



## **Rediseño de las rutas de recolección de residuos sólidos urbanos en el Municipio de León Guanajuato, México.**

José Luis Ramos<sup>a</sup>, Javier Yáñez<sup>a</sup>, Luis Ernesto Mancilla<sup>b</sup>,

<sup>a</sup>Centro de Innovación Aplicada en Tecnologías Competitivas, León, Gto., jramos.picyt@ciatec.mx

<sup>b</sup>Instituto Tecnológico de León (SEP)

### **RESUMEN:**

A pesar de que la recolección de residuos sólidos urbanos es una función gubernamental muy importante desde los puntos de vista económico y ambiental, tradicionalmente las rutas de recolección han sido diseñadas de manera intuitiva por parte de quienes tienen a cargo esa responsabilidad.

En este trabajo se tiene como objetivo proponer y aplicar una fusión de herramientas con el fin de rediseñar las rutas de recolección. El problema se abordó con el enfoque general para el tratamiento de los problemas de rutas de transporte conocido como el Problema del Agente Viajero (TSP por sus siglas en inglés). Se utilizaron conceptos de Programación Matemática como es el caso de los grafos, relacionándolo con la técnica metaheurística Algoritmos Genéticos. Estos elementos se complementaron con fuentes de información y herramientas informáticas disponibles a cualquier usuario como es el caso de Google Maps y su herramienta Daft Logic para medir distancias. Se desarrolló un programa de Algoritmos Genéticos en MatLab aplicándolo a una ruta del municipio. Con base en esta experiencia se desarrolló una metodología que se propone para el rediseño de rutas.

### **1. INTRODUCCIÓN**

Actualmente se generan un promedio de 991 toneladas de residuos sólidos urbanos diariamente en la Ciudad de León, Gto. (Sistema Integral de Aseo Público, SIAP León, 2014).

Las rutas de recolección de residuos sólidos urbanos tradicionalmente han sido diseñadas de manera intuitiva sin considerar aspectos importantes como el modelado del sistema y la optimización del mismo mediante técnicas científicas e ingenieriles que permitan tener rutas de recolección más eficientes.

Se tiene como objetivo aplicar diferentes herramientas de solución al problema del diseño de las rutas de recolección, con el fin de rediseñarlas. Se aplicó el enfoque general para el tratamiento de estos problemas conocido como el Problema del Agente Viajero (TSP por sus siglas en inglés).

Dado que en la literatura sobre el tema de ruteo de vehículos son abundantes los ejemplos, con resultados positivos, de aplicaciones de los Algoritmos Genéticos (AG), se eligió esta técnica metaheurística para ser aplicada en este proyecto. Más adelante se presentan algunos ejemplos.

En términos sencillos, el AG aplicado al TSP, consiste en la selección de los dos mejores padres, de entre una población, para crear dos hijos. Los hijos creados a partir de un cruce de los padres



son mejorados por medio de mutaciones, las cuales ocurren con una baja probabilidad. Posteriormente estos hijos reemplazan a los dos padres menos aptos. La aptitud se mide en función de la longitud de recorrido, a menor longitud mayor aptitud y viceversa. El proceso de crear hijos y retirar padres se repite hasta que se presenta una condición de terminación (Taha, 2012).

Lacomme, Prins & Ramdane-Cherif (2001), mencionan que los algoritmos exactos están aún limitados para problemas pequeños y que los metaheurísticos son requeridos para instancias de gran escala. Los autores publican el primer AG para el Problema de Ruteo de Arcos Capacitado (CARP, por sus siglas en inglés), planteando un híbrido que ataca extensiones realistas como vueltas prohibidas y grafos mixtos.

Posteriormente, continuando con el planteamiento del CARP, Lacomme, Prins & Sevaux, (2003), buscan minimizar la longitud del viaje más largo además de la longitud de la ruta total (el único criterio minimizado en el problema académico). Para el efecto, estos investigadores presentan un AG bi-objetivo para este CARP más realista nunca estudiado en la literatura hasta ese momento. Sobre esta base, incluyen dos características clave: uso de heurísticas constructivas para sembrar la población inicial y la hibridación con un procedimiento de búsqueda local.

Viotti, Poletini, Pomi, et al. (2003), argumentan que hasta la fecha, las rutas óptimas se han desarrollado de acuerdo con metodologías intuitivas y experiencia de campo, y que para analizar estas complejidades se usan aplicaciones que normalmente se basan en procedimientos heurísticos que permiten soluciones de alta calidad pero que en el tema computacional tienen una complejidad que es una limitación por la calidad de las soluciones calculadas y para la representación exacta de las zonas urbanas. Ellos emplean una metodología alternativa que se basa en un AG.

Zhu, Xia, Yang, et al. (2008), presentan un AG mejorado para resolver una versión extendida del CARP (ECARP) como vueltas prohibidas o problemas de encharcamientos. El nuevo algoritmo mejora su estructura, evitando el fenómeno de la convergencia prematura. La mejora propuesta puede resolver el ECARP eficazmente, y la comparación entre la mejora de AG y un algoritmo memético clásico, muestra que este algoritmo es más eficaz y puede conseguir mejores resultados en la resolución del CARP básico de gran tamaño.

Por su parte, Bonomo, Duran, Larumbe, et al. (2012), proponen un método que utiliza técnicas de investigación de operaciones para optimizar las rutas de los vehículos de recolección de residuos de contenedores. El problema de recolección de residuos se reduce al problema del viajante de comercio clásico (TSP). El enfoque de solución emplea la teoría de grafos y herramientas de programación matemática. También se discute el proceso de corrección de datos.

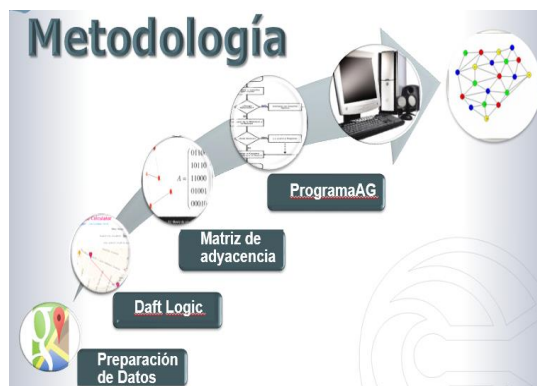
## 2. DESARROLLO

Después de revisar la literatura y las herramientas que se han empleado para caracterizar y resolver el problema de la recolección de residuos sólidos municipales, se plantea la siguiente metodología.



### Metodología para el rediseño de las rutas de recolección.

La figura muestra las fases de la metodología. A continuación se explican sus detalles.



Metodología para el rediseño de rutas de recolección

#### a) Preparación de datos básicos

Se obtuvo información del SIAP a través de la Unidad Municipal de Acceso a la Información, consistente en las zonas correspondientes a las rutas que se tienen implementadas en el Municipio.

Tomando como referencia esta información es posible descargar los mapas respectivos de Google Maps. Se elabora la relación de calles, se determinan sus cruces y se etiquetan los nodos correspondientes. Tomando en cuenta los sentidos de las calles se genera la matriz de adyacencias. Esto permite dibujar el grafo dirigido de la red.

#### b) Generación de la matriz de adyacencia ponderada

Se miden las aristas del grafo con la herramienta Daft Logic de Google Maps para obtener las distancias correspondientes. Con estos datos se convierte la matriz de adyacencias en la matriz de adyacencias ponderada. Para dirigir el programa en el manejo de los casos en los que no hay adyacencia entre nodos, se agrega un valor muy alto, 9999, en los elementos respectivos de la matriz.

#### c) Ejecución del programa AG.

Se desarrolló en MatLab el programa del AG propuesto por Taha (2012) para resolver el problema del TSP.

Se alimenta la matriz de adyacencias ponderada; se definen los parámetros correspondientes y se ejecuta el programa del AG. Si es necesario, dada la topología de la red, se calculan aristas auxiliares, las cuales se definen como aquellas en las que no hay adyacencia entre dos nodos, pero que el programa las requiere para cerrar un ciclo y terminar la ejecución del algoritmo. Esto se determina cuando el programa, en el resultado de una primera corrida incluye un valor muy alto (9999) en su recorrido. El valor de las aristas auxiliares se puede calcular por inspección, sumando los tramos correspondientes al recorrido más corto que une los dos nodos. Se dibuja el grafo de la solución.

#### d) Verificación de distancias de la ruta obtenida

Con base en el recorrido obtenido y los datos de la matriz de adyacencias ponderada se verifica la distancia total.

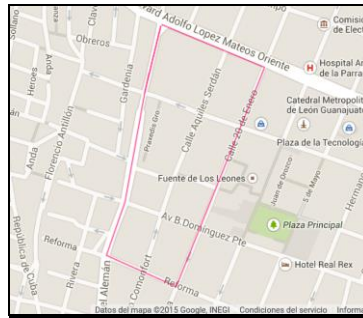
## 4. RESULTADOS

Se trabajó con la ruta 1 del municipio, de la cual se tomaron 16 nodos (cruces de calles).



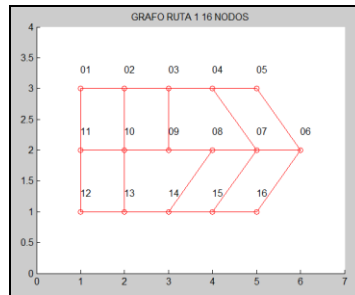
**a) Preparación de datos básicos**

A partir de la información del SIAP se descargó de Google Maps (2015) el mapa de la zona correspondiente a la ruta 1.



**Mapa de 16 nodos de la ruta 1**

Se elaboró la relación de calles y se determinaron los sentidos, restricciones y otras características de las vialidades. Se definieron los cruces de calles y se etiquetaron los nodos. Se generó la matriz de adyacencias a partir de los cruces de calles, tomando en cuenta los sentidos. Se dibujó el grafo de la red.



**Grafo de la ruta 1 16 nodos**

**b) Generación de la matriz de adyacencia ponderada**

Con la herramienta Daft Logic (2014) se midieron las aristas del grafo para obtener las distancias. Con estos datos se convirtió la matriz de adyacencias a matriz de adyacencias ponderada. Se incluyó el valor 9999 en los elementos de la matriz en los que no hay adyacencia entre nodos.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	9999	0150	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
2	9999	9999	0100	9999	9999	9999	9999	9999	9999	0100	9999	9999	9999	9999	9999	9999
3	9999	9999	9999	0210	9999	9999	9999	9999	0110	0800	9999	9999	9999	9999	9999	9999
4	9999	9999	9999	9999	0200	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
5	9999	9999	9999	9999	9999	0190	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999
6	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	0120
7	9999	9999	9999	9999	0160	9999	0190	9999	9999	9999	9999	9999	9999	9999	9999	9999
8	9999	9999	9999	9999	9999	9999	9999	0180	9999	9999	9999	9999	9999	9999	0120	9999
9	9999	9999	9999	9999	9999	9999	9999	9999	0020	9999	9999	9999	9999	9999	9999	9999
10	9999	9999	9999	9999	9999	9999	9999	9999	9999	0100	9999	9999	9999	9999	0110	9999
11	0100	9999	9999	9999	9999	9999	9999	9999	9999	9999	0160	9999	9999	9999	9999	9999
12	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	0100	9999	9999	9999	9999
13	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	0170	9999	9999
14	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	0120	9999
15	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	0180
16	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999

**Matriz de adyacencias ponderada**



**c) Ejecución del programa AG.**

Se alimentó la matriz de adyacencias ponderada y se definieron parámetros del programa:

Número de nodos = 16

Tamaño de la población = 6 individuos (recorridos)

Número de generaciones = 10

Tasa de probabilidad de mutación = 0.1

Dada la topología de la red, fue necesario alimentar en el programa la distancia entre los nodos 3 y 10, que no son adyacentes, pero que fue requerida para cerrar un ciclo y terminar la ejecución del algoritmo. El recorrido que une estos dos nodos es: 3, 9, 8, 14, 13, 12, 11, 10, el cual suma una distancia total de 800, misma que fue agregada a la matriz de datos.

El resultado de la ejecución fue el siguiente:

```

generacion =
10
individ =
16 15 14 13 12 11 1 2 3 10 9 8 7 4 5 6 16
16 15 14 13 12 11 1 2 3 10 9 8 7 4 5 6 16
16 15 14 13 12 11 1 2 3 10 9 8 7 4 5 6 16
9 8 7 4 5 6 16 15 14 13 12 11 1 2 3 10 9
16 15 14 13 12 11 1 2 3 10 9 8 7 4 5 6 16
16 15 14 13 12 11 1 2 3 10 9 8 7 4 5 6 16
distancia =
2870          2870          2870          2870          2870
>>
  
```

**Resultado de ejecución del programa Algoritmo Genético**

**d) Verificación de distancias de la ruta obtenida**

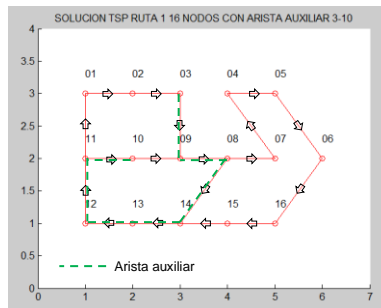
Siguiendo el recorrido obtenido y con los datos de la matriz de adyacencias ponderada se verifica la distancia total.

	Del nodo	Al nodo	Distancia
1	16	15	180
2	15	14	180
3	14	13	120
4	13	12	170
5	12	11	100
6	11	1	100
7	1	2	150
8	2	3	100
9	3	10	<b>800</b>
10	10	9	100
11	9	8	20
12	8	7	180
13	7	4	160
14	4	5	200
15	5	6	190
16	6	16	120
		suma	<b>2870</b>

**Distancia total del recorrido**



Se representó la solución obtenida mediante el dibujo del grafo correspondiente.



**Grafo de la solución**

## 5. CONCLUSIONES.

Al calcularse y agregarse el dato correspondiente a la arista artificial 3-10 se obtuvo el resultado del recorrido igual a 2870 el cual se considera correcto, mismo que fue validado mediante varias ejecuciones del programa.

El programa converge a una solución en 10 generaciones presentando un ciclo hamiltoniano, es decir, se visitan todos los cruces pero no todas las calles de la ruta. Esto es consistente con el hecho de que en la literatura frecuentemente se menciona como enfoque de solución para este tipo de problemas al llamado Problema de Ruteo de Arcos (ARP, por sus siglas en inglés). Lo anterior implica la sustitución de un algoritmo ad-hoc dentro del programa AG para etapas futuras. Así mismo se contempla la inclusión de una rutina que calcule en forma automática la distancia entre nodos de una arista artificial cuando sea requerido, mediante un algoritmo del camino más corto.

En este planteamiento se están obviando algunos detalles que pueden ser de importancia, como son: el volumen y peso estimado de recolección, tipo y capacidad de los vehículos, ubicación de los sitios de resguardo de los vehículos y de los vertederos de desechos.

Con base en las experiencias de este trabajo, se definió una propuesta de metodología práctica y accesible para el rediseño de rutas que se espera pueda ser replicada en otras.

## BIBLIOGRAFÍA

1. SIAP, León. 2014. Sistema Integral de Aseo Público de León, Gto. Municipio de León, Gto. México. <http://www.aseopublicoleon.gob.mx/#!servicios/c1iwz>.
2. Taha, H. 2012. 9a ed. Investigación de operaciones. Pearson Educación México.
3. Lacomme, P; Prins, C; Ramdane-Cherif, W. (2001). APPLICATIONS OF EVOLUTIONARY COMPUTING, PROCEEDINGS Book Series: LECTURE NOTES IN COMPUTER SCIENCE, 2037, 473-483.
4. Lacomme, P; Prins, C; Sevaux, M. (2003). Multiobjective Capacitated Arc Routing Problem. EVOLUTIONARY MULTI-CRITERION OPTIMIZATION, PROCEEDINGS Book Series: LECTURE NOTES IN COMPUTER SCIENCE, 2632, 550-564.
5. Viotti, P; Poletini, A; Pomi, R; et al. (2003). Genetic algorithms as a promising tool for optimisation of MSW collection routes. WASTE MANAGEMENT&RESEARCH 21(4) 292-98



6. Zhu, Zhengyu; Xia, Mengshuang; Yang, Yong; et al. (2008). An Improved Genetic Algorithm for the Extended Capacitated Arc Routing Problem. 7TH WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION, 1-23, 2017-2022.
7. Bonomo, Flavio; Duran, Guillermo; Larumbe, Frederico; et al. (2012). A method for optimizing waste collection using mathematical programming: a Buenos Aires case study. WASTE MANAGEMENT & RESEARCH, 30(3), 311-324
8. Google Maps. 2015. Recuperado de <https://www.google.com.mx/maps/@21.1193491,-101.683987,15z>
9. Daft Logic. 2014. Recuperado de <http://www.daftlogic.com/projects-google-maps-distance-calculator.htm>