica delegando la elección del par a de operadores de destrucción y reparación en un agente de *Q-learning* tabular [16] con política ε-voraz; decisión de diseño motivada por el hecho de que el rendimiento relativo de los operadores varía significativamente según la factibilidad del plan de TV y la trayectoria de búsqueda, dimensiones que la ruleta, al operar sin estado, no puede capturar. De este modo, la toma de decisiones del agente se desacopla por completo de la micro-configuración estructural interna de las rutas, abstrayendo el estado de la búsqueda mediante el producto cartesiano de dos dimensiones ortogonales. La dimensión de factibilidad comprende las categorías en que la solución actual posee o no un plan de TV factible, denominadas *TVFeasible* y *TVInfeasible*; por su parte, la dimensión de trayectoria mide la velocidad de la búsqueda mediante tres componentes determinados por la evolución del coste y un contador de estancamiento frente a un umbral fijo: *Improving*, donde el último movimiento aceptado mejoró el coste de la solución actual; *Neutral*, que actúa como un amortiguador de exploración al reconocer movimientos aceptados que no mejoran el coste de la solución actual pero que aún no alcanzan el límite de inactividad; y *Stagnant*, cuando la búsqueda no registra solución aceptada durante el número prefijado de iteraciones consecutivas. Los valores $Q(e, a)$ se almacenan en una matriz inicializada a cero, y se actualizan tras cada iteración

mediante la regla de Bellman de un paso:

$$Q(e,a) \leftarrow Q(e,a) + \theta\big[r + \gamma \max_{a'} Q(e',a') - Q(e,a)\big],$$

donde $\theta \in (0,1]$ es la tasa de aprendizaje, $\gamma \in [0,1]$ el factor de descuento, e' el estado siguiente, y r la recompensa devuelta al agente que refleja directamente el resultado del movimiento según una jerarquía de cuatro valores: el máximo cuando la candidata establece una nueva mejor solución global con plan TV factible; decrecientes para candidata aceptada con mejora de coste, aceptada sin mejora, y rechazada, respectivamente.

Asimismo, las transiciones que conducen al estado *TVInfeasible* reciben una penalización adicional r_{TV} proporcional al grado de infactibilidad:

$$r_{\mathrm{TV}} = \rho_0 + \rho_1 \cdot \frac{\max_k\big(t_k - t'_{\max}\big)}{t'_{\max}},$$

donde t_k es el tiempo total de desplazamiento del TV k, $t'_{\max}$ es el horizonte de planificación, y ρ_0, ρ_1 controlan la penalización base y su escalado con el exceso relativo, respectivamente. Nótese que con $\rho_1 = 0$ la penalización se reduce a la constante ρ_0.

Finalmente, la tasa de exploración del agente, ε, se inicializa en ε_0 y converge a ε_f de forma multiplicativa a lo largo de las $I^{\mathrm{A}}_{\max}$ iteraciones máximas fijadas mediante el factor $\delta_\varepsilon = (\varepsilon_f/\varepsilon_0)^{1/I^{\mathrm{A}}_{\max}}$.

Algoritmo 1: Smart Adaptive Large Neighborhood Search (S-ALNS)

Entrada: $\mathcal{I}$, instancia; parámetros R-GRASP y S-ALNS.
Salida: S^*, mejor solución VT-viable.

para 1 **hasta** $I^{\mathrm{G}}_{\max}$ **hacer**
  $S \leftarrow$ `VND(R-GRASP(`$\mathcal{I}, \alpha_i$`))`;
  Actualizar S^* y las probabilidades de selección de α_i;
$S \leftarrow S^*$;
Inicializar Q, F, ε, T;
para 1 **hasta** $I^{\mathrm{A}}_{\max}$ **hacer**
  $a = (a_1, a_2) \leftarrow \varepsilon$-voraz($Q$);
  $S' \leftarrow$ `VND(Reparar(Destruir(`S, a_1`),` a_2, F`))`;
  si S' *es TV-factible* $\wedge\ c(S') < c(S^*)$ **entonces**
    $S^* \leftarrow S'$;
    Actualizar F;
  Actualizar Q;
  si Aceptar(S', S, T) **entonces** $S \leftarrow S'$;
  Actualizar ε, T;
devolver S^*;

Una solución candidata solo puede reemplazar a la mejor solución global si dispone de planificación de TV factible y su coste es inferior. La planificación de

TV calculada por *TV Destroy* re reutiliza en este paso, evitando un cómputo redundante. Asimismo, el criterio de *simulated annealing* opera sobre el coste de la solución cuando tanto la solución actual como la candidata disponen de planificación de TV factible; en caso contrario, recae sobre el coste del primer nivel, corrigiendo el sesgo que rechazaría movimientos neutros en coste del primer nivel pero beneficiosos en el segundo nivel.

4. Resultados computacionales

Para validar el algoritmo propuesto, los experimentos se llevaron a cabo en un sistema Windows de 64 bits con procesador Intel Core i5-12500T (2 GHz) y 16 GB de memoria RAM, comparando el S-ALNS con Gurobi 11.0.3 [6] en Python sobre instancias de pequeño tamaño. Para ello, Gurobi resolvió una formulación matemática exacta del problema cuyos detalles teóricos se han omitido en esta ocasión por motivos de espacio.

Las instancias están fundamentadas en datos operativos reales de la ciudad de Saltillo, México. Los nodos corresponden a asentamientos urbanos cuyas coordenadas provienen de registros georreferenciados; la demanda y el tiempo de servicio se derivan de datos censales de población. Las distancias se calculan geodésicamente mediante la fórmula de Haversine, asumiendo una velocidad media de 60 km/h para los tiempos de viaje. La red comprende dos IF y un vertedero, fijos en todas las instancias. Se consideran tres distribuciones espaciales: aleatoria, agrupada y mixta; y seis tamaños de instancia: 10, 15, 20, 30, 50, 80; con cinco réplicas por combinación, obteniendo así un total de 90 instancias en total. La flota de CV es heterogénea, con capacidades de 8, 9 y 12 toneladas constantes en todas las instancias; la flota de TV es homogénea con capacidad de 30 toneladas. La duración máxima de ruta se fija en 480 minutos para ambos niveles, con un máximo de dos visitas por CV y TV a cada IF.

Además, se realizaron pruebas estadísticas no paramétricas para demostrar que las diferencias de rendimiento observadas son estadísticamente significativas. En particular, se aplicó un test de Wilcoxon de rangos con signo unilateral sobre las diferencias entre el mejor coste obtenido por el S-ALNS en cada ejecución y el valor objetivo correspondiente del solver exacto, formulándose la hipótesis direccional de que el S-ALNS produce soluciones de menor coste. El resultado fue estadísticamente significativo ya que existen diferencias significativas a favor de S-ALNS al 95 % con $p = 0{,}031$, lo que evidencia que el S-ALNS obtiene sistemáticamente soluciones de menor coste que Gurobi.

Dado que Gurobi no es capaz de resolver instancias de gran tamaño en un tiempo razonable con suficiente calidad de solución, la Tabla 1 presenta resultados agregados por distribución y tamaño para instancias pequeñas de $n \geq 20$ nodos, reportando la media aritmética sobre grupos de cinco instancias. Las dos primeras columnas indican las características de las instancias agregadas; las siguientes cuatro reportan el valor objetivo (*obj.*), el *gap* porcentual (*gap*), el tiempo de ejecución en segundos (t) y el número de óptimos confirmados (k) por Gurobi. Por su parte, las últimas siete columnas detallan los resultados de S-ALNS,

Tabla 1: Comparación entre Gurobi y S-ALNS en instancias de menor tamaño.

		Gurobi				S-ALNS						
Distribución	n	$\overline{Obj.}$	$\overline{Gap}$(%)	$\bar{t}$ (s)	k	$\overline{Obj.}$	$\overline{Dev.}$(%)	$\bar{\mu}$	$\bar{\sigma}$	$\bar{\tau}$ (s)	$\bar{t}$ (s)	$\hat{k}$
Agrupada	10	49,88	5,71	6018,51	3	48,46	−2,85	48,46	0,00	1,81	8,27	3
	15	90,70	17,25	14215,40	1	85,68	−5,53	85,72	0,08	6,37	16,34	1
	20	81,06	20,69	13717,03	1	76,60	−5,50	77,00	0,43	20,58	28,24	1
Mixta	10	68,99	10,61	8778,49	2	67,62	−1,99	68,24	1,00	3,59	7,51	2
	15	120,95	18,79	14402,43	0	119,46	−1,23	120,44	0,92	9,83	12,01	0
	20	74,02	15,04	12201,58	2	72,59	−1,93	72,93	0,31	5,83	25,90	2
Aleatoria	10	67,22	3,33	3704,97	4	65,94	−1,90	65,94	0,00	1,51	7,00	4
	15	83,67	12,15	12196,46	1	82,47	−1,43	82,47	0,00	7,34	13,33	1
	20	75,99	19,32	14463,60	0	75,02	−1,27	75,57	0,49	10,71	24,45	0

atendiendo a que cada instancia se ejecutó cinco veces con un criterio de parada de 1000 iteraciones por ejecución; de este modo, se reporta el mejor valor objetivo obtenido (*obj.*), la desviación porcentual respecto a Gurobi (*dev.*), la media (μ) y la desviación estándar (σ) de las cinco réplicas, el tiempo invertido en hallar la mejor solución (τ), el tiempo total de ejecución (t), ambos en segundos, y el número de óptimos alcanzados ($\hat{k}$).

Obsérvese que, si bien Gurobi agota el límite de tiempo de 14000 s en la mayoría de los casos, con brechas de optimalidad que oscilan entre el 3,33 % y el 10,61 % para $n = 10$, el 12,15–18,79 % para $n = 15$, y hasta el 20,69 % para $n = 20$, logra certificar la optimalidad en 14 de las configuraciones. Por su parte, el S-ALNS iguala el rendimiento de Gurobi en dichos óptimos certificados y lo supera en calidad en el resto, logrando desviaciones negativas entre el −1,33 % y el −5,53 %, alta estabilidad entre ejecuciones, dada la desviación estándar inferior a 1,00, y tiempos medios de 7 a 28,24 s, varios órdenes de magnitud inferiores a los de Gurobi, a la vez que consigue alcanzar los óptimos confirmados por Gurobi.

La Figura 2 presenta el estudio de ablación del S-ALNS en tres dimensiones: mejora relativa frente a R-GRASP, mejora relativa frente a ALNS, y mejora de tiempo de cómputo respecto a ALNS. En términos de calidad de solución, con respecto a todos los tipos de distribución de nodos, se supera consistentemente a R-GRASP, con ganancias que se acentúan a medida que crece el tamaño de instancia, alcanzando hasta un 12 % para instancias de 50 nodos. Frente a ALNS, las mejoras son más moderadas pero igualmente crecientes con n, situándose en el rango 0,5–2,5 % para instancias grandes, con la variante de mayor rendimiento destacándose de forma sostenida a partir de $n = 30$. Además, la disminución de tiempo de cómputo inducido sobre ALNS se mantiene acotado entre el 4 % y el 20 % en todas las instancias. En conjunto, los resultados validan la contribución de los componentes adicionales del S-ALNS frente a sus referencias, con una relación entre calidad y tiempo particularmente favorable en instancias de mayor tamaño.

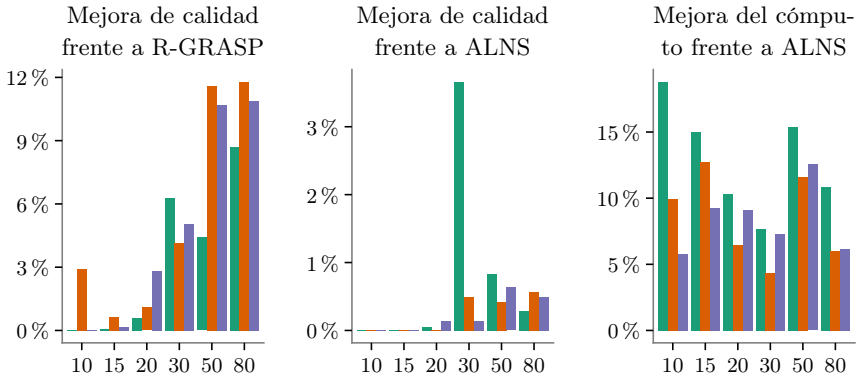


Figura 2: Estudio de ablación de las partes que conforman el S-ALNS, donde el eje x representa el tamaño de instancia n, el eje y el porcentaje de mejora relativa entre métodos, y las instancias se distinguen por tipo de distribución espacial, agrupadas, mixtas y aleatorias.

5. Conclusiones y futuras líneas

En este trabajo se ha propuesto el algoritmo híbrido S-ALNS para el problema de rutas de dos niveles con sincronización exacta de flotas en estaciones de transferencia sin almacenamiento en la recogida de residuos. Frente a las limitaciones de escalabilidad mostradas por Gurobi, el método propuesto, basado en aprendizaje por refuerzo y memoria de asignación, demuestra ser altamente eficaz y robusto al hallar soluciones óptimas en instancias pequeñas y superar drásticamente al *solver* en escenarios de gran tamaño, sin verse afectado por la distribución espacial de los nodos.

Como extensiones futuras, se identifican: el diseño de operadores de búsqueda local coordinados entre niveles; la incorporación de demanda estocástica y tiempos de viaje variables para obtener soluciones robustas; y la exploración de representaciones de estado más ricas y políticas transferibles dentro del esquema de aprendizaje por refuerzo del S-ALNS.

Agradecimientos. Los autores agradecen al Municipio de Saltillo, México, por su valiosa colaboración y por facilitar el acceso a los datos de recolección de residuos sólidos urbanos utilizados en este trabajo. Este trabajo ha sido financiado por la Agencia Estatal de Investigación (España; proyecto PID2019-104410RB-I00/AEI/10.13039/501100011033). Además, el trabajo se inscribe en el marco del contrato de concesión de servicios de gestión de residuos de la isla de Tenerife, entre el Cabildo Insular de Tenerife y la UTE Nivaria, integrada por Urbaser y FCC Medio Ambiente.

Declaración de Intereses. Los autores declaran no tener conflictos de interés que sean relevantes para el contenido de este artículo.

Referencias

1. Beltrami, E.J., Bodin, L.D.: Networks and vehicle routing for municipal waste collection. Networks **4**(1), 65–94 (1974). https://doi.org/10.1002/net.3230040106
2. Drexl, M.: Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints. Transportation Science **46**(3), 297–316 (Aug 2012). https://doi.org/10.1287/trsc.1110.0400
3. Duarte, A., Mladenović, N., Sánchez-Oro, J., ć, R.: Variable Neighborhood Descent, pp. 1–29. Springer Nature Switzerland, Cham (2016). https://doi.org/10.1007/978-3-319-07153-4_9-2
4. Ghiani, G., Manni, A., Manni, E., Moretto, V.: Optimizing a waste collection system with solid waste transfer stations. Computers & Industrial Engineering **161**, 107618 (Nov 2021). https://doi.org/10.1016/j.cie.2021.107618
5. Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.M.: An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. European Journal of Operational Research **254**(1), 80–91 (None 2016). https://doi.org/10.1016/j.ejor.2016.03.040
6. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2024), https://www.gurobi.com
7. Hess, C., Dragomir, A., Doerner, K., Vigo, D.: Waste collection routing: a survey on problems and methods. Central European Journal of Operations Research **32**, 1–36 (Nov 2023). https://doi.org/10.1007/s10100-023-00892-y
8. Pérez-Luis, O.J., Melián-Batista, B., Moreno Vega, J.M., García-Calvillo, I.D.: Instances for the capacitated vehicle routing problem with intermediate facilities (2e-cvrp-if) (2026). https://doi.org/10.5281/zenodo.21277368
9. Prais, M., Ribeiro, C.C.: Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. INFORMS Journal on Computing **12**(3), 164–176 (2000). https://doi.org/10.1287/ijoc.12.3.164.12639
10. Rahmanifar, G., Mohammadi, M., Hajiaghaei-Keshteli, M., Fusco, G., Colombaroni, C.: An integrated temporal and spatial synchronization for two-echelon vehicle routing problem in waste collection system. Journal of Industrial Information Integration **40**, 100611 (2024). https://doi.org/10.1016/j.jii.2024.100611
11. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science **40**(4), 455–472 (2006). https://doi.org/10.1287/trsc.1050.0135
12. Schiffer, M., Schneider, M., Walther, G., Laporte, G.: Vehicle routing and location routing with intermediate stops: A review. Transportation Science **53**(2), 319–343 (Mar 2019). https://doi.org/10.1287/trsc.2018.0836
13. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M., Puget, J.F. (eds.) Principles and Practice of Constraint Programming — CP98. pp. 417–431. Springer Berlin Heidelberg, Berlin, Heidelberg (1998). https://doi.org/10.1007/3-540-49481-2_30
14. Sluijk, N., Florio, A.M., Kinable, J., Dellaert, N., Van Woensel, T.: Two-echelon vehicle routing problems: A literature review. European Journal of Operational Research **304**(3), 865–886 (2023). https://doi.org/10.1016/j.ejor.2022.02.022
15. Soares, R., Marques, A., Amorim, P., Parragh, S.N.: Synchronisation in vehicle routing: Classification schema, modelling framework and literature review. European Journal of Operational Research **313**(3), 817–840 (2024). https://doi.org/10.1016/j.ejor.2023.04.007
16. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning **8**(3), 279–292 (May 1992). https://doi.org/10.1007/BF00992698