

INFORME DE EVALUACION DE TRABAJO FINAL

CODIGO DEL TRABAJO: CO19-C11

Marque la respuesta correcta	SI	NO
Cumple perfectamente con el Formato requerido para la presentación de trabajos Formato requerido: https://drive.google.com/file/d/1N-5uYFYt_Wk1M14VAJmJucUIFFiDvFBd/view?usp=sharing	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Contiene correctamente en su estructura:		
✓ Resumen	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Palabras claves	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Abstract	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Contenido (El trabajo posee un sólido marco teórico que lo avala, es concordante y sustenta el problema investigado.)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Ecuaciones, figuras y/o tablas en caso de ser necesarias	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Conclusiones	<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ Bibliografía	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Nota: es de suma importancia revisar el FORMATO del trabajo ya que dicho documento será parte de una publicación y todos los trabajos deben poseer el mismo formato.

Comentarios del Evaluador

Trabajo corregido.

Resultado de la evaluación

Aprobado = trabajo completo y listo para publicación. El evaluador aprueba los contenidos del trabajo y todas las respuestas del cuadro anterior son afirmativas.

Observado = trabajo incompleto o con observaciones y debe ser corregido por el/los autor/es a efectos de ser sometido nuevamente al proceso de evaluación.

Al recibir las correcciones se debe realizar un nuevo informe de Evaluación de trabajo final.

APROBADO

COMITÉ DE EVALUACIONES

Un estudio comparativo de algoritmos metaheurísticos sobre instancias reales de problemas de recolección de RSU

Fermani, Matías (1° Autor); Rossit, Diego Gabriel (2° Autor); Toncovich, Adrián Andrés (3° Autor)*

(1° Autor) Departamento de Ingeniería, Universidad Nacional del Sur (UNS).

Av. Alem 1253, 8000, Bahía Blanca, Argentina. *mati.ferma@gmail.com*.

(2° Autor) INMABB, Departamento de Ingeniería, Universidad Nacional del Sur (UNS)-CONICET.

Av. Alem 1253, 8000, Bahía Blanca, Argentina. *diego.rossit@uns.edu.ar*.

(3° Autor) *Departamento de Ingeniería, Universidad Nacional del Sur (UNS).

Av. Alem 1253, 8000, Bahía Blanca, Argentina. *atoncovi@uns.edu.ar*.

RESUMEN.

La gestión de la recolección de Residuos Sólidos Urbanos es una tarea compleja que deben enfrentar los gobiernos locales, consumiendo gran parte de su presupuesto. La utilización de herramientas computacionales que sirvan de apoyo a la toma de decisiones puede contribuir a mejorar la eficiencia del sistema y disminuir los costos asociados. En el presente trabajo se propone la evaluación de distintas herramientas informáticas exactas y metaheurísticas sobre casos reales de la ciudad de Bahía Blanca. Particularmente, se propone la utilización de CPLEX para resolver un problema de programación matemática y su comparación con algoritmos metaheurísticos basados en las técnicas de *Simulated Annealing* y *Large Neighborhood Search*. Los resultados muestran que CPLEX encuentra limitaciones para resolver los escenarios más grandes. Por otro lado, las herramientas heurísticas propuestas son competitivas, obteniendo valores cercanos a los métodos exactos con tiempos de cómputo mucho menores. Las herramientas heurísticas también son validadas con respecto a conocidos *benchmarks* de la literatura.

Palabras Claves: *Capacitated Vehicle Routing Problem*, Optimización, Algoritmos Metaheurísticos, *Simulated Annealing*, *Large Neighborhood Search*.

ABSTRACT.

The management of the Municipal Solid Waste collection is a complex task that local governments must face, consuming a large part of their budget. The use of computational tools that support decision-making can contribute to improve the efficiency of the system and reduce the associated costs. This paper proposes the evaluation of different exact and metaheuristic tools on real-world scenarios in the city of Bahía Blanca. In particular, the use of CPLEX is proposed to solve a mathematical programming problem and it is compared with two metaheuristic algorithms based on *Simulated Annealing* and *Large Neighborhood Search* techniques. The results show that the exact tool face limitations to solve the larger scenarios. On the other hand, the proposed heuristic tools are competitive, obtaining values that are close to the exact solution in much smaller computing times. The heuristic tools are also validated with respect to well-known benchmarks of the literature.

Keywords: *Capacitated Vehicle Routing Problem*, Optimization, Metaheuristic Algorithms, *Simulated Annealing*, *Large Neighborhood Search*.

1. INTRODUCCIÓN.

La generación de residuos sólidos urbanos (RSU) es una consecuencia inalienable del desarrollo de las sociedades modernas. Sin embargo, el manejo incorrecto de la creciente cantidad de residuos puede generar problemáticas no sólo ambientales, pero también sociales, como el problema de los recolectores informales [1] y económicos, como los costos de remoción luego de una mala disposición inicial que podrían haberse evitado. Las herramientas de apoyo a la toma de decisiones, como la que se presenta en el presente trabajo, son una estrategia válida para mitigar los efectos nocivos que pueden provenir de una mala gestión de los RSU y disminuir los costos que generan para los municipios, que son los principales encargados de la gestión [2].

Las etapas en la cadena logística inversa de los RSU son diversas. Según [3] ellas pueden clasificarse en generación de residuos; manejo, separación, acumulación y procesamiento de residuos en origen; prosiguiendo la recolección; transferencia y transporte; separación, procesamiento y transformación de residuos sólidos; y disposición final. El presente trabajo se enfoca en la segunda etapa de recolección; transferencia y transporte para casos de estudio del área de Bahía Blanca. El sistema actual de recolección de la ciudad de Bahía Blanca consiste en una recolección puerta a puerta sin ninguna clasificación de residuos en origen.

En trabajos previos ([4], [5]) se avanzó con una propuesta de un sistema de contenedorizado en la ciudad que permitiera, por un lado simplificar la logística de recolección debido a la disminución de puntos a visitar en las rutas de recolección [6], y a su vez permitir una clasificación en origen de los residuos. Particularmente, en dichos trabajos se evaluaron la optimización de la disposición de contenedores en algunos barrios de la ciudad de Bahía Blanca y propuestas iniciales para gestionar la logística del transporte donde se encontraron obstáculos para la resolución de los escenarios mediante algoritmos exactos. Debido a esto algunas regiones debieron particionarse en subregiones para disminuir el tamaño del problema.

El objetivo de este trabajo es avanzar en diseño las rutas de recolección. Para superar los obstáculos encontrados previamente, se trabajó con dos metodologías. Por una parte, se incorporaron inequaciones válidas en la formulación matemática para mejorar el desempeño de la resolución exacta. Por otra parte, se propusieron dos herramientas heurísticas abordar el problema, una basada en el recocido simulado (*Simulated Annealing*, SA) adaptada de los trabajos [7] y [8], y otra basada en la técnica *Large Neighborhood Search* desarrollada por [9]. Se espera que estas técnicas permiten abordar los escenarios reales planteados en los trabajos previos. Para validar las heurísticas se realizan pruebas sobre un conocido conjunto de instancias de la literatura.

1.1. Revisión bibliográfica.

La recolección de RSU en contenedores constituye un caso de aplicación del problema VRP (*Vehicle Routing Problem*). En términos generales, este tipo de problemas consiste en determinar la ruta de mínima distancia que recorra un determinado conjunto de puntos. Su complejidad radica en la gran cantidad de variables de naturaleza entera que contienen, creando así una región factible difícil de evaluar en tiempos razonables, siendo considerados en la literatura como problemas NP-*hard* [10]. En el caso de estudio que se desarrollará en este trabajo, se presenta un problema VRP en el que se incluye una restricción de capacidad para cada vehículo, conocido como *Capacitated VRP* (CVRP).

El problema de recolección de RSU ha sido abordado por numerosos autores con diversas técnicas, tanto por medio de métodos heurísticos como exactos. En [11] y [12] se desarrollan extensas revisiones sobre estos trabajos. En el caso de Argentina pueden encontrarse algunos trabajos sobre aplicaciones de modelos VRP para resolver el problema de diseño de rutas de recolección. Por ejemplo, en [13] y [14] se implementan modelos de programación matemática para planificar las rutas de recolección de contenedores de residuos en la zona sur de la Ciudad de Buenos Aires. Por otro lado, en [6] se trabaja optimizando las rutas en esta zona teniendo como objetivos minimizar las distancias de recorrido simultáneamente a la minimización del desgaste de los vehículos. En la ciudad de Concordia, [15] presenta una aplicación para diseñar las rutas de recolección de la ciudad haciendo un esfuerzo por minimizar la cantidad de giros para facilitar la implementación de las rutas por las autoridades. Por otra parte, [16] exhibe un algoritmo para resolver la zonificación de la ciudad de San Miguel de Tucumán a los efectos de optimizar el uso de recursos, reasignando camiones hacia la zona céntrica de la ciudad. En [17] y [18] mediante modelos de programación entera se planifica la recolección de residuos reciclables en la ciudad de Morón. En [19] se presenta un enfoque integral que determina la localización de los contenedores, así como el diseño de la ruta que efectúa la recolección de su contenido en instancias simuladas de Bahía Blanca. En la misma ciudad, [20] y [21] exponen modelos de ruteo de vehículos con balanceo de distancias entre las distintas rutas para diseñar los recorridos de los recolectores informales de residuos reciclables.

2. MÉTODOS DE RESOLUCIÓN UTILIZADOS.

El problema del tipo CVRP aquí estudiado puede formalizarse a través del siguiente modelo de programación matemática.

Definición de variables:

- K: conjunto finito de viajes a realizar por el camión.
- $i \in I$: nodos que hacen referencia a los contenedores.
- c: capacidad del vehículo.
- m: cantidad de puntos a visitar (excluido el origen ubicado en Av. Parchappe).
- f_i : volumen de residuos secos a recolectar en nodo i .
- d_{ij} : distancia a recorrer del nodo i hasta j .
- x_{ij}^k : variable binaria, 1 si el camión durante el viaje k circula desde i a j , 0 en el otro caso.
- u_i : variable auxiliar continua para la eliminación de *subtours*.

$$\min \sum_{\substack{i,j \in I \\ k \in K}} x_{ij}^k \cdot d_{ij} \quad (1)$$

Sujeto a:

$$\sum_{i,j \in I} x_{ij}^k \cdot f_i \leq c, \forall k \in K \quad (2)$$

$$\sum_{j \in I} x_{ij}^k \leq 1, \forall k \in K \quad (3)$$

$$\sum_{i \in I} x_{i1}^k \geq 1, \forall k \in K \quad (4)$$

$$\sum_{\substack{j \in I \\ k \in K}} x_{ij}^k = 1, \forall i \neq 1 \quad (5)$$

$$\sum_{\substack{i \in I \\ k \in K}} x_{ij}^k = 1, \forall j \neq 1 \quad (6)$$

$$\sum_{i \in I} x_{ij}^k - \sum_{i \in I} x_{ji}^k = 0, \forall k \in K, \forall j \in I \quad (7)$$

$$u_i - u_j + m \cdot x_{ij}^k \leq m - 1, \forall k \in K, \forall i, j \neq 1 \in I \quad (8)$$

$$x_{ij}^k \in \{0,1\}, \forall k \in K, \forall i, j \in I \quad (9)$$

$$u_i \geq 0, \forall i \in I \quad (10)$$

El objetivo planteado consiste en minimizar la distancia total recorrida durante la recolección de los residuos secos y húmedos, tal como lo muestra la Ecuación (1). Por su parte, las restricciones impuestas al modelo se presentan desde la Ecuación (2) hasta la (10).

La Ecuación (2) garantiza que la capacidad del vehículo, en cada viaje, no sea superada. Las Ecuaciones (3) y (4) aseguran que el inicio y el fin de cada viaje tenga lugar en las instalaciones de Bahía Ambiental SAPEM, ubicadas sobre Av. Parchappe N°782. Las Ecuaciones (5) y (6) garantizan que cada punto donde se ubican los contenedores, excepto aquel que oficia de inicio y fin de cada recorrido, sea visitado una única vez. La Ecuación (7) refiere a la conservación del flujo en el recorrido mientras que la Ecuación (8) impide la formación de *subtours*. La Ecuación (9) establece la naturaleza binaria de las variables x_{ij}^k . Por su parte, la Ecuación (10) expresa la naturaleza positiva de la variable u_i .

Luego de pruebas preliminares, al modelo presentado en las Ecuaciones (1)-(10), que fue utilizado en [5], se le adicionaron las expresiones válidas dadas por las Ecuaciones (11) y (12), las cuales han sido frecuentemente utilizadas en la literatura de VRP ([22], [23]). Esto permitió disminuir los tiempos requeridos para obtener las soluciones factibles o, en algunas ejecuciones, obtener el óptimo.

$$\sum_{i \in I} x_{i1}^k \geq \sum_i \frac{f_i}{c}, \forall k \in K, \forall j \in I \quad (11)$$

$$x_{ji}^k + x_{ij}^k = 1, \forall k \in K, \forall i, j \in I \quad (12)$$

La Ecuación (11) fija el mínimo número de viajes necesarios para satisfacer la demanda. La Ecuación (12) establece que, si un arco es atravesado en un sentido por un vehículo, este no debe ser atravesado en sentido contrario. La formulación matemática fue implementada y resuelta mediante el software CPLEX.

2.1 Metaheurísticas utilizadas.

En cuanto a las metaheurísticas propuestas, se utilizaron dos procedimientos distintos. El primero es un algoritmo que responde a una estrategia de recocido simulado (SA, *Simulated Annealing* en inglés) y fue adaptada de los trabajos [7] y [8]. SA es un método basado en la búsqueda local que se desarrolló a partir de una analogía con el fenómeno físico de recocido [24] para resolver problemas complejos de optimización. Los métodos de búsqueda local buscan la solución con un mejor valor del criterio elegido en el entorno de la solución actual, la aceptan como la solución actual y repiten este procedimiento hasta que no resulte posible mejorar la solución en el entorno explorado. Mediante la aplicación sistemática de este procedimiento se obtiene, en general, un óptimo local para el problema. Para evitar quedar atrapado en un óptimo local, se debe incorporar un mecanismo de diversificación con el fin de explorar adecuadamente el espacio de soluciones. En la metaheurística de recocido simulado, la estrategia de diversificación permite movimientos, con cierta probabilidad, hacia soluciones que empeoran el valor actual de la función objetivo. Para obtener una buena aproximación a la solución óptima del problema durante el proceso de búsqueda, resulta necesario reiniciar la búsqueda regularmente desde una de las soluciones aceptadas durante el proceso de búsqueda seleccionada al azar. El algoritmo SA incorpora los parámetros clásicos del recocido simulado. Seguidamente se indican los parámetros y variables correspondientes al algoritmo SA implementado:

- t: Iteración actual.
- S₀: Solución inicial.
- S_A: Solución actual.
- S_C: Solución candidata.
- V(S): Entorno de la solución S, dado por el conjunto de soluciones que pueden obtenerse a partir de S mediante un movimiento o perturbación elemental.
- T: Parámetro de control, que simula la temperatura en el proceso metalúrgico del recocido), valor positivo que varía desde un valor inicial mayor, T₀, a otro que es menor, T_f, durante la ejecución del algoritmo.
- N_T: Número de iteraciones realizadas por el algoritmo para cierto valor de T.
- α: Función en T, α = α(T), que determina la variación de T. En general α(T) = α T, en la práctica α ∈ [0,8;0,99].
- N_{cont}: Número de iteraciones sin mejora en la iteración t.
- N_{stop}: Número máximo de iteraciones permitidas sin mejora.

Se aplica el siguiente pseudo-código para determinar una solución potencialmente óptima.

i. Inicio

Se utiliza un procedimiento constructivo aleatorizado para generar una solución inicial, S₀.
 Se evalúa la función objetivo, FO(S₀), empleando la Ecuación (1).
 S₀ se adopta como solución actual S_A, N_{cont} = t = 0, T = T₀.

ii. Iteración t

Se genera aleatoriamente una solución candidata en el entorno de S_A, S_C ∈ V(S_A).

Se evalúa FO(S_C).

Se calcula ΔFO = FO(S_C) - FO(S_A).

Si ΔFO ≤ 0, la nueva solución es aceptada: S_A ← S_C, N_{cont} = 0.

En otro caso, S_C es aceptada con la siguiente probabilidad PA = e^{-ΔFO/T}.

Se genera un número aleatorio ξ uniformemente distribuido en el intervalo [0;1]:

$$\begin{cases} \text{si } \xi \leq PA, S_A \leftarrow S_C, N_{\text{cont}} = 0, \\ \text{si } \xi > PA, S_A \leftarrow S_A, N_{\text{cont}} = N_{\text{cont}} + 1. \end{cases}$$

t ← t + 1: Si t es un múltiplo de N_T, entonces T = α T, en otro caso el valor de T se mantiene. Si N_{cont} = N_{stop} o T < T_f, se detiene la ejecución, en otro caso se continúa la ejecución.

La otra metaheurística utilizada es la desarrollada por Erdoğan [9]. Este algoritmo fue adaptado a partir de la heurística conocida como Adaptive Large Neighborhood Search (ALNS) desarrollada por Pisinger y Ropke [25] que extiende la heurística original de Shaw [26]. Una explicación simplificada de su funcionamiento y pseudo-código (Figura 1) puede consultarse en [27].

El funcionamiento de la heurística está constituido por dos etapas principales. La primera, es la construcción de una solución inicial que se forma insertando clientes en las rutas disponibles de acuerdo con el objetivo a minimizar (operador *Sol-Inicial*). Luego, se realiza una mejora mediante el operador *Local-Search* que aplica cuatro técnicas de búsqueda local. Las primeras tres son conocidas como *Exchange* (se intercambian dos clientes), *1-OPT* (se extrae un cliente de una ruta y reinserta en otra) y *2-OPT* (se intercambian dos segmentos de rutas). Estos operadores se encuentran detallados en [28]. El cuarto operador de búsqueda local es *Vehicle-Exchange*. Este operador intenta intercambiar los vehículos asignados a dos rutas, por lo cual es útil cuando se trabaja con flota heterogénea.

La segunda etapa es donde se intenta un mejoramiento de la solución inicial. Se aplica, el operador *Destroy-And-Rebuild* que altera de manera considerable la solución actual para explorar otra región del espacio de búsqueda y así poder escapar de óptimos locales. Particularmente, se extraen tramos extensos de rutas de la solución actual y los clientes extraídos vuelven a reinsertarse en aquellas posiciones que minimicen la función objetivo (*Cost*). Una solución es aceptada si tiene un mejor costo que la mejor solución hallada hasta el momento o, en caso de que una solución no tenga mejor costo que la solución anterior, puede incluso ser aceptada con una probabilidad *p-value*. Este proceso se repite cíclicamente hasta el límite temporal impuesto por el usuario.

```

1: procedure LNS(input)           ▷ Como input se ingresa la información
   correspondiente a los clientes, vehículos y lugares de descarga (depots)
2:    $S \leftarrow SolInicial()$ 
3:    $S' \leftarrow LocalSearch(S)$ 
4:    $BestS \leftarrow S'$ 
5:   repeat
6:      $S' \leftarrow DestroyAndRebuild(S')$ 
7:      $S' \leftarrow LocalSearch(S')$ 
8:     if ( $Cost(S') \leq Cost(S')$ ) then
9:        $BestS \leftarrow S'$ 
10:    else
11:       $p \leftarrow Random(0, 1)$ 
12:      if ( $p \leq pvalue$ ) then
13:         $S' \leftarrow BestS$ 
14:      end if
15:    end if
16:  until  $Time\ elapsed > Time\ limit$ 
17:  return  $BestS$ 
18: end procedure

```

Figura 1 Algoritmo ALNS implementado en [9]. Fuente de la imagen: [27].

3.RESULTADOS.

Los resultados consistieron en la resolución de cuatro escenarios reales de la logística de recolección de RSU de Bahía Blanca construidos en los trabajos de [4] y [5]. Asimismo, la experimentación correspondiente a los enfoques metaheurísticos fue ampliada con la resolución de dos conocidos *benchmarks* de la literatura.

3.1 Pruebas realizadas sobre escenarios reales.

Los escenarios planteados en [4] y [5] constituyen casos reales diseñados en función del problema de RSU existente en la ciudad de Bahía Blanca.

Se estudian dos planificaciones distintas para la disposición de los contenedores, considerando la recolección de ambas fracciones de residuos: tanto los secos como los húmedos. Dichas planificaciones contemplan la opción de separar los contenedores en residuos húmedos y secos a la vez que consideran la opción de utilizar diferentes frecuencias de recolección. Las planificaciones difieren fundamentalmente en cuanto a la frecuencia con la cual se plantea recolectar los residuos secos, conformando de esta manera los siguientes cuatro escenarios:

Escenario 1:

- Frecuencia de recolección [veces/semana]: 6.
- Tipo de residuo a recolectar: húmedo.

Escenario 2:

- Frecuencia de recolección [veces/semana]: 4.
- Tipo de residuo a recolectar: seco.

Escenario 3:

- Frecuencia de recolección [veces/semana]: 6.

- Tipo de residuo a recolectar: húmedo.
- Escenario 4:
- Frecuencia de recolección [veces/semana]: 3.
 - Tipo de residuo a recolectar: seco.

Resulta necesario aclarar que las diferencias indicadas entre los distintos escenarios repercuten esencialmente en la cantidad de contenedores que se utilizarán, sin embargo, la explicación de este hecho no forma parte de los objetivos planteados para este trabajo.

La resolución exacta se implementó mediante el software CPLEX versión 12.6.0.0 en un entorno de GAMS, utilizando la versión de paralelismo oportunista de dicho software. Las ejecuciones fueron realizadas en un computador personal con un procesador Intel Core i7-4790 @ 3,60GHz con 32 GB de memoria RAM con sistema operativo Windows 10. Se fijó un límite de tiempo para cada ejecución de 15 000 segundos.

El algoritmo de *Simulated Annealing* fue programado mediante la interfaz de Visual Basic for Applications (VBA) de MSeExcel y se ejecutó en una PC con procesador i7-260M CPU @ 2,7GHz y 8,00 GB de memoria RAM con un sistema de operativo de 64 bits. En tanto que para la resolución de los escenarios mediante ALNS fue utilizado el mismo ordenador que para la resolución exacta descripto previamente.

Los resultados obtenidos se exponen a continuación en las Tablas 1 y 2. En la primera se puede evaluar la distancia obtenida para cada escenario, como así también sus respectivos errores (GAP), y en la segunda tabla se ven reflejados los tiempos insumidos, expresados en segundos, para llegar a la solución, al igual que la cantidad de iteraciones realizadas.

Tabla 1 *Resultados de la resolución de los escenarios reales.*

Instancia	Planteo exacto		SA		ALNS	
	Dist. óptima	Nº de rutas óptimo	Dist. SA	GAP*	Dist. ALNS	GAP*
Escenario 1	22613,9	3	23317,14	0,031096	23268,51	0,028946
Escenario 2	41533,28**	8	44156,44	0,063158	44194,28	0,064069
Escenario 3	27150	3	27829,61	0,025032	27728,61	0,021312
Escenario 4	62800,63**	12	66281,93	0,055434	65944,34	0,050059

*El GAP fue calculado como la diferencia entre los valores del planteo exacto y el heurístico dividida por el valor del planteo exacto.

**Debido a que no se encontró la solución óptima para estos escenarios, los valores corresponden al mejor valor posible estimado por CPLEX (*Upper Bound*).

Tabla 2 *Esfuerzo computacional requerido para la resolución de los escenarios reales.*

Instancia	CPLEX		SA		ALNS	
	Tiempo	Iteraciones	Tiempo (s)	Iteraciones	Tiempo (s)	Iteraciones
Escenario 1	1871,08	2506415	1223,55	10090887	1260	2798
Escenario 2	15000,00	12309162	1226,28	9429109	1260	4979
Escenario 3	4344,25	13475167	1411,52	10053621	1440	2110
Escenario 4	15000,00	5495456	1053,68	8910059	1440	3930

3.2. Pruebas realizadas sobre *benchmarks* de la literatura.

Para la validación de los resultados de las metaheurísticas fueron utilizados dos conjuntos de *benchmarks* ampliamente utilizados en la literatura, el *benchmark* propuesto en [29], denominado Conjunto E, y el *benchmark* propuesto en [30], denominado Conjunto M. Tanto las instancias como las mejores soluciones encontradas fueron descargadas del repositorio digital VRP-REP [31]. En la Tabla 3 se reflejan los rendimientos tanto de SA como de ALNS comparados contra la solución óptima conocida mediante el GAP.

A partir del análisis de los resultados se verifica que ambos procedimientos metaheurísticos resultan válidos para utilizarlos en la resolución de los problemas del tipo CVRP. A modo de evaluar su desempeño se registró la información correspondiente al tiempo, medido en segundos, que requirieron para resolver cada instancia, junto con el número de iteraciones que se realizaron (Tabla 4).

3. CONCLUSIONES.

Encontrar nuevas herramientas para volver más eficiente la logística de residuos sólidos urbanos (RSU) constituye una inquietud acuciante en las sociedades actuales. Este trabajo está centrado en la resolución de problemas de recolección de residuos para el área de Bahía Blanca. Particularmente se aborda una problemática encontrada en trabajos previos donde la resolución exacta de problemas de recolección de residuos resultaba ineficiente, debido a que se requería la

partición de los escenarios en porciones más pequeñas para lograr la convergencia a una solución factible aceptable.

Por ello, en este trabajo se presenta un estudio comparativo de tres métodos de resolución para el problema de la recolección de RSU en Bahía Blanca. Se aplican métodos de resolución exactos y heurísticos. La resolución exacta está basada en una formulación de programación matemática del modelo CVRP extendido con la incorporación de inecuaciones válidas que se implementa computacionalmente mediante el software CPLEX. Por otro lado, en lo que concierne a la resolución metaheurística se utilizan dos algoritmos, un algoritmo basado en el recocido simulado y otro tomado de la literatura basado en Large Neighborhood Search. Se encuentra que la resolución exacta con la incorporación de las inecuaciones válidas resulta más eficiente, pudiendo hallar una solución para todos los escenarios propuestos. Los dos métodos metaheurísticos también resuelven eficientemente las instancias, obteniendo valores cercanos a los obtenidos por CPLEX.

Tabla 3 Resultados de la experimentación realizada sobre los benchmarks.

Instancia	Solución óptima		SA		ALNS	
	Dist. óptima	Nº de rutas óptimo	Dist. SA	GAP*	Dist. ALNS	GAP*
E n13 k4	247	4	247	0	247	0
E n22 k4	375	4	382	0,018667	375	0
E n23 k3	569	3	569	0	619	0,087873
E n30 k3	534	3	534	0	534	0
E n31 k7	379	7	379	0	379	0
E n33 k4	835	4	837	0,002395	835	0
E n51 k5	521	5	521	0	521	0
E n76 k7	682	7	683	0,001467	692	0,014663
E n76 k8	735	8	739	0,005442	744	0,012245
E n76 k10	830	10	830	0	837	0,008434
E n76 k14	1021	14	1028	0,006856	1033	0,011753
E n101 k8	815	8	820	0,006135	827	0,014724
E n101 k14	1067	14	1082	0,014058	1084	0,015933
M n101 k10	820	10	820	0	865	0,054878
M n 121 k7	1034	7	1035	0,000967	1068	0,032882
M n151 k12	1015	12	1033	0,017734	1038	0,022660
M n200 k16	1274	16	1324	0,039246	1394	0,094192
M n200 k17	1275	17	1315	0,031373	1323	0,037647

*El GAP fue calculado como la diferencia entre los valores óptimo y heurístico dividida por el valor óptimo.

Tabla 4 Esfuerzo computacional requerido para la resolución de los benchmarks.

Instancia	SA		ALNS	
	Tiempo (s)	Iteraciones	Tiempo (s)	Iteraciones
E n13 k4	16,13	105817	234	85151
E n22 k4	24,60	165570	396	40246
E n23 k3	33,24	192076	414	26617
E n30 k3	373,48	3014484	540	13926
E n31 k7	73,34	579861	558	24155
E n33 k4	565,35	3852523	594	10760
E n51 k5	885,12	6808167	918	6847
E n76 k7	1341,16	9000600	1368	5262
E n76 k8	1331,62	11631256	1368	4183
E n76 k10	1333,98	10038124	1368	2768
E n76 k14	1347,98	6602943	1368	2098
E n101 k8	1734,77	9566979	1818	2514
E n101 k14	1785,07	15644844	1818	2263
M n101 k10	1781,86	15459233	1818	2105
M n 121 k7	2141,13	19895356	2178	352
M n151 k12	2691,35	24746588	2718	1035
M n200 k16	3582,00	33640699	3600	95
M n200 k17	3582,00	30403847	3600	294

Por otro lado, se extiende la experimentación de las metaheurísticas con *benchmarks* conocidos de la literatura. Se corrobora la potencialidad de ambos algoritmos, los cuales llegaron en varias

ocasiones a la solución óptima o, en su defecto, obtienen soluciones con un error relativo considerablemente reducido.

Luego del trabajo experimental realizado, se puede afirmar que la aplicación de estos algoritmos redanda en la obtención de resultados potencialmente aceptables y competentes, contando las metaheurísticas con la ventaja del menor esfuerzo de cómputo requerido para llegar a la solución. Como líneas de trabajo a futuro se tiene previsto experimentar con escenarios de mayor tamaño de la ciudad Bahía Blanca, utilizando las metaheurísticas propuestas, que fueron validadas en el presente informe. Por otro lado, se propone continuar mejorando la formulación de programación matemática, implementada a través de CPLEX, mediante la inclusión de inecuaciones válidas adicionales que permitan reducir los tiempos de resolución.

4. REFERENCIAS.

- [1] Wilson, D. C., Velis, C.; Cheeseman, C. (2006). "Role of informal sector recycling in waste management in developing countries". *Habitat International*. Vol. 30, no. 4, págs. 797-808.
- [2] Hoornweg, D.; Bhada-Tata, P. (2012). "What a waste: a global review of solid waste management". *World Bank*. Vol. 15, pág. 116. Washington DC, EUA.
- [3] Tchobanoglous, G., Kreith, F.; Williams, M. E. (2002). "Introduction". En: *Handbook of solid waste management*. G. Tchobanoglous, F. Kreith (Editores). McGraw-Hill, EUA. Segunda edición. Capítulo 1.
- [4] Herran Symonds, V. (2019). *Ubicación de contenedores diferenciados de RSU*. Tesis Final de Carrera. Universidad Nacional del Sur. Bahía Blanca, Argentina.
- [5] Signorelli Nuñez, M. (2019). *Análisis del sistema actual de recolección de RSU en el Barrio Universitario de la ciudad de Bahía Blanca*. Tesis Final de Carrera. Universidad Nacional del Sur. Bahía Blanca, Argentina.
- [6] Bonomo, F., Durán, G., Larumbe, F.; Marengo, J. (2012). "A method for optimizing waste collection using mathematical programming: a Buenos Aires case study". *Waste Management & Research*. Vol. 30, no. 3, págs. 311-324.
- [7] Toncovich, A., Burgos, T.; Jalif, M. (2017). "Planificación de la logística de recolección de miel en una empresa apícola". *Actas del X Congreso Argentino de Ingeniería Industrial*. Ciudad Autónoma de Buenos Aires, Argentina. Págs. 397-406.
- [8] Toncovich, A., Rossit, D. A., Frutos, M.; Rossit, D. G. (2019). "Solving a multi-objective manufacturing cell scheduling problem with the consideration of warehouses using a Simulated Annealing based procedure". *International Journal of Industrial Engineering Computations*. Vol. 10, no. 1, págs. 1-16.
- [9] Erdoğan, G. (2017). "An open source spreadsheet solver for vehicle routing problems". *Computers & Operations Research*. Vol. 84, págs. 62-72.
- [10] Lenstra, J. K.; Kan, A. R. (1981). "Complexity of vehicle routing and scheduling problems". *Networks*. Vol. 11, no. 2, págs. 221-227.
- [11] Beliën, J., De Boeck, L.; Van Ackere, J. (2012). "Municipal solid waste collection and management problems: a literature review". *Transportation Science*. Vol. 48, no. 1, págs. 78-102.
- [12] Han, H.; Ponce Cueto, E. (2015). "Waste collection vehicle routing problem: literature review". *PROMET - Traffic & Transportation*. Vol. 7, no. 4, págs. 345-358.
- [13] Bonomo, F., Durán, G., Larumbe, F.; Marengo, J. (2009). "Optimización de la Recolección de Residuos en la Zona Sur de la Ciudad de Buenos Aires". *Revista de Ingeniería de Sistemas*. Vol. 23.
- [14] Larrumbe, F. (2009). *Optimización de la Recolección de Residuos en la Zona Sur de la Ciudad de Buenos Aires*. Tesis de Grado. Universidad de Buenos Aires. Buenos Aires, Argentina.
- [15] Bertero, F. (2015). *Optimización de recorridos en ciudades. Una aplicación al sistema de recolección de residuos sólidos urbanos en el Municipio de Concordia*. Tesina de Grado. Universidad Nacional de Rosario. Rosario, Argentina.
- [16] Bianchetti, M. L. (2015). *Algoritmos de zonificación para recolección de residuos*. Tesis de Grado. Universidad de Buenos Aires. Buenos Aires, Argentina.
- [17] Braier, G., Durán, G., Marengo, J.; Wesner, F. (2015). "Una aplicación del problema del cartero rural a la recolección de residuos reciclables en Argentina". *Revista de Ingeniería de Sistemas*. Vol. 29.
- [18] Braier, G., Durán, G., Marengo, J.; Wesner, F. (2017). "An integer programming approach to a real-world recyclable waste collection problem in Argentina". *Waste Management & Research*. Vol. 35, no. 5, págs. 525-533.
- [19] Rossit, D. G., Broz, D., Rossit, D. A., Frutos, M.; Tohmé, F. (2015). "Modelado de una red urbana de recolección de residuos plásticos en base a optimización multi-objetivo". *Actas de la XXVI EPIO y VIII RED-M*. Bahía Blanca, Argentina.

- [20] Cavallin, A., Vigier, H. P.; Frutos, M. (2015a). "Aplicación de un modelo CVRP-RB a un caso de logística inversa". *Actas de la XXVI Escuela de Perfeccionamiento en Investigación Operativa (EPIO) y VIII RED-M*. Bahía Blanca, Argentina.
- [21] Cavallin, A., Vigier, H. P.; Frutos, M. (2015b). "Logística inversa y ruteo en el sector de recolección informal de residuos sólidos urbanos". En: *Avances en Gestión Integral de Residuos Sólidos Urbanos 2014-15*. Instituto Nacional de Tecnología Industrial, Buenos Aires, Argentina. Págs. 37-49.
- [22] Achuthan, N. R., Caccetta, L.; Hill, S. P. (2003). "An improved branch-and-cut algorithm for the capacitated vehicle routing problem". *Transportation Science*. Vol. 37, no. 2, págs. 153-169.
- [23] Dror, M., Laporte, G.; Trudeau, P. (1994). "Vehicle routing with split deliveries". *Discrete Applied Mathematics*. Vol. 50, no. 3, págs. 239-254.
- [24] Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. (1983). "Optimization by simulated annealing". *Science*. Vol. 220, no. 4598, págs. 671-680.
- [25] Pisinger, D.; Ropke, S. (2007). "A general heuristic for vehicle routing problems". *Computers & Operations Research*. Vol. 34, no. 8, págs. 2403-2435.
- [26] Shaw, P. (1998). "Using constraint programming and local search methods to solve vehicle routing problems". *Actas de la International Conference on Principles and Practice of Constraint Programming*. Springer. Berlin, Heidelberg. Págs. 417-431.
- [27] Cavallin, A., Rossit, D. G., Savoretti, A. A., Sorichetti, A. E.; Frutos, M. (2017). "Logística inversa de residuos agroquímicos en Argentina: resolución heurística y exacta". *Actas del XV Simposio en Investigación Operativa - 46 Jornadas Argentinas de Informática e Investigación Operativa*. Córdoba, Argentina.
- [28] Groër, C., Golden, B.; Wasil, E. (2010). "A library of local search heuristics for the vehicle routing problem". *Mathematical Programming Computation*. Vol. 2, no. 2, págs. 79-101.
- [29] Christofides, N.; Eilon, S. (1969). "An algorithm for the vehicle-dispatching problem". *Journal of the Operational Research Society*. Vol. 20, no.3, págs. 309-318.
- [30] Christofides, N., Mingozzi, A.; Toth P. (1979). "The vehicle routing problem". En: *Combinatorial optimization*. N. Christofides, A. Mingozzi, P. Toth, C. Sandi (Editores). John Wiley, Chichester. Capítulo 14.
- [31] Mendoza, J., Hoskins, M., Guéret, C., Pillac, V.; Vigo, D. (2014). "VRP-REP: a vehicle routing community repository". *Actas del Third meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog'14)*. Oslo, Noruega.

Agradecimientos

Los autores de este trabajo desean agradecer el financiamiento recibido de la Universidad Nacional del Sur para el desarrollo de los proyectos PGI 24/J084 y PGI 24/ZJ35. Además, el primer autor de este trabajo agradece el financiamiento recibido del Consejo Interuniversitario Nacional (CIN) a través de una Beca de Estímulo a las Vocaciones Científicas.