

---

# UNA APLICACIÓN DEL PROBLEMA DEL CARTERO RURAL A LA RECOLECCIÓN DE RESIDUOS RECICLABLES EN ARGENTINA

---

GUSTAVO BRAIER\*  
GUILLERMO DURÁN\*\*  
JAVIER MARENCO\*\*\*  
FRANCISCO WESNER\*\*\*

## Resumen

En este trabajo reportamos la aplicación de técnicas de programación matemática a la optimización de las rutas de vehículos de recolección de residuos reciclables en Morón, una municipalidad en el Gran Buenos Aires, Argentina. Este problema es un caso particular del problema del cartero rural abierto en grafos mixtos, y se resuelve por medio de un modelo de programación lineal entera. Las rutas generadas por este procedimiento son significativamente mejores que las rutas designadas a mano y que estaban en uso antes de la implementación del presente proyecto. El beneficio más importante consiste en que estas rutas cubren el 100 % del sector a ser recolectado, mientras que con las rutas diseñadas a mano se omitía hasta el 16 % de las cuadras. Las rutas generadas por el modelo fueron implementadas por la municipalidad en 2014.

**PALABRAS CLAVE:** Problema del cartero rural, Recolección de residuos, Ruteo de vehículos.

---

\*Braier & Asociados Consultores (papyro.com), Argentina.

\*\* CONICET, Argentina. Instituto de Cálculo, FCEyN, Universidad de Buenos Aires, Argentina. Departamento de Matemática, FCEyN, Universidad de Buenos Aires, Argentina. Departamento de Ingeniería Industrial, FCFM, Universidad de Chile, Chile.

\*\*\* Departamento de Computación, FCEyN, Universidad de Buenos Aires, Argentina.

---

## 1. Introducción

---

Este trabajo presenta un enfoque basado en programación lineal entera para mejorar el ruteo de vehículos de recolección de residuos reciclables en la municipalidad de Morón, al oeste de la Ciudad de Buenos Aires, Argentina. Morón tiene una población de 320,000 habitantes (de acuerdo con el Censo 2010) y cubre un área de 55.6 km<sup>2</sup>. La municipalidad está dividida en cinco distritos administrativos (Castelar, Morón centro, Haedo, Palomar y Villa Sarmiento), que a su vez están divididos en hasta siete sectores, de modo tal que cada sector es recorrido por un único camión de recolección. Los residuos reciclables en cada distrito se recolectan un día de la semana (de lunes a viernes). Como el sistema de recolección no cuenta con contenedores o basureros centralizados, los vehículos deben recorrer todos los frentes domiciliarios para realizar la recolección.

Las rutas de los camiones de la municipalidad para la recolección de los residuos reciclables habían sido diseñadas, previamente a la concreción de este proyecto, de manera manual. Como consecuencia de ello los recolectores terminaban no recorriendo algunas de las cuadras del municipio en las horas estipuladas. Esta situación generaba numerosas protestas de los vecinos, que motivaron la realización de este trabajo.

El problema de encontrar una ruta óptima que asegura que todas las cuadras de un sector son visitadas es un caso particular del problema del *cartero rural abierto* en grafos mixtos [15], que es NP-hard (en este trabajo, “cuadra” se refiere a un segmento de una calle a lo largo de una manzana entre dos esquinas consecutivas). Dado un grafo y un conjunto de arcos/aristas seleccionados, el problema del cartero rural solicita hallar un ciclo de costo mínimo que visite cada arco/arista seleccionado al menos una vez, y es una generalización del problema del cartero Chino. El problema del *cartero rural abierto* solicita un camino con las mismas propiedades. El problema del cartero rural ha sido objeto de mucho interés por parte de la comunidad de optimización combinatorial, tanto para grafos dirigidos [9, 13, 14, 4] como para grafos mixtos [6, 7, 5, 8].

Las reglas de tránsito complican el diseño de los recorridos de los camiones de recolección, y las reglas más relevantes en este contexto son la prohibición de girar a la izquierda en esquinas con semáforo y la prohibición de realizar “giros en U”. El enfoque propuesto en este trabajo utiliza un algoritmo de planos de corte sobre un modelo natural de programación lineal entera para el

problema. La principal contribución teórica de este trabajo es la introducción de un procedimiento para combinar *subtours*, que permite reducir la cantidad de rondas de planos de corte, reduciendo así los tiempos de ejecución necesarios para obtener soluciones óptimas. Este procedimiento no se puede aplicar en instancias generales del problema del cartero rural, aunque demostró ser útil en el caso particular considerado en este trabajo.

Un survey interesante de problemas de ruteo de vehículos para la recolección de residuos está dado en [11]. Se pueden hallar en la literatura aplicaciones de técnicas de programación matemática a la recolección de residuos en distintas ciudades, incluyendo Chicago (EEUU) [10], Kaoshiung (Taiwan) [3], Lisboa (Portugal) [12], Hamilton (Canadá) [16], Santiago (Chile) [1] y la Ciudad de Buenos Aires [2]. La Ciudad de Buenos Aires no incluye la municipalidad de Morón, y el problema de recolección de residuos estudiado en [2] tiene una estructura diferente a la considerada en este trabajo.

Este trabajo está organizado de la siguiente forma. La Sección 2 describe el problema a resolver y presenta un modelo de programación lineal entera. En la Sección 3 presentamos el procedimiento que utilizamos para resolver este modelo, obteniendo así las rutas de recolección en cada sector. La Sección 4 reporta los resultados computacionales sobre el municipio de Morón, y la Sección 5 cierra el trabajo con nuestras conclusiones.

---

## 2. El problema

---

Cada sector –que será recolectado por un único camión– está representado por un grafo mixto cuyos vértices corresponden a las esquinas del sector. Se tienen las coordenadas geográficas de cada vértice de este grafo, y se tiene también una indicación especificando si hay un semáforo en esa esquina. Los arcos del grafo representan calles de un sentido de circulación (calles de “mano única”). Las aristas del grafo representan calles con dos sentidos de circulación (calles “doble mano”) pero que son angostas y alcanza con recorrerlas en cualquiera de los dos sentidos para realizar la recolección de residuos. Hay también avenidas anchas con dos sentidos de circulación, pero en ese caso se representan las cuadras como dos arcos paralelos, uno en cada dirección. Cada arco y cada arista tiene asociada una distancia, que se calcula como la distancia en línea recta entre las dos esquinas.

Los vehículos de recolección no pueden realizar giros en U. Para incorporar esta restricción, expandimos el grafo dividiendo cada vértice en varios vértices que representan las formas posibles de arribar a la esquina. Se agregan arcos

auxiliares entre estos nuevos vértices, que representan las transiciones permitidas entre los vértices de la esquina. La Figura 1 muestra un ejemplo de esta expansión en una esquina de una calle de un sentido con una calle de doble mano. En la figura, se muestran con líneas punteadas los arcos auxiliares. De acuerdo con los funcionarios de la municipalidad, los camiones de recolección no tienen restricciones en cuanto a la posibilidad de girar en las esquinas, incluso en calles angostas. Esto implica que un giro en una esquina no tiene un costo adicional demasiado alto, y entonces se asigna una distancia pequeña y positiva a los arcos auxiliares. Esta distancia es la misma para todos los arcos auxiliares.

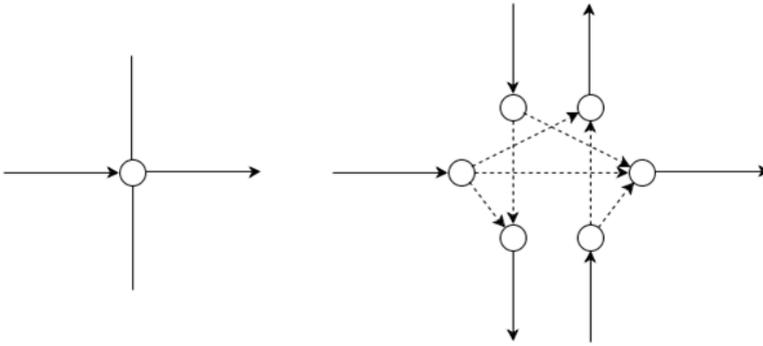


Figura 1: Expansión de un vértice del grafo para representar los giros permitidos.

La prohibición de realizar giros en U no aplica para las calles sin salida. Para permitir esta situación, se modifica levemente el grafo agregando arcos auxiliares entre los vértices finales de una calle sin salida, como en la Figura 2.

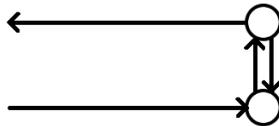


Figura 2: Arcos auxiliares para permitir giros en U en calles sin salida.

Además de los giros en U, las reglas de tránsito en Argentina prohíben girar a la izquierda en esquinas con semáforos de calles de doble mano, excepto cuando el semáforo permite explícitamente este giro con una luz especial. La expansión del grafo descrita arriba permite incorporar estas restricciones, eliminando los arcos auxiliares que representan giros ilegales. Un ejemplo de esta situación está dado en la Figura 3, para la intersección de dos calles de

doble mano.

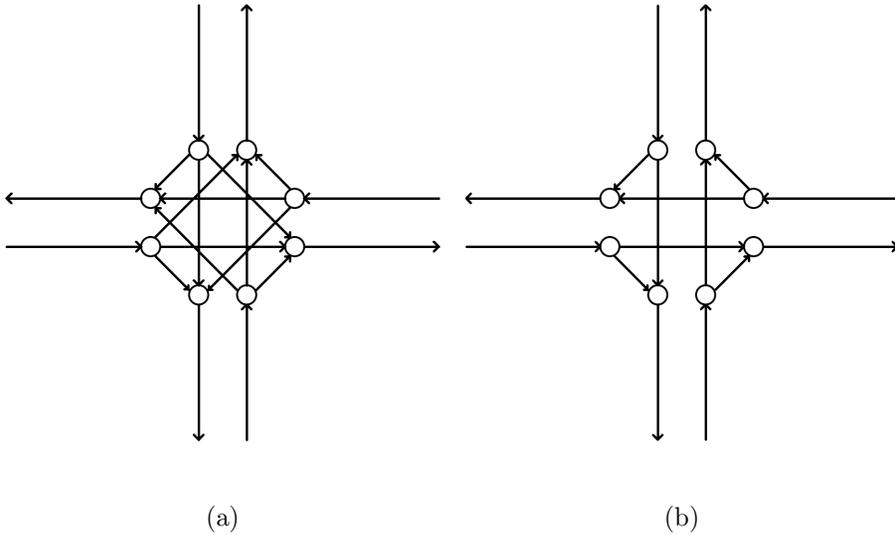


Figura 3: Giros permitidos en la intersección (a) sin semáforos y (b) con semáforos.

El punto de inicio de la ruta de recolección de un sector debe estar ubicado en el borde del sector más cercano al depósito de origen, desde donde parten los camiones. Cualquier vértice en este borde puede ser seleccionado como el punto de inicio del recorrido. La selección del vértice final del recorrido no tiene un impacto significativo en la performance total del mismo, dado que una vez que la recolección ha sido completada el camión vuelve directamente al centro de reciclado. Como los sectores no son muy extensos y el centro de reciclado se encuentra bastante alejado de los diferentes sectores, se puede despreciar este costo adicional.

En algunas instancias, en especial en sectores con muchas calles de mano única, los camiones de reciclado pueden tener que realizar pequeñas desviaciones para ingresar a algunas calles. Para reducir este efecto y evitar desvíos innecesarios, es posible que los conductores deban ingresar brevemente a un sector vecino. Para contemplar estas situaciones, se agregan al grafo las manzanas inmediatamente contiguas al sector en cuestión, formando una banda perimetral de manzanas en sectores adyacentes que no se deben recolectar pero que pueden ser utilizadas para optimizar el recorrido. Estos nuevos arcos y aristas se tratan como arcos auxiliares, con el costo adecuadamente calculado.

La Figura 4 muestra un ejemplo de esta situación para un sector en el distrito de Villa Sarmiento. Las cuadras que conforman la banda perimetral están mostradas con líneas oscuras. En esta figura no se agregaron manzanas



entera para este problema. Para cada arista  $ij \in E$  introducimos las variables enteras  $x_{ij}$  y  $x_{ji}$ , que representan la cantidad total de veces que se recorren  $ij$  y  $ji$ , respectivamente. Para cada arco  $ij \in A$  introducimos la variable entera  $y_{ij}$ , que representa la cantidad total de veces que se recorre el arco  $ij$ . Finalmente, para cada vértice  $i \in I$  definimos la variable binaria  $s_i$  que especifica si  $i$  es el primer vértice de la ruta, y para cada vértice  $j \in V$  definimos la variable binaria  $t_j$  que especifica si  $j$  es el último vértice de la ruta. Para cualquier conjunto  $S \subseteq V$ , definimos  $E(S, \bar{S}) = \{ij \in E : i \in S, j \notin S\}$  y  $A(S, \bar{S}) = \{ij \in A : i \in S, j \notin S\}$ . Podemos ahora formular el modelo como sigue.

$$\text{mín} \sum_{ij \in E} w_{ij}(x_{ij} + x_{ji}) + \sum_{ij \in A} w_{ij}y_{ij} \tag{1}$$

$$y_{ij} \geq 1 \quad \forall ij \in A_M \tag{2}$$

$$x_{ij} + x_{ji} \geq 1 \quad \forall ij \in E \tag{3}$$

$$s_i + \sum_{j:ji \in E} x_{ji} + \sum_{j:ji \in A} y_{ji} = \sum_{j:ij \in E} x_{ij} + \sum_{j:ij \in A} y_{ij} + t_i \quad \forall i \in I \tag{4}$$

$$\sum_{j:ji \in E} x_{ji} + \sum_{j:ji \in A} y_{ji} = \sum_{j:ij \in E} x_{ij} + \sum_{j:ij \in A} y_{ij} + t_i \quad \forall i \in V \setminus I \tag{5}$$

$$\sum_{i \in I} s_i = 1 \tag{6}$$

$$\sum_{i \in V} t_i = 1 \tag{7}$$

$$\sum_{ij \in E(S, \bar{S})} x_{ij} + \sum_{ij \in A(S, \bar{S})} y_{ij} \geq 1 \quad \forall S \subset V, S \neq \emptyset \tag{8}$$

$$x_{ij} \in \mathbb{Z}_+ \quad \forall ij \in E \tag{9}$$

$$y_{ij} \in \mathbb{Z}_+ \quad \forall ij \in A \tag{10}$$

$$s_i \in \{0, 1\} \quad \forall i \in I \tag{11}$$

$$t_i \in \{0, 1\} \quad \forall i \in V \tag{12}$$

La función objetivo intenta minimizar el costo total de la ruta de recolección, es decir, la suma de los costos individuales de las cuadras recorridas por el vehículo. Las restricciones (2) solicitan que cada arco sea visitado al menos una vez, mientras que las restricciones (3) requieren que cada arista sea visitada al menos una vez en alguna de las dos direcciones. Las restricciones (4)-(5) aseguran que la solución es un camino, imponiendo la conservación de flujo en cada vértice con excepción de los vértices de inicio y finalización del recorrido. Las restricciones (6) y (7) garantizan que estos dos vértices son únicos.

A pesar de que la incorporación de arcos auxiliares permite modelar las restricciones de tránsito, también agrega una complicación en el modelado. Dado que no es necesario cubrir todos los arcos en el grafo, entonces requerir que el grado de entrada sea igual al grado de salida no es suficiente para garantizar que las soluciones factibles del modelo sean rutas conexas. De hecho, las restricciones (2)-(7) permiten la formación de subtours, que deben ser evitados.

Por ejemplo, si se resuelve el modelo dado por (1)-(7) y (9)-(12) sobre la instancia dada por la Figura 5, se puede obtener una solución como la especificada en la Figura 6. Esta solución tiene un camino principal (la línea punteada) formado por los vértices  $C_1 = (4, 1, 2, 3, 6, 9, 8, 7, 4, 5, 2)$  y un subtour (la línea rayada) formado por los vértices  $C_2 = (5, 6, 9, 8, 5)$ . Recordemos que como el grafo está expandido, cada vértice en la figura en realidad representa varios vértices en el grafo. Una vista detallada del vértice 5 revela cómo se encuentra separado el subtour del resto de la ruta, y se tiene una situación similar en los vértices 6, 9 y 8. Para evitar esta situación, las *restricciones de rompimiento de subtours* (8) impiden la formación de subtours. Estas restricciones imponen la existencia de al menos un arco o arista conectando el subtour al resto de los vértices. Finalmente, las restricciones (9)-(12) especifican los valores posibles para las variables.

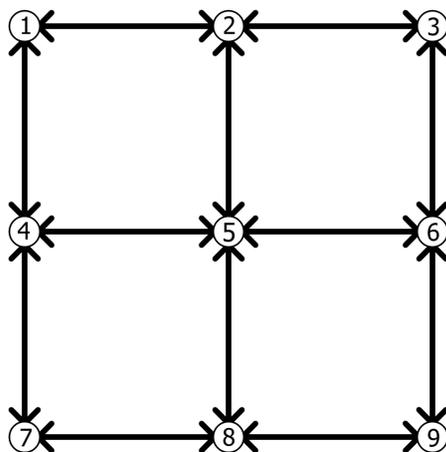


Figura 5: Grafo de un sector compuesto por cuatro manzanas y calles de doble mano.

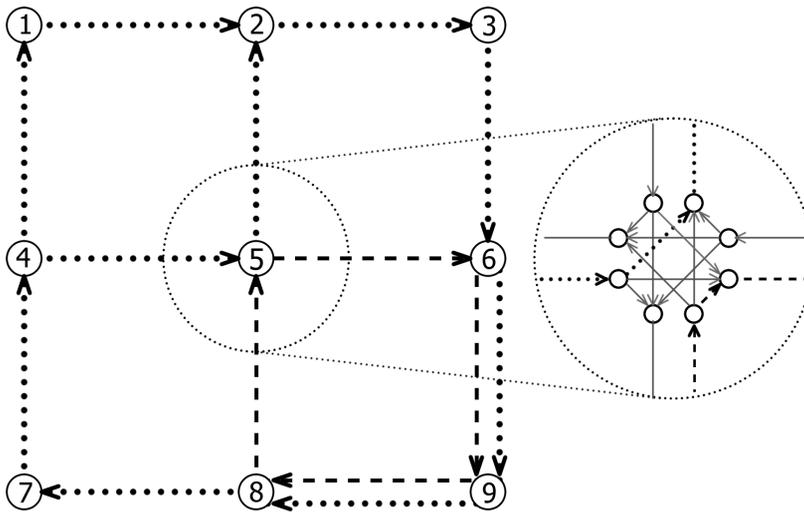


Figura 6: Formación de un subtour en una solución para el modelo sin las restricciones de eliminación de subtours, para la instancia de la Figura 5.

---

### 3. Procedimiento propuesto

---

Proponemos en esta sección un procedimiento sencillo basado en planos de corte para resolver el modelo de programación entera presentado en la sección anterior. La Figura 7 muestra el pseudocódigo de este algoritmo. Se resuelve un modelo relajado, compuesto por las restricciones (2)-(7) y (9)-(12) (Pasos 1 y 2). Si la solución obtenida no contiene subtours (Paso 4), entonces es óptima y el algoritmo termina. En caso contrario, la solución contiene un subtour que debe ser evitado.

Para excluir esta solución, una técnica estándar consiste en agregar la restricción de rompimiento de subtours (8) asociada con el subtour, cortando así esta solución. Sin embargo, esta estrategia tiene el problema de que puede requerir muchas iteraciones hasta hallar una solución sin subtours. Para manejar esta situación, primero intentamos combinar el subtour con el camino principal por medio de intercambio de arcos auxiliares. Por ejemplo, la solución que se muestra en la Figura 6 se puede modificar intercambiando los arcos auxiliares en el vértice 5, como en la Figura 8, obteniendo así una ruta conexas. Como el costo de los arcos auxiliares entre vértices de una misma esquina es el mismo, la nueva ruta obtenida por este procedimiento también es óptima.

- 
1. Inicializar el modelo relajado  $\mathcal{M} := (1)-(7)$  y  $(9)-(12)$ .
  2. Resolver el modelo  $\mathcal{M}$ .
  3. Si  $\mathcal{M}$  es no factible, retornar “no factible” y terminar.
  4. Si la solución óptima de  $\mathcal{M}$  no tiene subtours, retornar esta solución y terminar.
  5. Si los subtours se pueden combinar con el camino principal entre el nodo de inicio y el nodo de finalización, combinarlos, retornar la solución obtenida y terminar.
  6. En caso contrario, agregar a  $\mathcal{M}$  una restricción de rompimiento de subtours (8) para cada subtour en la solución y volver al Paso 2.
- 

Figura 7: Algoritmo de planos de corte para resolver en forma exacta el caso particular del problema del cartero rural planteado en este trabajo.

A pesar de su gran eficiencia computacional, esta técnica de *combinación de subtours* no funciona en todos los casos. Por ejemplo, consideremos el mapa de la Figura 9 y la solución (con un subtour) de la Figura 10. En este caso, el subtour (línea rayada) no se puede combinar con el camino principal (línea punteada) por el método descrito arriba.

La estrategia definitiva consiste entonces en utilizar el procedimiento de combinación de subtours en conjunto con el agregado dinámico de restricciones de rompimiento de subtours. Si una solución no tiene subtours, se retorna la solución como resultado (Paso 4). Si la solución tiene uno o más subtours que se pueden combinar con el camino principal, se realiza esta combinación y se retorna la solución obtenida (Paso 5), que es óptima. Finalmente, si la solución tiene subtours que no se pueden combinar con el camino principal, se agrega al modelo una restricción de rompimiento de subtours (8) por cada subtour en la solución y se repite el procedimiento (Paso 6).

La idea no es eliminar completamente el agregado dinámico de restricciones de rompimiento de subtours, sino reducir la cantidad de rondas de planos de corte necesarias para obtener una solución factible (es decir, conexa). El algoritmo propuesto garantiza que se encuentra una solución óptima en un número finito de pasos, dado que en el peor caso se agregan todas las restricciones de rompimiento de subtours y el número de estas restricciones es

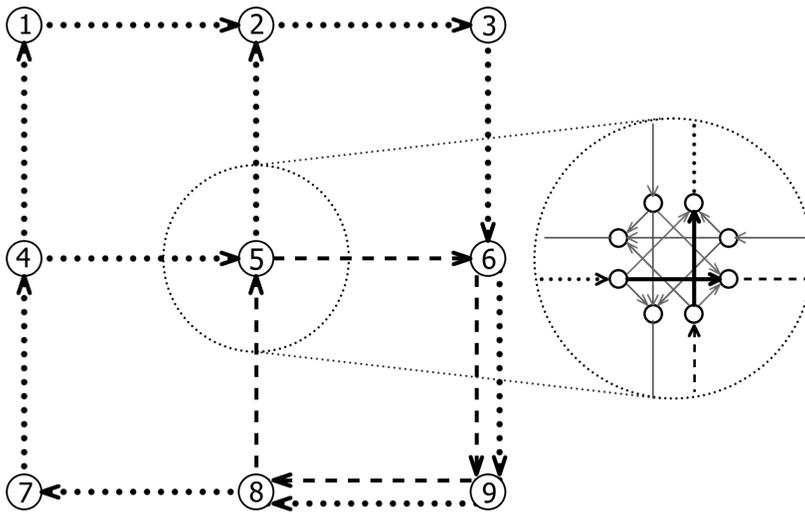


Figura 8: Combinación de un subtour con el camino principal para la ruta de la Figura 5.

exponencial pero finito. La intención es que esto no sea necesario, buscando una solución factible y óptima por medio del procedimiento de combinación de subtours.

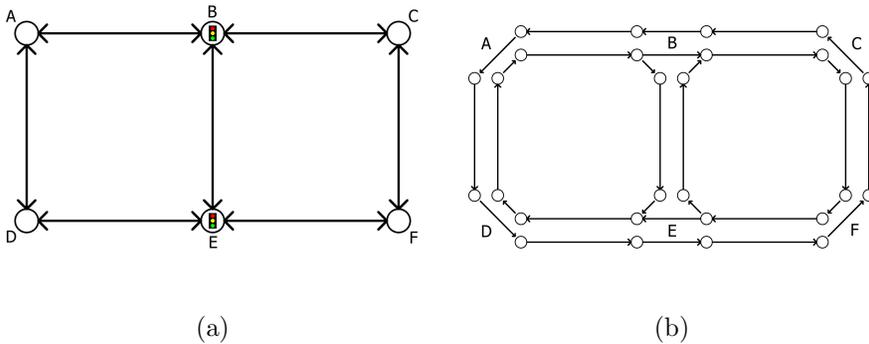


Figura 9: Un grafo simple y el grafo expandido correspondiente.

El agregado dinámico de restricciones de rompimiento de subtours se realiza una vez que el procedimiento de resolución del modelo  $\mathcal{M}$  ha terminado. Esto permite identificar los subtours rápidamente, dado que las variables en la solución obtenida toman valores enteros. Un enfoque alternativo consiste en separar las restricciones (8) durante el procedimiento branch-and-bound (convirtiendo así este procedimiento en un algoritmo de tipo branch-and-cut), pero la detección de restricciones de rompimiento de subtours violadas pasa a ser

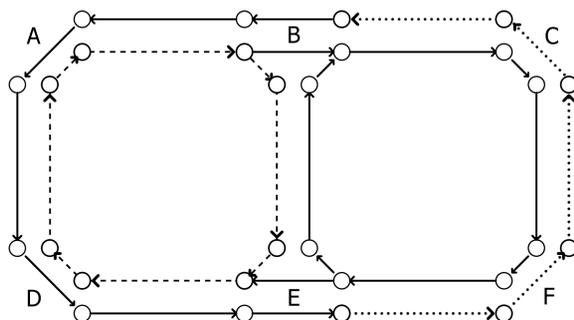


Figura 10: Una ruta con subtours que no se pueden combinar utilizando la técnica propuesta, para el grafo de la Figura 9.

más complicada. Por este motivo, decidimos en este trabajo buscar subtours una vez que el modelo se resuelve en forma óptima y se obtiene una solución entera.

---

## 4. Resultados computacionales

---

En esta sección presentamos y analizamos los resultados de los experimentos realizados para evaluar la efectividad del algoritmo propuesto en la sección anterior. Se fijó un máximo de 10 minutos para la resolución del modelo de programación entera. Se determinó este límite luego de comprobar que la mayoría de los sectores considerados en este trabajo se resuelven en forma óptima en menos de 10 minutos, y para el resto de los sectores el gap de optimalidad es muy bajo.

Los experimentos fueron realizados en una PC con un procesador AMD Phenom II x4 945 corriendo a 3GHz, y 4GB de memoria RAM. El sistema operativo es Linux Ubuntu 12.04 (32 bits) y el solver para los modelos de programación entera es SCIP. A pesar de que existen solvers comerciales con mejor performance que SCIP sobre bancos de prueba estándar, el uso de SCIP no trajo aparejados problemas importantes, y su performance fue adecuada para este trabajo.

La Tabla 1 muestra los experimentos realizados para determinar la efectividad del procedimiento de combinación de subtours propuesto en este trabajo. Todos los datos corresponden al distrito de Castelar, que fue seleccionado para estas pruebas debido a la gran diversidad de sectores que presenta. Los sectores de este distrito varían en tamaño, forma (algunos son rectangulares y otros son más irregulares) y tipos de calles. Para cada sector, se muestra el

número de iteraciones y los tiempos de resolución sin y con el procedimiento de combinación de subtours, respectivamente.

En Castelar1 y Castelar5, los dos sectores con la menor mejora, el procedimiento de combinación de subtours redujo de 2 a 1 la cantidad de iteraciones y el tiempo de resolución en aproximadamente la mitad. Por su parte, en Castelar3 el algoritmo sin combinación de subtours requiere de 59 iteraciones y casi 10 horas de ejecución para encontrar una solución factible, mientras que en la primera iteración el procedimiento de combinación de subtours permite encontrar una solución conexa y óptima. En otros sectores se observaron resultados similares.

Sector	V	E	A <sub>M</sub>	A <sub>AUX</sub>	Sin el Paso 5		Con el Paso 5	
					Iteraciones	Tiempo	Iteraciones	Tiempo
Castelar1	1372	658	28	1570	2	5.2 seg	1	2.4 seg
Castelar2	650	100	225	559	5	19 seg	1	3 seg
Castelar3	852	200	226	782	59	9.7 hs	1	10 min
Castelar4	878	354	85	876	6	52.4 seg	1	5.6 seg
Castelar5	1090	508	37	1416	2	5.7 seg	1	3 sec
Castelar6	738	166	203	650	22	3.7 hs	1	10 min
Castelar7	1082	524	17	1341	35	5.9 hs	2	20 min
Castelar8	1276	622	46	1574	24	3.7 hs	1	17.9 seg

Tabla 1: Contribución del procedimiento de combinación de subtours para el distrito de Castelar.

La Tabla 2 presenta información provista por la municipalidad antes del proyecto descrito en este trabajo. Las rutas informadas en esa tabla corresponden a los recorridos diseñados manualmente sin más herramientas que un mapa del sector a recorrer. Como puede verse en la Tabla en todos los sectores quedan cuadras sin recorrer, llegando a 60 en Castelar4.

Para comparar esta situación con nuestros resultados, la Tabla 3 muestra las características de las rutas obtenidas por el procedimiento propuesto en este trabajo. Por construcción, las rutas propuestas por el algoritmo de la sección anterior recorren todas las cuadras, visitando así el 100 % del sector en cuestión. En algunos casos, las rutas generadas por nuestro procedimiento son más largas que las rutas manuales, debido a la restricción que solicita recorrer todas las cuadras. Una estimación sencilla nos permitió ver que las nuevas rutas, aún siendo algunas de ellas más largas que las actuales, se podían recorrer en los tiempos asignados para el recorrido (5 horas en total). Por ejemplo, para la instancia Castelar4 se recorrían 46km en 4:10 horas, y el algoritmo propuso una ruta de 50km. Es prácticamente imposible estimar con precisión el tiempo de recorrida de la nueva ruta, pero las 4:10 horas necesarias

Sector	Fecha	Distancia viajada	Cuadras salteadas	Tiempo de viaje	% del sector visitado
Castelar4	26/04/2013	46 km	60	4:10hs	88,5 %
Castelar7	19/04/2013	43,1 km	6	3:40hs	98,6 %
Haedo2	23/04/2013	17,5 km	20	2:26hs	89,7 %
Palomar1	10/04/2013	25 km	46	3:38hs	84,5 %
Palomar2	10/04/2013	24,5 km	35	4:10hs	87,5 %
Palomar3	24/04/2013	31,2 km	17	3:12hs	94,8 %
Palomar4	27/03/2013	35,2 km	35	4:46hs	91,0 %
VillaSarmiento1	06/04/2013	24,6 km	10	3:09hs	96,1 %
VillaSarmiento2	13/04/2013	23,7 km	9	2:44hs	96,3 %
VillaSarmiento3	15/04/2013	23,2 km	17	3:13hs	93,2 %

Tabla 2: Datos de GPS proporcionados por la municipalidad.

para recorrer 46 km nos permitieron asumir que la nueva ruta se podía recorrer en las 5 horas disponibles. Estas estimaciones realizadas a priori mostraron ser razonables cuando se aplicaron las nuevas rutas en la práctica.

Sector	$ V $	$ E $	$A_M$	$A_{AUX}$	Tiempo	Gap	Iteraciones	Costo
Castelar4	878	354	85	876	600 seg	0.008	1	50.45 km
Castelar7	1082	524	17	1341	646 seg	0.003	2	40.45 km
Haedo2	1154	396	181	1256	0.3 seg	0	1	18.31 km
Palomar1	1046	502	21	1251	30 seg	0	1	33.51 km
Palomar2	822	316	95	875	512 seg	0	1	26.90 km
Palomar3	1254	608	19	1596	1200 seg	0.003	2	30.53 km
Palomar4	1090	500	45	1353	1200 seg	0.003	2	30.53 km
VillaSarmiento1	304	12	140	258	1 seg	0	1	24.17 km
VillaSarmiento2	600	134	166	494	1 seg	0	1	24.25 km
VillaSarmiento3	384	76	116	322	1 seg	0	1	25.08 km

Tabla 3: Resultados del procedimiento para los sectores de la Tabla 2.

Los tiempos de resolución no superaron los 20 minutos para ninguno de los sectores. Estos tiempos son razonables para los requerimientos del sistema de reciclado, dado que una vez que se define una ruta, solamente se requieren cambios ocasionales en el futuro. Los gaps de optimalidad son pequeños o nulos, mostrando que el límite de 10 minutos para la resolución de SCIP es adecuado para estas instancias.

Es importante notar que las rutas obtenidas por este procedimiento fueron bien recibidas por la municipalidad. Se realizaron estudios de campo antes de implementar efectivamente las rutas para analizar su factibilidad, y las reacciones de los usuarios fueron positivas.

Cabe notar que en todos los casos las rutas óptimas obtenidas pueden ser recorridas en el tiempo disponible. Si variaran las instancias a resolver se podría dar la situación de que algunos recorridos óptimos fueran demasiado largos y no alcanzara el tiempo para realizarlos. En ese caso una posible variante sería maximizar el número de cuadras a ser visitadas incorporando al modelo una restricción que fije como cota superior el mayor número de cuadras que pueden ser recorridas en el tiempo disponible (y obviamente no obligando a pasar por todas las cuadras).

---

## 5. Conclusiones

---

Se propuso en este trabajo una metodología basada en programación lineal entera para optimizar las rutas de recolección de residuos reciclables en una municipalidad del Gran Buenos Aires. El problema a resolver corresponde a una variante del problema del cartero rural abierto, con restricciones que reflejan las reglas de tránsito de Argentina. El procedimiento presentado en este trabajo permite manejar prohibiciones de giros en algunas esquinas con semáforos, prohibiciones de giros en U, la selección del punto de inicio de la ruta, la inclusión de bandas perimetrales alrededor del sector a recolectar, y el sentido de circulación de las calles.

Las rutas de los vehículos de reciclado generadas por esta metodología e implementadas por la municipalidad tienen una performance significativamente mejor que las rutas diseñadas manualmente que estaban en uso antes de este trabajo. El mayor avance está dado en la cobertura provista por el servicio de recolección, que ahora llega a todas las cuadras de la municipalidad. Antes de esta implementación, hasta 60 cuadras se salteaban en algunos sectores. En algunos sectores, se logró esta mejora disminuyendo también la distancia total recorrida, con relación a las rutas manuales. En unos pocos casos la distancia total aumentó levemente.

Luego de varios meses de testeo de las rutas definidas por el procedimiento propuesto en este trabajo, la municipalidad comenzó su implementación a mediados de 2014. Esta implementación llevó a una mejor cobertura de la municipalidad, y a una notable reducción en los reclamos relacionados con la recolección de residuos reciclables. Debido al éxito de este proyecto, los autores han sido invitados por el Ministerio de Interior y Transporte de la Nación para reproducir la experiencia en otras municipalidades a lo largo del país.

**Agradecimientos:** Este estudio fue financiado parcialmente por los proyectos UBACyT 20020130100808BA (Argentina), ANPCyT PICT 2012-1324 (Argentina) y FONDECyT 1140787 (Chile), por el Instituto Milenio de Sistemas Complejos de Ingeniería (Chile) y por la firma papyro.com (Argentina). Los autores quisieran agradecer a la municipalidad de Morón, por su apoyo a este proyecto, y al revisor anónimo, por sus muy interesantes observaciones que permitieron mejorar la versión final de este trabajo.

## Referencias

- [1] C. Arribas, C. Blazquez, y A. Lamas. Urban solid waste collection systems using mathematical modelling and tools of geographic information systems. *Waste Management & Research*, 24(4):355–363, 2010.
- [2] F. Bonomo, G. Durán, F. Larumbe, y J. Marengo. A method for optimizing waste collection using mathematical programming: A buenos aires case study. *Waste Management & Research*, 30(3):311–324, 2012.
- [3] N. Chang, H. Lu, y L. Wei. GIS technology for vehicle routing and scheduling in solid waste collection systems. *Journal of Environmental Engineering*, 123:901–933, 1997.
- [4] N. Christofides, V. Campos, A. Corberán, y R. Mota. An algorithm for the rural postman problem on a directed graph. *Mathematical Programming Study*, 26:155–166, 1986.
- [5] A. Corberán, R. Martí, E. Martínez, y D. Soler. The rural postman problem on mixed graphs with turn penalties. *Computers and Operations Research*, 29(7):887–903, 2002.
- [6] A. Corberán, R. Martí, y A. Romero. Heuristics for the mixed rural postman problem. *Computers and Operations Research*, 27(2):183–203, 2000.
- [7] A. Corberán, E. Motta, y J. Sanchis. A comparison of two different formulations for arc routing problems on mixed graphs. *Computers and Operations Research*, 33(12):3384–3402, 2006.
- [8] A. Corberán, I. Plana, y J. Sanchis. The rural postman problem on directed, mixed, and windy graphs. In: *Arc Routing: Problems, Methods, and Applications, MOS-SIAM Series on Optimization, MO20*, pages 101–127, 2014.

- [9] H. Eiselt, M. Gendreau, y G. Laporte. Arc routing problems, part ii: The rural postman problem. *Operations Research*, 43(3):399–414, 1995.
- [10] D. Eisenstein y A. Iyer. Garbage collection in Chicago: a dynamic scheduling model. *Management Science*, 43:922–933, 1997.
- [11] B. Kim, S. Kim, y S. Sahoo. Waste collection vehicle routing problem with time windows. *Computers and Operations Research*, 33:3624–3642, 2006.
- [12] M. Mourao y M. Almeida. Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. *European Journal of Operational Research*, 121:420–434, 2000.
- [13] W. Pearn y T. Wu. Algorithms for the rural postman problem. *Computers and Operations Research*, 22(8):819–828, 1995.
- [14] A. Rodrigues y J. Ferreira. Solving the rural postman problem by memetic algorithms. *Proceedings of MIC'2001 - 4th Metaheuristics International Conference*, pages 679–683, 2001.
- [15] H. Thimbleby. The directed chinese postman problem. *Software - Practice and Experience*, 33(11):1081–1096, 2003.
- [16] J. Yeomans, G. Huang, y R. Yoogalingam. Combining simulation with evolutionary algorithms for optimal planning under uncertainty: An application to municipal solid waste management planning in the regional municipality of hamilton-wentworth. *Journal of Environmental Informatics*, 2(1):11–30, 2003.

