
ALGORITMOS DE ZONIFICACIÓN PARA EL PROBLEMA DE LA RECOLECCIÓN DE RESIDUOS URBANOS: EL CASO DE ESTUDIO DE UNA CIUDAD ARGENTINA

MARCELO BIANCHETTI *
GUILLERMO DURÁN **
IVO KOCH ***
JAVIER MARENCO ****

Resumen

La programación de los aspectos logísticos de la recolección de residuos urbanos plantea una serie de problemas de optimización combinatorial de difícil resolución en la práctica. Uno de estos problemas consiste en la división del área a recorrer en zonas, cada una de ellas correspondiente a un camión recolector. Para ello es preciso respetar restricciones de volumen de recolección por cuadra y sobre la flota de vehículos, así como también restricciones cualitativas de similitud de las zonas entre sí. En este artículo analizamos el problema aplicado a la recolección de residuos en la ciudad de San Miguel de Tucumán en Argentina. Estudiamos para ello distintos algoritmos heurísticos de zonificación que nos permiten proponer una partición en zonas alternativa a la actual, de forma tal de minimizar la cantidad de camiones utilizados.

Palabras Clave: Recolección de residuos, Zonificación, Heurísticas.

* Instituto de Industria, Universidad Nacional de General Sarmiento, Buenos Aires, Argentina

** Departamento de Matemática e Inst. de Cálculo FCEN-UBA y CONICET, Argentina, Departamento de Ingeniería Industrial, FCFM, Universidad de Chile, Santiago, Chile

*** Instituto de Industria, Universidad Nacional de General Sarmiento, Buenos Aires, Argentina

**** Instituto de Ciencias, Universidad Nacional de General Sarmiento y Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina

1. Introducción

La recolección de residuos ha sido objeto de numerosos trabajos de investigación debido a la amplitud de problemas que surgen de la misma, tanto para el área de grafos como de la programación lineal entera; pero el mayor esfuerzo se ha hecho sobre la optimización del ruteo de vehículos y la asignación de horarios (ver por ejemplo [1, 5, 9, 13, 15, 19, 20, 21, 23, 27]). El presente trabajo aborda un problema distinto: el de la organización en zonas de una región de manera tal de minimizar la flota de camiones a utilizar. Este problema de *diseño territorial*, esto es, de agrupamiento de áreas geográficas básicas en zonas más grandes que sean aceptables de acuerdo a determinados criterios de planeamiento fue estudiado como parte de numerosas aplicaciones. Aparece en diseño de distritos electorales [16], de áreas de ventas [6], de distritos escolares [10], de territorios de servicio para prestaciones sociales como hospitales o unidades administrativas [22], y de distritos policiales [4], entre otros. Puede encontrarse una muy buena reseña en [17]. El problema de diseño territorial aplicado a la recolección de residuos y a operaciones de remoción de nieve y esparcimiento de sal en las calles fue analizado en [14] y [24], respectivamente. Los autores abordaron el problema de la recolección de residuos utilizando otras técnicas a las estudiadas aquí en [2] y [3].

El caso de estudio de del presente artículo será el Municipio de San Miguel de Tucumán, ciudad capital de la Provincia de Tucumán, situada en el noroeste de la Argentina.

El servicio de recolección municipal de residuos en esta ciudad se lleva a cabo en una franja horaria diurna y una nocturna. Esto se debe en parte a la falta de camiones suficientes para cubrir todas las zonas en paralelo, y en parte a cuestiones de tránsito y seguridad.

De martes a sábados, la cantidad de camiones con la que se cuenta alcanza para cubrir íntegramente las zonas; pero al no haber recolección los domingos, los lunes se incrementa el volumen de residuos a recolectar, formando un cuello de botella en el área del microcentro, que es un área densamente poblada delimitada por cuatro grandes avenidas. Aquí sus recorridos se trazan de una manera especial, haciendo un barrido horizontal y otro vertical. Es decir que los camiones sólo la recorren en líneas rectas que entran o salen de la misma, girando sólo fuera de ésta. La manera requerida por parte del municipio para optimizar este problema es asignar mayor cantidad de camiones a esta área. Como la cantidad de camiones es fija, para ello es preciso retirarlos de otras zonas de la misma franja horaria. A tal fin, desarrollamos en este trabajo heurísticas basadas en programación lineal entera con el objetivo de rezonificar el área nocturna recuperando la mayor cantidad de camiones posible para destinarlos al área del microcentro. Las heurísticas desarrolladas son originales

y difieren de las halladas en la bibliografía. En [14] los autores plantean técnicas metaheurísticas basadas en modificar sucesivamente los “centros” de las zonas: asignar el centro de una zona a otra o intercambiar dos centros, para intentar mejorar la zonificación; en nuestro trabajo los centros son un dato de entrada para el algoritmo y no se modifican dinámicamente. Por otra parte, el trabajo [24] coincide en una de las ideas de este artículo, de hacer crecer las zonas agregando ciclos. Sin embargo, esta estrategia fue reemplazada en este estudio por la agregación de aristas individuales a las zonas, por dar mejores resultados computacionales.

El presente trabajo se estructura de la siguiente manera: en el resto de esta sección precisamos en detalle los requerimientos del problema y damos algunas definiciones preliminares. En la Sección 2 desarrollamos los algoritmos que empleamos en la resolución. A continuación, la Sección 3 muestra los resultados computacionales obtenidos. La última sección indica algunas posibles líneas de trabajo futuro.

1.1. Consideraciones particulares del problema

El servicio se cumple en 40 zonas diurnas y nocturnas disjuntas, que abarcan 12.320 cuadras aproximadamente. Actualmente el área diurna cubre un total de 6.982 cuadras, de las cuales 1.710 pertenecen a calles o avenidas mano única mientras que 5.272 a calles o avenidas doble mano. Esta región se divide en 19 zonas con un camión asignado a cada una. El servicio nocturno cubre 21 zonas ubicadas en el centro de la ciudad con un total de 5.338 cuadras, de las cuales 3.515 pertenecen a avenidas y calles mano única, y 1.823 doble mano. Dentro de estas 21 zonas céntricas, el microcentro de la ciudad delimitado por las avenidas Sarmiento al Norte, Avellaneda - Sáenz Peña al Este, Roca al Sur y Alem - Mitre al Oeste es cubierto por 9 camiones. Este microcentro está compuesto por 2.092 cuadras. Las 3.246 cuadras restantes de la región, de las cuales 1.696 pertenecen a calles o avenidas mano única, y 1.550 a avenidas o calles doble mano son cubiertos por los 12 camiones restantes.

Procuraremos que las zonas en que buscamos dividir el área nocturna del municipio sean similares; es decir, que un camión tarde aproximadamente el mismo tiempo en recolectar los residuos en cada una de ellas. Buscamos entonces la menor varianza del conjunto de los tiempos que tardan todos los camiones en realizar sus servicios respectivos.

En las áreas a zonificar de la ciudad en cuestión hay contenedores y recolección domiciliaria. Existen ambas modalidades porque no en todas las cuadras existen contenedores; en las que carecen de ellos, la recolección es por cada domicilio. Por ello, es necesario que los camiones recorran absolutamente todas las calles.

Según los datos brindados por el municipio, el tiempo que consume un camión en recolectar residuos en una cuadra de una zona nocturna es 5 veces mayor en promedio que el que tarda en atravesarla sin recolectar, y 3 veces

mayor en el caso de la zona diurna. La razón de esta diferencia es que las zonas nocturnas cuentan con edificios altos y son más densamente pobladas, por lo que el tiempo de recolección en cada cuadra se incrementa. Se considera que el volumen de recolección dentro de cada una de las zonas (excluido el microcentro) es aproximadamente lineal con respecto a las longitudes de las cuadras a recolectar.

Sea la velocidad promedio de un camión V_p y V_r en cuadras de paso y recolección respectivamente. Se estima que $V_p = 30$ km/h mientras que $V_{r_{nocturna}} = 6$ km/h y $V_{r_{diurna}} = 9$ km/h, aproximadamente. Como el personal de recolección trabaja 8 hs diarias, la zona de cobertura de un camión en la región nocturna se limita a 48 km, mientras que en las diurnas ésta puede tener una longitud de 72 km.

Un camión puede recorrer calles de zonas vecinas si de esa manera reduce el tiempo total empleado para recorrer su zona. Asumimos que la capacidad de los camiones es suficiente para recolectar los residuos de su zona sin necesidad de descargar. Del mismo modo se considera que la autonomía de los vehículos es suficiente para salir del depósito, recolectar y volver a descargar, sin necesidad de recargar combustible.

1.2. Descripción formal del problema

Para las definiciones, propiedades y algoritmos de teoría de grafos no precisadas aquí, remitimos al lector a [26]. El mapa M de la región a estudiar se modelará como un grafo mixto $G = (V_G, E_G, A_G)$, tal que el conjunto de aristas no dirigidas $E_G \subseteq (V_G \times V_G)$ representará los segmentos de las calles y avenidas doble mano, sus arcos dirigidos $A_G \subseteq (V_G \times V_G)$ serán los segmentos de calles y avenidas mano única (su sentido estará dado por el orden de sus nodos), y el conjunto de vértices V_G representará los extremos e intersecciones de segmentos de caminos vehiculares.

Un k zonificado será una partición del grafo G en k subgrafos conexos (denominados *zonas* o *áreas*) $G_i = (V_i, E_i, A_i)$ tales que $\forall i: V_i \subseteq V_G, E_i \subseteq E_G \cap (V_i \times V_i), A_i \subseteq A_G \cap (V_i \times V_i)$, cumpliendo además que $\forall i \neq j: E_i \cap E_j = \emptyset, A_i \cap A_j = \emptyset$.

Para calcular el tiempo que consumiría un camión en recolectar todos los residuos de su zona es necesario resolver el problema del *cartero chino* (CPP) [7, 8, 18, 25] sobre el subgrafo asociado. Formalmente, dado un grafo mixto $G = (V, E)$ con longitudes positivas asignadas a sus aristas $L: E \rightarrow \mathbb{R}_+$, el problema del cartero chino consiste en encontrar un circuito de longitud mínima que pase por cada arista de G al menos una vez.

En un grafo dirigido o no dirigido el CPP puede ser resuelto en tiempo polinomial. Por otro lado, si el grafo es mixto el problema de decisión asociado es NP-Completo, aún si todas las longitudes de las aristas son iguales, el grafo es planar y su grado máximo es 3 [12]. Como la zonificación involucra resolver el CPP sobre grafos mixtos, nuestro problema es a su vez NP-Completo.

Una condición necesaria y suficiente para que exista una solución del CPP, es que G sea fuertemente conexo. Un grafo con estas características se denominará *factible*.

Como el origen del grafo es un mapa, debe ser factible. Además, como la simplificación preserva factibilidad, entonces para zonificar es necesario dividir un grafo G , que se sabe factible, en k subgrafos (G_1, \dots, G_k) disjuntos que se desean factibles. Este objetivo no siempre es alcanzable. Por ejemplo, si G es un grafo consistente en un único circuito simple (sin nodos repetidos), no es posible dividirlo en k subgrafos factibles disjuntos, para cualquier valor de $k > 1$.

Para cualquier particionamiento (G_1, \dots, G_k) del grafo factible $G = (V_G, E_G, A_G)$ en k subgrafos disjuntos, existen k conjuntos de aristas (C_1, \dots, C_k) tales que $G_i \cup C_i$ es factible para todo $i = 1 \dots k$. Sea $G_i = (V_{G_i}, E_{G_i}, A_{G_i})$, basta tomar $C_i = (A_G \cup E_G) \setminus (A_{G_i} \cup E_{G_i})$. Luego, siempre es posible factibilizar las zonas agregándoles aristas del grafo G . Por eso en este trabajo se definió que una zona está compuesta por aristas *obligatorias*, que son las que representan los caminos que un camión *debe* recolectar, y *optativas*, que son las que representan los caminos que *puede* atravesar (sin recolectar).

Definimos en este trabajo el peso $L : (E_G \cup A_G) \rightarrow \mathbb{R}^+$ para las aristas optativas de G como la distancia euclideana de los nodos que componen la arista sobre la cual se evalúa. El peso de las aristas obligatorias será 5 veces este valor (aplicaremos nuestros algoritmos a las zonas nocturnas). El tiempo que consume una arista en el recorrido de un camión entonces estará dado por $T : E_G \cup A_G \rightarrow \mathbb{R}$, tal que:

$$T(e) = \begin{cases} L(e)/V_r & \text{si } e \text{ es arista obligatoria} \\ L(e)/V_p & \text{si } e \text{ es arista optativa.} \end{cases}$$

Para calcular el *costo* de la zona, es decir el tiempo consumido por el camión en recorrerla recolectando los residuos de sus calles obligatorias, es necesario resolver esta variante del cartero chino sobre el subgrafo mixto asociado factibilizado con arcos y aristas optativas. Como un arco o una arista obligatoria, una vez recolectados los residuos, se convierte en arco o arista de paso, se determinó para el cálculo del circuito que todos los arcos y las aristas fueran tratadas como de paso, diferenciando únicamente las obligatorias forzando a que aparezcan al menos una vez. Luego, al circuito resultante se le suma el tiempo que consume la recolección. Si denominamos $f = \frac{\text{costo}_{\text{recoleccion}}}{\text{costo}_{\text{paso}}}$ entonces este tiempo estará dado por $f - 1$ veces el costo de pasar por los arcos y aristas obligatorias de la zona (se decrementa en 1 por ya haber contado una vez cada arista obligatoria).

Dos zonas se dirán *similares* si un camión tarda aproximadamente el mismo tiempo en recolectar los residuos en cada una de ellas.

Una *zonificación óptima* de la región en k zonas se definirá como aquel particionamiento del grafo asociado en k subgrafos conexos similares, siendo

además el promedio de los resultados del CPP en todos ellos el menor posible y dando prioridad a un grado de similitud mayor.

El requisito informal de la forma de las áreas se traduce en buscar que los subgrafos sean conexos, y que la varianza de las distancias euclidianas de los nodos más alejados del centro de la misma en las distintas direcciones sea pequeña.

2. Algoritmos desarrollados

La idea principal del método de resolución es generar las áreas en forma iterativa e incremental, de manera tal que en cada una de estas iteraciones la similitud entre ellas sea la mayor posible.

Como los subgrafos que representan las áreas generadas hasta el momento deben representar submapas del mapa original, buscamos que sea posible recorrerlos, i.e., que sea posible resolver en ellos el problema del cartero chino. Por este motivo se pedirá que dichos subgrafos sean fuertemente conexos. La manera de garantizar ese requisito será demostrar que cada subgrafo se puede describir como una unión de circuitos.

De aquí surgen dos de los principales criterios analizados en este trabajo: agregar en cada iteración o bien un circuito, o bien una arista (o arco). En el segundo caso será necesario agregarle al área como optativos un conjunto de circuitos que la cubran completamente para garantizar factibilidad.

Para decidir a qué zona agregar la arista o arco o circuito la hipótesis fundamental del presente trabajo es que la solución del CPP está fuertemente ligada con la cantidad de aristas y arcos que posee el grafo a resolver. Por este motivo se analizaron diversos criterios de manera de intentar agregar la arista o circuito a aquel grafo cuyo resultado del CPP fuera menor. Se procura además minimizar la cantidad de ejecuciones del CPP dado su costo computacional.

El Algoritmo 1 resume estas ideas. La resolución comienza convirtiendo un mapa de la región del municipio actualmente cubierta por el turno nocturno en formato *openStreetMap* (.osm) [11] al digrafo G que lo modela (línea 2), seguido de una simplificación del mismo que da lugar a un grafo equivalente H (línea 3). Posteriormente se verifica la factibilidad de H , se generan las distintas zonas H_i de forma heurística (línea 9), se factibilizan (línea 11) y resuelven los modelos de programación lineal entera mixta (PLEM) asociados para obtener una solución del problema del cartero chino en cada una (líneas 12 y 13). Esas soluciones son por último reescritas como recorridos (línea 14). Dedicamos el resto de esta sección al análisis de cada uno de estos pasos.

2.1. De un mapa al modelo

Los datos del mapa M del municipio en formato .osm constan de un conjunto de nodos $N_M = \{n_1, \dots, n_j\}$, un conjunto de caminos $W_M = \{w_1, \dots, w_k\}$,

Algoritmo 1 Algoritmo de zonificación

```

1: function ZONINGAMAP( $M : osmMap, K : osmKnodes, H : heuristic$ )
2:    $G := osmToGraph(M)$ 
3:    $S := simplifyGraph(G)$ 
4:    $S, B, E := graphByCircuits(S)$ 
5:   if not  $checkFactibility(S)$  then
6:      $printErrors(S, G, B)$ 
7:     STOP
8:   end if
9:    $GrownAreas := growAreas(S, K, B, H)$ 
10:  for  $k \in K$  do
11:     $Area_k := zoning(GrownAreas, k)$ 
12:     $LP_k := graphToModel(Area_k)$ 
13:     $Solution_k := LPSolver.Solve(LP_k)$ 
14:     $Path_k := solutionToPath(Solution_k, S, G)$ 
15:  end for
16: end function

```

y un conjunto de restricciones R_M . Cada *nodo* $n_i \in N_M$ tiene asociado un par de coordenadas $(latitud_i, longitud_i) \in \mathbb{R}^2$ y un conjunto de atributos. Un *camino* $w_i \subseteq N_M \times \dots \times N_M$ está definido como una sucesión ordenada de un subconjunto del conjunto de nodos y un listado de atributos. Dependiendo de éstos, el camino representará una calle doble mano, una avenida, un río, una vía de tren, el perímetro de un edificio público, etc. En caso de ser una calle mano única, el orden de los nodos indicará el sentido de la misma. Las *restricciones* $r_i \subseteq W_M \times N_M \times W_M$ determinan los giros prohibidos de un camino a otro camino a través de un nodo.

La conversión de este formato al grafo G sobre el cual se trabajará vendrá dada por:

- el conjunto de aristas no dirigidas $E_G \subseteq (V_G \times V_G) \subseteq (N_M \times N_M)$ cumplirá que $\forall (x, y) \in E_G : \exists w_i \in W_M$ tal que $w_i = (n_1, \dots, x, y, \dots, n_n)$ y cuyos atributos indican avenida o calle residencial de doble sentido.
- sus arcos dirigidos $A_G \subseteq (V_G \times V_G)$ cumplirán que $\forall (x, y) \in A_G : \exists w_i \in W_M$ tal que $w_i = (n_1, \dots, x, y, \dots, n_n)$ y cuyos atributos indican avenida o calle residencial de un solo sentido.
- el conjunto de vértices $V_G \subseteq N_M$ cumplirá que $v \in V_G \leftrightarrow \exists y : (v, y) \in (E_G \cup A_G) \vee (y, v) \in (E_G \cup A_G)$.

Se utilizará la distancia terrestre entre dos nodos para la función de peso $L : (E_G \cup A_G) \rightarrow \mathbb{R}^+$ para las aristas de G .

En este trabajo realizamos una simplificación del mapa original, para obtener instancias equivalentes más reducidas. En los mapas utilizados las curvas se representan mediante varios segmentos rectos para representar más fielmente la realidad, y para evitar superposición de caminos, por lo que una cuadra puede comprender un conjunto de aristas o arcos; reemplazamos estos casos por un solo segmento recto. Además, eliminamos los nodos *redundantes*, que unen exactamente dos segmentos con el mismo sentido y categoría (calle residencial o avenida). Denominaremos $H = (V_H, E_H, A_H)$ al grafo simplificado que resulta de la transformación, que describimos en detalle debajo:

i) El conjunto de aristas no dirigidas $E_H \subseteq (V_H \times V_H) \subseteq (V_G \times V_G)$ representan los segmentos comprendidos entre los extremos, intersecciones o cambios de categoría (esto es, avenida o calle) de los caminos doble mano. Cumplen la propiedad de que para toda arista (x, y) del grafo H , con $x \neq y$, existe un camino $C = \{n_1, \dots, n_k\} \subseteq V_G$ compuesto por aristas doble mano de G tal que todos los nodos internos de C son redundantes salvo sus extremos n_1 y n_k . Además se debe cumplir una de las tres condiciones siguientes: o bien $C = (x, y)$ o bien $(x, y) = (n_1, n_k) \notin E_G$ y ninguna de las aristas que forman a C están en H , o bien (x, y) pertenece a C siendo que o C es un circuito o la arista que reemplazaría al camino C ya existe en el grafo H .

ii) Los arcos dirigidos $A_H \subseteq (V_H \times V_H) \subseteq (V_G \times V_G)$ representan los segmentos comprendidos entre los extremos, intersecciones o cambios de categoría (esto es, avenida o calle) de los caminos de única mano. Verifican que para todo $(x, y) \in A_H$, con $x \neq y$ existe un camino dirigido $C = \{n_1, \dots, n_k\} \subseteq V_G$ compuesto por arcos pertenecientes a A_G con todos sus nodos internos redundantes, sus extremos n_1 y n_k no redundantes. Además, debe cumplirse que $C = (x, y)$, o bien $(x, y) = (n_1, n_k) \notin A_G$ y ninguno de los arcos que forman a C están en H , o bien (x, y) pertenece a C siendo que o C es un circuito o el arco que reemplazaría al camino C ya existe en el grafo H .

iii) El conjunto de vértices $V_H \subseteq V_G$ representa únicamente los extremos e intersecciones de caminos vehiculares, o la unión de dos segmentos de los mismos con distinto sentido y/o categoría. Se cumplirá que un vértice v pertenecerá a V_H si y sólo si existe un vértice y tal que la arista o arco (v, y) o su inversa es un arco o arista del grafo H .

2.2. Verificación de factibilidad general

Para verificar la factibilidad del grafo asociado al mapa a zonificar, se han utilizado dos algoritmos de complejidad polinomial que recurren a la siguiente propiedad, que puede probarse fácilmente.

Proposición 2.1. *Sea $G = (V_G, E_G, A_G)$ un grafo mixto, si vale que el grafo no dirigido $F = (V_F, E_F) : V_F = V_G$ y $E_F = (A_G \cup E_G)$ es conexo, y si toda arista de G tiene un circuito que la contiene, es decir $\forall (x, y) \in (E_G \cup A_G)$,*

existe un camino C tal que $C = \{(y, n_1), \dots, (n_k, x)\} \subseteq (E_G \cup A_G)$, entonces G es fuertemente conexo.

Por lo tanto lo que se buscó fue que existiera un circuito para cada arista y arco de G , y que el grafo F asociado fuera conexo. Para hallar un circuito para cada arista del grafo mixto $H = (V_H, E_H, A_H)$ se utilizó el algoritmo de Dijkstra para camino mínimo entre dos nodos (es decir, la versión que detiene el algoritmo original al hallar la solución para los nodos en cuestión) aplicado entre cada par distinto (x, y) con $(y, x) \in A_H$. El camino C hallado que une y con x es mínimo. Un arco sin circuito implica que el problema del cartero chino no tiene solución. Un mapa de una ciudad tiene que ser factible por ser la representación de las calles utilizadas para ir de algún lugar a cualquier otro. De existir arcos sin circuito, puede ser que los datos sean incorrectos, con lo que debe corregirse el mapa; o bien puede darse que este conjunto de calles intransitables se deba a un mal recorte del mapa original. Esto es, que se le haya quitado un segmento a algún camino que sale y entra a la ciudad. En el grafo simplificado H , el conjunto de arcos sin circuito está compuesto por aquellos que representan calles mano única pertenecientes a un camino sin salida o sin entrada. A su vez dichos arcos pueden particionar el grafo original generando circuitos desconectados del resto del grafo. La manera de solucionar este inconveniente es eliminar las aristas sin circuito.

Verificamos si el grafo H es conexo mediante el algoritmo *Breadth First Search (BFS)*, aplicado al grafo no dirigido $F = (V_F, E_F) : V_F = V_H \wedge E_F = (A_H \cup E_H)$, revisando que todos los nodos sean alcanzables a partir de uno cualquiera.

2.3. Generación de los subgrafos

Sean un grafo mixto $F = (V_F, E_F, A_F)$ y un circuito mixto $C = (V_C, E_C, A_C)$, definimos la *unión* de ambos como el grafo $(C \cup F) = (V_F \cup V_C, E_F \cup E_C, A_F \cup A_C)$. Decimos que dicha unión es *conexa* si $(V_C \cap V_F) \neq \emptyset$.

Proposición 2.2. *Sea G un grafo mixto factible, y C un circuito mixto, si la unión $G \cup C$ es conexa entonces el grafo resultante es factible.*

Corolario 2.1. *Una unión finita conexa de circuitos determina un grafo factible.*

En consecuencia, partiendo de un subgrafo H_i del grafo factible H , es posible agregar circuitos de H a H_i hasta dejarlo factible si éstos contienen las aristas que en H_i no pertenecen a ningún circuito.

Para generar una partición del grafo H que se sabe factible en subgrafos similares, se evaluó un conjunto de heurísticas que coinciden estructuralmente en la serie de pasos especificada en el Algoritmo 2, pero difieren en la combinación de criterios utilizados.

Estudiamos a continuación los criterios para dividir las aristas del grafo original en conjuntos.

Algoritmo 2 growAreas

- 1: Partir de una ubicación tentativa de los k centros de las zonas a generar. Estos centros, que tienen latitud y longitud, se denominan k -nodos. Asignar a cada uno un grafo vacío.
 - 2: Dividir las aristas del grafo original en h conjuntos según algún criterio (el particionamiento por aristas o por circuitos descrito debajo).
 - 3: Calcular las distancias de dichos conjuntos a los k -nodos. Para esto es necesario definir una noción de distancia.
 - 4: Mientras sea posible, seleccionar el k -nodo con el grafo más reducido (también debe definirse este criterio), y agregarle los nodos y aristas correspondientes al conjunto de aristas más cercano al mismo, de forma que lo mantengan conexo. Si no hay, marcar dicho k -nodo como *terminado* e intentar con el siguiente. El procedimiento continúa hasta marcar todos los k -nodos como terminados.
-

- **Particionamiento por Circuitos.** Este procedimiento utiliza la división en circuitos mencionada anteriormente. Este criterio garantiza factibilidad, pero genera zonas no disjuntas. Luego de finalizada su ejecución, debemos determinar a qué zona pertenece cada una de las aristas de las intersecciones.
- **Particionamiento por Aristas.** Con este criterio en el que cada arista es un conjunto, se garantiza que las zonas generadas sean disjuntas, pero no se tiene noción sobre su factibilidad. Por este motivo será necesario garantizar la factibilidad mediante el agregado de circuitos opcionales. Para ello se agregan para cada arista sin circuito, en carácter de optativas las aristas pertenecientes al circuito mínimo que la contiene.

Definimos además dos nociones de distancia entre un punto k del espacio y un conjunto no nulo de aristas C , en ambos $dist(x, y)$ definida entre dos puntos es la distancia euclideana.

- **Lejano:** $dist(k, C) = \text{máx}\{dist(k, y) | y \in e \wedge e \in C\}$.
- **Promedio:** $dist(k, C) = \text{prom}\{dist(k, y) | y \in e \wedge e \in C\}$.

Como en cada paso es necesario determinar si un grafo es menor que otro en relación con el problema a resolver se han definido tres criterios. Para las siguientes definiciones se utiliza que C es la solución óptima del CPP en el grafo G , $obl(G)$ y $opt(G)$ son los conjuntos de aristas y arcos obligatorios y optativos respectivamente de G , y $peso(e)$ determina el peso de una arista o arco.

- **CPP (G)** = $F_{obl} \sum_{e \in obl(C)} peso(e) + F_{opt} \sum_{e \in opt(C)} peso(e)$.

Con este criterio, es necesario factibilizar el grafo cada vez que se lo quiera utilizar. Requiere definir dos factores $F_{obl}, F_{opt} \in \mathbb{R}$ que marcan la importancia que se le da a las aristas optativas u obligatorias al determinar el peso del grafo. En este trabajo se deja fijo F_{opt} y se varía solamente F_{obl} .

- **Peso Total (G)** = $F_{obl} \sum_{e \in obl(G)} peso(e) + F_{opt} \sum_{e \in opt(G)} peso(e)$.

Más simple que el criterio anterior, reemplaza resolver el CPP por sumar las aristas del grafo factibilizado.

- **Peso Parcial (G)** = $\sum_{e \in obl(G)} peso(e)$.

Este criterio toma en cuenta solamente las aristas obligatorias. Puede verse como un caso particular de Peso Total en el que $F_{obl} = 0$ y $F_{opt} = 1$, pero se ha considerado aparte porque no requiere que el grafo G sea factible, lo cual acelera su cálculo.

Por último, una heurística determina cuál de esos criterios se aplica en cada paso, es decir para seleccionar el menor subgrafo al cual agregarle el conjunto de aristas más cercano. Las desarrolladas en el presente trabajo han sido las siguientes:

- **CPP/Total/Parcial:** Usar en todos los pasos el mismo criterio según se decida al inicio del algoritmo de generación de las zonas: si las áreas son G_i para $i = 1, \dots, k$, en cada paso el grafo al cual agregar el conjunto de aristas (según el criterio elegido) estará dado por

$$\min_{i=1}^k \text{PesoParcial}(G_i) \quad \text{ó} \quad \min_{i=1}^k \text{PesoTotal}(G_i) \quad \text{ó} \quad \min_{i=1}^k \text{CPP}(G_i).$$

- **CPP-N y Total/Parcial:** Usar el criterio CPP cada N pasos, y en los $N - 1$ intermedios usar Total ó Parcial según se decida al inicio de la generación de áreas. En la iteración j el grafo elegido estará dado por

$$\begin{cases} \min_{i=1}^k \text{CPP}(G_i) & \text{si } j \text{ mód } N = 0 \\ \min_{i=1}^k \text{PesoParcial}(G_i) \quad \text{ó} \quad \min_{i=1}^k \text{PesoTotal}(G_i) & \text{sino.} \end{cases}$$

- **CPP-N, Total/Parcial y Feedback:** Usar el criterio CPP cada N pasos, y en los $N - 1$ pasos intermedios utilizar una función de realimentación basada en los resultados del último paso en el que se resolvió CPP y el resultado de Total o Parcial según se decida al inicio del algoritmo de generación. Sea j un paso de la iteración, $h = j \text{ mód } N$. La selección de la menor área $G_i \in \{G_1, \dots, G_k\}$ estará dada por

$$\min_{i=1}^k \text{CPP}_{feed}(G_i),$$

donde

$$CPP_{feed}(G_i) = \begin{cases} F_m(G_i) & \text{si } h > 0 \\ P_{eso_j}(G_i) & \text{si } h = 0, \end{cases}$$

tal que F_m es una de las 4 funciones de realimentación desarrolladas con el fin de extender el impacto del resultado del cartero chino durante las iteraciones que siguen a su cómputo. La cantidad de estas iteraciones en las cuales se modifica el resultado del criterio común con la aplicación de la función de realimentación dependerá de una estimación de la cantidad de aristas o arcos a agregar para equilibrar todos los resultados del CPP calculado. Asimismo el grado de influencia de dicha función sobre cada área dependerá de la diferencia entre el resultado del criterio común en el comienzo de la realimentación con el resultado del mismo en la iteración actual. Definimos a continuación las funciones F_m .

Sea $n = \frac{j}{N}$. Para la primera función de realimentación, el valor resultante para el paso $n \times N$ debe ser el de resolver el CPP. En los pasos intermedios $nN + h < (n + 1) \times N$ este valor va aumentando junto con la diferencia entre el resultado de la heurística en el paso actual y el resultado en el paso nN . La forma de estimar la cantidad de aristas o arcos a agregar se aclara a continuación.

En primer lugar se intentó precisar de forma experimental cuánto influye una arista o arco del grafo en el recorrido solución del CPP. Se obtuvo como resultado

$$edge_w = \frac{CPP_{result}}{|edges|} \cong 0,7586.$$

Es decir que agregar una arista en promedio aumenta en 0,7586 el valor del resultado del CPP. Luego se buscó determinar la cantidad de aristas que se deben agregar en el paso nN para igualar todos los resultados de resolver el CPP en las áreas, lo cual se representa como:

$$|edgesToAdd|_{nN} = \sum_{x \in CPP_{results_{nN}}} \frac{[\text{máx}(CPP_{results_{nN}})] - x}{edge_w}.$$

Como en cada iteración se agrega una arista al sistema, dicho valor es disminuído en 1 cada paso hasta llegar a 0. Aquí termina la realimentación y se vuelve al criterio intermedio seleccionado. Entonces en los pasos j no múltiplos de N

$$|edgesToAdd|_j = \max(0, |edgesToAdd|_{(j-1)} - 1).$$

Con esos valores, la primer función de realimentación utilizada para calcular el valor resultante en cada zona en la iteración j queda definida como

$$F_1 = \begin{cases} CPP_{nN} + F_{feed} \times (P_{eso_j} - P_{eso_{nN}}) & \text{si } h > 0 \\ P_{eso_j} & \text{si } h = 0, \end{cases}$$

donde

$$\begin{aligned} CPP_{nN} &= CPP \text{ de la iteración } nN, \\ Peso_j &= \text{PesoParcial o } PesoTotal \text{ de la iteración } j, \\ Peso_{nN} &= \text{PesoParcial o } PesoTotal \text{ de la iteración } nN, \\ F_{feed} &= |edgesToAdd|_j. \end{aligned}$$

La segunda fórmula busca generar una combinación lineal entre los resultados de las dos heurísticas utilizadas, de manera tal que a medida que se suceden las iteraciones intermedias, disminuye el peso de la realimentación hasta resolver nuevamente el CPP (los parámetros fueron determinados mediante experimentación)

$$F_2 = \begin{cases} CPP_{nN} \times (F_{feed} - h) + 9 \times F_{feed} \times (Peso_j - Peso_{nN}) & \text{si } h > 0 \\ Peso_j & \text{si } h = 0. \end{cases}$$

La tercera y la cuarta función utilizan la misma idea

$$F_3 = \begin{cases} (CPP_{nN} + Peso_j - Peso_{nN}) \times F_{feed} + h \times Peso_j & \text{si } h > 0 \\ Peso_j & \text{si } h = 0, \end{cases}$$

$$F_4 = \begin{cases} (CPP_{nN} + done_j) \times F_{feed} + h \times Peso_j \times \frac{F_{obl}}{F_{opt}} & \text{si } h > 0 \\ Peso_j & \text{si } h = 0, \end{cases}$$

donde $done_j = (Peso_j - Peso_{nN}) \times \frac{F_{obl}}{F_{opt}}$, es decir cuánto aumentó el área desde la iteración nN hasta la j .

2.4. Factibilización de los subgrafos

Luego de generar las k áreas, éstas deben factibilizarse para cada iteración en la cual se agreguen aristas sueltas. Dada un área, se recorre su conjunto de aristas obligatorias y para cada una de éstas se agregan como optativas todas las aristas de alguno de los circuitos que la contienen. Estos circuitos son precomputados al dividir el grafo simplificado en circuitos en el tercer paso del algoritmo general.

En caso de que se esté utilizando el criterio que agrega circuitos, las zonas al finalizar cada iteración son ya factibles; lo que resta es determinar qué aristas son obligatorias y cuáles optativas en cada una de ellas. En el presente trabajo, por motivos que se detallarán en la sección de experimentos y resultados, no se desarrolló dicho algoritmo diferenciador, por lo que este criterio sólo es posible aplicarlo en los pasos intermedios tomando todas las aristas como obligatorias. Una vez generadas las áreas, para calcular los recorridos se da la opción de determinar manualmente los límites de las zonas marcando optativas y obligatorias.

2.5. Resolución del CPP

Sea el grafo $G = (V, E, A)$ la representación de un área con calles obligatorias y optativas, mano única y doble mano, en la cual E será un conjunto de pares ordenados, a pesar de representar aristas no dirigidas. Sean también las funciones *required* y *optional*, que evaluadas sobre una arista indican obligatoriedad o no de incluirla en el recorrido,

$w : (E \cup A) \rightarrow \mathbb{R}^+$ la función de peso de las aristas de G .

$$OBL_A = \{a \in A : \text{required}(a)\},$$

$$OBL_E = \{e \in E : \text{required}(e)\},$$

$$OBL_{Einv} = \{e_{ij} | e_{ji} \in E \wedge \text{required}(e_{ji})\},$$

$$OPT = \{e \in (A \cup E) : \text{optional}(e)\}.$$

El modelo tendrá una variable entera $obl_{uij} \geq 1$ para cada arista $e_{ij} \in OBL_A$ y una variable entera $opt_{ij} \geq 0$ para cada $e_{ij} \in OPT$, que denotarán la cantidad de apariciones de las aristas correspondientes en el circuito solución. Se contará además con dos variables enteras $obl_{dij}, obl_{dji} \geq 0$ para cada $e_{ij} \in OBL_E$ que representarán la cantidad de apariciones de cada arista obligatoria.

La función objetivo buscará minimizar la longitud de dicho circuito:

$$\sum_{e_{ij} \in OBL_A} w(e_{ij}) \times obl_{uij} + \sum_{e_{ij} \in OBL_E} w(e_{ij}) \times (obl_{dij} + obl_{dji}) + \sum_{e_{ij} \in OPT} w(e_{ij}) \times opt_{ij}.$$

Al ser conexo el conjunto de aristas obligatorias, la forma de obligar a que la solución sea un circuito es forzando la conservación del flujo en cada nodo:

$$\forall i \in V :$$

$$\sum_{j: e_{ji} \in OBL_A} obl_{uji} + \sum_{j: e_{ji} \in (OBL_E \cup OBL_{Einv})} obl_{dji} + \sum_{j: e_{ji} \in OPT} opt_{ji} = \sum_{j: e_{ij} \in OBL_A} obl_{uij} + \sum_{j: e_{ij} \in (OBL_E \cup OBL_{Einv})} obl_{dij} + \sum_{j: e_{ij} \in OPT} opt_{ij}.$$

Se pide además que una calle doble mano sea recorrida al menos una vez en alguno de los dos sentidos:

$$obl_{dij} + obl_{dji} \geq 1 \quad \forall i, j : e_{ij} \in E \wedge \text{required}(e_{ij}) .$$

Como resultado de resolver el problema en cada área se obtiene el conjunto de las aristas y arcos que la componen (obligatorias y optativas) con la cantidad de apariciones de éstas en el circuito solución.

2.6. Obtención de los recorridos

Para obtener los recorridos de las distintas zonas se genera un multigrafo dirigido $R = (V, A)$ (no mixto) en el cual cada arista se repite tantas veces como indica la variante asociada en dicha solución. Se sabe que este grafo

posee un circuito euleriano, se obtiene el mismo con una implementación del algoritmo de Euler. Luego (comparando con el subgrafo de la zona) se marca una de las apariciones de cada arista obligatoria (dirigida o no dirigida) como *de recolección* y el resto como *de paso*.

3. Resultados computacionales

Estudiamos a continuación los experimentos realizados. La implementación se realizó en el lenguaje de programación *Python*, se utilizó IBM CPLEX 12.5 como *solver* de programación lineal entera, con todos sus parámetros en los valores por defecto. Los experimentos se realizaron en una PC Intel Core i5 con 8Gb de memoria RAM en un sistema operativo Linux de 64bits.

3.1. Heurísticas de agregado

Se analizó la posibilidad de hacer crecer las áreas garantizando la factibilidad en cada paso. Para esto se consideró en primer lugar la heurística que agrega en cada iteración un circuito al área más cercana según los criterios de distancia *lejano* y *promedio* de la sección anterior. La generación de los circuitos se hizo mediante el algoritmo presentado en la Sección 2, y se obtuvieron los resultados que resumimos en el Cuadro 1. En el mismo se muestra para los distintos tamaños de grafo, medidos en cantidad de aristas, la cantidad de circuitos generados, el promedio de sus longitudes en kilómetros y en cantidad de aristas, la longitud máxima del mayor de ellos en kilómetros y la mínima en metros, junto con el tiempo consumido en computarlos. La principal desventaja observada en esta heurística es la dificultad para automatizar la delimitación de las zonas una vez generadas. Al no ser disjuntos los circuitos agregados, hay que definir a qué zona pertenecen las aristas que tienen en común, lo cual afecta significativamente los tiempos de recolección. Además, el grado de entrelazamiento de las zonas resulta en aristas obligatorias de una zona encerradas en el interior de otra. Este problema crece en dificultad junto con el número de zonas en contacto.

$\#aristas$	$\#circ.$	$ circuito $ [aristas]	$ circuito $ [km]	$ circuito _{max}$ [km]	$ circuito _{min}$ [mts]	$tiempo$ [min]
1885	2961	3.8888	0.2703	2.4588	0.8909	00:24.69
2077	3261	3.8749	0.2741	3.0915	0.8909	00:34.14
2329	3660	3.8694	0.2756	3.0915	0.8909	00:45.42
2536	3935	3.9532	0.2879	3.2355	0.8909	00:54.64
2743	4189	3.9957	0.2955	3.2355	0.8909	01:06.11
3274	4700	4.2834	0.3264	3.2712	0.8909	01:32.71
6982	12180	3.4756	0.2467	4.7809	3.8970	12:59.65

Cuadro 1: División de grafos en circuitos.

Como la modificación manual de las zonas resulta en un aumento notable de la varianza de sus recorridos, se reemplazará este criterio por el de agregar en cada iteración una arista o arco. Al agregar siempre una arista distinta, las zonas terminan siendo disjuntas, y lo único que resta es factibilizarlas mediante el agregado de los circuitos opcionales, resultando además sus contornos más simples.

Debido a que la finalización del algoritmo está dada por la imposibilidad de agregar más aristas a ninguna zona, el inconveniente ahora radica en la dependencia de la ubicación inicial de los centros de las áreas, ya que lo más probable es que un área se termine encerrando cuando aún quedan aristas por agregar. De este modo la similitud de las áreas generadas, definida como la diferencia de sus resultados finales de resolver el CPP, será inversamente proporcional a la cantidad de aristas sin asignar al momento del encierro de la primer zona.

En el presente análisis, la similitud entre las áreas se analizará hasta el momento en el que una de las áreas no pueda crecer más. Dejamos para un trabajo futuro la búsqueda de la ubicación óptima de los centros.

3.2. Heurísticas base

Para seleccionar el criterio de selección de la menor de las áreas analizamos en primer lugar las tres heurísticas base de manera independiente, esto es CPP, parcial y total.

La heurística CPP requiere resolver el CPP en cada iteración, por lo que fue evaluado únicamente sobre instancias pequeñas.

Las formas de las zonas generadas mediante este método son simples y uniformes, como se observa en la Figura 1 que representa un paso intermedio en la zonificación de 8 zonas nocturnas. En esta instancia, cuando las áreas cubren un promedio de 22 km cada una aproximadamente, el tiempo requerido para resolver el CPP es tan elevado que no es razonable continuar.

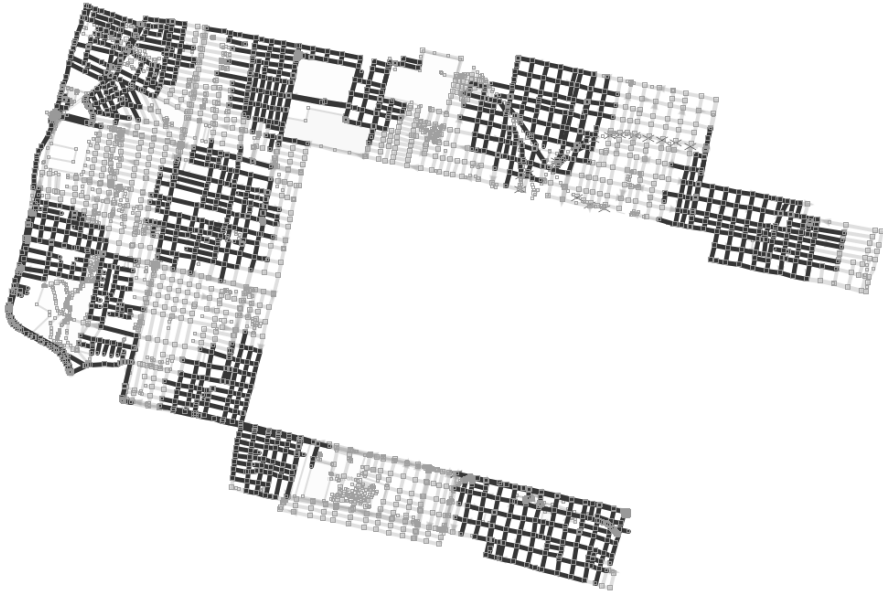


Figura 1: Ejemplo del crecimiento de las 8 zonas (en tono más oscuro) en las cuales está particionándose la región nocturna modificada, utilizando el criterio base CPP.

Los tiempos de ejecución de esta heurística en función de los tamaños de las zonas evaluadas son los representados en la Figura 2; en ella se observa el promedio del tiempo que tarda en resolverse la factibilización, la generación del modelo y el CPP para diversos tamaños de instancia. Estos datos provienen del proceso de zonificación de la región nocturna en 8 zonas. Como se puede ver, el tiempo requerido para resolverla crece exponencialmente con el tamaño de los subgrafos que representan las áreas crecientes, llegando a demorar salvo casos excepcionales, 1 minuto en promedio por cada una cuando éstas llegan a 20 km obligatorios. Nuevamente sin tener en cuenta los casos excepcionales (en los cuales llega a requerir varias horas para resolver una sola iteración), proyectando dicha curva para los tamaños de zona que se desea generar (entre 40 y 60 km) el tiempo consumido es inaceptable, por lo que no es posible utilizar este método en forma exhaustiva.

8 Areas - Tiempo de resolución de la heurística

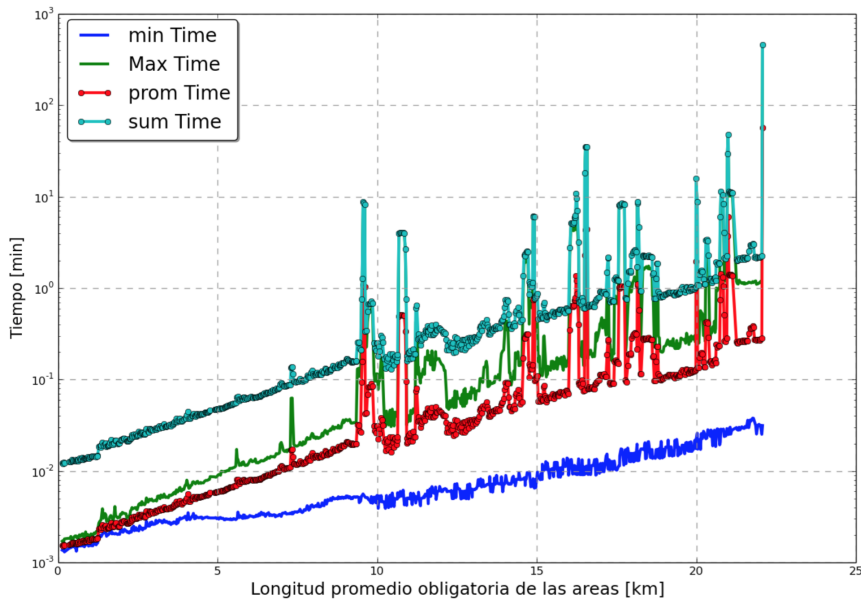


Figura 2: Variación del máximo, mínimo, promedio y suma de los tiempos que demoran en resolverse 8 zonas utilizando el criterio CPP, en función del tamaño de las áreas (medido como el promedio de la suma de los kilómetros de sus aristas obligatorias).

La segunda heurística base denominada *Parcial* se funda en la estrecha relación entre la longitud del recorrido y el peso del grafo analizado. En ella el criterio de selección de la menor de las áreas se basa en la sumas de los pesos de las aristas obligatorias de las mismas, lo cual permite resolver cada iteración sin necesidad de factibilizar el grafo, de generar el modelo ni de resolver el CPP. Aplicado sobre los mismos mapas genera zonas de contornos tan simples y uniformes como la heurística CPP pero, previsiblemente, en un tiempo varios órdenes por debajo de ésta.

En la Figura 3 se observa el costo temporal de una iteración a medida que aumenta el promedio del tamaño de las áreas. Dicha medición está hecha en un total de 8 zonas deteniendo la ejecución al bloquearse una de ellas luego de 20.96 segundos habiendo cubierto 283.68 *km* obligatorios de la región nocturna. Por un lado se observa el tiempo requerido en promedio, máximo y mínimo para resolver cada una de las áreas, y por otro lado la suma de resolver las 8 áreas.

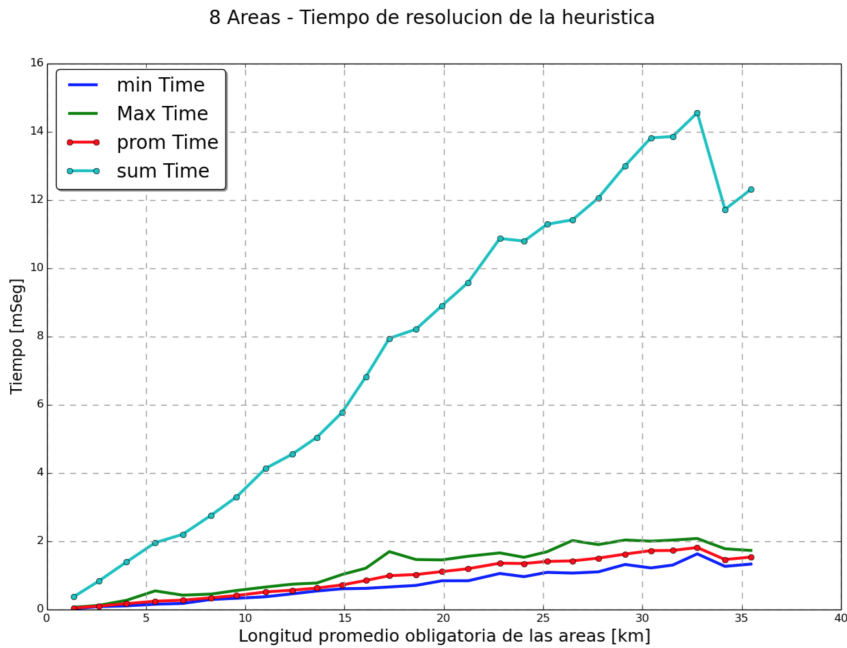


Figura 3: Costo temporal de resolver una iteración con la heurística base Partial sobre 8 zonas en función del promedio de la suma de los pesos de las aristas obligatorias de las mismas.

La principal desventaja de esta heurística en comparación con la heurística CPP consiste en la disminución de la similitud de las áreas durante y al finalizar su generación. En la Figura 4 se observa cómo varía la máxima diferencia entre los tiempos de recolección para una zonificación de la región nocturna en 8 áreas utilizando la heurística Partial, en función del promedio de los tamaños de las zonas generadas. La manera de calcular la variación de tiempo, para cada valor del eje x es restarle a las 8 áreas el tiempo consumido por la menor de ellas. Se grafica además el promedio de los valores resultantes y la media. Si bien al final se estabiliza alrededor de los 8 minutos, por momentos llega a superar los 15 minutos. A pesar de que este último resultado es hasta 4 veces peor que el de la anterior, la zonificación es resuelta en un tiempo extremadamente corto.

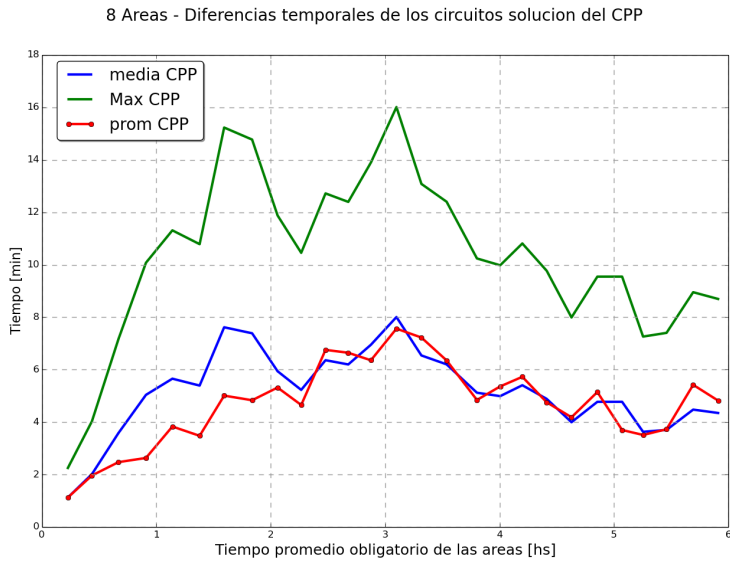
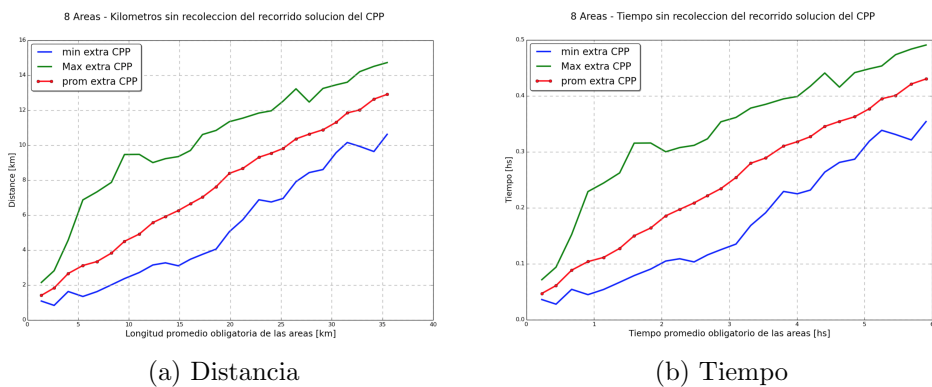


Figura 4: Detalle de la variación de la diferencia máxima en tiempo entre los recorridos de 8 zonas en función del tamaño que van adoptando mediante la inclusión de aristas utilizando el criterio base Parcial.



(a) Distancia

(b) Tiempo

Figura 5: Diferencia de velocidades promedio de los camiones en aristas de recolección o de tránsito.

Se observó en los experimentos la estrecha relación entre el peso obligatorio del grafo (definido como la suma de los pesos de sus aristas) y la longitud del recorrido solución del CPP. La Figura 5 muestra el comportamiento típico observado en todas las instancias analizadas. En la Figura 5b se ilustra que a pesar de que la cantidad de kilómetros de tránsito es aproximadamente la mitad de los kilómetros de recolección, como puede verse en la Figura 5a, el tiempo que requiere recorrer los primeros es poco menos que la doceava parte de lo que demandan los segundos. De aquí se desprende que la diferencia de

velocidades de recolección juega un papel muy importante en la heurística a utilizar.

En la heurística *Total* el criterio utilizado para seleccionar la menor de las áreas tiene en cuenta, además de la suma de los pesos de las aristas y arcos obligatorios, el peso de los circuitos optativos agregados al factibilizar. Como se los recorre a una velocidad varias veces mayor se les asignó a las aristas de estos circuitos un peso menor que a los obligatorios.

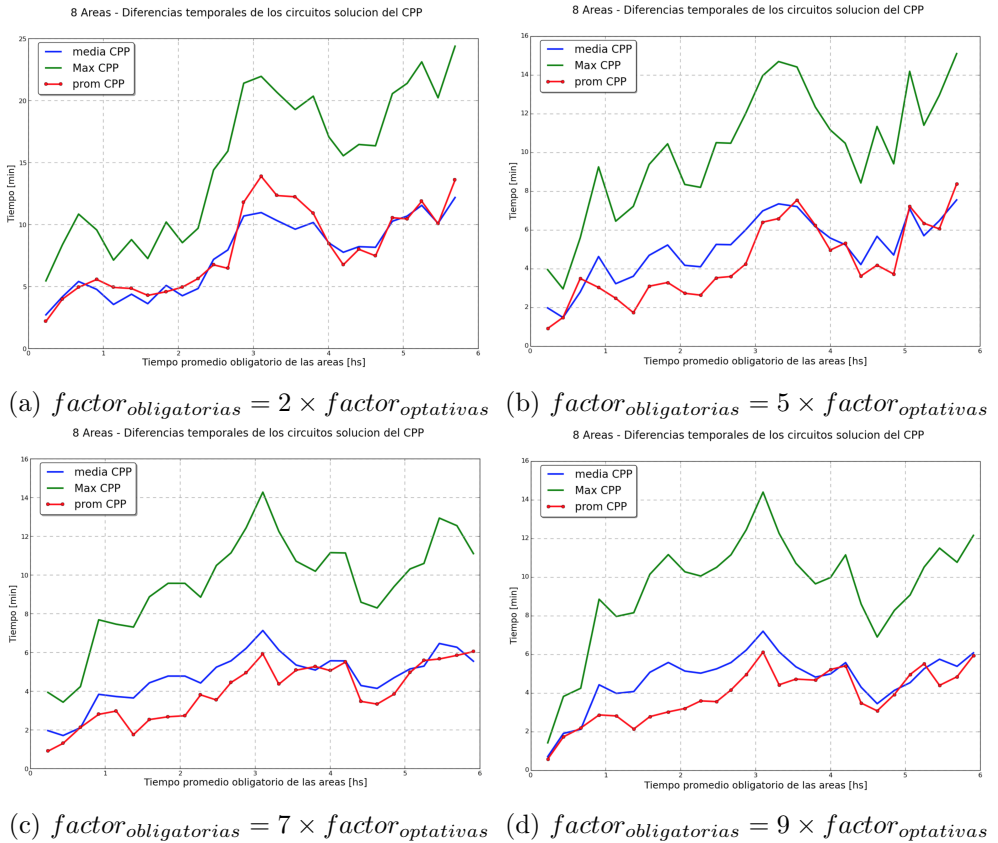


Figura 6: Detalle de la variación de la diferencia máxima en tiempo entre los recorridos de 8 zonas en función del tamaño que van adoptando mediante la inclusión de aristas utilizando el criterio Total con 4 valores distintos de factor.

A pesar de que para el cálculo del tiempo consumido por el recorrido resultante se utiliza un factor fijo de 5, se analizó asignarle distintos valores al mismo dentro de la heurística para ver si de ese modo mejoraba su comportamiento. Como se observa en los gráficos de la Figura 6, asignando menor prioridad al peso de los circuitos optativos mejora levemente la heurística al disminuir la máxima diferencia de los recorridos solución de las áreas en la segunda mitad de su generación, a la vez que empeora la primera. Por lo tanto, cuanto mayor prioridad se asigne a los circuitos optativos, mejores resultados

arrojará la heurística para grafos pequeños. Cuanta menor prioridad se les asigne, se tendrán mejores resultados sobre zonas ya más grandes, obteniendo resultados similares a la heurística anterior.

En lo que respecta a la forma de las áreas resultantes, el resultado fue el mismo que con las dos heurísticas anteriores, se generan contornos simples. La diferencia radica más en los tiempos de resolución que en los resultados. Si bien está dentro de lo razonable en contraposición de la primera (CPP), esta heurística demora bastante más tiempo que la segunda.

El Cuadro 2 sintetiza los tiempos de resolución de las tres heurísticas. La cantidad de aristas y los kilómetros hacen referencia al promedio de las áreas, mientras que los tiempos están dados por la suma de lo que consume la generación de todas ellas.

	zonas	aristas	longitud[km]	Parcial[seg]	Total[hs]	CPP[hs]
pequeñas	8			8,25	0 : 12 : 00,65	23 : 31 : 20
	9			9,21	0 : 16 : 21,98	89 : 57 : 23
	10	270	22,45	12,99	0 : 22 : 17,66	---
	11			14,51	0 : 27 : 39,91	---
	12			20,67	0 : 34 : 53,60	---
nocturnas	12	367	31,12	25,43	1 : 13 : 26,39	---
	11	400	33,92	25,41	1 : 09 : 04,08	---
	10	440	37,31	25,16	1 : 04 : 38,74	---
	9	489	41,47	23,49	0 : 59 : 08,61	---
	8	550	46,64	23,29	0 : 53 : 59,12	---

Cuadro 2: Tiempos de resolución de las tres heurísticas base.

A partir de estos resultados iniciales se exploraron heurísticas que simularan el criterio CPP para conseguir varianzas casi nulas y a la vez que utilizaran una combinación de los otros dos criterios para obtener resultados en corto plazo.

3.3. Heurísticas combinadas

La idea es utilizar alguno de los dos criterios veloces (Parcial ó Total), y cada N iteraciones corregir mediante el criterio CPP. Surgen así dos heurísticas que combinan Parcial con CPP, y Total con CPP, y utilizan las funciones de realimentación de la sección anterior.

Los cuatro gráficos de la Figura 7 muestran el detalle de la variación de la diferencia máxima en tiempo entre los recorridos de 8 zonas en función del tiempo promedio que lleva recorrer sus arcos y aristas obligatorias (a 6 km/h y $F_{obl}/F_{opt} = 5$) a medida que crecen éstas en número con el criterio Parcial utilizando 4 funciones de realimentación para $N = 100$. Fueron tomadas

muestras cada 100 iteraciones. Es importante notar que la cuarta función de realimentación logra una mejora notable bajando 2 minutos en promedio la diferencia máxima entre las áreas a lo largo de su generación.

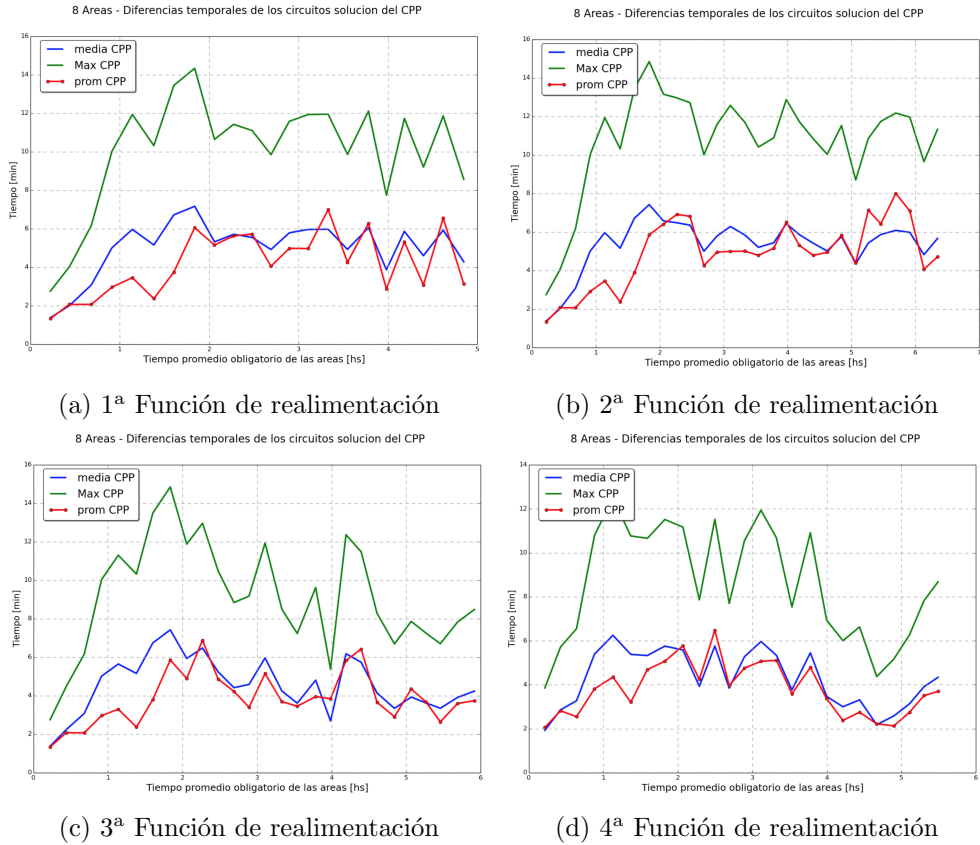


Figura 7: Detalle de la variación de la diferencia máxima en tiempo entre los recorridos de 8 zonas con el criterio Parcial.

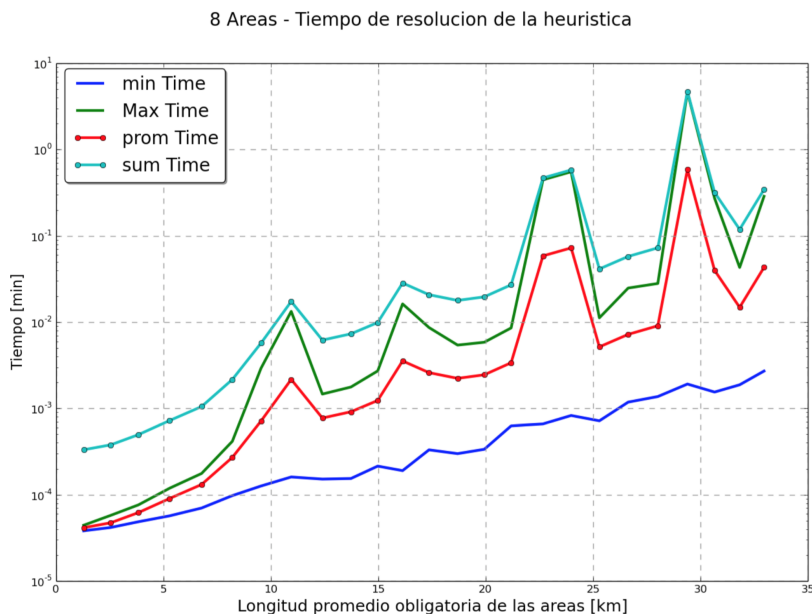
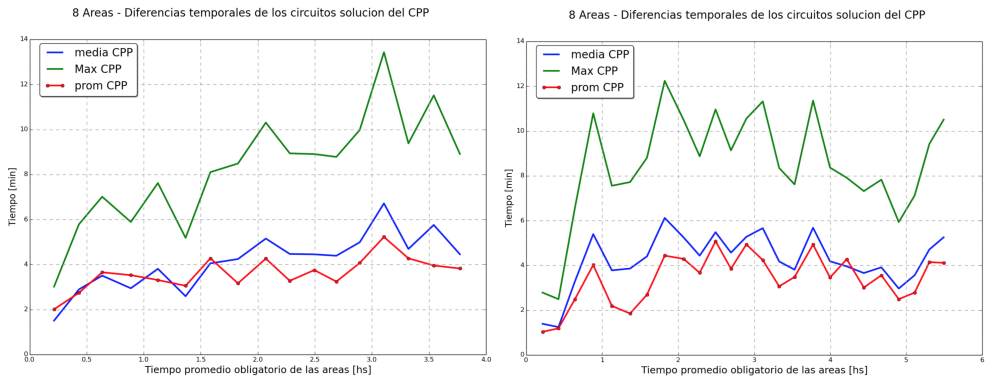


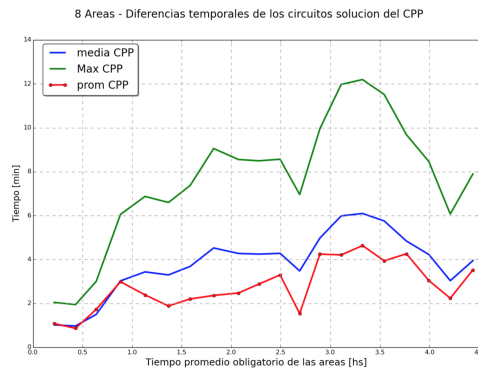
Figura 8: Variación del tiempo promedio requerido para el zonificado a medida que crecen las áreas utilizando la función de realimentación.

Los tiempos de resolución de estas heurísticas con realimentación se ven aumentados respecto de lo que era la heurística Parcial, como se observa en la Figura 8, siendo cada vez más grandes a medida que disminuye el N . Cada valor representa el promedio del tiempo empleado en resolver las 100 últimas iteraciones incluyendo la última en la que se resuelve el CPP. Este gráfico no está orientado a mostrar los valores máximos, sino dar una idea de la variación del tiempo promedio requerido para el zonificado a medida que crecen las áreas utilizando las funciones de realimentación.

Con respecto a las heurísticas con realimentación que utilizan el criterio Total en las iteraciones intermedias, sólo la cuarta obtuvo mejores resultados. Fijando $N = 50$ y variando la relación entre el factor multiplicador de las aristas obligatorias y las optativas se consiguieron las curvas que se muestran en la Figura 9. Se toman muestras cada 100 iteraciones. El tamaño de las zonas está representado por las horas promedio que demora recorrer las aristas y arcos obligatorios a la velocidad de recolección (6 km/h).



(a) $factor_{obligatorias} = 3 \times factor_{optativas}$ (b) $factor_{obligatorias} = 9 \times factor_{optativas}$



(c) $factor_{obligatorias} = 5 \times factor_{optativas}$

Figura 9: Detalle de la variación de la diferencia máxima en tiempo entre los recorridos de 8 zonas en función del tamaño que van adoptando mediante la inclusión de aristas utilizando el criterio Total con una función de realimentación para 3 valores distintos de factor.

Fijando el factor en 5, que es el que en promedio mantiene una diferencia menor, se varió N para intentar obtener mejoras en los resultados, pero sin éxito con valores mayores a 30, al igual que sucedía con la heurística anterior. Recién con $N = 20$ se ha logrado una mejora notable en cuanto a similitud de las áreas, pero con tiempos incluso mayores a los de la anterior heurística, por ende no razonables para los tamaños de zonas que requiere la región de San Miguel de Tucumán.

Los centros de las áreas se ubicaron manualmente en posiciones tales que éstas se encerraran entre ellas lo más cerca de finalizar el algoritmo, logrando un 99% de cobertura para 8 zonas, un 98,45% para 9 zonas, y un 98.71% para 10 zonas utilizando la heurística Parcial. Con esas posiciones de los centros se han realizado 6 últimos experimentos: la zonificación de la región nocturna para 8, 9 y 10 zonas utilizando la heurística Parcial, y luego la heurística Total + CPP + Realimentación con un factor de 5 y $N = 100$. Estos resultados se

resumen en el Cuadro 3.

	Región Nocturna	Parcial	Total + CPP + Feed
8 zonas	Tiempo promedio de los recorridos	07:36:50.19	07:36:50.61
	Diferencia Máxima de tiempos	00:13:03.96	00:12:03.81
	Desviacion estándar tiempos	00:03:56.39	00:04:07.64
	Tiempo Total de Resolución	03:39:08.72	37:25:15.78
	Tiempo Generacion de Areas	00:00:19.26	36:00:24.75
9 zonas	Tiempo promedio de los recorridos	06:47:05.95	06:47:26.78
	Diferencia Máxima de tiempos	00:14:47.84	00:19:21.82
	Desviacion estándar de tiempos	00:05:26.14	00:06:04.90
	Tiempo Total de Resolución	00:41:29.77	11:14:37.74
	Tiempo Generación de Areas	00:00:24.74	10:29:34.58
10 zonas	Tiempo promedio de los recorridos	06:06:50.43	06:06:35.46
	Diferencia Máxima de tiempos	00:19:01.94	00:23:12.60
	Desviacion estándar de tiempos	00:06:24.89	00:06:42.64
	Tiempo Total de Resolución	00:25:46.15	02:39:45.96
	Tiempo Generación de Areas	00:00:39.81	02:30:07.90

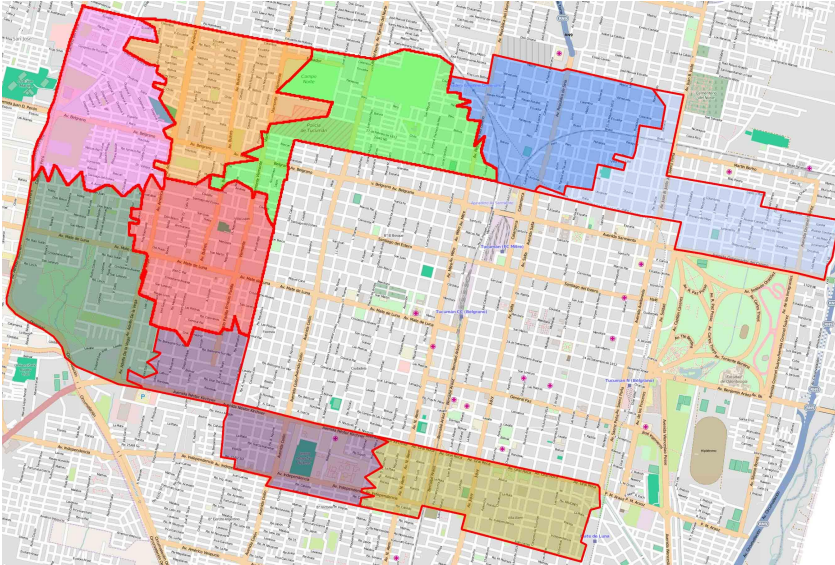
Cuadro 3: Comparación temporal en horas de la zonificación de la región nocturna en 8, 9 y 10 zonas utilizando las dos heurísticas con mejores resultados.

Si bien la zonificación en 8 áreas da como resultado recorridos menores a las 8 horas, al no tener en cuenta factores que pueden hacer demorar más tiempo a los camiones, éstos no parecen viables en la práctica. Por este motivo la zonificación que se utilizará es la que se observa en la Figura 10a en la cual se logra reducir la cantidad de camiones necesarios de 12 a 9, con lo que se contaría con 3 camiones más para cubrir la zona céntrica.

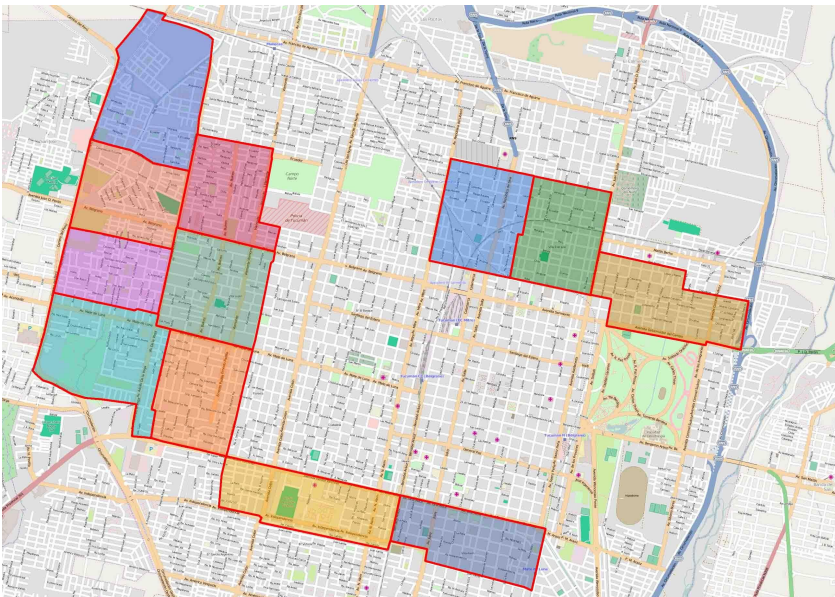
3.4. Calidad de las soluciones

Recordemos que la solución óptima según la definición del problema es una división del grafo en k subgrafos conexos, similares (en el sentido que un camión tarda aproximadamente el mismo tiempo en recolectar los residuos en cada una de ellas), tal que el promedio de los resultados del CPP en todos ellos sea el menor posible, con la restricción adicional de las formas sencillas de las áreas. Si bien no fue posible obtener resultados de *gap* de optimalidad para los experimentos, las soluciones obtenidas se consideran de calidad aceptable para los requerimientos del Municipio de San Miguel de Tucumán, ya que las zonas son lo suficientemente similares: la diferencia máxima entre tiempos de recorrido entre zonas oscila entre el 2.86 % para 8 zonas y el 6.33 % para 10 zonas, y el desvío estándar entre el 0.86 % y el 1.83 %, respectivamente (ver Cuadro 3). Por otro lado, los contornos de las zonas obtenidas resultan más complejos

que en la zonificación actual, pero son todavía aceptablemente sencillos (ver Figuras 10a y 10b). Puede observarse además que la región nocturna es ligeramente más grande en la zonificación propuesta por el algoritmo; incluimos este fragmento adicional en los experimentos (ver zona de color verde claro en la Figura 10a) para dar mayor flexibilidad al algoritmo.



(a) Posible zonificado de la región nocturna en 9 zonas resuelto con el algoritmo desarrollado en el presente trabajo con la heurística total + CPP + feedback con $N = 100$, y un factor de 5.



(b) Zonificado actual de la región nocturna de San Miguel de Tucumán según los datos provistos por el Municipio.

4. Conclusiones

En este trabajo se estudiaron algoritmos heurísticos para resolver el zonificado de la recolección de residuos del municipio de San Miguel de Tucumán. Buscamos minimizar la cantidad de camiones recolectores, y dividir el municipio en zonas similares, definiendo similitud entre zonas como el parecido entre los resultados de resolver el problema del cartero chino mixto en cada uno de los subgrafos asociados.

La idea central de los algoritmos desarrollados consistió en partir de un conjunto de subgrafos nulos del grafo inicial e ir agregando a los mismos conjuntos de aristas hasta formar el particionamiento.

Se desarrollaron dos heurísticas base, que demostraron dar muy buenos resultados en muy corto tiempo. Se intentó mejorar estos algoritmos por medio de funciones de realimentación basadas en la resolución del problema del cartero chino mixto cada cierta cantidad de pasos, y mediante la variación de sus parámetros. Esta estrategia aportó mejoras con respecto a las heurísticas iniciales.

Mediante las heurísticas desarrolladas, se ha logrado cumplir con el requerimiento más importante: la disminución de la cantidad de zonas de la región nocturna. Al reducir de 12 a 9 esta cantidad, se liberan tres camiones para mitigar la problemática del área delimitada por las cuatro avenidas. Con respecto a los requerimientos sobre los contornos de las zonas, fue posible obtener zonas comparables a las utilizadas hasta el momento.

Como trabajo a futuro, podrían desarrollarse heurísticas para mejorar las formas de las zonas generadas que redistribuyan las aristas y arcos de las intersecciones buscando mantener en la medida de lo posible los resultados del CPP.

Adicionalmente, podrían modificarse las posiciones de los centros de las áreas para generar zonas más equilibradas que cubran toda la región sin bloquearse mutuamente en el crecimiento. En este trabajo, los centros de las zonas a generar son asignados en forma manual; este comportamiento podría automatizarse, esto es, iterativamente resolver el zonificado con alguna heurística sencilla e ir variando las posiciones de los centros intentando generar espacio para la primer zona que se bloquea en la ejecución anterior.

Finalmente, en este trabajo no se han tenido en cuenta las reglas de tránsito al momento de armar los recorridos: podrían agregarse restricciones de giros explotando los nodos y conectando sólo las aristas asociadas a las calles que permitieran la circulación entre ellas.

Los resultados de este trabajo fueron entregados al municipio para que se evalúe su implementación en el caso real.

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*, (28):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] G. Braier, G. Durán, J. Marengo, y F. Wesner. An integer programming approach to a real-world recyclable waste collection problem in Argentina. *Waste Management and Research*. Aceptado para su publicación.
- [4] V. Bucarey, F. Ordóñez, y E. Bassaletti. Shape and balance in police districting. In *Applications of Location Analysis*, páginas 329–347. Springer International Publishing, 2015.
- [5] 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.
- [6] A. Drexel y K. Haase. Fast approximation methods for sales force deployment. *Management Science*, (45):1307–1323, 1999.
- [7] J. Edmonds. The chinese postman problem. *Operations Research*, (5):88–124, 1965.
- [8] J. Edmonds y E. L. Johnson. Matching, Euler tours and the chinese postman. *Mathematical Programming*, (5):88–124, 1973.
- [9] D. Eisenstein y A. Iyer. Garbage collection in Chicago: a dynamic scheduling model. *Management Science*, (43):922–933, 1997.
- [10] J. Ferland y G. Guénette. Decision support system for a school districting problem. *Operations Research*, 38(6):15–21, 1990.
- [11] The OpenStreetMap Foundation. <http://www.openstreetmap.org>, 2015.
- [12] M. Garey y D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Company, 1979.
- [13] B. Golden, A. Assad, y E. Wasil. Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industrie. In Toth P. y Vigo D., editors, *The Vehicle Routing Problem*, página 245–286. SIAM, 2002.

- [14] S. Hanafi, A. Freville, y P. Vaca. Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search methods. *INFOR*, (37):236–254, 1999.
- [15] P.E. Hart, N.J. Nilsson, y B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, (4):100–107, 1968.
- [16] M. Hojati. Optimal political districting. *Comput Oper Res*, 23(12):1147–1161, 1996.
- [17] J. Kalcsics, S. Nickel, y M. Schröder. Towards a unified territory design approach – Applications, Algorithms and GIS Integration. *Top*, 13(1):1–56, 2005.
- [18] H. C. Kappauf y G. J. Koehler. The mixed postman problem. *Discrete Applied Mathematics*, (1):89–103, 1976.
- [19] B. Kim, S. Kim, y S. Sahoo. Waste collection vehicle routing problem with time windows. *Computers and Operations Research*, (33):3624–3642, 2006.
- [20] J. Liebman. Models in solid waste management. In S. Gass y R. Sisson, editors, *A Guide to Models in Governmental Planning and Operations*. Sauger Books, 1975.
- [21] J. Male y J. Liebman. Districting and routing for solid waste collection. *Journal of the Environmental Engineering Division*, (104):1–14, 1978.
- [22] R. Minciardi, P. Puliafito, y R. Zoppoli. A districting procedure for social organizations. *European Journal of Operational Research*, 8:47–57, 1981.
- [23] 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.
- [24] L. Muyldermans, D. Cattrysse, D. van Oudheusden, y T. Lotan. Districting for salt spreading operations. *European Journal of Operational Research*, 139(3):521–532, 2002.
- [25] T. K. Ralphs. On the mixed chinese postman problem. *Operations Research*, (14):123–127, 1993.
- [26] D. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, 2000.
- [27] 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):11–30, 2003.