Automatic Dwelling Segmentation of the Buenos Aires Province for the 2010 Argentinian Census

Flavia Bonomo  
Dto. de Computación, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina; and  
IMAS–CONICET, Buenos Aires, Argentina, fbonomo@dc.uba.ar

Diego Delle Donne  
Dto. de Computación, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina; and  
Instituto de Ciencias, Universidad Nacional de General Sarmiento, Buenos Aires, Argentina, ddelledo@ungs.edu.ar

Guillermo Durán  
Dto. de Matemática, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina;  
Dto. de Ingeniería Industrial, FCEyN, Universidad de Chile, Santiago de Chile, Chile;  
Instituto de Cálculo, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina; and  
CONICET, Buenos Aires, Argentina, gduran@dm.uba.ar

Javier Marenco  
Dto. de Computación, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina; and  
Instituto de Ciencias, Universidad Nacional de General Sarmiento, Buenos Aires, Argentina, jmarenco@ungs.edu.ar

In planning for a population census, determining which dwellings within a census tract each enumerator must visit is a logistical challenge. This challenge, which we call the dwelling segmentation problem, generally includes a set of constraints on the enumerators’ assigned routes and various criteria regarding the homogeneity and uniformity of the segmentation solutions. In this paper, we present a computational approach to solve this problem. We successfully applied our solution, which is a substantial improvement over manual methods, to the Province of Buenos Aires in the 2010 Argentinian census.

Key words: population census; segmentation.

History: This paper was refereed. Published online in Articles in Advance May 24, 2013.

A population census is a procedure for acquiring demographic information on the population within a geographical area, generally an entire country. The information gathered typically includes data on housing, educational, and employment characteristics of the area’s inhabitants. In Argentina, census enumerators visit each dwelling in the area covered and usually complete a census within one day.

To ensure the smooth and successful completion of such a major logistical operation within a few hours, census planning usually begins several years in advance. Determining which dwelling units each enumerator must visit is crucial. This task, which we refer to as the dwelling segmentation problem, consists of partitioning a set of dwellings into subsets that satisfy specific constraints. This problem generalizes the set-partitioning problem; hence, it is NP-complete.

As with the redistribution of electoral divisions, a relatively high partition level gives rise to the redistricting problem. Considerable literature addresses this issue (Altman 1997, Altman et al. 2005, Bozkaya et al. 2003, Fleischmann and Paraschis 1988, Garfinkel and Nemhauser 1970, Helbig et al. 1972, Hess et al. 1965), and various software packages are available to perform redistricting, either automatically or semi-automatically. For example, Altman and McDonald (2009) present an open-source tool implemented in the R statistical environment. However, because the constraints in electoral redistricting generally differ from those imposed by the dwelling segmentation problem,
the latter requires specifically designed algorithms. Dwelling segmentation also involves homogeneity and uniformity criteria that a solution must satisfy, as we describe in the Problem Description section. Satisfying the criteria was almost impossible using previous manual solution methods, because they required human operators who construct segmentations in a greedy fashion, without any prespecified rules or algorithmic instructions. Using such a scheme, an operator might find a good segmentation, but could fail if his or her first decision resulted in a poor final solution.

Our study analyzes the dwelling segmentation problem in the Province of Buenos Aires in the 2010 Argentinian census, and reports on our development of a computational approach for solving it, which we applied during the census planning process. Based on its experience in the previous census, Dirección Provincial de Estadística, the province’s statistical authority, decided to use operations research (OR) techniques to address its dwelling segmentation problem, and contacted the University of Buenos Aires to explore a collaboration. The study we discuss in this paper is the result of this collaboration. We are not aware of any other studies in the OR literature that have investigated dwelling segmentation in a census context.

We organized the remainder of this paper as follows. In Problem Description, we review the relevant characteristics of the 2010 census and discuss the dwelling segmentation problem in this context. In The Segmentation Algorithm, we propose a solution algorithm and discuss the main difficulties we encountered during its implementation. In Computational Results, we describe the results we obtained by using this tool. Conclusions contains some final remarks. In the appendix, we describe the algorithm and its parameters in detail.

Problem Description

Argentina has 23 provinces and the autonomous city of Buenos Aires (formerly known as the Federal Capital), which enjoys special administrative status. The Province of Buenos Aires is Argentina’s largest geographical province (304,907 square kilometers) and the most populated, with approximately 15.6 million inhabitants, 39 percent of the country’s population.

It excludes the autonomous city of Buenos Aires but includes Greater Buenos Aires—the outer belt of suburbs surrounding the autonomous city—which has about nine million people. Together the two areas form a single large population nucleus. With the exception of Greater Buenos Aires, the province is predominantly rural and agricultural, although some areas depend primarily on tourism (the Atlantic coast), mining (the south), or steelmaking (the northeast along the Parana River). By contrast, Greater Buenos Aires is primarily urban and industrial.

Each province is subdivided into “partidos” or “departamentos.” Buenos Aires province includes 134 of these divisions; for census purposes, they are subdivided into 19,577 “radios censales” or census tracts. Each census tract contains about 300 dwellings distributed across a set of contiguous blocks that can number from 1 to 50, depending on the population density. The focus of our study is the 16,216 census tracts that are located in urban areas. The remaining census tracts are divided into 2,886 rural tracts and 475 intermediate tracts. The rural and intermediate tracts are sparsely populated and far apart geographically (the census takers cover these tracts by car); in addition, the corresponding maps are not completely updated. The geographical information systems team from the Dirección Provincial de Estadística believed that an automated solution was not suitable for these tracts, because it would require substantial manual preprocessing. Therefore, the rural and intermediate tracts are not part of this study.

For any given census tract, the segmentation problem involves partitioning the constituent blocks into disjoint sets of dwellings, which we call segments, each of which is assigned to a single census enumerator. As we will see in this study, the sides of a block may under certain conditions be split across different segments. The blocks in Argentinian urban centers typically have four sides, although other block morphologies are found occasionally.

A principal requirement for the feasibility of a segment is that its various blocks or block sides must be contiguous. According to criteria set by the Dirección Provincial de Estadística, each block or side must directly face another block or side, or (in the case of a side only) must continue directly on from a side in an immediately adjacent block. Figure 1 illustrates...
Figure 1: Feasible segments must contain contiguous block sides and cannot cross diagonally from one block to another.

Figure 2: After enumerating block b, the enumerator should give preference to the block sides marked with a 0, then the block sides marked with a 1, and finally the block sides marked with a 2.
side of a block (e.g., side b in Figure 2), the enumerator, rather than crossing a street to continue, should either proceed from that corner along the connecting side of the same block or cross over to the side of the directly-facing adjacent block. These options are marked with a 0 in Figure 2. If this is impossible, the enumerator should continue in the same direction (i.e., on the same side of the street) to the next block (one of the sides marked 1 in Figure 2). If this is also impossible, the enumerator should proceed to one of the sides marked with a 2 in the adjacent blocks. Thus, the numbers 0, 1, and 2 define the levels of adjacency, which in turn determine the order of the preferences described previously.

Additionally, the following compactness preferences, in order of importance, should be considered in a segmentation:

1. Preference should be given to segments consisting only of whole blocks.
2. Segments should contain complete sides and be as compact as possible. These criteria are defined informally and refer to the breadth of a segment across the various blocks it contains (see Figure 3). For example, a segment consisting only of whole blocks is considered to be compact; however, a segment that has various blocks in which at least one but not all of whose sides belong to another segment is not considered compact. We try to define segments that are as compact as possible because they tend to minimize the number of enumerator route errors.
3. If the segmentation cannot be completed using entire block sides, some block sides can be divided between two or more segments. This situation may occur if the block side (1) contains a number of dwellings that exceed the maximum allowed per segment, or (2) contains fewer than the corresponding lower bound, but the combination with any adjacent side exceeds the upper bound, thus violating the feasibility constraints. Typically, this occurs when a block has one or more apartment building; in this case, preference should be given to segments that include all the apartments in any single apartment building, so that one enumerator can cover the building.
4. If one enumerator cannot take the census in all the apartments in a single building (e.g., if the building contains more dwellings than the maximum number of dwellings per segment), preference should be given to segments that do not split up the apartments on a single floor. That is, a single enumerator should cover each floor, if possible. Please note that if an apartment spans more than one floor, its address typically specifies a single floor; the apartment is considered as a single dwelling for census purposes.

The Segmentation Algorithm

To solve the dwelling segmentation problem in each census tract, we developed an algorithm that tries to find a feasible segmentation (see the appendix).

The algorithm iteratively considers sets of candidate segments. At each iteration, it generates a set of feasible segments and tries to find a complete segmentation involving segments from this set. Therefore, we solve an integer programming (IP) model that takes the set of candidate segments as input, and seeks a feasible segmentation that maximizes the global compactness. If such a segmentation exists (i.e., if the IP model is feasible), then the algorithm stops and returns the segmentation that the model solution found. However, if the model is infeasible, the algorithm moves on to the next iteration, in which a new (usually larger) set of candidate segments is generated and the procedure is repeated.

The first iterations consider feasible segments whose block sides are joined by the adjacencies marked with 0 in Figure 2. At the i-th iteration, all feasible segments involving these adjacencies and at most i blocks are generated, and the model takes such a set of segments as input. If all these iterations fail to find a feasible
segmentation, the adjacencies marked with a 1 in Figure 2 are also considered and the process is repeated. If this procedure fails again, all feasible adjacencies from Figure 2 are considered when generating the set of candidate segments and the process is repeated.

If the previous procedure does not find a feasible solution, we divide the block sides into parts. We introduce a parameter that sets the maximum number of dwellings a block side can have, and we divide all sides that exceed this number into two or more parts. The block-side division algorithm greedily generates two or more parts from each block side; each part has no more than the number of dwellings specified by the parameter. Once the block sides have been divided, we repeat the previous procedure, which then takes the generated parts as input instead of the original block sides. If this procedure fails to find a complete segmentation, we decrease this parameter, such that each (divided) block side has a smaller number of dwellings, and repeat the aforementioned procedure.

In the preliminary testing and adjustment stages, this algorithm solved the problem for a large number of the highly urban census tracts; however, it had difficulty segmenting tracts with relatively low population densities. For example, the algorithm did not find a feasible solution for the tract in the city of Olavarria (see Figure 4). The major obstacle in these cases is the presence of many blocks with few, if any, dwellings (i.e., the algorithm must execute a large number of iterations to generate enough feasible segments to cover the whole tract). Furthermore, with few dwellings per block, the number of feasible segments is very high. For the tract in Figure 4, the candidate set after seven iterations has more than 100,000 such segments, resulting in excessively high segment-generation and solutions times. To address this situation, we (1) treat blocks with few dwellings as a single segment, and (2) arbitrarily combine side blocks with few dwellings. We describe the details and related parameters in the section Improvements for Low-Density Census Tracts in the appendix.

Computational Results

Given the number of parameters and their significance for the algorithm’s execution time, we conducted preliminary tests using various census tracts to identify the best set of parameter values for different tract types. Based on the results, we grouped tracts by population density into three categories: high density (up to 10 blocks), medium density (11 to 30 blocks), and low density (more than 30 blocks) tracts.

We coded the algorithms in C++ and solved the IP models using CPLEX. We took the data on blocks, sides, and dwellings from the geographical database maintained by the Province of Buenos Aires. We implemented the necessary interfaces using a
geographic information system to export the data and import and view the segmentations that our algorithm generated.

Execution times for generating the segments were relatively short; the worst case was about two minutes using the parameters specified in the appendix. More than 99 percent of the IP models for the various tracts were solved in a few seconds, implying that the linear relaxation is very tight and the first feasible solution that CPLEX found was usually optimal. In a few cases, the execution time reached the imposed time limit; the appendix shows the actual value of this parameter, depending on the tract density.

The largest number of segments for each solved census tract (i.e., the number of feasible segments in the last IP model in the execution of the segmentation algorithm) is on average 1,305 segments for high-density tracts (with a standard deviation (SD) of 4,424). For medium-density tracts, the average number of segments in the largest IP model is 3,826 (SD = 8,709); for low-density tracts, the average number of segments is 10,314 (SD = 11,265).

As a comparison, the segmentation of the province for the previous (2001) census, which was done manually, required 25 employees working full time for 30 days (i.e., 6,000 person-hours). In this paper we address urban tracts; these tracts correspond to 80 percent of the segmentation time for the 2001 census. For the 2010 census, automated software tools were used for the first time to determine the dwelling segmentation. Our algorithm delivered satisfactory results for 96 percent of the urban census tracts in about 320 CPU hours running on a computer with a 2.4 GHz Intel Celeron processor with 2 GB of RAM (the equivalent of less than one day on a cluster of 15 computers).

For approximately 600 census tracts (four percent of the urban tracts), the algorithm did not find a feasible solution. In these cases, we solved the segmentation problem manually or by using the tool with slightly relaxed constraints. As an example, consider the medium-density tract in Figure 5. The preprocessing stage combined the blocks with few dwellings to form indivisible groups, as we describe previously. The contiguous blocks with 14, 13, and 14 dwellings, respectively (see the bottom left of Figure 5), are surrounded by very sparsely populated blocks, which we combined into a single group and that isolate the aforementioned three blocks. These blocks include 41 dwellings that, because of the upper bound on the number of dwellings per segment, could not be put into a single segment; and because of the lower bound on the number of dwellings per segment, two segments could not be constructed from them. We solved this tract in a few seconds by relaxing the upper bound by one unit.

Many of the census tracts that the algorithm could not solve had characteristics similar to those in Figure 5. That is, they were usually sparsely populated but had denser areas along their boundaries (i.e., they usually bordered more heavily populated tracts). When faced with these results, the planning staff at the census authority asserted that the boundaries of these tracts were poorly drawn, because they include areas of widely varying densities that should be avoided. This was the result of population growth.

A comparison with available data from the 2001 census shows that the manually generated segmentations frequently violated the lower and upper bounds for the number of dwellings assigned to each census enumerator—a situation that did not hold for any census tract that our algorithm solved in the 2010 census (i.e., 96 percent of the census tracts). Furthermore, segments in software-generated segmentations tend to be...
Figure 6: Segmentations generated by the algorithm we discuss in this paper satisfy all constraints and tend to have more compact segments than manually generated segmentations.

more compact than those in manual segmentations. For example, in Figure 6, the graphic on top shows the manual segmentation for a census tract in Azul City for the 2001 census; the graphic on the bottom shows the segmentation we obtained using the same input data. The software-generated segmentation contains more complete blocks and satisfies all the constraints. This is a consequence of the greedy nature of the method that the manual operators used; this method tends to complete the current segment by including adjacent block sides without revising its previous decisions. Therefore, the lower and upper bounds on the number of dwellings per segment greatly constrain the operator’s freedom to choose compact segments, although they were frequently violated—a situation that the procedure we propose in this paper addresses.

Conclusions

In this paper, we present a computational tool that we used in the 2010 Argentinian census to address the dwelling segmentation problem for the Province of Buenos Aires. We completed the implementation in time to meet the two-month deadline that the census planning process requires. In the previous census, operators were not given any rules or algorithmic instructions, thus allowing different operators to generate different segmentations for the same census tract. In contrast, our algorithm’s automated procedure generated uniform results for the entire province, ensuring that similar workloads could be assigned to all census enumerators. Finally, processing times were also considerably less than those of the manual method.

The algorithm’s performance proved to be sensitive to the parameter values we selected. Using the values we show in the appendix, the algorithm arrived at solutions in a matter of seconds; however, poorly chosen values risked diminishing the feasibility of the model or led to the generation of hundreds of thousands of segments, thus extending execution times to several hours. The categorization of the census tracts by population density greatly facilitated the identification of suitable parameter levels. Incorporating a sequential procedure in the proposed algorithm was fundamental to guaranteeing that the preferences regarding desirable segment characteristics were also met.

Despite these highly satisfactory results, we note that the algorithm we implemented provides a heuristic solution. Therefore, exploring an integrated IP model that addresses all the problem characteristics is an interesting task for future research. A natural generalization of the model in the appendix will contain a huge number of variables; hence a column-generation approach (Barnhart et al. 1998) should be considered. The major challenge in developing this approach will be solving the column-generation subproblem.

From a theoretical point of view, exploring the computational complexity of the segmentation problem for particular classes of instances would be interesting. If the streets may define an arbitrary graph, then
the problem is NP-complete; however, to the best of our knowledge, its computational complexity for more regular morphologies is open.

Because the census tracts contained about 300 dwelling units and each enumerator had to be assigned a number of dwellings within a prespecified range, the number of enumerators per tract varied little (approximately eight to ten). Therefore, we did not pursue the objective of minimizing this number; doing so would have been of little value. Fernández Slezak (2012) explored this issue and showed that by minimizing the number of enumerators, only a two percent improvement could have been attained for high-density tracts, and a five percent improvement could have been attained for low-density tracts.

Fernando Aliaga, head of geographical information systems for the 2010 census in the Province of Buenos Aires, stated that the “use of this computer tool allowed us to produce a homogeneous segmentation with uniform compactness criteria, unlike the manual segmentation method, which depends in large measure on operator decisions” (F. Aliaga, pers. comm.). The provincial authorities pronounced the census, conducted on October 27, 2010, an organizational success (La Voz de Tandil 2010).

Appendix. The Segmentation Algorithm

In this appendix, we provide the full details of the algorithm for the dwelling segmentation problem in an individual census tract. We consider a segment as exceeded if it has more dwellings than the maximum number of dwellings allowed per segment or is longer than the upper bound \( L \). The algorithm uses nonexceeded segments; however, these may (or may not) be feasible depending on whether they satisfy the minimum number of dwellings per segment. When trying to construct a feasible solution, we consider only feasible segments. By defining adjacency type \( \delta \in \{0, 1, 2\} \) (see Figure 2), a segment is said to be \( \delta \)-connected if it is connected by adjacencies of levels up to \( \delta \).

We set out the proposed segmentation procedure in Algorithm 1. The procedure uses geographical data on the census tract as inputs, and the value of \( \delta \in \{0, 1, 2\} \) as a parameter. Denote as \( S_i \) the set of nonexceeded (although not necessarily feasible) segments that have no more than \( i \) blocks, \( i \geq 1 \). At the \( i \)-th iteration, Algorithm 1 tries to find a feasible segmentation using the subset \( S_i \) of feasible segments from \( S_i \). We solve an IP model that tries to maximize the global compactness, as described in the following section. The algorithm iterates until (1) it finds a feasible solution or \( S_i \) does not vary with respect to \( S_{i+1} \), or (2) it reaches a previously specified iteration limit \( MI \). When it finds a feasible solution for the model for the given subset \( S_i \) of segments, the algorithm returns the solution obtained and terminates.

Algorithm 1. (Segmentation algorithm for \( \delta \)-connected segments)
1. \( S_b \leftarrow \{\} \) // base set
2. for each block \( b \) do
3. \( S_b \leftarrow S_b \cup \{\} \) 
4. end for
5. \( i \leftarrow 1 \)
6. \( S_i \leftarrow S_b \)
7. repeat
8. Run the integer linear programming (ILP) segmentation model with all segments from \( S_i \) satisfying the constraint that imposes a minimum number of dwellings per segment (i.e., feasible segments)
9. if solution found then
10. End (with solution)
11. end if
12. \( S_{i+1} \leftarrow S_i \)
13. for each \( (s_i, s_b) \in S_i \times S_b \) do
14. if \( s_i \cup s_b \) is a nonexceeded \( \delta \)-connected segment then
15. \( S_{i+1} \leftarrow S_{i+1} \cup \{s_i \cup s_b\} \)
16. end if
17. end for
18. if \( S_{i+1} = S_i \) then
19. End (without solution)
20. end if
21. \( i \leftarrow i + 1 \)
22. until \( i > MI \)
23. End (without solution).

We now describe the construction of the set \( S_i \) of nonexceeded feasible segments that have no more than \( i \) blocks. In lines 2–4 of Algorithm 1, the algorithm generates a base set of segments \( S_b \) using complete block sides. For each block, all possible nonexceeded segments \( S_b \) contained within it are generated. If the IP procedure cannot find a solution using the feasible segments from \( S_b \) (i.e., the segments from \( S_i \) that satisfy the lower bound on the number of dwellings per segment), then in lines 12–17 we add new segments to the current set constructing \( S_{i+1} \). In these lines, each existing segment \( s_i \in S_i \) is connected with each base segment \( s_b \in S_b \) from a neighboring block, such that the resulting segment \( s_i \cup s_b \) is a nonexceeded \( \delta \)-connected segment. Following line 17, the set \( S_{i+1} \) contains all nonexceeded \( \delta \)-connected segments that involve at most \( i + 1 \) blocks.

Because no formal objective function is attached to the dwelling segmentation problem, the solution that Algorithm 1 provides may not be a good solution. However,
the sequential procedure readily incorporates the compactness preferences of the selected segments, thus increasing the likelihood that the preferred solution will be found first.

To find solutions with the most compact segments possible, we sequentially run Algorithm 1 for \( \delta = 0, 1, \) and 2, and we interrupt the procedure when we find the first feasible solution.

If the preceding process does not find a feasible solution, we divide the block sides into parts. In this setting, the parameter \( P \) specifies the maximum number of dwellings a block side can have, with all sides that exceed this number divided into two or more parts. Each part can have at most \( P \) dwellings. The block-side division algorithm uses a greedy strategy, and attempts to keep all apartments in a single building together in the same part. Starting from one of the corners of the block side to be divided, this procedure constructs a part by advancing along that side until either an apartment building is encountered or \( P \) dwellings have been counted. In either case, construction of the block part is completed and a new one is started; the procedure is repeated until the algorithm obtains an entire set of parts for the divided side. The division algorithm also considers apartments within buildings by greedily trying to keep together apartments in the same floor. Finally, we specify new adjacencies for the divided sides (see Figure A.1).

Note that the smaller \( P \) is, the greater both the number of block-side parts in the tract and the size of the base set \( S_P \).

Once the block sides have been divided as described previously, we execute Algorithm 1 again for \( \delta = 0, 1, \) and 2 to find a feasible solution. If we find no solution, we lower the value of \( P \) and repeat the process. We explain the entire procedure in Algorithm 2. We enter into the algorithm a list \( PL \) containing all (nonnegative integer) values to be considered for \( P \), and the algorithm successively uses these values in order of decreasing size, given that the greater the \( P \), the less divided the solutions will be. If upon completion the procedure has not found a solution, it terminates and informs the user that no segmentation could be identified. Note that \( P \) is initially set at less than or equal to the maximum allowed number of dwellings per segment; however, the parameter value will depend on the population density of the census tract to be segmented.

Algorithm 2 (Segmentation algorithm)

1. For each \( P \in PL \) do
2. Divide sides with number of dwellings greater than \( P \).
3. for \( \delta = 0 \) to 2 do
4. Run Algorithm 1 for \( \delta \)
5. if Algorithm 1 found a solution then
6. Return solution and end
7. end (if)
8. end (for)
9. end (for)
10. End (without solution).

Improvements for Low-Density Census Tracts

As we state in the Segmentation Algorithm section, the algorithm must be modified to successfully address low-density tracts. We can modify Algorithm 1 as follows:

- We treat blocks with few dwellings as a single nonexceeded segment for purposes of adding segments to the base set \( S_P \) (line 3 in Algorithm 1). We add a parameter \( MP \) that sets the minimum number of dwellings a block must have before it can generate more than one base segment. If a block falls short of this number, we add only one segment for the complete block to \( S_P \). This considerably reduces the size of \( S_P \) for low-density tracts.
- We generate the base segments \( S_P \); if any base segment has few dwellings, we arbitrarily combine it with an adjacent segment to form a single base segment. To do this, we add a parameter \( MH \) to set the minimum number of dwellings a segment must have to be added to \( S_P \). We insert this treatment of \( S_P \) in Algorithm 1 following line 4.

The values of the various parameters for low-density tracts must be identified carefully, because inappropriate choices may increase execution times (e.g., if \( P \) is very low) or reduce the model’s feasibility (e.g., if values \( MP \) and \( MH \) are too high). Table A.1 gives the parameter values for each category. Recall that the categories are given by population density: high-density tracts have up to 10 blocks, medium-density tracts have 11–30 blocks, and low-density tracts have more than 30 blocks.

The Integer Programming Model

We now formulate an ILP model for selecting a set of segments that covers all the dwellings within a census tract. Let \( R \) be the census tract to be segmented and let \( S \) be the (input) set of feasible segments to be considered. For each segment \( s \in S \), introduce the binary variable \( x_s \), such that \( x_s = 1 \) if and only if segment \( s \) is included in the solution.

To maximize the compactness of the segments selected by the model, we specify the following objective function. Given a segment \( s \in S \), we define its compactness to be \( \text{comp}(s) = \text{sides}(s)/\text{blocks}(s) \), where \( \text{sides}(s) \) and \( \text{blocks}(s) \) are the number of sides and blocks, respectively, in segment \( s \). Thus, a segment consisting of a single complete rectangular block will have a compactness of 4, and a segment...
the valuation of a segment the sense of desirability and avoid undesirable ones, define the most desirable. To prioritize segments that are compact in or four segments, each consisting of a quarter block (a single side). Clearly, the first of these three possibilities is the most desirable. To prioritize segments that are compact in the sense of desirability and avoid undesirable ones, define the valuation of a segment \( s \) as \( \text{val}(s) = k^{\text{comp}(s)} \). Any value \( k \geq 3 \) is a reasonable choice for \( k \). In this particular application, we selected \( k = 10 \), a value that provided good results; however, Fernández Slezak (2012) shows a posteriori that \( k = 3 \) would have given better results for some census tracts.

Now let \( V \) be the set of dwellings in \( B \), and let \( L_0 \) be the set of block sides without dwellings in \( B \). For each \( v \in V \), denote as \( S_v \subseteq S \) the set of feasible segments that includes dwelling \( v \), and for each \( l \in L_0 \), denote as \( L_l \subseteq S \) the set of feasible segments that includes side \( l \). With these definitions, we can formulate a simple ILP model for the segmentation problem:

\[
\begin{align*}
\text{max} & \quad \{ \sum_{s \in S} \text{val}(s) \cdot x_s \} \\
\text{s.t.} & \quad \sum_{s \in S_v} x_s = 1 \quad \forall \, v \in V, \quad (1) \\
& \quad \sum_{s \in S_l} x_s = 1 \quad \forall \, l \in L_0, \quad (2) \\
& \quad x_s \in \{0,1\} \quad \forall \, s \in S. \quad (3)
\end{align*}
\]

Constraints (1) ensure that each dwelling is covered by exactly one segment, and constraints (2) guarantee that block sides with no dwellings are also covered by exactly one segment, because sides with no dwellings must be visited by an enumerator. Recall that the model includes no preferences regarding adjacency levels for segments that cross streets (i.e., extend beyond a single block), which are addressed within the global procedure.

Acknowledgments
The authors thank the Dirección de Estadística from the Province of Buenos Aires for taking the initiative to explore the concept of dwelling segmentation with automated techniques. The authors also thank Fernando Aliaga, head of geographical information systems, for the 2010 census in the Province of Buenos Aires, and to his team for their collaboration in key aspects of this study. The authors gratefully acknowledge Rodrigo Sotelo, Andrés Weintraub, Kenneth Rivkin, and Jaime Miranda for their insightful remarks on this paper. Finally, the authors thank the anonymous reviewers, the associate editor, and the editor-in-chief for their constructive comments, which greatly helped to improve this work. This project was partially funded by the Ministerio de Economía de la Provincia de Buenos Aires. Partial funding was also provided by ANPCyT [PICT-2007-00518], CONICET [PIP 112-200901-00178], and UBACyT [2002009030094 and 20020100100980] (Argentina), and by FONDECyT [1110797] and the Millennium Science Institute Complex Engineering Systems in Chile.

References

Aliaga F (2010) Province of Buenos Aires 2010 census information provided via personal communication with the authors, November 5.

Altman M (1997) Is automation the answer: The computational complexity of automated redistricting. Rutgers Comput. Law Tech. J. 23(1):81–141.

Altman M, McDonald MP (2009) Bard: Better automated redistricting. J. Statist. Software 42(4):1–28.

Altman M, MacDonald K, McDonald MP (2005) From crayons to computers: The evolution of computer use in redistricting. Social Sci. Comput. Rev. 23(3):334–346.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. Oper. Res. 46(3):316–329.

Bozkaya B, Erkut E, Laporte G (2003) A tabu search heuristic and adaptive memory procedure for political districting. Eur. J. Oper. Res. 144(1):12–26.

Fernández Slezak F (2012) Programación matemática para distribución eficiente de censistas en censos nacionales: El censo 2010 de la Provincia de Buenos Aires como caso de estudio (in Spanish). Master’s thesis, University of Buenos Aires, Buenos Aires.

| Parameter | High-density | Medium-density | Low-density |
|-----------|---------------|----------------|-------------|
| MI        | 4             | 7              | 9           |
| MP        | 1             | 2              | 10          |
| MH        | 0             | 1              | 5           |
| PL        | [32, 16, 10]  | [32, 16]       | [40, 32, 20]|
| MT        | 60            | 60             | 120         |

Table A.1: The table lists algorithm parameter values by type of census tract.
Fleischmann B, Paraschis JN (1988) Solving a large scale districting problem: A case report. Comput. Oper. Res. 15(6):521–533.

Garfinkel RS, Nemhauser GL (1970) Optimal political districting by implicit enumeration techniques. Management Sci. 16(8): B495–B508.

Helbig RE, Orr PK, Rodiger RR (1972) Political redistricting by computer. Comm. ACM 15(8):735–741.

Hess SW, Weaver JB, Siegfeldt HJ, Whelan JP, Zitlau PA (1965) Nonpartisan political redistricting by computer. Oper. Res. 13(6):998–1006.

La voz de Tandil (2010) Se censó más del 95% de las viviendas en la provincia. Accessed November 15, 2010, http://www.lavozdetandil.com.ar/ampliar_nota.php?id_n=20090&com=1.

Verification Letter
Rodrigo Sotelo, Planning Director of the Dirección Provincial de Estadística, Torre Gubernamental II, Calle 12 esq. 53–Piso 10. CP 1900–La Plata–Bs As: Writes:

"I am writing in regard to the impact of work performed for the DPE, the statistical agency of the Province of Buenos Aires, by the Graph Theory and Optimization Research Group, a unit within the Department of Mathematics and Computer Science at the University of Buenos Aires (UBA). The Group’s work involved the automation of certain pre-census survey activities undertaken by the DPE in advance of the 2010 Argentinean Census.

"Let me point out first of all that the DPE, whose origins go back to the Statistical Registry created in 1821, is part of Argentina’s national official statistics system. Its duties as defined under both national and provincial legislation (National Law No. 17.622 and Provincial Decree 206/07) include representing the province in dealings with INDEC, the national statistical agency, and coordinating all census-related operations concerning the province within its territory.

"Although any census is a highly complex undertaking, this is particularly true in the case of the Province of Buenos Aires given that it is Argentina’s largest territorial division in both area and population. Covering 304,907 km², it was home in 2010 to 15,625,084 of 40,117,096 inhabitants, or almost 39% of the country’s total population (INDEC 2011).

"The census design considerations established by INDEC for the 2010 census, coupled with the particular characteristics of the province and the tight deadlines for carrying out all the many necessary activities to ensure an optimal preparation for census day, posed a serious challenge for the DPE. Officials at the provincial agency thus found themselves obligated to take on responsibilities going beyond the routine coordination and execution of the tasks initially set for it. This led to the decision to define specific methodologies that would address realities peculiar to the province.

"Some idea of the scale of the job may be had simply by considering that in the 2001 census, more than 178,000 workers had to be recruited in the province to carry out the many pre-census, census and post-census tasks. Preliminary calculations based on this figure suggested that for the 2010 census day alone, the number of people required would be about 220,000, consisting of 185,000 census enumerators, 25,000 heads of census tracts, 2,500 heads of census subdivisions and 134 heads of census divisions. All of these personnel would have to be trained and supplied with the necessary materials within a period of eight months.

"In light of the foregoing, the DPE decided to link its staff teams with those of other provincial and local governments as well as entities in the country’s information industry and academic institutions.

"One of the methodological decisions emerging from this process was to automate all cartographical procedures. This included the segmentation of the territory in such a way that the census-taking workload would be distributed as equally as possible across all enumerators. The DPE’s view that this segmentation could be accomplished only by automation ran counter to the widespread conviction in statistical circles that it was an impossible task, a belief partly based on the failure of previous segmentation efforts by the agency in preparation for the 2001 census. As it turned out, the research initiated by the DPE for the 2010 version demonstrated that the segmentation problem could in fact be solved.

"To carry out this research the DPE partnered with the Graph Theory and Optimization Research Group at UBA, which was charged with developing an automatic segmentation algorithm. The algorithm that resulted was integrated into the agency’s geographical information system early enough in its workflow schedule to ensure all work teams were able to perform their activities in full coordination and at maximum capacity.

"The application of the segmentation algorithm enabled 96 percent of the urban and mixed census tracts in the province to be constructed automatically in approximately 320 processing hours (less than a single day with a cluster of 15 PC’s).

"The impact of the Group’s work can be appreciated quantitatively in terms of a comparison with previous experience. In the 2001 census, for example, the DPE needed 6,000 person-hours (the equivalent of 25 persons working full time for 25 days) to perform the segmentation task manually, excluding the times involved in training and support provision. Note also that this comparison takes no account of the risk of assignment errors inherent in a manual process."

Flavia Bonomo holds a PhD in computer science and a degree in mathematics from the University of Buenos Aires. She is an assistant professor in the Computer Science Department of the School of Sciences at the University of Buenos Aires and a researcher at IMAS-CONICET. Her research interests include graph theory, optimization, and applications of OR.
Diego Delle Donne holds a degree in computer science from the University of Buenos Aires, where he is a PhD student. He is a teaching assistant at the Sciences Institute at the National University of General Sarmiento. His research interests include integer programming, graph theory, and OR.

Guillermo Durán holds a PhD in computer science from the University of Buenos Aires. He is head of the Instituto de Cálculo and assistant professor in the Mathematics Department of the School of Sciences at the University of Buenos Aires (Argentina), adjunct professor in the Industrial Engineering Department at the University of Chile (Chile), and researcher at CONICET (Argentina). His research interests include operations research, graph theory, combinatorial optimization, and applications of OR.

Javier Marenco holds a PhD and a degree in computer science from the University of Buenos Aires. He is assistant professor at the Sciences Institute at the National University of General Sarmiento and a private consultant in OR. His research interests include integer programming and applications of OR.