UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales
Departamento de Matemática

Tesis de Licenciatura

Programación Matemática para distribución eficiente de censistas en censos nacionales: el Censo 2010 de la Provincia de Buenos Aires como caso de estudio

Florencia Fernández Slezak

Director: Dr. Guillermo A. Durán
Codirector: Lic. Diego Delle Donne

Marzo 2012

Resumen

La planificación de un *censo poblacional* implica diversos desafíos logísticos, uno de los cuales es determinar qué hogares debe visitar cada censista. A este problema se lo conoce como el *problema de segmentación de viviendas* y generalmente involucra un conjunto de restricciones sobre el recorrido de cada censista. Durante el Censo Nacional 2010 en Argentina, se aplicó exitosamente una solución para este problema en las zonas urbanas, semi-urbanas y rurales de la Provincia de Buenos Aires, basado en programación lineal entera [5]. En este trabajo se analiza la solución propuesta y se desarrollan mecanismos alternativos para la resolución de dicho problema. Dentro de estas soluciones alternativas, se sugieren nuevos objetivos y se analiza el impacto económico que podría haber tenido la implementación de éstos en el caso real.

Por un lado, se identifican varios problemas que se han suscitado con la solución propuesta originalmente y se proponen diversas formas de corregirlos, consiguiendo soluciones aún más satisfactorias. Por otro lado, se propone otro objetivo del mismo problema donde se le da importancia a la economía de la Provincia de Buenos Aires consiguiendo una reducción potencial de hasta un 5,55 %, según la zona de la provincia en donde se implementen estos nuevos modelos. Esta reducción, en general, debido a la forma de construir las soluciones, no ha empeorado visiblemente la bondad de las soluciones halladas originalmente, lo que hace que valga la pena la implementación de las mejoras propuestas.

Agradecimientos

Es muy difícil en tan pocas líneas sintetizar los recuerdos y nombrar a toda la gente que de algún modo me acompañó en este dificultoso camino. Sin la luz de cada uno de ellos, transitarlo hubiese sido prácticamente imposible.

Primero y principal, quiero agradecer a mis directores Willy y Diego por guiarme a través de esta tesis y empujarme cada vez que me estancaba. Ellos supieron *sacarme la ficha* y saber cómo obtener lo mejor de mí. Gracias por su paciencia infinita.

A mis viejos, que me mostraron desde chica el mágico y hermoso mundo de la ciencia. Siempre acompañándome y apoyándome en mis decisiones sean acertadas o no. Estuvieron presentes en todo momento ayudándome desde una mudanza hasta en un problema de la guía práctica, por más que siempre terminara diciéndoles "no entienden nada, ¡si son físicos!".

A Pol, por siempre sacarme una sonrisa sea como sea, no importa el momento por el que esté pasando. Sabés que te gané. A Die y Vicky, los newly weds, siempre alentándome, ayudándome, mostrando la confianza ciega que me tuvieron en cada etapa de la carrera. A la flia. Robledo en general, por ser tan buena onda conmigo, compartiendo asaditos y tardes tigrenses. A Athilín, por las tardes de frisbee interrumpiéndome mis largas jornadas de reventarme la cabeza con algún ejercicio.

A la chicas de la facu, Anita, Ro, Katri, Lulú, Pau, Carito, Manu, Georgi, Lau, Juli, Mari, Lucas, Marian, Alita y a todos los que de alguna manera compartieron materias o simplemente un mate en el bar. Gracias por las eternas tardes de estudio y las fabulosas noches de salidas. Cada uno aportó su granito de arena para que ahora esté acá.

A los chicos del cole, Rena, Orne, Jime, Ceci y Ramos. Desde el cole que vengo soñando con este momento y ustedes estuvieron siempre acompañándome, alentándome, apoyándome. Fueron un gran empuje en los distintos momentos de la carrera. Desde mis clases de matemática en el secundario dado que teníamos una profe *fatal*, hasta yéndonos a anotar juntos al CBC.

A los chicos de baile, Charly, Lu, Belu, la Matri, Pau, Mich y el resto del *team*, serán siempre mi cable a tierra. Las horas de ensayo fueron siempre disparadoras de noches de inspiración para mi estudio, ese corte diario de baile, disfrute y risas es necesario para cerrar cada día.

A los profes de la facu, Constanza, Juli Molina, Gabriel y todos los que de alguna manera me instruyeron y mostraron el verdadero trabajo de un matemático. Muchas gracias por la paciencia y amor a este arte.

A mi gordito Tuco. El gran boicoteador de estudio, sin él no hubiese sido lo mismo estudiar para los finales. Esas ganas de acostarse justo arriba de mi cuaderno lo hicieron por momentos amoroso y por momentos insoportable. Por ese amor incondicional que me da todos los días, ¡gracias chanchito!

A Javier y Flavia, por sus comentarios y observaciones constructivos sobre este trabajo, es realmente enriquecedor escucharlos. A Fernando Aliaga, por contestar mis consultas y aportar los datos necesarios relativos al Censo de forma tan desinteresada.

Gracias a cada uno de ustedes por acompañarme a lo largo de todos estos años.

Índice general

1. Introducción **1**
1.1. La historia de los censos en la Argentina 1
1.2. Descripción general del problema 2

2. Programación matemática **5**
2.1. Programación lineal 6
2.2. Programación lineal entera 7
2.3. Métodos de resolución 7
2.3.1. *Branch and Bound* 8
2.3.2. Planos de corte 9
2.3.3. *Branch and Cut* 9

3. Descripción del problema original **11**
3.1. Marco del problema 11
3.2. Segmentación de viviendas 12
3.3. Modelo de programación lineal entera 15
3.4. Algoritmos utilizados 18
3.5. Resultados 22

4. Nuevos enfoques del problema **25**
4.1. Minimización de censistas 25
4.1.1. Algoritmo 26
4.2. Problema de minimax 27
4.2.1. Algoritmo 28
4.3. Profundización de la búsqueda 29
4.3.1. Algoritmo 29
4.4. Nuevas funciones de valuación 30
4.4.1. Base de la función de valuación 30
4.4.2. Definición de compacidad 31

5. Resultados **35**
5.1. Impacto económico 35
5.1.1. Minimización de censistas 36
5.1.2. Problema de minimax 37

5.2. Profundizar la búsqueda 37
5.3. Mejoras en la valuación de segmentos 40
5.3.1. Base de la función de valuación 40
5.3.2. Definición de compacidad 41

6. Conclusiones y trabajo futuro **45**

Capítulo 1

Introducción

Un *censo poblacional* requiere de un proceso de adquisición de información demográfica de un área geográfica (habitualmente un país). El objetivo principal es relevar datos estadísticos sobre los habitantes, incluyendo información sobre sus viviendas, su nivel educativo y su realidad laboral. Típicamente, un censo se realiza a lo largo de un sólo día declarado no laborable, y la información es recabada por *censistas* que visitan casa por casa del área censada.

En este capítulo se hace una breve reseña histórica sobre la ejecución de los censos de viviendas en la República Argentina y se presenta el problema que suscitó al momento de organizar el Censo Nacional de Población, Hogares y Viviendas en el año 2010.

1.1. La historia de los censos en la Argentina

Desde su creación en 1968, el INDEC está a cargo del diseño metodológico, la planificación, implementación y evaluación del operativo censal. En esta sección se detallan ciertos Censos relevantes en la historia de la República Argentina [16].

En 1776, año de la creación del Virreinato del Río de la Plata, el Rey Carlos III mandó a realizar censos anuales en las colonias. En 1778, registró 186.526 habitantes en el Virreinato, de los cuales más del 30 % fueron catalogados como afrodescendientes.

El primer Censo Nacional en la Argentina se realizó en el año 1869 bajo la presidencia de Domingo Faustino Sarmiento, limitado a un peculiar relevamiento demográfico, que permitió saber que la población *civilizada* del país era entonces de 1.830.214 habitantes. Alcanzó sólo a una parte del territorio actual porque excluyó regiones dominadas entonces por grupos indígenas considerados *ilegales*, que no tenían apellido porque no eran *civiles* dentro de la ley, no eran bautizados, ni hablaban español. En otras regiones, incluyó a los indígenas *civilizados*, como los guaraníes en el litoral y los collas en el noroeste.

El censo de 1947 fue realizado durante la primera presidencia de Juan Do-

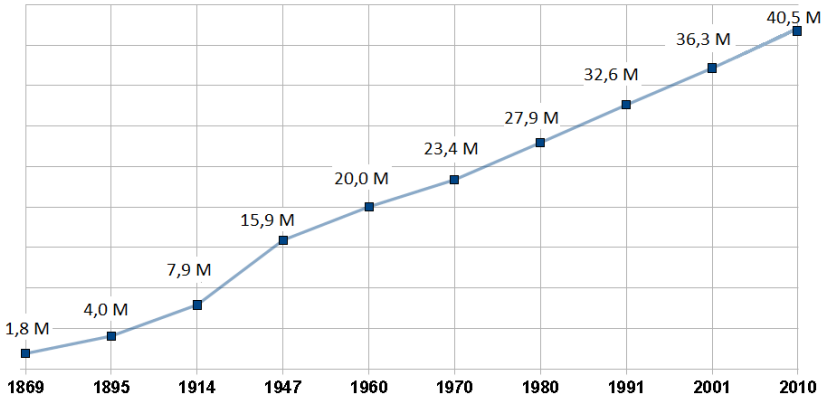

Figura 1.1: Cantidad de habitantes en la Argentina en función de cada año censado.

mingo Perón. Fue el primero en considerar a la familia y al hogar como unidad de análisis privilegiada. Se realizaron preguntas sobre educación y ocupación laboral. También fue la primera vez que se incorporó tecnología computacional para el procesado de los datos, considerado por los especialistas uno de los mejores censos realizados en el país.

El de 1970 se realizó durante el gobierno de facto del General Levingston. Este censo tuvo la novedad del uso de las lectoras ópticas de caracteres manuscritos. Sin embargo, fue el más impreciso de los realizados en el país, pues las cédulas censales fueron provistas por una empresa y las máquinas lectoras por otra compañía, que resultaron incompatibles y los resultados terminaron calculándose años después en base a la proyección del 10 % de las planillas confeccionadas.

En la Figura 1.1 se muestra un gráfico de la cantidad de población en el país en función de los años censados.

1.2. Descripción general del problema

Las actividades de planificación de un censo suelen comenzarse con varios años de anticipación dado que un censo involucra un importante movimiento logístico realizado durante unas pocas horas. Una tarea crucial dentro de la planificación de un censo es determinar qué viviendas deberá visitar cada censista, que en este contexto se denomina *problema de segmentación de viviendas*. Ello consiste en obtener una *partición* del conjunto de viviendas en subconjuntos que cumplan ciertas restricciones pre-establecidas.

Cuando la partición del conjunto de viviendas se realiza a mayores escalas, por ejemplo, para determinar distritos electorales dentro de un área geográfica, esto es conocido como *problema de re-zonificación* (*redistricting problem*), tema

que posee abundante literatura [2, 3, 6, 9, 11, 13, 14]. Existen varios paquetes de software que permiten realizar este tipo de tareas en forma automática o semi-automática. Sin embargo, las restricciones presentes en los problemas de re-zonificación en general difieren de los requerimientos de un caso típico de segmentación de viviendas, con lo cual se hace necesario recurrir a algoritmos específicos.

En este trabajo se describe el problema original de segmentación de viviendas en la Provincia de Buenos Aires para el Censo Nacional de Población, Hogares y Viviendas transcurrido en el año 2010 en Argentina (ver Figura 1.2) y la herramienta basada en programación lineal entera que se utilizó para su resolución [5]. Este aporte fue utilizado exitosamente durante el proceso de planificación del Censo Nacional 2010 en esta provincia.

Como objeto de estudio de esta tesis se presentan variantes de este problema, como así también sus resoluciones, donde además de buscar una segmentación satisfactoria se busca minimizar los costos económicos incurridos por la misma. Asímismo, se han detectado fallas en el planteo original, a las que se les ha buscado distintas formas de corregirlas obteniendo resultados satisfactorios. Además de [5], no se han encontrado otros trabajos en la literatura de investigación operativa que hayan encarado el problema de segmentación de viviendas en el contexto de un censo.

El presente trabajo está organizado del siguiente modo: en el Capítulo 2 se introduce la base teórica de las herramientas matemáticas utilizadas en este trabajo, que se utilizará en el resto de los capítulos. En el Capítulo 3, se detalla el problema original impuesto por la Provincia de Buenos Aires, el modelo que se utilizó para resolverlo, con sus respectivos algoritmos y una breve descripción de los resultados conseguidos. En el Capítulo 4 se abordan nuevos enfoques del problema, se proponen distintas soluciones y se describen los algoritmos desarrollados. Además, se detectan problemas con ciertas definiciones y se proponen soluciones a los mismos. En el Capítulo 5 se muestran los resultados obtenidos. Por último, en el Capítulo 6 se exhiben las conclusiones y el posible trabajo futuro.

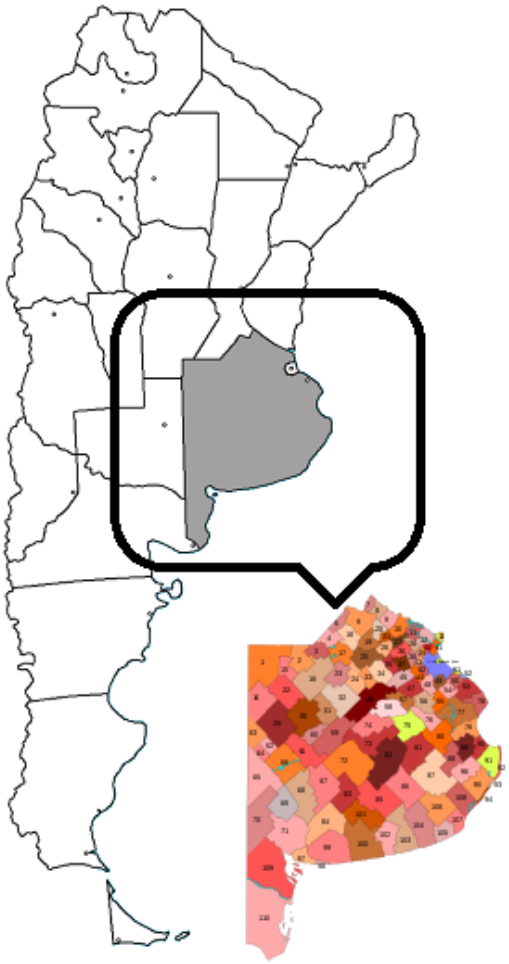

Figura 1.2: Mapa de Argentina, con la división política de la Provincia de Buenos Aires.

Capítulo 2

Programación matemática

La principal herramienta de programación matemática que se usará en este trabajo es la programación entera, que trata con problemas de maximización o minimización de una función en varias variables sujeto a:

- restricciones de igualdad y/o desigualdad y
- condiciones de integralidad de algunas o todas las variables.

Dada la robustez del modelo general, una vasta variedad de problemas del mundo real pueden ser trabajados en este contexto.

Una importante y reconocida área de aplicación es la que concierne al manejo y uso eficiente de recursos acotados para aumentar la productividad, tales como la distribución de bienes, la programación de una producción en serie y el ordenamiento de tareas a realizar en máquinas. También se modelan problemas de planificación como, por ejemplo, uso eficiente de capital o locación de facilidades, además de problemas de diseño tales como el armado de redes de comunicación y de transporte, la distribución de sistemas de circuitos electrónicos y la automatización de sistemas de producción.

En matemática se utiliza este tipo de modelos en materias como combinatoria, teoría de grafos y lógica. Más aún, se incluyen problemas de estadística como análisis y fiabilidad de datos. Estos modelos se utilizan también en campos científicos más amplios como en biología molecular, en física de altas energías y en cristalografía de rayos X [8, 20]. Por último, una aplicación política podría ser la de dividir una región en distritos electorales o censales o, como es nuestro caso, distribuir censistas para cubrir una zona determinada.

El principal propósito de este capítulo es presentar la base matemática de los modelos de optimización entera y los algoritmos más utilizados para resolver estos problemas, los cuales se describen en la Sección 2.3. A lo largo de todo este trabajo, tanto la función a maximizar como las restricciones serán lineales. Notar que minimizar una función es equivalente a maximizar el opuesto de ella y que una restricción de igualdad es equivalente a dos de desigualdad. Dado el tipo de problemas descriptos en esta sección, en general las variables son no negativas.

2.1. Programación lineal

Al problema de hallar el vector $x = (x_1, \ldots, x_n)$ que maximice

$$z_{PL} = \text{máx}\{cx : Ax \leq b, x \in \mathbb{R}^n_+\}, \tag{2.1}$$

donde $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$ y $b \in \mathbb{R}^m$, se lo llama *problema de programación lineal*. Una *instancia* del problema está determinada por los datos c, A y b.

Definición 1 *Sea*

$$\mathcal{S}_{PL} = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$$

el conjunto de todas las posibles soluciones de (2.1). Cuando no haya necesidad de distinguir a qué tipo de problemas se refiere este conjunto, simplemente lo llamaremos $\mathcal{S}$. Definimos:

- *a $\mathcal{S}$ se lo llama* región de factibilidad*,*
- *a un punto $x \in \mathcal{S}$ se lo define como* solución factible*,*
- *una instancia se dice* factible *si $\mathcal{S} \neq \varnothing$, e* infactible *si $\mathcal{S} = \varnothing$,*
- *a la función a maximizar $z = c.x$ se la llama* función objetivo*,*
- *se le dice* solución óptima *a un punto factible $x^0 \in \mathcal{S}$ en el cual la función objetivo alcanza un máximo global, es decir:*

$$z^0 := cx^0 \geq cx \ \forall x \in \mathcal{S},$$

- *$z^0 = c.x^0$ es el* valor óptimo*,*
- *una instancia factible puede no tener solución óptima. En ese caso, se dice que la instancia es* no acotada *pues $\forall \omega \in \mathbb{R}$ hay una solución factible $x \in \mathcal{S}$ tal que $cx > \omega$. Se lo nota con $z = \infty$.*

Con estas definiciones, resolver una instancia de (2.1) es encontrar una solución óptima o mostrar que la misma es no acotada o infactible. Este problema está bien definido en el sentido de que, si es factible y acotado, entonces tiene solución óptima.

Este tipo de problemas fue estudiado intensivamente y existen varios métodos para su resolución, siendo el más reconocido de ellos el método *simplex* [7], propuesto por George Dantzig en 1947. Si bien este método tiene un peor caso de complejidad exponencial, es el más utilizado en la práctica ya que en la mayoría de los casos resulta muy eficiente. Se sabe además que la programación lineal pertenece a la clase de problemas P [12].

Quien demuestra en 1979 que el problema de programación lineal es resoluble en tiempo polinomial es el matemático ruso Leonid Kchachiyan, mediante el llamado *Algoritmo del Elipsoide* [18]. Éste prácticamente ya no se utiliza debido a las buenas implementaciones de otros algoritmos. Más tarde, en 1984, Narendra Karmarkar introduce un nuevo método de punto interior para resolver problemas de programación lineal, lo que constituiría un enorme avance en los principios teóricos y prácticos en el área [17]. Existen muy buenas implementaciones de los algoritmos del punto interior competitivas con el simplex.

2.2. Programación lineal entera

Definición 2 *Un* problema de programación lineal entera *es*

$$z_{PLE} = \text{máx}\{cx : Ax \leq b, x \in \mathbb{Z}^n_+\}, \tag{2.2}$$

donde $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$ *y* $b \in \mathbb{R}^m$. *A la región de factibilidad se la notará* $\mathcal{S}_{PLE}$ *o, simplemente,* $\mathcal{S}$.

Un modelo aún más general es aquel en el que sólo algunas variables son enteras. Lo definimos de la siguiente manera:

Definición 3 *Un* problema de programación lineal entera mixto *es*

$$z_{PLEM} = \text{máx}\{cx + hy : Ax + Gy \leq b, x \in \mathbb{Z}^n_+, y \in \mathbb{R}^n_+\},$$

donde $A \in \mathbb{R}^{m\times n}$, $G \in \mathbb{R}^{m\times p}$, $c \in \mathbb{R}^n$, $h \in \mathbb{R}^p$ *y* $b \in \mathbb{R}^m$. *A la región de factibilidad se la notará* $\mathcal{S}_{PLEM}$ *o* $\mathcal{S}$.

En muchos modelos, a algunas o todas las variables se las utiliza para representar decisiones lógicas, lo que las hace valer simplemente 0 ó 1. En caso de ser íntegramente binarios, se obtiene el 0-1 PLE donde el vector $x \in \mathbb{Z}^n_+$ es en realidad $x \in \{0,1\}^n$.

Observación 1 *Dados* $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$ *y* $b \in \mathbb{R}^m$, *se tiene que* $z_{PLE} \leq z_{PL}$ *ya que* $\mathbb{Z}^n_+ \subseteq \mathbb{R}^n_+$. *La cota superior* z^0_{PL} *a veces es utilizada para probar la optimalidad de (2.2).*

En efecto, si x^0 *es una solución factible de (2.2) y además* $z^0_{PLE} = z_{PL}$, *es decir que se alcanza la cota, entonces claramente* x^0 *es una solución óptima de (2.2).* $k = 10$

Otro resultado que se puede inducir del mismo modo es que si (2.1) no tiene solución, entonces (2.2) tampoco. Por lo tanto, resolviendo el problema (2.1) permite acercarse al valor óptimo buscado para el problema (2.2), o incluso podría dar la solución óptima. A (2.1) se lo llama *relajación lineal* de (2.2), pues lo único que difiere entre uno y otro es la condición de integralidad. La Figura 2.1 muestra la diferencia entre ambos problemas.

Una conexión más profunda entre programación lineal y programación entera es que, para cada problema entero, existe uno de programación lineal $\text{máx}\{cx : Ax \leq b, A'x \leq b', x \in \mathbb{R}^n_+\}$, donde ambos comparten la misma solución óptima. Más aún, el conjunto de desigualdades dado por $Ax \leq b, A'x \leq b'$, corresponde a la descripción de la cápsula convexa de S_{PLE}.

A diferencia de la programación lineal, la programación lineal entera pertenece a los problemas NP-completos [10], problemas para los cuales no se conoce un algoritmo polinomial para resolverlos.

2.3. Métodos de resolución

En esta sección se enuncian los métodos de resolución para la programación lineal entera más utilizados en la actualidad.

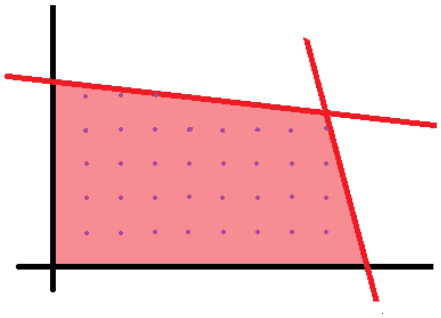

Figura 2.1: En rojo se muestra la región de factibilidad $\mathcal{S}_{\text{PL}}$ y los puntos aislados en violeta muestran la región de factibilidad $\mathcal{S}_{\text{PLE}}$. A $\mathcal{S}_{\text{PL}}$ se lo llama la *relajación lineal* de $\mathcal{S}_{\text{PLE}}$.

2.3.1. *Branch and Bound*

Uno de los métodos más conocidos para resolver problemas de programación lineal entera se basa en dividir el problema en pequeños sub-problemas utilizando un método llamado *Branch and Bound*. Este método consiste en particionar el espacio de soluciones en subproblemas, y a su vez particionar cada subproblema en más subproblemas siempre que no sea posible resolver los mismos (*branching*). De este modo se va construyendo un árbol desde el nodo raíz, el cual contiene al problema original, y en cada ramificación se plantea un subproblema asociado a éste. Finalmente las hojas del árbol son soluciones factibles del problema original. Este método así descripto puede generar una cantidad intratable de subproblemas para ser analizados, sin embargo posee una herramienta eficiente que permite eliminar tempranamente algunas de sus ramas.

Dado un nodo fijo del árbol, a éste le corresponde un subproblema de PLE del cual se resuelve su relajación lineal. Si se encuentra una solución entera el nodo se cierra. Si no, si el problema es infactible o bien su valor óptimo es peor que la mejor solución entera hallada hasta el momento en cualquier otro nodo del árbol (*bounding*), se procede a "podar " esa rama, ya que ningún subproblema suyo obtendrá una mejor solución. En cualquier otro caso, se prosigue con la ramificación.

El proceso de generación de los subproblemas no está determinado a priori y es específico a cada implementación. En el caso de los problemas del tipo 0-1 PLE, es natural idear un árbol binario (i.e.: de dos ramas por nodo) cuya ramificación consista en tomar una variable y fijarla en alguno de sus dos valores posibles.

Con respecto a la *exploración* del árbol, hay distintas variantes muy usadas en la práctica:

- **Búsqueda en profundidad** (*Depth-First-Search*). Se explora un subproblema del último problema analizado. De este modo se asegura llegar pronto a una hoja.
- **Búsqueda en anchura** (*Breadth-First-Search*). Se exploran todos los subproblemas del último problema analizado y luego, los subproblemas de cada uno de ellos. Así, se busca en todas las "direcciones" a la vez.
- **Mejor cota** (*Best-Bound-First*). Se explora el subproblema que tenga mejor valor objetivo de la relajación.

2.3.2. Planos de corte

Otra alternativa que existe para resolver problemas de programación lineal entera (PLE) es la técnica llamada de *planos de corte*. Se basa en el agregado de desigualdades válidas, llamados *cortes*, hasta que la solución óptima sea alcanzada con valores enteros.

El algoritmo se basa en resolver la relajación lineal del problema. Si es infactible o no acotado, entonces el problema original también lo será. Sin embargo, en caso de conseguir solución, puede ser de los siguientes tipos: todos sus valores enteros o alguno de sus valores real. En caso de que sea de valores enteros íntegramente, esta solución resulta la óptima del problema de PLE que se está resolviendo. El desafío se encuentra en el caso de que la solución tenga al menos una de sus variables en $\mathbb{R}$.

En este caso, se debe conseguir separar esa solución del conjunto de factibilidad del PLE. Se busca entonces una desigualdad que cumpla las siguientes condiciones:

- cada solución factible y entera deber seguir siendo factible luego del corte, es decir que no se reduzca el conjunto de factibilidad S_{PLE}, así la desigualdad es válida,
- la actual solución real no debe satisfacer la desigualdad.

Es fundamental para este algoritmo la eficiencia de encontrar *planos de corte* que *separen* soluciones con valores reales. A esto se lo llama *algoritmo de separación*. Las técnicas que han producido mejores resultados para hallar planos de corte se basan en el estudio de los poliedros asociados a cada problema. Estos procedimientos son en general de carácter heurístico, pues muchas veces el problema de separación resulta ser NP-completo.

En la práctica no suele usarse este método únicamente, ya que no son eficientes. Sin embargo, al combinarse con técnicas del estilo del *Branch and Bound* se logran los mejores resultados. A esta combinación se la llama *Branch and Cut*.

2.3.3. *Branch and Cut*

Es una generalización de *Branch and Bound* agregando planos de corte a las relajaciones resueltas en cada subproblema.

Al igual que en *Branch and Bound*, se arma un árbol en donde cada nodo representa un subproblema del original y sus ramificaciones son subproblemas de él.

Dado un nodo fijo, se resuelve la relajación lineal de su subproblema. En caso de que la solución hallada sea a valores enteros, se actualiza la cota para la poda en caso de que supere la obtenida hasta el momento. En este caso, si no se supera la mejor solución hallada hasta el momento o si la relajación lineal no es factible, se descarta el nodo sin ramificarse (poda) y se avanza al siguiente, donde la exploración del árbol se realiza en algún orden predeterminado como los detallados en la Sección 2.3.1.

Hasta aquí se procede del mismo modo que *Branch and Bound*. Si no ocurrió ninguno de los casos mencionados en el párrafo anterior, éste es el momento en el que se buscan planos de corte que separen a la solución encontrada y se agregan las desigualdades válidas que se encuentren. Luego, como el nodo cambió su formulación, nuevamente se resuelve la relajación lineal y se hace un análisis análogo al descripto a este mismo nodo actualizado.

En caso de que no se encuentren planos de corte, se procede a ramificar armando nuevos subproblemas de este nodo y se avanza a analizar al próximo.

Éste es básicamente el algoritmo de *Branch and Cut*. Cuántos planos de corte incorporar a cada formulación depende de muchos factores pero, en gran medida, depende del problema particular que se quiere resolver.

Capítulo 3

Descripción del problema original

En este capítulo se describe el problema original que se suscitó en la Provincia de Buenos Aires para llevar a cabo el Censo Nacional 2010. Se detallan las condiciones particulares del problema y el objetivo impuesto por las autoridades. Además, se plantea el modelo, los algoritmos utilizados y los resultados hallados entonces.

3.1. Marco del problema

El territorio argentino está dividido en 23 *provincias* además de la Ciudad Autónoma de Buenos Aires, que tiene un régimen jurídico especial. A su vez, cada provincia está dividida en *partidos* o *departamentos*. A los fines del censo, cada partido se divide en conjuntos de manzanas contiguas denominados *radios censales*. Cada radio censal contiene aproximadamente 330 hogares y, dependiendo de su densidad poblacional, entre 1 y 50 manzanas. A lo largo de este trabajo se referirá a los radios censales simplemente como "radios".

Buenos Aires es la provincia de mayor superficie (307.571 km^2) y mayor cantidad de habitantes (unos 15,6 millones de habitantes) de Argentina. No incluye a la Ciudad Autónoma de Buenos Aires pero sí al conurbano bonaerense, que forma parte del Gran Buenos Aires (GBA), un cordón de unos 10 millones de habitantes que rodea al territorio de la Ciudad de Buenos Aires y conforma con ella un núcleo poblacional único. La Provincia de Buenos Aires completa está dividida en 135 *partidos* donde se ubican 16.691 radios, lo que da un total de aproximadamente 5,5 millones de viviendas.

3.2. Segmentación de viviendas

Dado un radio, el problema de segmentación de viviendas consiste en particionarlas en conjuntos disjuntos de viviendas –llamados *segmentos*– de modo tal que cada segmento sea asignado a un censista. No todo conjunto de viviendas es un segmento válido, pues la Dirección Provincial de Estadísticas de Buenos Aires ha determinado un manual de procedimientos censales, donde se detallan las siguientes restricciones para la *segmentación*:

- Cada segmento debe estar compuesto por viviendas *contiguas*. Dos viviendas se consideran contiguas si están en la misma manzana y son adyacentes, o bien pertenecen a esquinas de dos manzanas distintas y dichas esquinas están enfrentadas sobre una misma calle, es decir, un segmento no puede cruzar una esquina en diagonal. La Figura 3.1 muestra algunos ejemplos de segmentos factibles y no factibles de acuerdo con este criterio

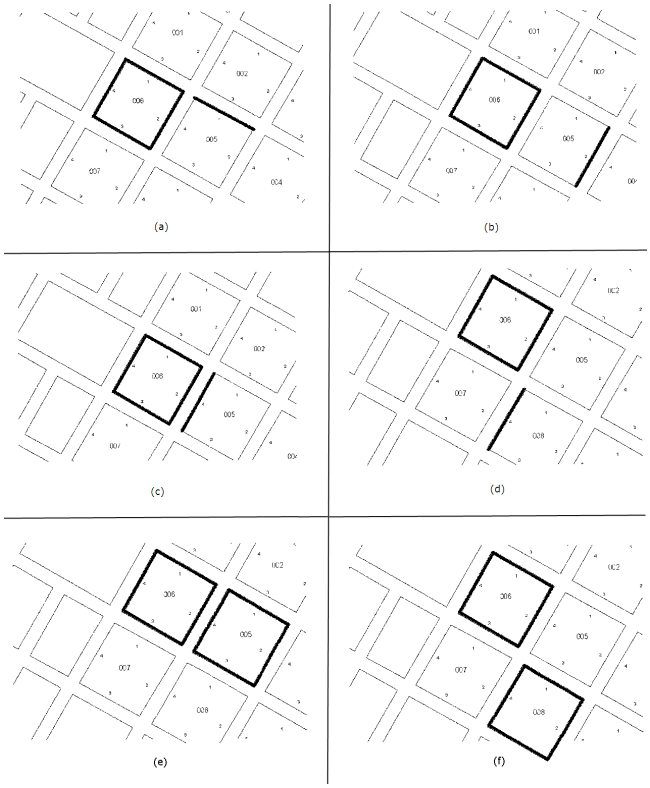

Figura 3.1: Los segmentos mostrados en (a), (c) y (e) son ejemplos de segmentos factibles que cruzan la calle. El segmento mostrado en (b) no es factible dado que las viviendas no son contiguas. Los segmentos mostrados en (d) y (f) no son factibles dado que cruzan una esquina en diagonal.

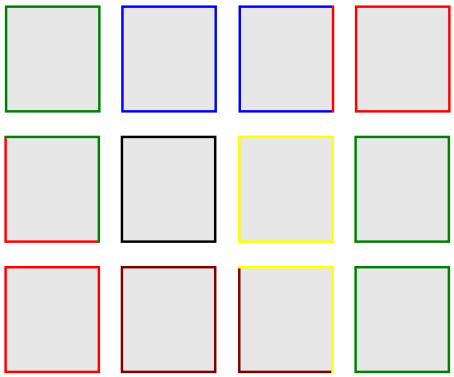

Figura 3.2: Un ejemplo de segmentación de un radio dado. Cada color representa un censista.

de contigüidad.

- Cada vivienda debe pertenecer a un único segmento y el conjunto de segmentos debe cubrir a todas las viviendas del radio.
- Si un *lado* de una manzana no tiene ninguna vivienda, ese lado también debe formar parte de algún segmento. Es importante notar que se denominan "lados" a los bordes rectos de una manzana. En los trazados urbanos típicos de las ciudades argentinas, una manzana es un rectángulo que contiene cuatro lados, aunque es frecuente encontrar manzanas con morfologías distintas.
- Cada segmento debe contener entre 32 y 40 viviendas.
- Cada censista debe recorrer una distancia inferior a cierta cota L, que depende de la densidad poblacional del radio en cuestión.
- Los censistas no pueden cruzar avenidas, vías del ferrocarril ni cursos de agua.
- Un segmento debe estar contenido dentro de su radio. Esta restricción es muy importante dado que como un segmento no puede involucrar viviendas de dos radios distintos, entonces el problema de segmentar todas las viviendas de las provincia se reduce a 16.691 instancias individuales e independientes entre sí.

Como ejemplo, en la Figura 3.2 se muestra una segmentación válida para un radio típico de 12 manzanas.

Además de las restricciones mencionadas, se establece un orden de preferencia de segmentación. El orden impuesto es el siguiente:

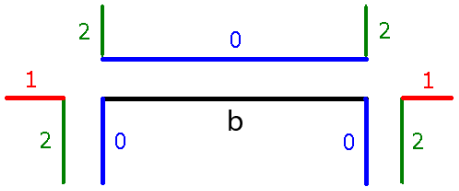

Figura 3.3: Niveles de adyacencia para el lado base b

1. Se deben privilegiar los segmentos que consten de manzanas completas.

2. Los segmentos deben consistir de lados completos y ser tan "compactos" como sea posible. Esto hace referencia a la amplitud de un segmento a lo largo de las manzanas del radio al que pertenece. Por ejemplo, un segmento constituído por una o varias manzanas completas es considerado muy compacto, mientras que un segmento que recorre varias manzanas utilizando pocos lados por manzana es considerado poco compacto. El objetivo de emplear segmentos tan compactos como sea posible es el de minimizar la posibilidad de que los censistas cometan errores en sus recorridos.

3. Si no es posible contar con lados completos (por ejemplo, si un lado tiene más de 40 viviendas), se lo puede partir y distribuirlo en más de un segmento. En este caso se prefiere que los segmentos contengan todos los departamentos de un mismo edificio, de modo tal que cada edificio sea atendido por un censista.

4. Si un edificio tiene más de 40 departamentos, entonces se debe dar preferencia a los segmentos que contengan pisos completos. En otras palabras, idealmente cada piso de un edificio debe ser atendido por un mismo censista.

Como preferencia adicional, para un lado fijo b, sus vecinos tienen distintos *niveles de adyacencia* según su posición relativa. Por ejemplo, luego de recorrer el lado b de la Figura 3.3, es preferible no cruzar la calle y continuar por los lados adyacentes a b en la misma manzana o bien cruzar al lado directamente enfrentado a b, estos lados se los denomina de *nivel de adyacencia* 0 (cero) con respecto al lado b. Si esto no es posible, es preferible continuar el segmento por el lado de la siguiente manzana sin doblar en la esquina; éstos tienen *nivel de adyacencia* 1 (uno). Si esto no es posible, se admite continuar en los lados de las manzanas adyacentes y dar un giro al recorrido, con *nivel de adyacencia* 2 (dos).

3.3. Modelo de programación lineal entera

Para este problema, se formuló el modelo de programación lineal entera que se detalla a continuación. Como se explicó en la Sección 3.2, se resolverá cada radio por separado ya que son instancias individuales. Sea $\mathcal{R}$ el radio a segmentar y sea $\mathbb{S}$ el conjunto de todos los segmentos factibles sobre el radio $\mathcal{R}$. Para cada segmento $s \in \mathbb{S}$, se introduce la variable binaria x_s, de modo tal que $x_s = 1$ si y sólo si el segmento s participa de la solución.

Con el objetivo de maximizar la compacidad de los segmentos seleccionados, se propone la siguiente función de *compacidad*, dado un segmento $s \in \mathbb{S}$:

$$\text{comp}(s) = \frac{\text{lados}(s)}{\text{manzanas}(s)}, \tag{3.1}$$

donde lados(s) representa la cantidad de lados en el segmento s y manzanas(s), la cantidad de manzanas en el segmento s. Observar que lados(s) $\geq$ manzanas(s) y este resultado hace que comp(s) ≥ 1 $\forall s \in \mathbb{S}$.

Notar que maximizar la compacidad es equivalente a buscar la máxima cantidad de lados en la menor cantidad de manzanas posibles, lo que logra reflejar el deseo de que el censista se mantenga en la misma manzana o recorra muchas viviendas en pocas manzanas. Por ejemplo, en la Figura 3.4, el segmento (a) tiene comp(s_a) $= \frac{5}{2}$ mientras que el segmento (b) posee comp(s_b) $= \frac{4}{4} = 1$. Aquí se puede apreciar el hecho de que cuanto más compacto sea el segmento, mejor será el recorrido según lo detallado anteriormente.

Con esta definición de compacidad, un segmento que involucra una manzana rectangular completa tendrá una compacidad de 4, mientras que un segmento que involucre cuatro lados en cuatro manzanas distintas tendrá una compacidad de 1 y, como se detalló en la Sección 3.2, es deseable que un censista recorra manzanas completas o lo más completas posibles. Sin embargo, analizando la Figura 3.5 cabe notar que, de esta manera, un segmento formado por una man-

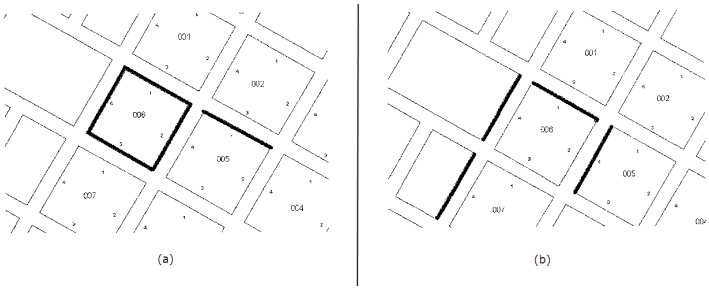

Figura 3.4: Las dos segmentaciones son válidas, pero la (a) es más deseable que la (b) pues en la (a) se toma una manzana completa más un lado de otra mientras que en la (b) se toman lados sueltos de distintas manzanas. La segmentación (b) haría un trazado muy confuso para ser llevado a cabo exitosamente por los censistas.

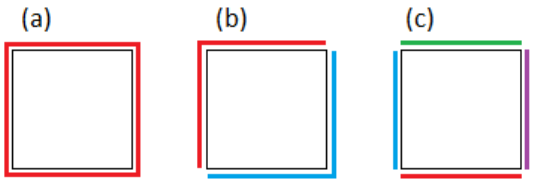

Figura 3.5: Segmentaciones en distintas manzanas, en (a) se tiene un segmento que abarca la manzana completa, en (b) se divide la manzana en mitades formando 2 segmentos y en (c) se divide en 4 partes. Se prefiere la segmentación (a) por sobre el resto, pues se desea que un censista recorra la manzana completa o lo más completa posible, luego se prefiere la segmentación (b) y, por último, la (c).

zana completa logra la misma compacidad que sumar dos segmentos formados por media manzana cada uno, o bien cuatro segmentos formados por cuartos de manzana, sin embargo el primer caso es el más deseado. Para priorizar los segmentos de mayor compacidad y evitar estos casos, se define la *valuación* de un segmento s como

$$\text{val}(s) = k^{\text{comp}(s)},$$

donde $k > 1$. En [5] se indica que tras algunas pruebas preliminares para determinar el valor de k, se fijó este valor en 10, pues con este valor se obtuvieron los mejores resultados. Este criterio fue definido en conjunto con los responsables de la planificación del censo en la Provincia de Buenos Aires, y proporcionó resultados satisfactorios. Veremos más adelante en este trabajo algunos problemas que supone esta decisión.

Como se describió en la Sección 3.2, en caso de no poder asignar un lado completo a un segmento, por ejemplo porque está excedido en cantidad de viviendas, se recurre a partirlo. Es decir, se generan dos o más fragmentos de ese mismo lado, denominados *partes*, a los que se los trabaja como si fueran lados independientes, siempre respetando las condiciones de vecindad. En la Sección 3.4 se describirá con más detalle este método de fragmentación de lados.

La definición de compacidad propuesta no refleja el deseo de obtener, para un mismo segmento, la mayor cantidad de partes posibles de un lado para así conseguir que un segmento contenga al lado completo o casi completo. Es decir, es necesario corregir esa definición. Finalmente, se decidió que la siguiente idea de compacidad discriminaría mejor estos casos:

$$\text{comp}(s) = \frac{\frac{\text{partes}(s)}{\text{lados}(s)}}{\text{manzanas}(s)} \tag{3.2}$$

Al querer maximizar esta función de compacidad, lo que se consigue es obtener la mayor relación de partes por lado, con la cantidad mínima de manzanas.

Con estas definiciones, la función objetivo de modelo de programación lineal entera busca maximizar la suma de la valuación (definida con la función de compacidad (3.2)) de los segmentos que se elijan para ser recorridos (o sea, en los que $x_s = 1$),

$$F(x) = \sum_{s \in \mathbb{S}} \text{val}(s) \cdot x_s$$

Para construir la región factible, se recurre a las restricciones impuestas en el inicio de este capítulo:

Cada vivienda debe ser cubierta por exactamente un segmento

Sea $\mathbb{V}$ el conjunto de viviendas de $\mathcal{R}$, para cada vivienda $v \in \mathbb{V}$, llamamos $S_v \subseteq \mathbb{S}$ al conjunto de segmentos factibles que incluyen a la vivienda v. Las ecuaciones que reflejan esta restricción son:

$$\sum_{s \in S_v} x_s = 1 \ \forall v \in \mathbb{V}.$$

Observar que la restricción $\sum_{s \in S_v} x_s \geq 1$ a pesar de asegurarse de cubrir la vivienda v del radio, puede ocasionar que una vivienda sea censada más de una vez, y esto no está permitido.

Cada lado sin viviendas debe ser recorrido por exactamente un segmento

Esto se pide para corroborar que lados aparentemente sin viviendas no hayan cambiado de situación. Sea $\mathbb{L}_0$ el conjunto de lados sin viviendas de $\mathcal{R}$, para cada lado sin viviendas $l \in \mathbb{L}_0$, llamamos $L_l \subseteq \mathbb{S}$ al conjunto de segmentos factibles que incluyen al lado l. Las ecuaciones que reflejan esta restricción son:

$$\sum_{s \in L_l} x_s = 1 \ \forall l \in \mathbb{L}_0.$$

Dominio del problema

Dado que x_s representa una decisión binaria, los valores que puede tomar son $\{0, 1\}$ y, por lo tanto, el dominio del problema es:

$$x \in \{0, 1\}^{|\mathbb{S}|}$$

Con estas definiciones se obtiene un modelo de programación lineal entera del tipo 0-1 PLE.

Modelo

Finalmente, el modelo de programación lineal entera de segmentación para este problema es el siguiente:

$$\max F(x) = \sum_{s \in \mathbb{S}} \text{val}(s) \cdot x_s \tag{3.3}$$

$$\text{sujeto a } \mathcal{S} \begin{cases} \sum_{s \in S_v} x_s & = & 1 & \forall v \in \mathbb{V} \\ \sum_{s \in L_l} x_s & = & 1 & \forall l \in \mathbb{L}_0 \\ x_s & \in & \{0,1\} & \forall s \in \mathbb{S} \end{cases}$$

Este modelo tiene una cantidad muy grande de variables, dado que la cantidad de segmentos factibles crece en forma exponencial a medida que aumenta la cantidad de manzanas en el radio en cuestión. Por este motivo, no resulta práctico ejecutar el modelo con todos los segmentos factibles. Una opción natural es recurrir a un algoritmo de generación de columnas, aunque este enfoque tiene el inconveniente de que no resulta claro a priori cómo resolver satisfactoriamente el subproblema de generación de columnas (segmentos válidos, en nuestro caso) sin recurrir a heurísticas o búsquedas exhaustivas.

Es importante observar que no se está teniendo en cuenta en este modelo las preferencias en cuanto a los niveles de adyacencia utilizados en los segmentos que cruzan las calles. Es decir, si se encuentra la solución óptima del problema (3.3), ésta no reflejará la preferencia de los distintos niveles de adyacencia. Estas preferencias se podrían incorporar en la función objetivo a través de la función de valuación de los segmentos, aunque esto agregaría parámetros adicionales que deberían ser ajustados para obtener resultados satisfactorios.

Por estos motivos, se implementó un enfoque basado en la resolución secuencial de este modelo para subconjuntos cada vez más grandes de segmentos, teniendo en cuenta explícitamente los niveles de adyacencia utilizados por los segmentos. Este esquema resultó exitoso por el tipo de soluciones que se hallaron, de acuerdo con los resultados reportados en la Sección 3.5.

3.4. Algoritmos utilizados

En esta sección se describe el algoritmo que se implementó para encarar la resolución del problema de segmentación de viviendas para el Censo 2010.

Se dirá que un segmento se encuentra *excedido* si contiene más de 40 viviendas o su longitud supera el límite L. Se dice que un segmento es *factible* si no está excedido y tiene al menos 32 viviendas. En el contexto del algoritmo, los segmentos **no** necesariamente son factibles. Para $\delta \in \{0, 1, 2\}$, se dice que un segmento es δ-*conexo* si sus lados están conectados por adyacencias de tipo a lo sumo δ, de acuerdo con la especificación de la Sección 3.2.

Algoritmo 1 Algoritmo de segmentación para nivel δ de adyacencias.

Entrada: Radio $\mathcal{R}$
1: $S_b \leftarrow \{\}$ /* *conjunto base* */
2: **para** toda manzana q **hacer**
3: $S_b \leftarrow S_b \cup \{$segmentos de q no excedidos$\}$
4: **fin para**
5: $i \leftarrow 1$
6: $S_i \leftarrow S_b$
7: **repetir**
8: Ejecutar el modelo de programación lineal entera de segmentación con los segmentos factibles de S_i
9: **si** hay solución **entonces**
10: Terminar (con solución)
11: **fin si**
12: $S_{i+1} \leftarrow S_i$
13: **para** todo $(s_i, s_b) \in S_i \times S_b$ **hacer**
14: **si** $s_i \cup s_b$ es un segmento δ-conexo no excedido **entonces**
15: $S_{i+1} \leftarrow S_{i+1} \cup \{(s_i \cup s_b)\}$
16: **fin si**
17: **fin para**
18: **si** $S_{i+1} = S_i$ **entonces**
19: Terminar (sin solución)
20: **fin si**
21: $i \leftarrow i + 1$
22: **hasta** $i > MI$
23: Terminar (sin solución)

El Algoritmo 1 ilustra el procedimiento propuesto, tomando los datos geográficos del radio como datos de entrada y el valor $\delta \in \{0, 1, 2\}$ como parámetro. Para $i \geq 1$, se denomina S_i al conjunto de segmentos no excedidos (aunque no necesariamente factibles) que involucran no más de i manzanas. En la i-ésima iteración, el Algoritmo 1 ejecuta el modelo de programación lineal entera de segmentación utilizando solamente los segmentos factibles $S'_i \subseteq S_i$. Si el modelo no tiene solución factible, se continúa iterando hasta que el modelo tenga solución factible, o S_i no varíe con respecto a S_{i-1}, o bien hasta llegar a un límite M de iteraciones especificado de antemano. En cuanto el modelo se torna factible, el algoritmo termina devolviendo la solución obtenida por el modelo.

Es importante observar que la solución obtenida por este algoritmo puede no ser óptima para el modelo de programación lineal entera de segmentación con *todos* los segmentos, dado que se ejecuta este modelo con un subconjunto de segmentos $S'_i \subseteq \mathbb{S}$. Sin embargo, el procedimiento secuencial propuesto permite tener en cuenta de una manera muy natural las preferencias sobre la compacidad de los segmentos seleccionados, aumentando la posibilidad de encontrar primero

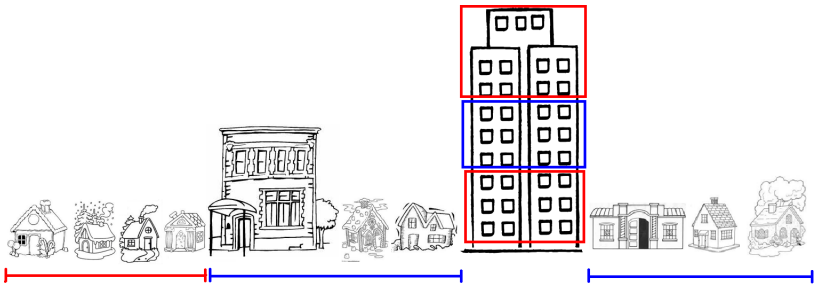

Figura 3.6: Ejemplo de fragmentación de un lado, de modo que cada parte no se exceda en P viviendas.

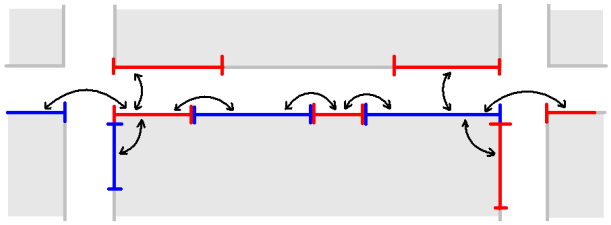

Figura 3.7: Adyacencias para lados partidos. Las flechas indican las partes que se consideran vecinas.

las soluciones preferidas.

En las líneas 2-4, el algoritmo genera el conjunto base de segmentos S_b utilizando lados de manzanas completos. En este paso, se generan para cada manzana todos los posibles segmentos no excedidos contenidos en la manzana. Con el objetivo de hallar soluciones formadas por segmentos lo más compactos posible, se ejecuta el Algoritmo 1 secuencialmente para $\delta = 0, 1, 2$, interrumpiendo el procedimiento en cuanto se encuentre la primera solución factible.

Si luego de todo el proceso mencionado no se encuentra solución factible, se activa la opción de partir lados. Para ello, se introduce un parámetro P que indica el número máximo de viviendas que puede tener un lado y se fragmentan en dos o más partes los lados que superen en viviendas a este valor, como se muestra en la Figura 3.6. Cada una de estas *partes* debe tener a lo sumo P viviendas. El algoritmo de fragmentación de lados es un procedimiento goloso, e intenta dejar en una misma parte los departamentos de un mismo edificio. Para los lados partidos se especifican nuevas adyacencias, de acuerdo con la Figura 3.7. Vale aclarar que mientras más chico sea P, mayor será la cantidad de lados (i.e., partes) de las manzanas del radio y mayor será el tamaño del conjunto base S_b.

Una vez partidos los lados de acuerdo con la descripción anterior, se ejecuta el Algoritmo 1 nuevamente secuencialmente para $\delta = 0, 1, 2$, intentando encontrar una solución factible. Si aún así no se encuentra una solución factible, se reduce aún más el valor de P consiguiendo partes aún más chicas y se repite el proceso. El Algoritmo 2 ilustra este procedimiento, recibiendo como parámetro una lista, llamada PL, que contiene los sucesivos valores a utilizar para el parámetro P (en orden decreciente). Si luego de todo este procedimiento no se encuentra una solución factible, el procedimiento termina informando al usuario que no se pudo encontrar una segmentación. Si la lista PL concluye con un 1, se terminan hallando todos los segmentos del radio $\mathcal{R}$, es decir $S_i = \mathbb{S}$ lo que haría que el modelo sea intratable dada la cantidad de segmentos a generar. Cabe mencionar que inicialmente el valor de P es menor o igual a 40.

Algoritmo 2 Algoritmo de segmentación.

Entrada: Radio $\mathcal{R}$, Lista PL
1: **para** cada $P \in PL$ **hacer**
2: Partir los lados que superen en viviendas al valor P.
3: **para** $\delta = 0$ **hasta** 2 **hacer**
4: Ejecutar el Algoritmo 1 para δ.
5: **si** el Algoritmo 1 encuentra una solución **entonces**
6: **devolver** la solución
7: Terminar (con solución)
8: **fin si**
9: **fin para**
10: **fin para**
11: Terminar (sin solución)

Todos los parámetros que se han definido en esta sección deben ser manejados con cuidado ya que según el radio a resolver, valores erróneos pueden incrementar los tiempos de ejecución. Los parámetros requieren de distintos valores según el tipo de radio que se esté trabajando. Con la aprobación de las autoridades de la Provincia de Buenos Aires, se resolvió clasificar los radios del siguiente modo:

A *urbanos*, radios de hasta 10 manzanas,

B *semi-urbanos*, radios de entre 11 y 30 manzanas,

C *rurales*, radios de más de 30 manzanas.

La Figura 3.8 muestra ejemplos de radios para las categorías descriptas y el Cuadro 3.1 muestra los valores utilizados para cada cateogría en las ejecuciones de estos algoritmos para algunos de los parámetros del modelo. El parámetro MI representa la cantidad máxima de veces que se generan nuevos segmentos y se vuelve a ejecutar el modelo, es decir los conjuntos de segmentos generados

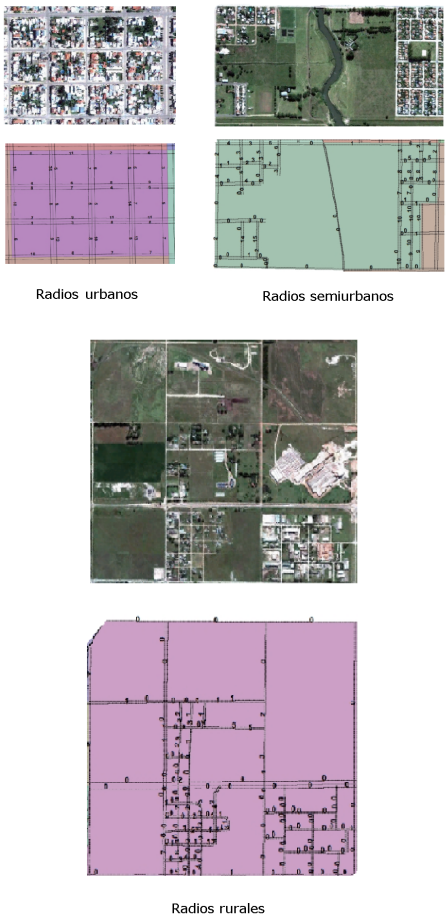

Figura 3.8: Ejemplo de cada tipo de radio.

durante el algoritmo son $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_{MI} \subseteq \mathbb{S}$. Si se alcanza a ejecutar el modelo de programación lineal entera con S_{MI} y no se encuentra solución, entonces el algoritmo termina sin solución. El parámetro MT determina el límite de tiempo para la resolución de cada modelo. En caso de que se alcance este límite, de haber hallado alguna solución, se devuelve la misma y se toma como solución del problema.

3.5. Resultados

El proceso de segmentación en la Provincia de Buenos Aires para el Censo Nacional 2001 (el anterior censo realizado en Argentina) se realizó manualmente, demandando 25 operadores a tiempo completo que trabajaron durante 30

	Parám.	**A** Urbanos	**B** Semi-urbanos	**C** Rurales
Cantidad máxima de iteraciones en la generación de segmentos	MI	5	8	9
Número máximo de viviendas por parte	PL	[32, 15, 10, 5]	[32, 15, 10]	[40, 32, 15]
Límite de tiempo para la ejecución del modelo de PLE (seg)	MT	60	60	120

Cuadro 3.1: Valores utilizados para los parámetros del algoritmo según el tipo de radio.

días (es decir, unas 6.000 horas-hombre). En cambio, el Censo Nacional 2010 fue la primera oportunidad en la que la segmentación de viviendas en la Provincia de Buenos Aires se realizó por medio de herramientas computacionales automáticas. El algoritmo original obtuvo segmentaciones satisfactorias para el 96 % de los radios en aproximadamente 320 horas de CPU sobre un computador con procesador Intel Celeron © a 2,4 GHz y memoria RAM de 2 GB.

En aproximadamente unos 600 radios (el 4 % del total) el algoritmo no encontró una solución factible en forma automática. Estos radios fueron resueltos utilizando la herramienta pero relajando levemente las restricciones del problema, o bien en forma manual.

La performance del algoritmo demostró ser muy sensible a la parametrización utilizada. Con un conjunto adecuado de parámetros, la resolución puede demorar unos pocos segundos, mientras que un juego de parámetros mal seleccionado puede comprometer la factibilidad del problema o bien llevar la cantidad de segmentos a varias centenas de miles, haciendo que la resolución demore varias horas. La clasificación de los radios de acuerdo con su densidad permitió manejar adecuadamente estos parámetros. Por último, el procedimiento secuencial dado por los algoritmos fue fundamental para cumplir con las preferencias en cuanto a las características deseables para los segmentos.

La generación de los segmentos requiere de un tiempo relativamente corto, siendo el peor caso de unos 2 minutos con los parámetros especificados en el Cuadro 3.1. Más del 99 % de los modelos de programación lineal entera se resuelve en pocos segundos, implicando que la relajación lineal resulta muy ajustada y la primera solución factible hallada por Cplex suele ser óptima. En los casos donde se halló solución, son pocos los que se llega al límite de tiempo MT especificado en el Cuadro 3.1 con solución sub-óptima, la cual se toma dentro del Algoritmo 1 como la segmentación del radio.

El algoritmo descripto permitió realizar el proceso de segmentación de viviendas de la Provincia de Buenos Aires dentro de los plazos que imponía la planificación del censo. A diferencia del proceso de segmentación manual empleado en el censo anterior, en el que cada operador podía introducir un sesgo en la segmentación realizada, el proceso automático a cargo del algoritmo permitió generar una segmentación con criterios uniformes para toda la provincia,

contribuyendo así a una división del trabajo pareja entre los censistas. Además, los tiempos de procesamiento disminuyeron considerablemente con relación al de la operatoria manual.

Al respecto, el responsable de los sistemas de información geográfica para el Censo Nacional 2010 en la Provincia de Buenos Aires, Fernando Aliaga, comenta que "el uso de esta herramienta computacional nos permitió una segmentación homogénea con criterios de compacidad uniformes, a diferencia de la segmentación manual que depende en gran medida de las decisiones de los operadores" [1]. El Censo Nacional 2010 se realizó exitosamente el 27 de Octubre de 2010, y fue calificado por las autoridades de la Provincia de Buenos Aires como un éxito organizativo [19].

Capítulo 4

Nuevos enfoques del problema

En este capítulo se proponen nuevas variantes del problema original (3.3), el cual busca conseguir una solución factible con segmentos lo más compactos posibles, sin importar, por ejemplo, la cantidad total de censistas requeridos. Esto hace que se consigan soluciones deseadas en cuanto a "belleza" de los segmentos, sin embargo, quizás alguna otra solución con valor objetivo del mismo orden siga obteniendo segmentos satisfactorios y, a su vez, cumpla nuevos requerimientos. Además, se detallan y resuelven algunos problemas encontrados en la formulación original.

4.1. Minimización de censistas

La actividad de los censistas es una tarea remunerada (entre AR$250 y AR$300 por censista). Una propuesta interesante es la de dar un enfoque económico al problema minimizando la cantidad de censistas requeridos en la segmentación. Luego, el problema a resolver es:

$$\min \sum_{s \in \mathbb{S}} x_s \tag{4.1}$$

$$\text{sujeto a } \mathcal{S} \begin{cases} \sum_{s \in S_v} x_s = 1 & \forall v \in \mathbb{V} \\ \sum_{s \in L_l} x_s = 1 & \forall l \in \mathbb{L}_0 \\ x_s \in \{0,1\} & \forall s \in \mathbb{S} \end{cases}$$

Notar que la región de factibilidad no cambia con respecto al problema inicial (3.3), sólo se modifica la función objetivo.

Este planteo deja de lado la noción de compacidad, que es un hecho que no se quiere perder, ya que conseguir menos cantidad de censistas a costas de obtener un trazado de segmentos "dificultoso" para ser transitado no es algo deseable. Es importante no perder el requisito de que los segmentos sigan siendo compactos para facilitar los recorridos.

Para esto, se agregó una segunda fase en esta implementación que consiste en lo siguiente: una vez hallada la cantidad mínima de censistas a la que notaremos $\mathcal{M}$, se buscará maximizar la valuación definida en la Sección 3.3, con la restricción de que la cantidad de segmentos de la solución no supere $\mathcal{M}$. Es decir, se buscará solución al siguiente problema:

$$\max F(x) = \sum_{s \in \mathbb{S}} \text{val}(s) \cdot x_s \tag{4.2}$$

$$\text{sujeto a } \mathcal{S}_{min} \begin{cases} \sum_{s \in S_v} x_s = 1 & \forall v \in \mathbb{V} \\ \sum_{s \in L_l} x_s = 1 & \forall l \in \mathbb{L}_0 \\ \sum_{s \in \mathbb{S}} x_s = \mathcal{M} & \\ x_s \in \{0,1\} & \forall s \in \mathbb{S} \end{cases}$$

4.1.1. Algoritmo

En el problema de minimizar censistas, al problema original se le agrega la restricción $\sum_{s \in \mathbb{S}} x_s = \mathcal{M}$ donde $\mathcal{M}$ es la cantidad mínima de censistas. Es por esto que, para resolver (4.2), se necesita primero encontrar la mínima cantidad de censistas, para luego poder construir la restricción adicional.

Por lo tanto, como primera fase, se corre el Algoritmo 2 con el modelo (4.1). De allí, en caso de encontrar solución, se extraerán los siguientes datos que serán necesarios para la segunda fase: el valor objetivo, que será la cantidad mínima de censistas M, y el subconjunto de segmentos S_i en donde se halló el mínimo, ya que ahí mismo es donde se buscará la máxima valuación. El propósito de guardar ese subconjunto es el de agilizar la segunda fase, ya que la generación de segmentos hasta encontrar una solución factible ya estará concluida. Notar que la solución hallada en la primera fase no se utilizará en absoluto pues no refleja ningún nivel de compacidad, como se explicó en el planteo de este modelo. Entonces, se pasa a la segunda fase.

Una vez hallada la cantidad mínima de segmentos $\mathcal{M}$, simplemente queda por ejecutar el modelo (4.2) con los segmentos $\overline{S_i} \subseteq \mathbb{S}$. Vale aclarar que si se encontró solución en la primera fase, *debe* haber solución en ésta, ya que la región de factibilidad no es vacía, pues al menos contiene a la hallada en la primera fase. Luego, se garantiza que, si se llega a esta fase, entonces habrá solución óptima.

Ésta es la solución del modelo propuesto, pues tiene la mínima cantidad de censistas en S_i y a su vez la suma de las valuaciones es máxima.

4.2. Problema de minimax

La restricción adicional del modelo (4.2) puede llevar a empeorar mucho la valuación de los segmentos utilizados en la solución óptima encontrada con respecto al valor de (3.3), obteniendo soluciones con menos cantidad de censistas pero perdiendo la importancia de la calidad de soluciones que se aspira conseguir. Por esta razón se buscó un abordaje intermedio entre los problemas (3.3) y (4.2), en el cual se mantiene el objetivo de minimizar censistas aún respetando la búsqueda de una alta compacidad de segmentos.

La propuesta es establecer una cota inferior al minimizar censistas, sobre la suma de la valuación de los segmentos de la solución, por ejemplo, una fracción p sobre el valor objetivo óptimo $F(x^0)$ obtenido al resolver el problema original (3.3). Es decir, dada la máxima suma de valuaciones de una segmentación, se busca minimizar la cantidad de censistas acotando inferiormente la valuación total de los segmentos de la nueva solución. El valor de p es un parámetro del modelo. Luego, el problema a resolver es equivalente a (4.1) pero con una restricción adicional que impone dicha cota sobre la valuación. Así, se plantea el siguiente problema de programación lineal entera:

$$\min \sum_{s \in \mathbb{S}} x_s \tag{4.3}$$

$$\text{sujeto a} \begin{cases} \displaystyle\sum_{s \in S_v} x_s = 1 & \forall v \in \mathbb{V} \\ \displaystyle\sum_{s \in L_l} x_s = 1 & \forall l \in \mathbb{L}_0 \\ \displaystyle\sum_{s \in \mathbb{S}} \text{val}(s) \cdot x_s \geq p \cdot F(x^0) & \\ x_s \in \{0,1\} & \forall s \in \mathbb{S} \end{cases}$$

En este planteo, se encuentra un problema; como $\text{val}(s) = k^{\text{comp}(s)}$ es una función exponencial, la diferencia entre la valuación de dos segmentos distintos puede ser muy grande aún con leves cambios en sus compacidades. Así, pierde algo de sentido acotar la suma de las valuaciones con un porcentaje del valor óptimo de (3.3). Lo que en realidad se está buscando es que p actúe directamente sobre la compacidad de los segmentos, ya que no se quiere permitir que la nueva solución dé segmentos poco compactos. Una forma de evitar esto es comparar las compacidades de los segmentos, o sea, revertir el hecho de que $\text{val}(s)$ trabaja con funciones exponenciales con base $k \in \mathbb{N} - \{1\}$. Para esto, se propone reemplazar la tercera desigualdad con la siguiente:

$$\min \sum_{s \in \mathbb{S}} x_s \tag{4.4}$$

$$\text{sujeto a } \mathcal{S}_{minimax} \begin{cases} \sum_{s \in S_v} x_s = 1 & \forall v \in \mathbb{V} \\ \sum_{s \in L_l} x_s = 1 & \forall l \in \mathbb{L}_0 \\ \sum_{s \in \mathbb{S}} \text{val}(s) \cdot x_s \geq k^{p \cdot \log_k F(x^0)} & \\ x_s \in \{0,1\} & \forall s \in \mathbb{S} \end{cases}$$

Aquí entonces se encuentra la cantidad mínima de censistas con una cierta cota de "belleza" sobre la valuación total de la solución. Nuevamente, surge la posibilidad de que con la misma cantidad de censistas $\mathcal{M}$ se consiga una mayor valuación, luego se procede a ejecutar el modelo (4.2) con el valor objetivo $\mathcal{M}$ hallado en (4.4).

4.2.1. Algoritmo

Este algoritmo es similar al de minimizar censistas, ya que, nuevamente, requiere de datos que se extraen de corridas previas. Se procede a segmentar hasta encontrar una región factible y luego, fijado ese subconjunto de segmentos $S_i \subseteq \mathbb{S}$, se ejecutarán los modelos propuestos para este problema.

En primer lugar, se correrá el Algoritmo 2 con el modelo original (3.3), para así obtener una solución de máxima valuación, cuyo valor objetivo se utilizará para acotar inferiormente la valuación de la solución al momento de minimizar la cantidad de censistas. En caso de encontrar solución, al igual que al minimizar la cantidad de censistas, para las siguientes ejecuciones se necesitan los siguientes resultados de esta primera fase: S_i y el valor objetivo $F(x^0)$.

Con estos dos datos, se procede a ejecutar el modelo (4.4). Nuevamente, se puede asegurar a priori que este paso encontrará solución, pues la hallada al ejecutar el modelo original sigue siendo una solución factible de este nuevo modelo. Aquí se encontrará entonces la mínima cantidad de censistas $\mathcal{M}$ garantizando a su vez una solución suficientemente compacta, debido a la cota inferior impuesta. Sin embargo, no necesariamente se encontró la solución de segmentos de mayor valuación posible, pues la función objetivo $\sum_{s \in S_i} x_s$ no tiene en cuenta la valuación de los segmentos. Así, dado el $\mathcal{M}$ hallado, es atinado conseguir la solución que maximice la suma de las valuaciones con la mínima cantidad de censistas $\mathcal{M}$. Con este fin, como tercer paso, se ejecuta el modelo (4.2), para así obtener con la mínima cantidad de censistas la solución de segmentos más compactos posibles.

Tanto el algoritmo para minimizar censistas como el de minimax concluyen en una fase en donde se ejecuta el modelo (4.2), el cual maximiza la suma de las valuaciones sujeto, entre otras restricciones, a que la cantidad de segmentos

utilizados sea igual a $\mathcal{M}$, para así asegurarse de tomar la mínima cantidad.

Para concluir con estas variantes, vale recordar la existencia del parámetro de límite de tiempo MT (ver Cuadro 3.1), que interrumpe el algoritmo de *Branch and Cut* en caso de ser excedido, retornando la mejor solución hallada hasta ese momento. Debido a esto, podría darse el caso en que la solución hallada $\mathcal{M}$sea sub-óptima, es decir, podría ocurrir que exista en la región factible alguna solución con una menor cantidad aún de censistas y no se haya encontrado. Por esta razón, podría ocurrir que en la siguiente etapa, aun utilizando la misma región factible, se reduzca la cantidad de censistas hallada. Entonces, resulta natural relajar la restricción sobre la cantidad de censistas a considerar en la segmentación de la siguiente forma

$$\sum_{s \in \mathbb{S}} x_s \quad \leq \quad \mathcal{M},$$

para dar lugar a que esta situación no sea pasada por alto.

4.3. Profundización de la búsqueda

El algoritmo implementado en el Censo Nacional 2010, que se describió en detalle en la Sección 3.4, es básicamente un "generador" de subconjuntos de segmentos válidos los cuales se agregan al modelo en cada iteración hasta encontrar solución al mismo, en cuyo caso termina, incluso sabiendo que la solución hallada podría no ser la óptima para el modelo completo (3.3).

Análogamente, cuando se quiere encontrar la mínima cantidad de censistas, quizás no se halle la óptima ya que se está trabajando con un subproblema que contiene menos soluciones factibles que la región de factibilidad completa $\mathcal{S}$. Por la forma de construir estos subconjuntos, lo que sí se puede esperar es que se encuentre una solución con compacidad alta. Luego, lo que se conseguirá realmente en la implementación del algoritmo es una solución que minimizará la cantidad de censistas entre un subconjunto de segmentos bien compactos.

Una manera de buscar una menor cantidad de censistas aún es permitir que el algoritmo prosiga generando más segmentos, incluso luego de haber encontrado una solución. Esto agranda la región de factibilidad, aumentando la posibilidad de encontrar alguna segmentación con menor cantidad de censistas que la hallada en la iteración anterior. A continuación se describe una implementación para esta variante.

4.3.1. Algoritmo

Lo que se modifica en la implementación de esta propuesta es el Algoritmo 1, pues no se concluye en caso de hallar solución. Lo que se hace en cambio es avanzar una generación más de segmentos, es decir, se construye una vez más S_{i+1} del modo que está descripto en las líneas 12-17 (ver página 19). De esta

forma, se generan más segmentos y se tiene la posibilidad de mejorar la solución encontrada.

Luego, el momento de avanzar con la generación de segmentos es cuando se halla solución para el problema de encontrar la mínima cantidad de censistas, es decir, al resolver el modelo (4.1) o el modelo (4.4) en caso de estar resolviendo el problema de minimax. Si no mejora la solución inicial, lo mejor sería volver al subconjunto de segmentos S_i ya que tiene menos cantidad de elementos y hace que la ejecución del modelo de maximizar la valuación (4.2) sea más rápida.

En conclusión, los cambios del Algoritmo 1 son los que se detallan a continuación. En la línea 8 se ejecuta el modelo (4.1), o (4.4) si se está trabajando con una cota de la valuación. También se debe reemplazar la línea 10 por el Algoritmo 3. O sea, si se encuentra solución, en vez de terminar, ejecutar este nuevo algoritmo.

Algoritmo 3 Algoritmo para profundizar la búsqueda.

1: $\mathcal{M} \leftarrow$ valor objetivo de la solución.
2: Ejecutar líneas 12-17 del Algoritmo 1 /* *Avanzar una generación más de segmentos* */
3: Ejecutar el mismo modelo con los segmentos factibles de S_{i+1}
4: $\mathcal{M}_{\text{aux}} \leftarrow$ valor objetivo de la nueva solución
5: **si** $\mathcal{M}_{\text{aux}} < \mathcal{M}$ /* *Se mejoró la solución* */ **entonces**
6: $\quad \mathcal{M} \leftarrow \mathcal{M}_{\text{aux}}$
7: $\quad$ Ejecutar el modelo (4.2) con los segmentos factibles de $\overline{S_{i+1}}$ /* *Maximizar la valuación con la "nueva" cantidad mínima de censistas* */
8: **si no**
9: $\quad$ Ejecutar el modelo (4.2) con los segmentos factibles de $\overline{S_i}$
10: **fin si**
11: Terminar (con la última solución hallada)

4.4. Nuevas funciones de valuación

Con la actual función de valuación se han encontrado soluciones satisfactorias en cuanto a la simpleza del recorrido de cada censista, lo que ha resultado en que sea aceptada por el gobierno de la Provincia de Buenos Aires para ser utilizada en el Censo Nacional 2010. Sin embargo, analizando las soluciones halladas se han encontrado algunos problemas, los cuales se describen en esta sección acompañados por propuestas para resolverlos.

4.4.1. Base de la función de valuación

Como se definió en la Sección 3.3, $\text{val}(s) = k^{\text{comp}(s)}$ es la valuación de un segmento $s \in \mathbb{S}$. En el modelo implementado en el Censo Nacional, se utilizó la

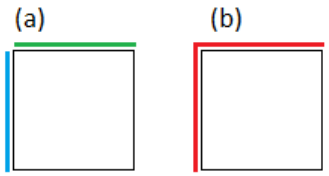

Figura 4.1: En la segmentación (a), se tienen dos segmentos de 1 lado y 1 manzana cada uno, lo que da una valuación de $2^1 + 2^1 = 4$, sin embargo la segmentación (b), que resulta preferida sobre (a) dado que toma mayor parte de la manzana, tiene valuación $2^2 = 4$. Este simple ejemplo, muestra que no es conveniente tomar $k = 2$.

base $k = 10$, sin embargo ello hace que las valuaciones de ciertos segmentos fueran números extremadamente grandes en comparación con otros segmentos. El problema que surge con el hecho de trabajar con números de órdenes tan distintos es que la función objetivo es suma de éstos y se encontraron varios casos donde se suman números de gran cantidad de cifras con otros de muy pocos dígitos, incurriendo así en errores numéricos relevantes para el problema.

Esto genera que, con el truncamiento, se pierdan los valores más pequeños de la suma. Entonces quizás distintas soluciones factibles que involucran a un segmento de valuación extremadamente alta den el mismo valor objetivo en la máquina a pesar de que en realidad una solución es mejor que otra.

Una forma de reducir estas magnitudes es utilizar un valor más pequeño de k, siempre por supuesto mayor que 1. Una primera propuesta fue la de tomar $k = 2$, pero falla en casos muy comunes, ver Figura 4.1. Entonces, se decidió tomar $k = 3$, lo que funcionó satisfactoriamente, eliminando ese tipo de problemas y a su vez reduciendo notablemente el orden de los números de valuación de los segmentos.

4.4.2. Definición de compacidad

La definición de compacidad de un segmento relaciona la cantidad de partes utilizadas, con la de lados y manzanas. Pero, como vimos en la Sección 3.2, podría ocurrir que no haga falta partir los lados de las manzana (esto se da mucho en las zonas rurales y semi-urbanas, pues la cantidad de viviendas por lado es baja). En estos casos, partes(s) = lados(s) pues una parte es un lado completo de la manzana. Allí, la compacidad tendrá el siguiente valor:

$$\text{comp}(s) = \frac{\frac{\cancel{\text{partes}(s)}}{\cancel{\text{lados}(s)}}}{\text{manzanas}(s)} = \frac{1}{\text{manzanas}(s)}.$$

Luego, en este contexto, maximizar la suma de las valuaciones será equivalente a buscar la mínima cantidad de manzanas por segmento. Si bien esto es

algo deseable, de esta manera no se está buscando tener la mayor cantidad de lados por manzana. Peor aún, un segmento formado por un lado tiene la misma compacidad que una manzana completa. Se encontraron ejemplos en los que esta definición de compacidad es tan mala como uno quisiera, ver Figura 4.2.

Es por esto que la siguiente función de compacidad es más general y cubre ambos tipos de casos, aquellos en donde se parten los lados y aquellos en donde no:

$$\text{comp}(s) = \frac{\text{lados}(s) + \frac{\text{partes}(s)}{\text{lados}(s)} - 1}{\text{manzanas}(s)} = \underbrace{\frac{\text{lados}(s)}{\text{manzanas}(s)}}_{(3.1)} + \frac{\frac{\text{partes}(s)}{\text{lados}(s)} - 1}{\text{manzanas}(s)}. \tag{4.5}$$

En el caso donde no se parten los lados, se consigue que la compacidad esté definida por (3.1). De ser necesario partir los lados, se estará maximizando la suma de (3.1) junto con $\frac{\frac{\text{partes}(s)}{\text{lados}(s)} - 1}{\text{manzanas}(s)}$, que es equivalente a (3.2), es decir, al buscar maximizar la valuación con esta nueva definición de compacidad se cumple el mismo objetivo inicial de obtener la mayor cantidad de partes por lado en la menor cantidad de manzanas posibles. En conclusión, se partan o no los lados, se está realizando lo deseado.

Definiendo la compacidad de un segmento con (4.5) se logra apreciar en mayor medida los distintos tipos de segmento utilizados. Sin embargo, quedan

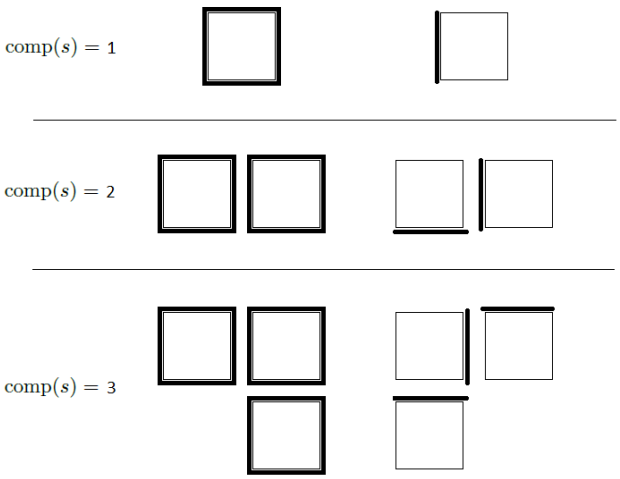

Figura 4.2: En cada fila se muestran dos segmentaciones distintas que tienen la misma compacidad según la definición (3.1), sin embargo la columna izquierda es la situación favorita para cada valor de compacidad, mientras que la situación de la columna derecha es la menos deseada.

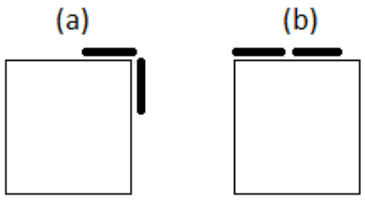

Figura 4.3: Ejemplo de dos segmentos distintos con misma compacidad (4.5) donde se recurrió a partir lados. Sin embargo, es preferible el segmento (b) sobre (a) pues (a) dobla en la esquina.

aún ejemplos en los cuales un segmento deseable puede no destacarse frente a otros. Por ejemplo, la Figura 4.3 muestra dos segmentos formados por dos partes de una misma manzana. En el primero, ambas partes pertenecen a un mismo lado, mientras que en el segundo no. Claramente el primer segmento es preferible al segundo, sin embargo ambos poseen compacidad igual a 2, según la definición de (4.5). Éste fue el único problema que se logró detectar con esta definición de compacidad.

Una forma de resolver este problema es dándole mayor importancia al término que involucra la proporción de partes por lado en la definición, obteniendo así la siguiente variante:

$$\text{comp}(s) = \frac{\text{lados}(s) + \left(\frac{\text{partes}(s)}{\text{lados}(s)}\right)^t - 1}{\text{manzanas}(s)}, \tag{4.6}$$

donde $t > 1$.

Esta nueva fórmula de compacidad evita los problemas descriptos en el párrafo anterior. Sin embargo, posee un nuevo problema y es que puede llegar a involucrar números muy grandes. Además, se debe tener en cuenta que la valuación es una función exponencial cuyo exponente está dado por la compacidad. Todo esto conlleva a que se trabaje con números de muy distintos órdenes de magnitud y esto trae problemas de operaciones de punto flotante cuando se usa aritmética finita. En la experimentación se probaron ambas funciones de compacidad descriptas en esta sección obteniendo distintos resultados según los segmentos analizados.

Capítulo 5

Resultados

Los algoritmos propuestos fueron codificados en C++ y los modelos de programación lineal entera fueron resueltos con Cplex 12.1. Los datos de manzanas, lados y viviendas se tomaron de la base de datos geográfica de la Provincia de Buenos Aires, implementando en el sistema de información geográfica ESRI ArcGIS, las interfaces necesarias para exportar la información e importar y visualizar las segmentaciones obtenidas por los algoritmos propuestos.

En este capítulo, se mostrarán los resultados obtenidos para los distintos enfoques propuestos. Además, se analizarán las distintas soluciones halladas para un mismo radio. Para diversificar los resultados, las pruebas se hicieron sobre un 4,78 % del total de radios de la Provincia de Buenos Aires, en municipios representativos de cada tipo de zona, los cuales se listan a continuación:

ZONA A *urbana* Tigre y Vicente López, con un total de 292 radios.

ZONA B *semiurbana* Tigre y Ezeiza, con un total de 440 radios.

ZONA C *rural* Ezeiza y Marcos Paz, con un total de 65 radios.

5.1. Impacto económico

En el modelo original, en donde sólo se maximiza la valuación de los segmentos, no se considera de ninguna forma la cantidad de censistas. Esto hace que se tome una solución de valuación máxima, pero se puede estar perdiendo alguna otra solución con compacidad suficientemente alta, es decir cuyos segmentos trazan también caminos sencillos para los censistas y además la segmentación contiene menor cantidad de segmentos.

En esta sección se muestran los resultados de los modelos propuestos para reducir la cantidad de censistas y, por consiguiente, reducir el presupuesto invertido en llevar a cabo el censo.

En el Cuadro 5.1 se listan los porcentajes de reducción de cantidad de censistas discriminados por zonas para cada tipo de modelo propuesto. De la misma

	Con prof. de búsqueda		Sin prof. de búsqueda	
	Min. censistas	Minimax, $p = \frac{85}{100}$	Min. censistas	Minimax, $p = \frac{85}{100}$
Urbana	1,96 %	1,55 %	1,59 %	1,27 %
Semi-urbana	4,37 %	4,46 %	2,97 %	2,79 %
Rural	5,55 %	5,55 %	3,7 %	4,63 %

Cuadro 5.1: Porcentaje de ahorro de censistas para cada modelo propuesto.

	Con prof. de búsqueda		Sin prof. de búsqueda	
	Min. censistas	Minimax, $p = \frac{85}{100}$	Min. censistas	Minimax, $p = \frac{85}{100}$
Urbana	17,8 %	14,1 %	14,4 %	11,5 %
Semi-urbana	44,9 %	43,9 %	29,9 %	28 %
Rural	60 %	60 %	50 %	40 %

Cuadro 5.2: Porcentaje de radios donde se disminuye la cantidad de censistas con respecto a la solución dada por el modelo original.

manera, en el Cuadro 5.2 se muestra el porcentaje de radios en el que la solución dada por cada uno de los nuevos modelos difiere de la solución del modelo original, es decir, se muestra el porcentaje de radios en los que efectivamente se logró disminuir la cantidad de censistas.

Se puede observar que la mayor reducción de censistas se produjo en zonas rurales. Es probable que esto se deba a que las manzanas en estas zonas suelen tener muy poca cantidad de viviendas y por lo tanto es posible minimizar la cantidad de segmentos utilizados simplemente combinando manzanas completas. De esta manera, las soluciones encontradas mantienen siempre una alta compacidad.

Por el contrario, en las zonas urbanas, las manzana suelen tener una gran cantidad de viviendas y esto hace que reducir la cantidad de segmentos utilizados requiera una mayor fragmentación en las manzanas, e incluso en los lados mismos. Esto implica que las primeras soluciones halladas, es decir, las de mayor compacidad según el algoritmo de generación de segmentos, no reduzcan demasiado la cantidad de segmentos utilizados. Para este tipo de zonas, es probable que se requiera profundizar demasiado la búsqueda hasta poder encontrar el valor óptimo para $\mathcal{M}$. Sin embargo, como se comenta más adelante no es recomendable implementar tal profundización en la búsqueda, pues los modelos generados serían de un tamaño intratable en la práctica.

5.1.1. Minimización de censistas

De todas las pruebas realizadas, se logró diminuir en a lo sumo un censista por radio. Es decir, de los radios en los que se logra reducir la cantidad de censistas, ver Cuadro 5.2, la cantidad de segmentos en las soluciones dadas por los modelos (3.3) y (4.2) difiere en una unidad.

Dado que se logró reducir la cantidad de censistas entre un 1,27 % y un

5,55 % según el radio, que la remuneración por censista fue entre $250 y $300 y que en la Provincia de Buenos Aires participaron aproximadamente 230.000 censistas [21], se puede estimar que se podría haber ahorrado aproximadamente $ 4.000.000 en la ejecución del Censo Nacional realizado en la Provincia de Buenos Aires. Esta estimación tiene en cuenta el hecho de que la Provincia de Buenos Aires es una provincia predominantemente rural.

5.1.2. Problema de minimax

Utilizando la fracción $p = \frac{85}{100}$ para establecer la cota inferior de la suma de valuaciones, fueron menor a un 1 % los casos en los que al minimizar la cantidad de censistas se redujo la valuación en la solución obtenida por debajo de la cota, con respecto a la valuación usando el modelo original (3.3). Es decir, de los radios en donde se logra reducir la cantidad de censistas con respecto al modelo original, menos del 1 % está por debajo de la cota determinada por p. Más precisamente, el porcentaje de radios con solución de valuación menor a la cota determinada por $p = \frac{85}{100}$ fue de: 0,42 % para la zona urbana, 0,19 % para la zona semi-urbana y 0,97 % para la zona rural.

De todas maneras, en estos radios de baja valuación con respecto a la solución óptima de (3.3), la mayoría de las soluciones halladas contienen igualmente trayectos bastantes sencillos de transitar. Esto fue corroborado inspeccionando visualmente algunas de estas soluciones en la herramienta GIS utilizada en este trabajo.

Como ejemplo, la Figura 5.1 muestra el radio para el cual se obtuvo la mayor diferencia entre la valuación de la solución al minimizar censistas con respecto a la hallada al maximizar la valuación. Más precisamente, para que la segmentación hallada al minimizar censistas sea solución del problema de minimax, es necesario poner $p = \frac{66}{100}$, cota demasiado permisiva. Es decir, si se modelara el problema de minimax con el parámetro $p > \frac{66}{100}$, esta segmentación no formaría parte de la región de factibilidad y la solución coincidiría con la de maximizar la valuación. Observar que la solución obtenida al maximizar la valuación es notablemente mejor en cuanto al trazado de segmentos, ya que muchos son manzanas completas o lados completos, mientras que la solución obtenida al minimizar censistas tiene un trazado algo más complicado. De todas maneras, se puede considerar que, siendo éste el peor caso encontrado, el modelo de minimización propuesto en este trabajo resulta bastante favorable.

Este ejemplo muestra que es interesante y acertado proponer el modelo de minimax, con el fin de evitar este tipo de soluciones de características poco deseables para el trazado.

5.2. Profundizar la búsqueda

La variante de profundización de la búsqueda consiste en la generación de un nuevo conjunto de segmentos luego de haber hallado una primera solución para el problema. No es recomendable repetir demasiado este procedimiento (i.e.,

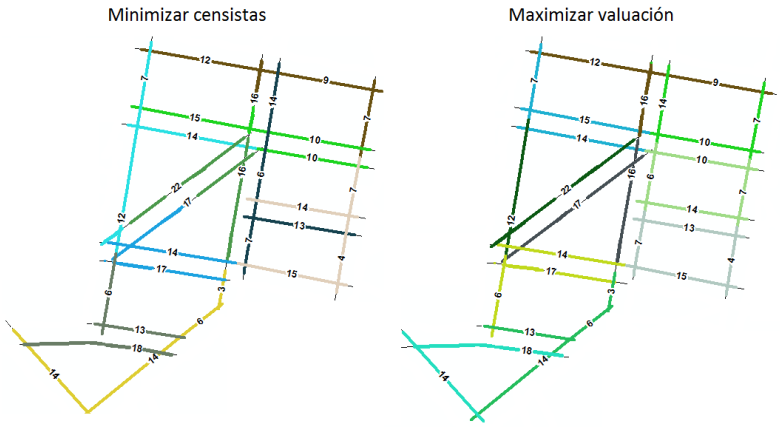

Figura 5.1: Peor caso hallado al comparar soluciones entre minimizar censistas y maximizar la valuación. Los valores sobre cada lado indican la cantidad de viviendas en él.

generar más y más segmentos) ya que cada iteración genera una vasta cantidad de segmentos nuevos lo que hace que haya un excesivo uso de memoria, además de demorarse más en dar una solución por cada radio. Una experimentación preliminar mostró que con avanzar una generación más de segmentos se logra conseguir en distintos radios llegar hasta aproximadamente 100.000 de segmentos válidos, lo que hace que la generación del modelo se demore demasiado o incluso, en algunos casos, se agote la memoria. Con las pruebas realizadas se llegó a que modelos de hasta 75.000 segmentos son manejables en cuanto al uso de memoria.

En los radios de prueba donde se ejecutó este modelo de profundización de la búsqueda se encontró que una, vez hallada una solución sobre el subconjunto de segmentos S_i, la cantidad de elementos que se incrementan en el subconjunto S_{i+1} es, en promedio, la que se detalla a continuación, según el tipo de radio:

Urbano 107,6 %

Semi-urbano 391,3 %

Rural 215,5 %

Por este motivo, dependiendo el radio y su densidad poblacional, la profundización de la búsqueda, una vez hallada una solución, puede llevar a no encontrar ninguna en lugar de mejorar (o igualar) la obtenida. Sin embargo, la experimentación demostró que en los casos en los que la cantidad de segmentos es *manejable*, vale la pena realizar esta generación de segmentos adicional, pues se consigue así disminuir la cantidad de censistas requeridos.

Más precisamente, en el Cuadro 5.3 se muestra el porcentaje de reducción de censistas con respecto a la mínima cantidad hallada antes de profundizar la búsqueda discriminado por zona y por modelo de PLE. En promedio se disminuye la cantidad de censistas en aproximadamente un 1 % con respecto a la solución sobre S_i.

	Urbana	Semi-urbana	Rural
Minimización de censistas	0,37 %	1,53 %	0,97 %
Minimax con $p = \frac{85}{100}$	0,29 %	1,63 %	1,92 %

Cuadro 5.3: Porcentaje de censistas reducidos con respecto a la cantidad mínima de censistas sobre S_i luego de realizar una profundización de búsqueda.

En la Figura 5.2, se muestra un radio con dos soluciones. Primero, en (a), se ejecutó el algoritmo profundizando la búsqueda, en donde el subconjunto de segmentos S_{i+1} tiene un 31 % más de segmentos que S_i. En (a) se muestra la solución hallada resolviendo el modelo de minimización de censistas con S_{i+1} y en (b), se muestra la solución del mismo modelo pero sin profundizar la búsqueda, es decir minimizando censistas sobre el subconjunto S_i. Al profundizar la búsqueda en (a), se encuentra una solución con menor cantidad de censistas, pero se puede observar que en (b) la solución es más satisfactoria que la de (a), pues se toman manzanas completas o casi completas. De todas maneras, la solución dada en (a) es aceptable por el tipo de trazado y a su vez, se tiene un menor número de censistas.

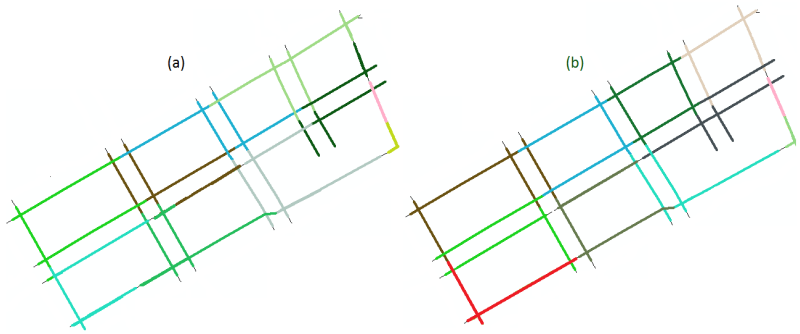

Figura 5.2: Mismo radio resuelto minimizando censistas. En (b) se muestra la solución con 10 cesistas sin profundizar la búsqueda y en (a) son 11 censistas, luego de una iteración más de segmentación.

5.3. Mejoras en la valuación de segmentos

En esta sección se comparan las distintas propuestas de valuación y compacidad de un segmento entre ellas, como también con las definiciones originales de dichas funciones. A lo largo de la experimentación se encontraron muchos ejemplos con resultados satisfactorios, lamentablemente es imposible mostrar estos resultados en forma global pues las mejoras obtenidas son puramente visuales y hay que analizarlas individualmente en cada radio.

5.3.1. Base de la función de valuación

El cambio de base k de la valuación de los segmentos fue escencial, dado que las soluciones que se obtienen con $k = 10$ tienen un rango de valores objetivos que va desde 2 hasta 22 cifras. Mientras que con $k = 3$, los valores objetivos van desde 1 hasta 7 cifras con escasas soluciones de 10 cifras. Con este cambio, se logró evitar por completo los problemas de truncamiento previamente descriptos.

En la Figura 5.3, se muestra un radio resuelto con la función de valuación original (3.2), pero con distinta base k. Notar que la solución con $k = 3$, comparada con la de $k = 10$, tiene segmentos más satisfactorios, pues contienen manzanas completas, caso del segmento verde oscuro, o más lados por manzana, caso del segmento amarillo. También el segmento celeste en el caso de $k = 3$ no cruza la calle, situación que es deseada, no así en $k = 10$. En la solución con $k = 10$, el valor objetivo fue de $1 \cdot 10^9$, mientras que la solución con $k = 3$ obtuvo un valor objetivo de $19738,2$.

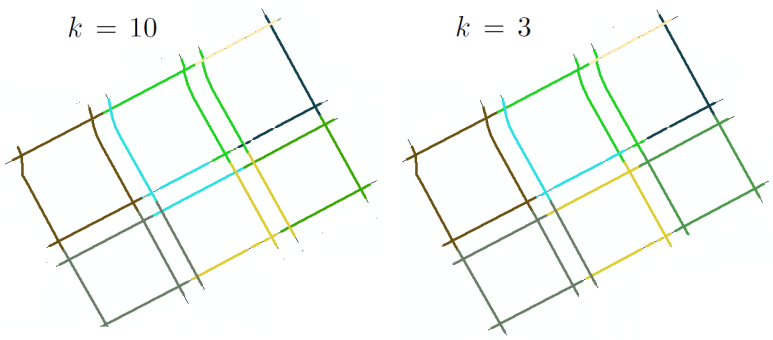

Figura 5.3: Mismo radio resuelto con distinta base k de la función de valuación original (3.2).

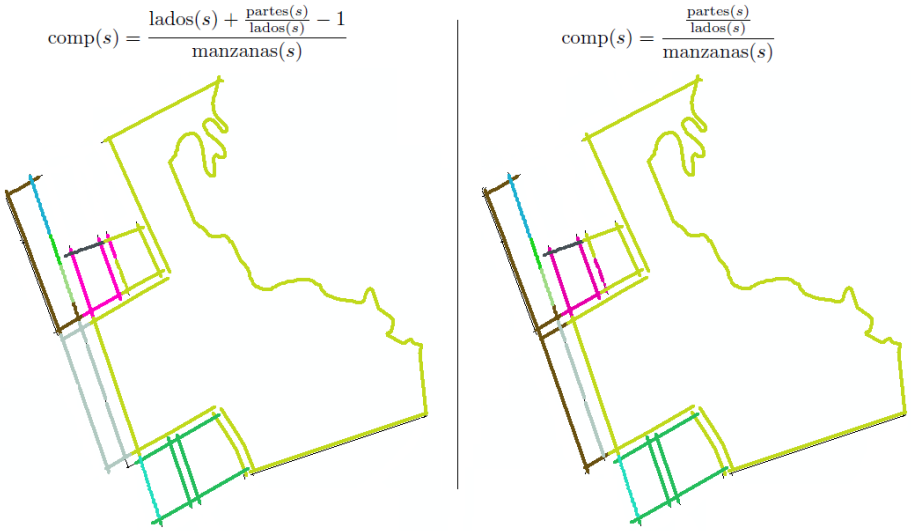

Figura 5.4: Comparación entre definiciones de compacidad, en un radio de baja densidad poblacional.

5.3.2. Definición de compacidad

La función de valuación propuesta en (4.5) resultó más atinada con respecto a la función de valuación original. En la mayor parte de los radios, se encontró la misma solución para ambas, pero en aquellos donde se consiguieron nuevas soluciones se ve reflejada en la precisión de esta nueva definición.

Por ejemplo, en la zona costera de Vicente López, donde no se recurre a partir los lados debido a la baja densidad de población, ocurrió lo descripto en la Sección 4.4.2: la función de compacidad original no refleja la preferencia de obtener mayor cantidad de lados por manzana. Como se ve en la Figura 5.4, el segmento gris, que corresponde a la manzana donde se encuentra la seguridad de la Residencia Presidencial, está poco poblado y no es necesario partir sus lados. Sin embargo, con la definición original de compacidad se obtiene el valor 1 en el segmento de cualquiera de las dos soluciones encontradas, a pesar de que es preferible la segmentación hallada con la definición de compacidad propuesta (4.5), pues el segmento gris contiene a la manzana completa y el marrón no cruza calles. Si bien este ejemplo es un caso particular, en la mayor parte del territorio de la Provincia de Buenos Aires, no es raro encontrar radios con estas características.

A su vez, al utilizar la función de compacidad definida en (4.6) con distintos valores de t, los resultados obtenidos han sido los esperados. En las Figura 5.5 se muestra la cantidad de dígitos de la mediana de los valores óptimos obtenidos

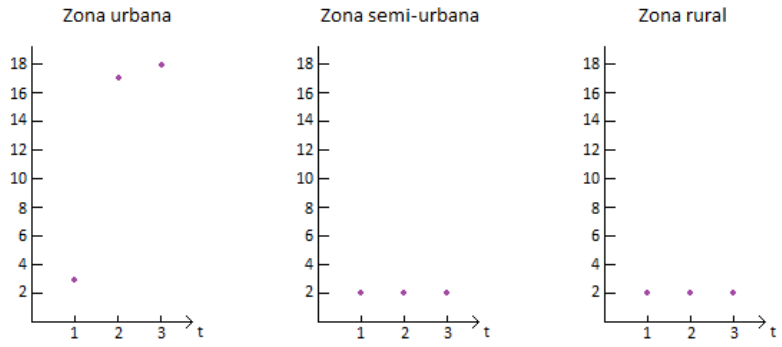

Figura 5.5: Comparación entre distintos valores de t para la función de compacidad (4.6).

para distintos t, calculada por tipo de zona. Se puede observar que tanto en la zona rural como semi-urbana no cambia ese valor ya que no se parten muchos lados y el cociente $\frac{\text{partes}(s)}{\text{lados}(s)} \cong 1$.

En la Figura 5.6 se muestra un caso donde se hallan distintas soluciones. La solución con $t = 2$ exhibe una segmentación más compacta, dado que los segmentos están más concentrados en la manzana en la que se encuentran. Se observó que en este caso, la solución obtenida para $t = 1$ coincide con aquella de maximizar la valuación original.

En la Figura 5.7 se muestra un radio donde las tres definiciones de compacidad dieron soluciones distintas pero similares e igualmente satisfactorias. Es ligeramente superior la solución con $t = 2$ ya que sus segmentos contienen vecinos de nivel de adyacencia 0 y 1. Mientras que las otras dos soluciones llegan al nivel de adyacencia 2.

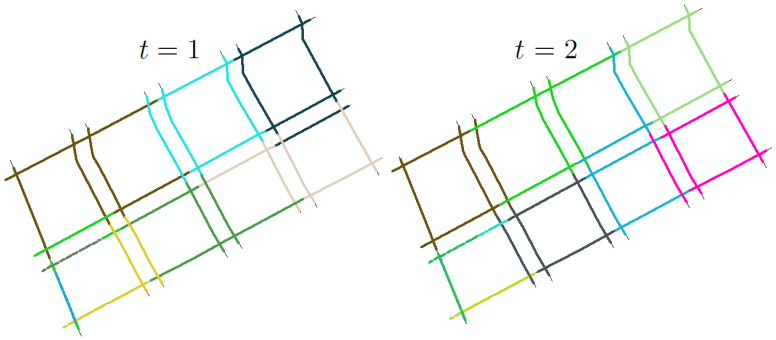

Figura 5.6: Mismo radio resuelto con la función de compacidad (4.6) para distintos valores de t.

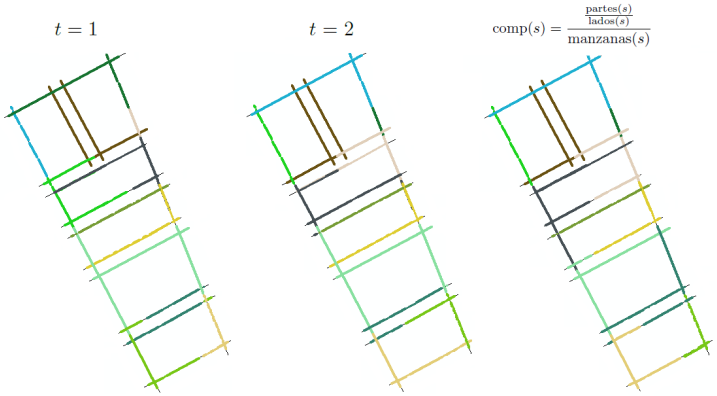

Figura 5.7: Un radio resuelto utilizando las distintas definiciones de compacidad descriptas: (4.5), (4.6) y (3.2).

Capítulo 6

Conclusiones y trabajo futuro

Se trabajó a lo largo de esta tesis con una aplicación de Investigación Operativa del mundo real: la distribución de censistas en la Provincia de Buenos Aires en el Censo Nacional de Viviendas del 2010 de la Argentina. Es interesante notar cómo a partir de una aplicación real surgen problemas teóricos y metodológicos muy interesantes. Los resultados encontrados muestran que se pueden obtener soluciones igual o más satisfactorias que las obtenidas con el modelo original.

Más precisamente, se planteó un modelo para minimizar la cantidad de censistas totales, con la condición de que, de todas las soluciones posibles con esa cantidad, se tome aquella que maximiza la valuación de sus segmentos. Aquí se encontraron soluciones que reducen la cantidad de censistas hasta casi un 6 % sobre la solución propuesta para el modelo implementado en el Censo Nacional 2010.

Una variación interesante es la de minimizar con una cierta cota de "belleza" de los segmentos. La cota se determina a partir del valor objetivo del modelo que maximiza la valuación de los segmentos y acota inferiormente a la valuación de la solución del nuevo modelo. Con una cota exigente del 85 % se arribó a soluciones similares a las de minimizar sin esta cota, aumentando la cantidad de censistas en menos de un 1 %.

Un cambio en el algoritmo implementado en el Censo Nacional 2010 fue el de agrandar el conjunto de segmentos luego de haber hallado una solución con mínima cantidad de censistas y volver a ejecutar el modelo. Este cambio se implementó tanto en el problema estándar de minimización de censistas como en el de minimax, obteniendo resultados igualmente satisfactorios en ambos. Esto logró reducir aún más la cantidad de censistas en un factor de alrededor del 1 %.

Por último, se analizó la función objetivo propuesta en el modelo original a la que se le encontraron distintos conflictos. Por un lado, no reflejaba exactamente lo impuesto por las autoridades para ciertos tipos de segmentos, es decir,

para ejemplos generales de segmentos se obtenían valuaciones cruzadas entre uno aceptado y uno no satisfactorio. Por otro lado, se observó que dado que la valuación es una función exponencial, los valores obtenidos arrastraban problemas numéricos importantes. Para estos inconvenientes, se propusieron distintas funciones de valuación que resultaron atinadas.

Consideramos que sería muy beneficioso que los resultados obtenidos en este trabajo sean conocidos por las autoridades de la Provincia de Buenos Aires y nacionales, de modo de ser tenidos en cuenta para su implementación en próximos censos.

Como posible trabajo futuro queda por resolver estos modelos mediante otros algoritmos para programación matemática, como, por ejemplo, algoritmos de generación de columnas [7] o de tipo *branch and price* (los cuales combinan generación de columnas con *branch and bound*). [4]. También, queda por seguir analizándose las funciones de compacidad y valuación. Éstas se han logrado mejorar notablemente, pero aún se encuentran esporádicamente radios en los que la condición de exponencialidad de la función de valuación hace que se trabaje con números de muy alto orden con otros de pocos dígitos, lo que puede distorsionar la solución final.

A su vez, el concepto de *belleza* de un segmento tiene cierta ambigüedad ya que no está dado por una expresión matemática sino por el tipo de recorrido que dibuja, es decir, como se explicó en este trabajo, se desea que un censista doble lo menos posible en las esquinas, que abarque la menor cantidad de manzanas y/o que tome lados completos. Esto hace que todas las funciones de compacidad propuestas cumplan con lo pedido, algunas más precisas que otras en ciertos ejemplos de radios.

Bibliografía

[1] Aliaga, F., 2010. *Personal communication.* Buenos Aires, Argentina (November 5, 2010).

[2] Altman, M., 1997. *Is Automation the Answer: The Computational Complexity of Automated Redistricting.* Rutgers Computer and Law Technology Journal **23**(1) 81–141.

[3] Altman, M., MacDonald, K., and McDonald, M.P., 2005. *From Crayons to Computers: The Evolution of Computer Use in Redistricting.* Social Science Computer Review **23**(3) 334–346.

[4] Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. and Vance, P. H., 1998. *Branch-and-Price: Column Generation for Solving Huge Integer Programs.* Operations Research **46**(1) 316–329.

[5] Bonomo, F., Delle Donne D., Durán G. and Marenco J, 2011. *Segmentación automática de la Provincia de Buenos Aires para el Censo Nacional Argentino 2010.* Revista Ingeniería de Sistemas **25** 29–45.

[6] Bozkaya, B., Erkut, E., and Laporte, G., 2003. *A tabu search heuristic and adaptive memory procedure for political districting.* European Journal of Operational Research **144**(1) 12–26.

[7] Chvâtal, V., 1983. *Linear Programming.* W.H. Freeman & Company.

[8] Dantzig, G.B, 1963. *Linear Programming and Extensions.* Princeton University Press.

[9] Fleischmann, B. and Paraschis, J.N., 1988. *Solving a large scale districting problem: a case report.* Comput. Oper. Res. **15**(6) 521–533.

[10] Garey, M.R. and Johnson D.S., 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W.H.Freeman & Co Ltd.

[11] Garfinkel, R.S. and Nemhauser, G.L., 1970. *Optimal Political Districting by Implicit Enumeration Techniques.* Management Science **16**(8) B495–B508.

[12] Grötschel, M., Lovász, L. and Schrijver, A., 1993. *Geometric algorithms and combinatorial optimization.* 2 edition, Springer/Sci-Tech/Trade.

[13] Helbig, R.E., Orr, P.K., and Roediger, R.R., 1972. *Political redistricting by computer*. Comm. ACM **15**(8) 735–741.

[14] Hess, S.W., Weaver, J.B., Siegfeldt, H.J., Whelan, J. N., and Zitlau, P.A., 1965. *Nonpartisan Political Redistricting by Computer*. Operations Research **13**(6) 998–1006.

[15] IBM ILOG, 2009. *User's Manual for CPLEX*.

[16] INDEC. `http://www.indec.gov.ar/`

[17] Karmarkar, N., 1984. *A New Polynomial-Time Algorithm for Linear Programming*. Combinatorica **4**(4) 373–395.

[18] Khachiyan, L., 1979. *A polynomial algorithm in linear programming*. Soviet Math. Doklady **20** 191-194.

[19] La voz de Tandil, 2010. *Se censó más del 95 % de las viviendas en la provincia*. Retrieved November 15^{th}, 2010. `http://www.lavozdetandil.com.ar/ampliar_nota.php?id_n=20090`.

[20] Nemhauser G.L., Wolsey, L.A., 1988. *Integer and Combonatorial Optimization*. Wiley Interscience, John Wiley and Sons.

[21] Secretaría de Comunicación Pública, Presidencia de la Nación, 2010. `http://www.prensa.argentina.ar/2010/09/12/11742-avanza-la-preparacion-de-los-censistas.php`. Sala de Prensa.