Solving the segmentation problem for the 2010 Argentine census with integer programming

Flavia Bonomo, Diego Delle Donne
Guillermo Durán, Javier Marenco

Computer Sciences and Mathematics departments, FCEyN, Universidad de Buenos Aires.
Sciences Institute, Universidad Nacional de General Sarmiento.

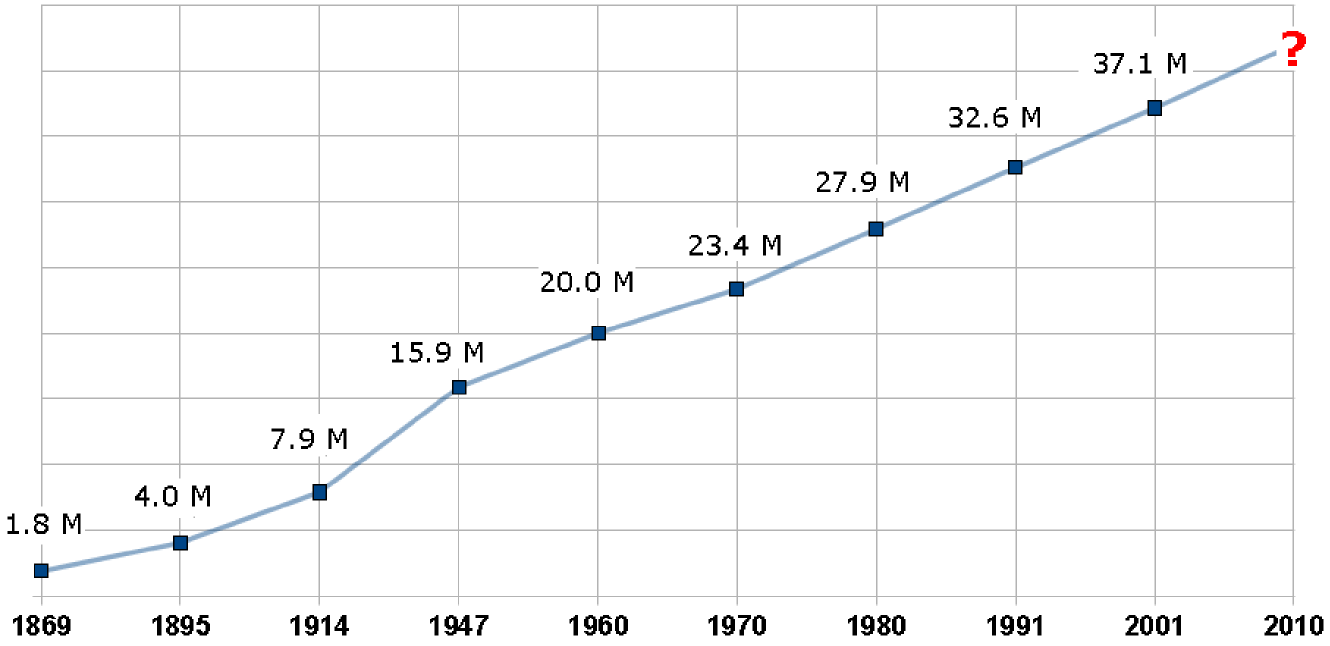
IFORS '11 - Melbourne, July 2011

Outline

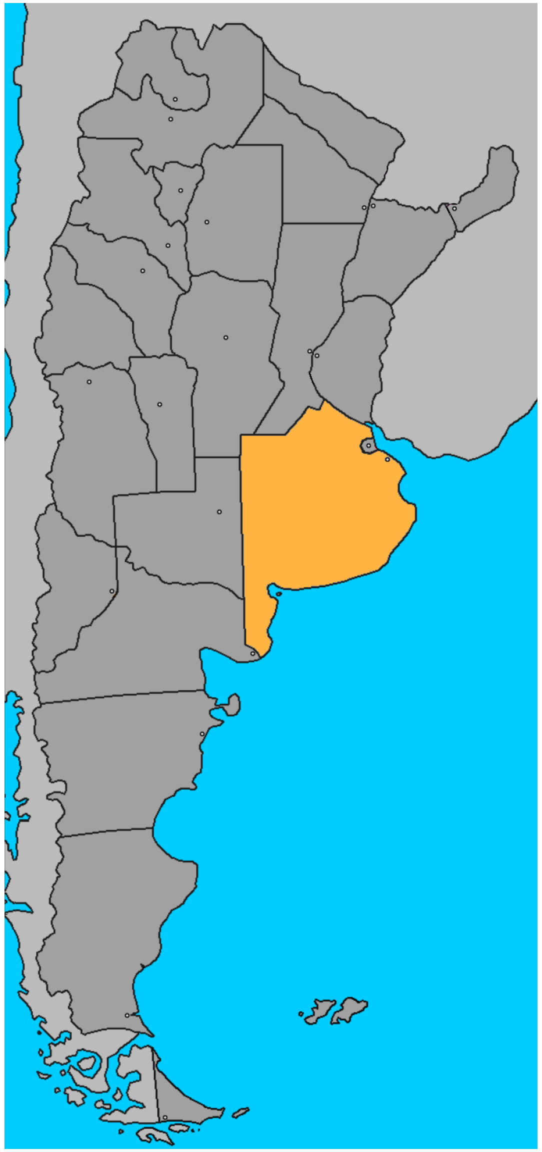
- Context and problem definition
  - Populational census in Argentina
  - The segmentation problem
- Resolution strategy
- Results
- Conclusions

Context and problem definition

- National Populational Census:
  Demographic survey conducted house to house.
- Includes employment, health and education, plus questions about disabilities, native people and access to technology.



Context and problem definition



- **Problem:** Decide which houses must visit each census taker.
- Buenos Aires Province:
  - 15.300.000 inhabitants
  - 307.571 km^2
- Predominantly rural province with towns of 10,000 to 100,000 inhabitants.

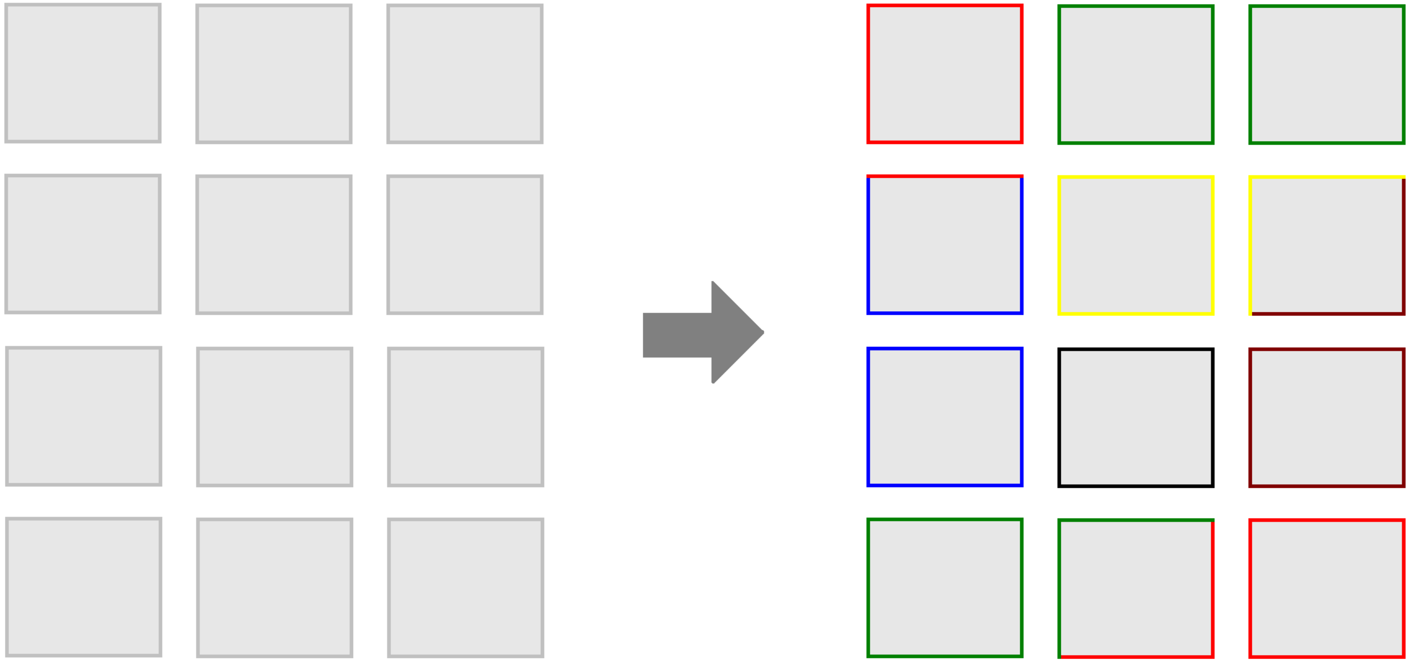
Context and problem definition

PCIA. DE SANTA FE
PCIA. DE CORDOBA
PCIA. DE ENTRE RIOS
URUGUAY
BENTOS
MERCEDES
TRINIDAD
CNIA. DEL SACRAMENTO
BUENOS AIRES
RIO DE LA PLATA
LA PLATA
Bahía Samborombón
MAR DEL PLATA
BAHIA BLANCA
VIEDMA
PCIA. DE LA PAMPA
RIO NEGRO
PCIA. DE RIO NEGRO
OCÉANO ATLÁNTICO SUR
MAR ARGENTINO
34°
36°
38°
40°
62°
60°
58°

- The Buenos Aires province is divided into 137 counties.
- Each county is divided into census tracks.
- Each track contains approx. 300 houses (⇒ between 1 y 40 blocks)
  - 16.691 urban tracks

Context and problem definition

Objective: Divide each track into segments, which will be visited by the census takers.



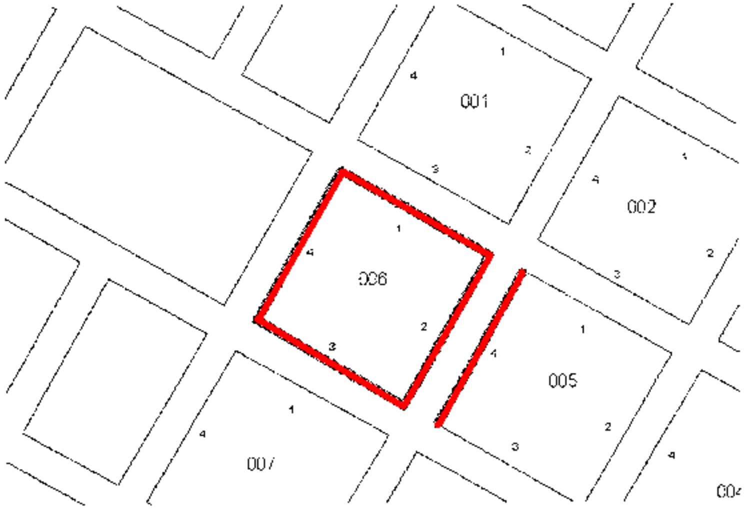
Context and problem definition

- The segmentation must satisfy the following **restrictions**:
  - Each segment must have between 32 y 40 houses.
  - A block side cannot be splitted, unless there is no solution (the same for buildings).
  - Empty block sides must also be covered by the segmentation.
  - A segment must be contained in one track.
  - Segments must be "as compact as possible".

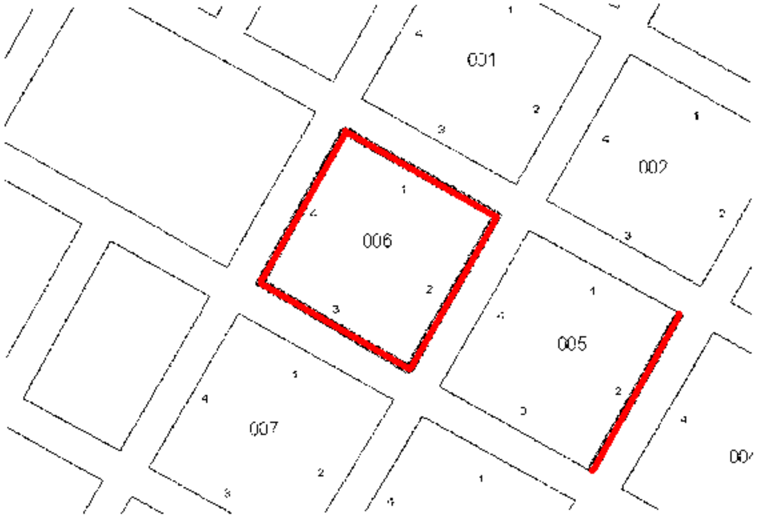
Context and problem definition

If a segment cross the street, it must cross to an adjacent block side (1/2).

SI



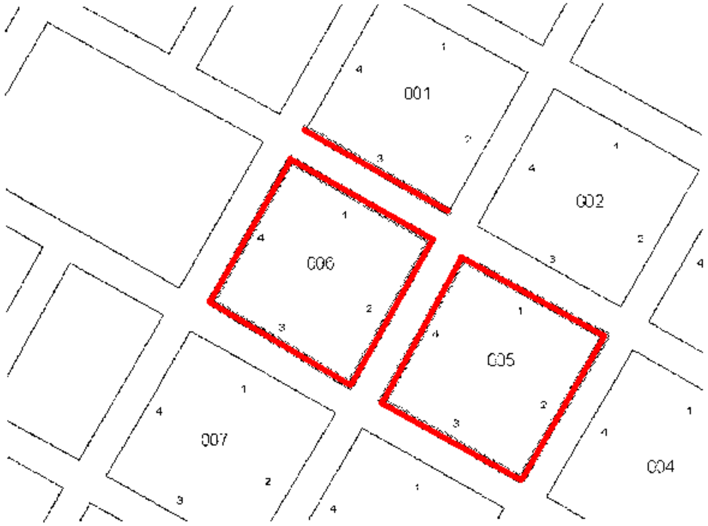
NO



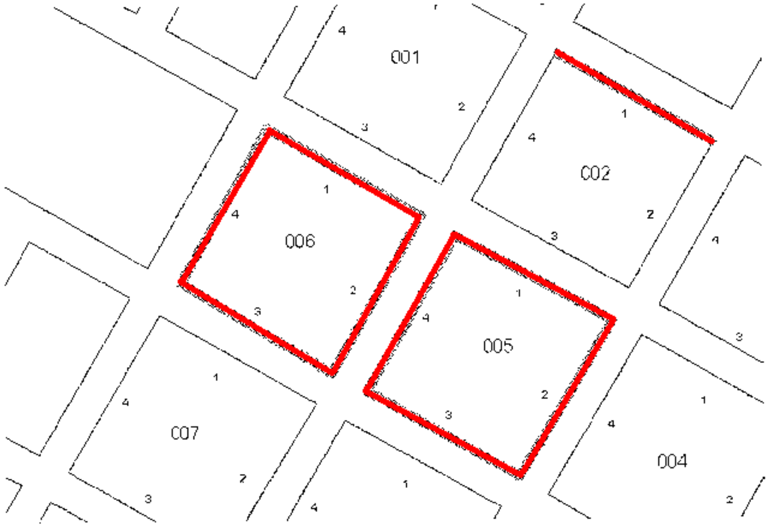
Context and problem definition

If a segment cross the street, it must cross to an adjacent block side (2/2).

SI

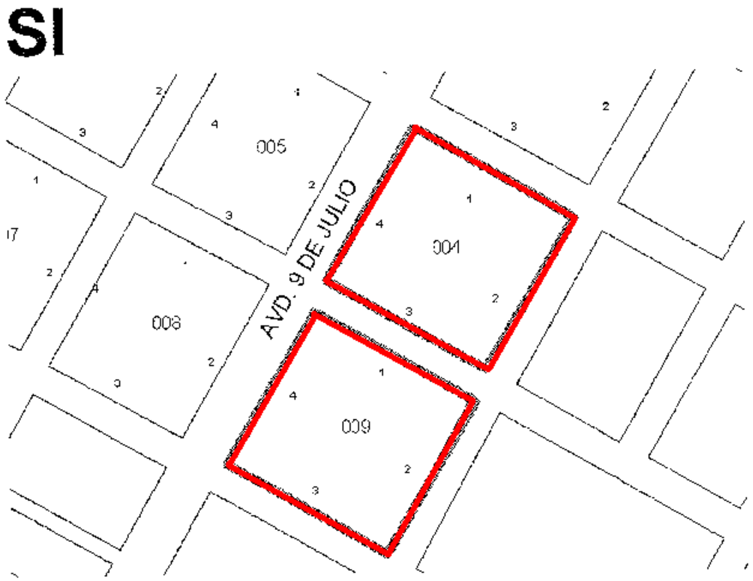


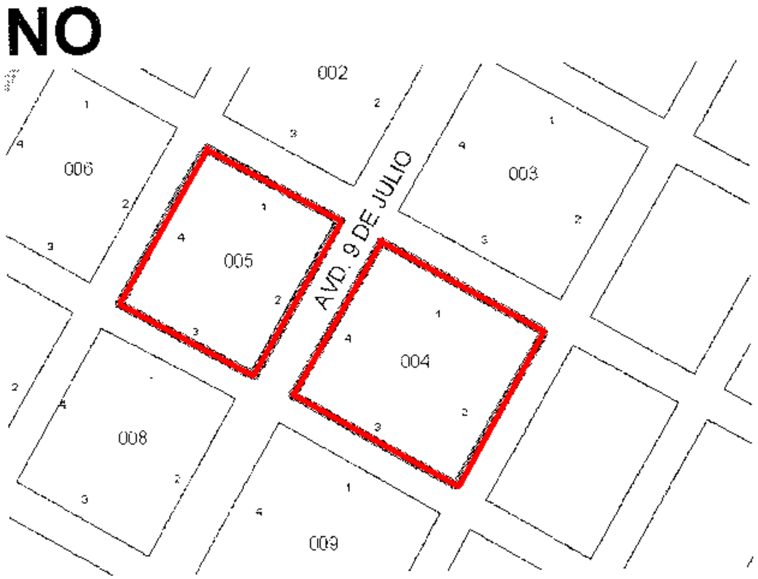
NO



Context and problem definition

An avenue cannot be crossed.

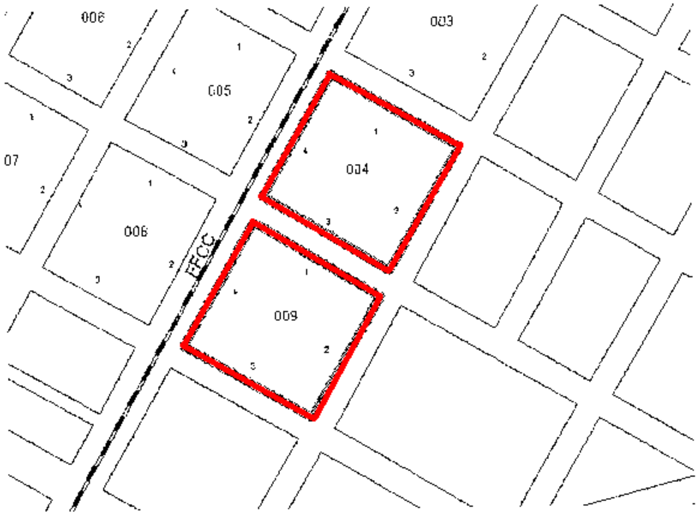




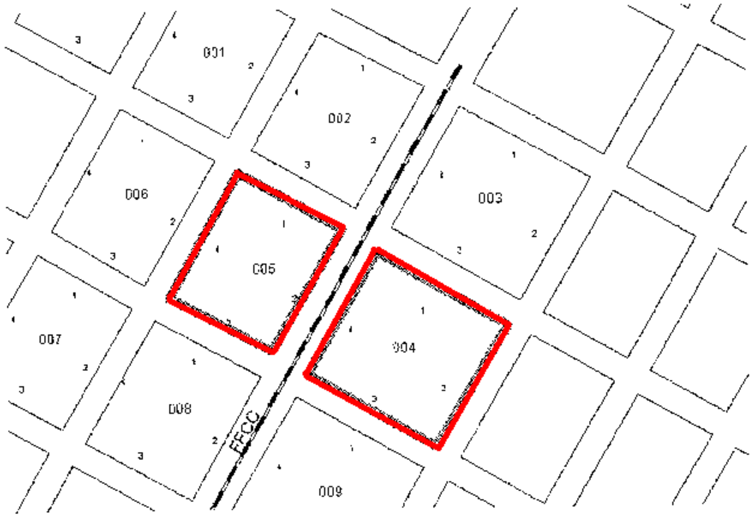
Context and problem definition

A railroad cannot be crossed.

SI

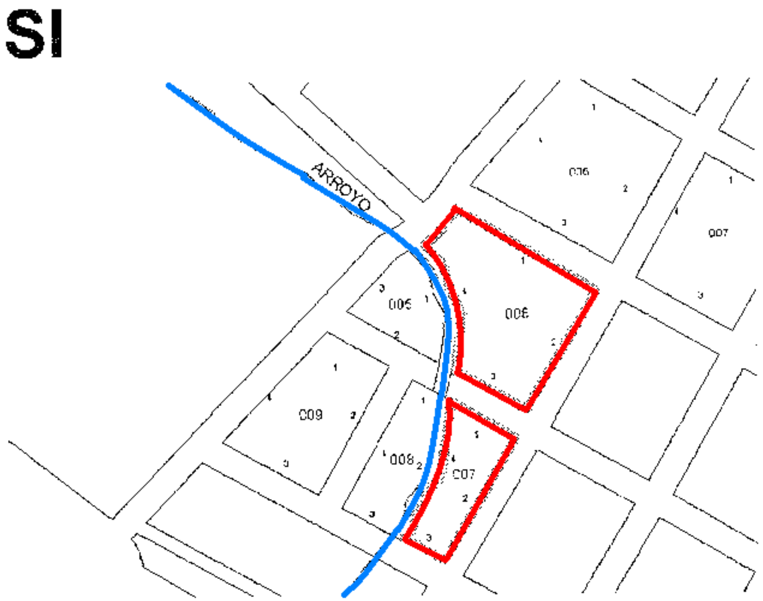


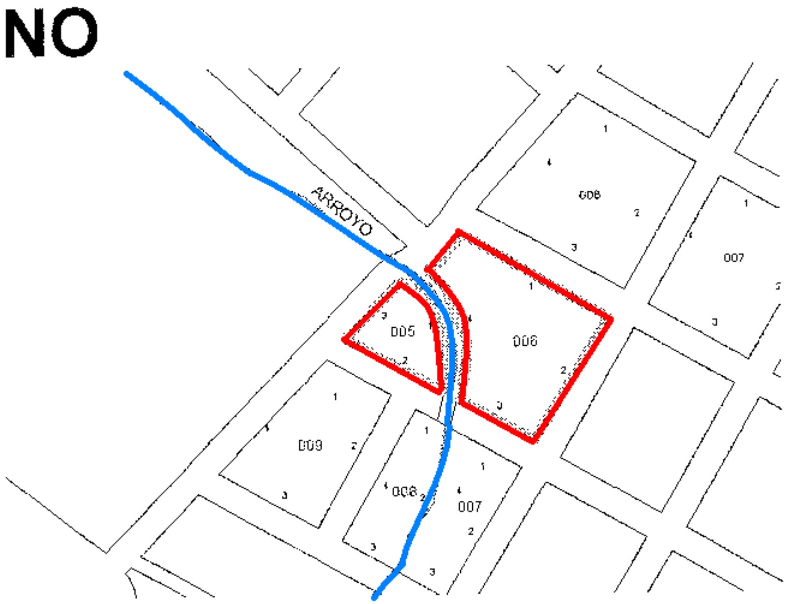
NO



Context and problem definition

Rivers or water courses cannot be crossed.





Outline

- Context and problem definition
  - Populational census in Argentina
  - The segmentation problem
- Resolution strategy
- Results
- Conclusions

Resolution strategy

Integer programming model:

- $S =$ valid segments set.
- For every $s \in S$, binary variable x_s specifying whether segment s is used or not.

$$
\begin{array}{llcll}
\max & \displaystyle\sum_{s \in S} \text{values}_s \, x_s & & & \\
\text{s.a.} & \displaystyle\sum_{s \in S_h} x_s & = & 1 & \forall\ h \in \text{Houses} \\
& \displaystyle\sum_{s \in S_l} x_s & = & 1 & \forall\ l \in \text{Empty block sides} \\
& x_s & \in & \{0,1\} & \forall\ s \in S
\end{array}
$$

Resolution strategy

The valuation coeficient of a segment s in the objective function is:

$$\text{value}_s \;=\; 10^{\frac{\#\text{sides}_s}{\#\text{blocks}_s}}.$$

Resolution strategy

- **Problem:** Too many segments!
  - Over 100.000 segments for simple tracks.
  - Too much time spent for the generation of the segments, before the model resolution.

Resolution strategy

- **Problem:** Too many segments!
  - Over 100.000 segments for simple tracks.
  - Too much time spent for the generation of the segments, before the model resolution.
- Imposing a limit on the number of segments may leave many uncovered houses.

Resolution strategy

- **Problem:** Too many segments!
  - Over 100.000 segments for simple tracks.
  - Too much time spent for the generation of the segments, before the model resolution.
- Imposing a limit on the number of segments may leave many uncovered houses.
- A column generation approach may be too risky (complicated implementation and risky results).

Resolution strategy 1/4:

1. Generate S_1 with **all** the (not exceded) segments which span over one block, and solve the model using every valid segment from S_1.

Resolution strategy

Resolution strategy 1/4:

1. Generate S_1 with **all** the (not exceded) segments which span over one block, and solve the model using every valid segment from S_1.

2. If no solution exists, generate $S_2 = \{s \in S_1 \times S_1 \ / \ s$ *conected* $\}$ and solve using the valid segments from $S_1 \cup S_2$.

Resolution strategy 1/4:

1. Generate S_1 with **all** the (not exceded) segments which span over one block, and solve the model using every valid segment from S_1.
2. If no solution exists, generate $S_2 = \{s \in S_1 \times S_1 \ / \ s$ *conected* $\}$ and solve using the valid segments from $S_1 \cup S_2$.
3. If no solution exists, generate $S_3 = \{s \in S_2 \times S_1 \ / \ s$ *conected* $\}$ and solve using the valid segments from $S_1 \cup S_2 \cup S_3$.

Resolution strategy 1/4:

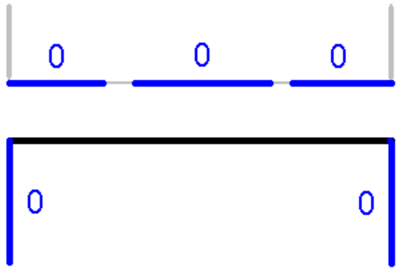
1. Generate S_1 with **all** the (not exceded) segments which span over one block, and solve the model using every valid segment from S_1.
2. If no solution exists, generate $S_2 = \{s \in S_1 \times S_1 \ / \ s$ *conected* $\}$ and solve using the valid segments from $S_1 \cup S_2$.
3. If no solution exists, generate $S_3 = \{s \in S_2 \times S_1 \ / \ s$ *conected* $\}$ and solve using the valid segments from $S_1 \cup S_2 \cup S_3$.
4. until the obtained set is empty (or a prespecified limit is reached)

Resolution strategy

Resolution strategy 1/4:

1. Generate S_1 with **all** the (not exceded) segments which span over one block, and solve the model using every valid segment from S_1.
2. If no solution exists, generate $S_2 = \{s \in S_1 \times S_1 \ / \ s \textit{ conected } \}$ and solve using the valid segments from $S_1 \cup S_2$.
3. If no solution exists, generate $S_3 = \{s \in S_2 \times S_1 \ / \ s \textit{ conected } \}$ and solve using the valid segments from $S_1 \cup S_2 \cup S_3$.
4. until the obtained set is empty (or a prespecified limit is reached)

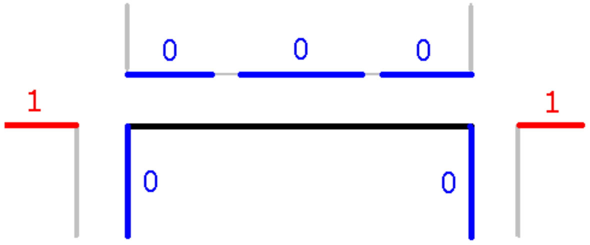
On each step, only level 0 neighbours are considered:



Vecinos de Nivel 0

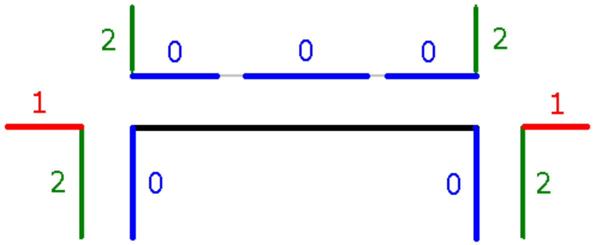
Resolution strategy 2/4:

$\boldsymbol{N_1}$. If there is no feasible solution, the process is repeated but using neighbours from levels 0 and 1:



Vecinos de Nivel 0 y 1

$\boldsymbol{N_2}$. If there is no feasible solution, the process is repeated but using neighbours from levels 0, 1 and 2:



Vecinos de Nivel 0, 1 y 2

Resolution strategy

Resolution strategy 3/4:

- If there is no solution, enable the side splitting option, leaving buildings unsplitted (if possible):

Resolution strategy 3/4:

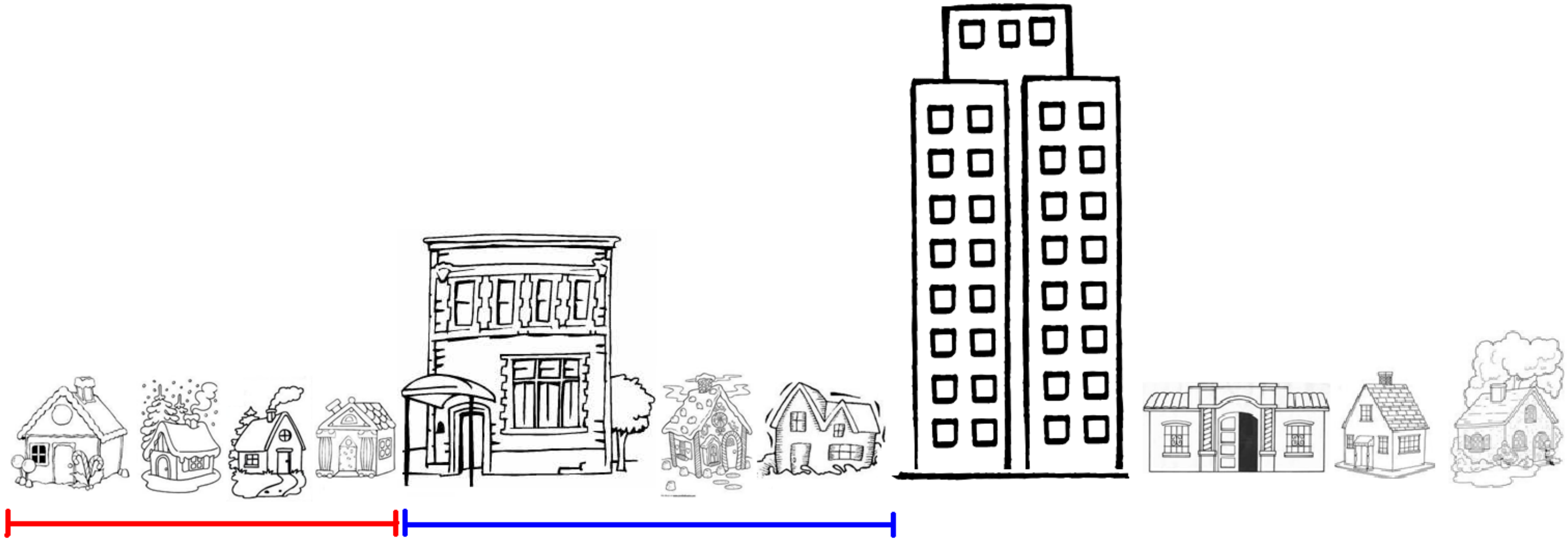
- If there is no solution, enable the side splitting option, leaving buildings unsplitted (if possible):

Resolution strategy 3/4:

- If there is no solution, enable the side splitting option, leaving buildings unsplitted (if possible):

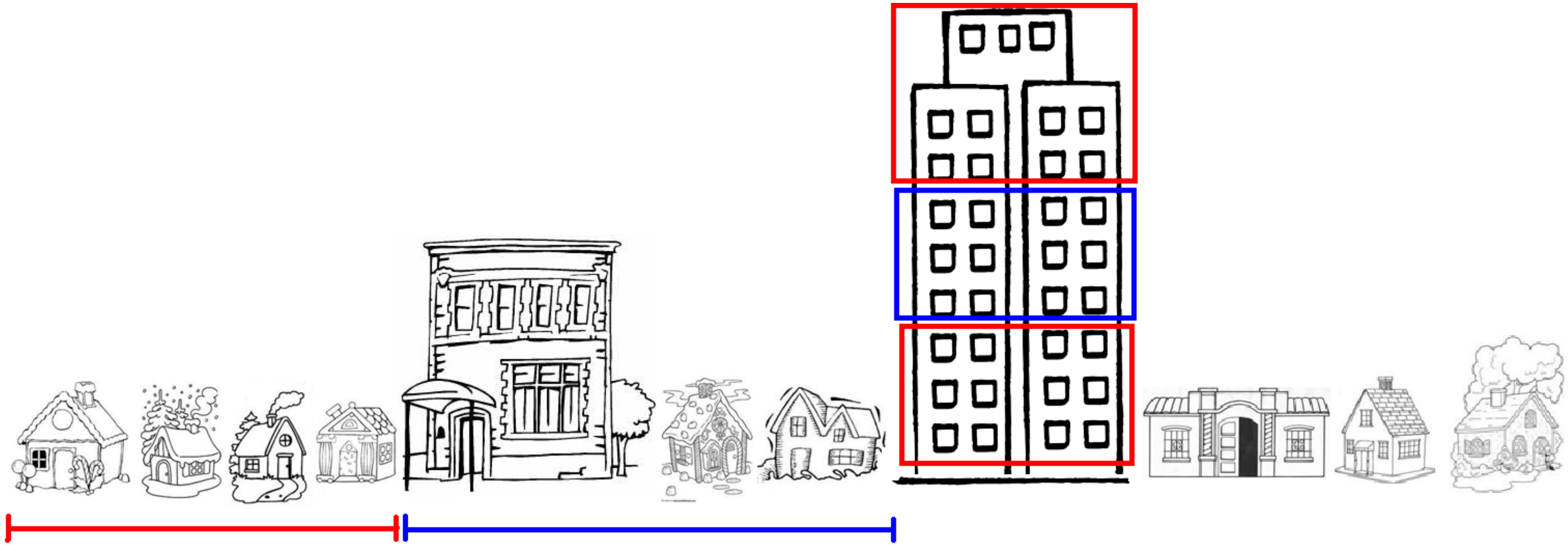
Resolution strategy 3/4:

- If there is no solution, enable the side splitting option, leaving buildings unsplitted (if possible):



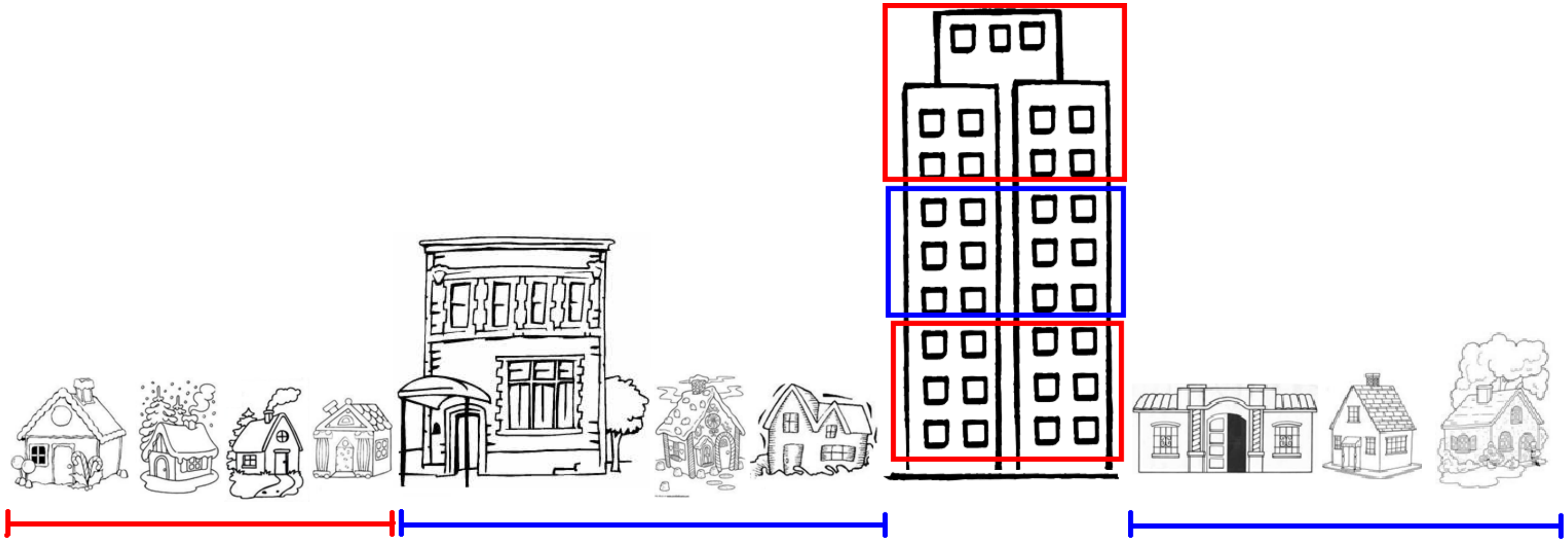
Resolution strategy 3/4:

- If there is no solution, enable the side splitting option, leaving buildings unsplitted (if possible):



Resolution strategy 3/4:

- If there is no solution, enable the side splitting option, leaving buildings unsplitted (if possible):



Resolution strategy 4/4:

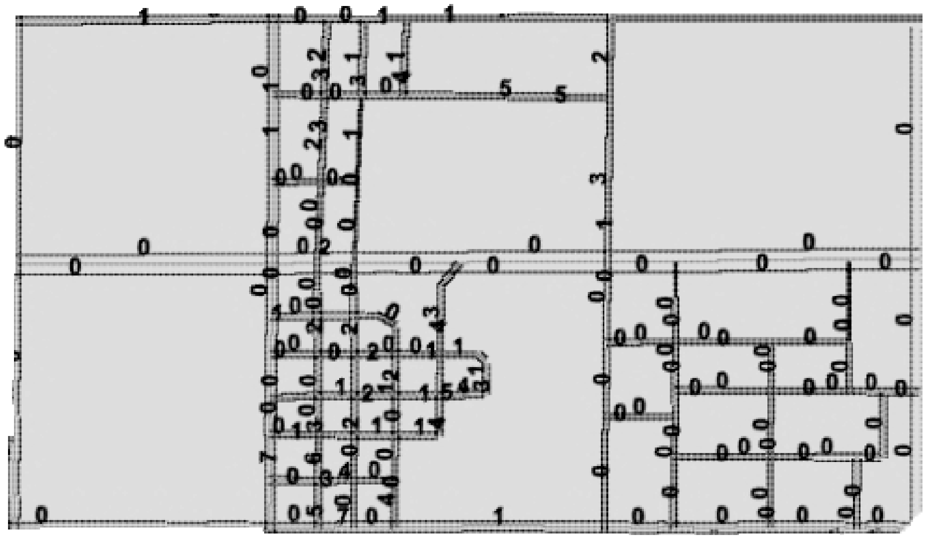
- Resolution process is repeated but using a new set S_1' of base segments.

Resolution strategy

- This approach allowed us to solve a large number of tracks, especially in urban areas.

Resolution strategy

- This approach allowed us to solve a large number of tracks, especially in urban areas.
- **Problem:** In rural areas, the number of segments may be too large!



- More than 100.000 segments, and up to 10 minutes to generate them! (implementation using C++)

Resolution strategy

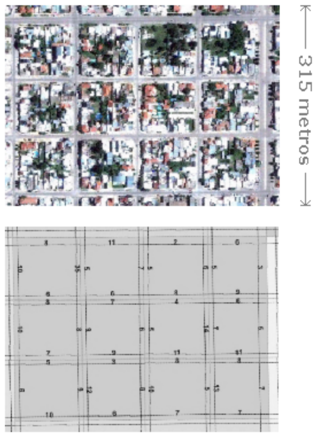
- **Solution:** We implemented the following parameters, in order to solve rural areas:
  - Blocks having a number of houses below a given number, are not splitted in parts.
  - Base segments S_1 must have at least a minimum number of houses (if not, they are arbitrarily grouped in order to reach this number).
- Handling this new parameters, we were able to solve almost every rural tracks.

Resolution strategy

- **Solution:** We implemented the following parameters, in order to solve rural areas:
  - Blocks having a number of houses below a given number, are not splitted in parts.
  - Base segments S_1 must have at least a minimum number of houses (if not, they are arbitrarily grouped in order to reach this number).
- Handling this new parameters, we were able to solve almost every rural tracks.
- **New problem:** Too many parameters to set!

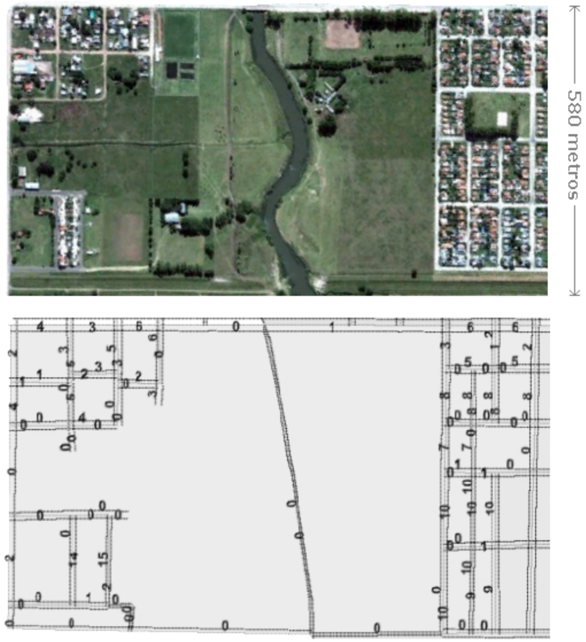
Resolution strategy

Solution: Classify tracks in three categories, with a different parameter set for each case:



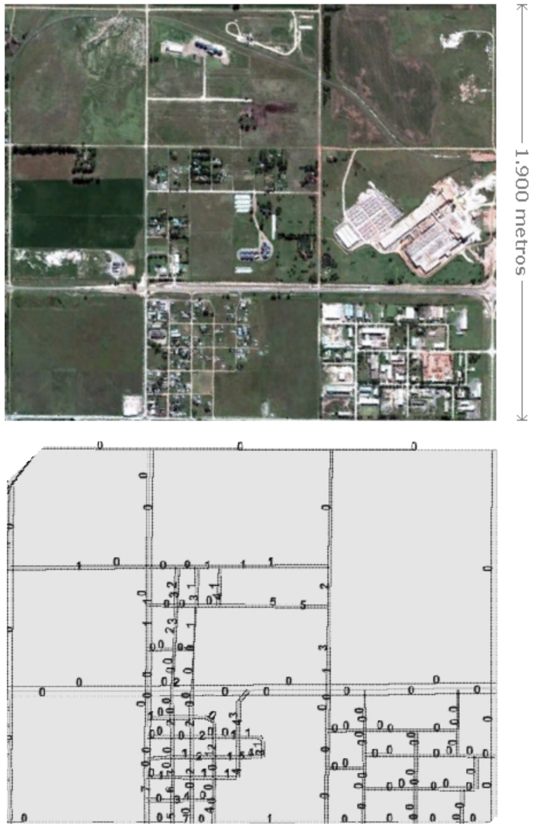
Radios urbanos

(hasta 10 manzanas, sin manzanas "grandes")



Radios semiurbanos

(entre 10 y 30 manzanas)



Radios rurales

(más de 30 manzanas)

Resolution strategy

According to the track category, the following parameter sets are used for the resolution:

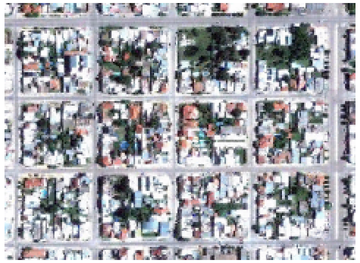
	Urban	Semiurb.	Rural
Maximum iterations in segment generation stage:	4	7	9
Minimum number of houses for a block to be divisible:	1	2	10
Minimum number of houses for the base segments	0	1	5
Maximum number of houses in a part (when side-splitting is applied)	32	32	40
Time limit for the IP model (sg):	60	60	120

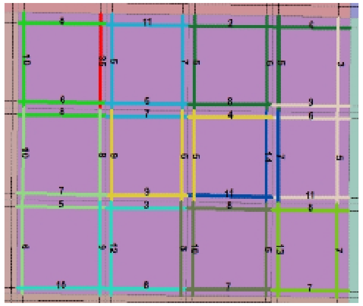
Outline

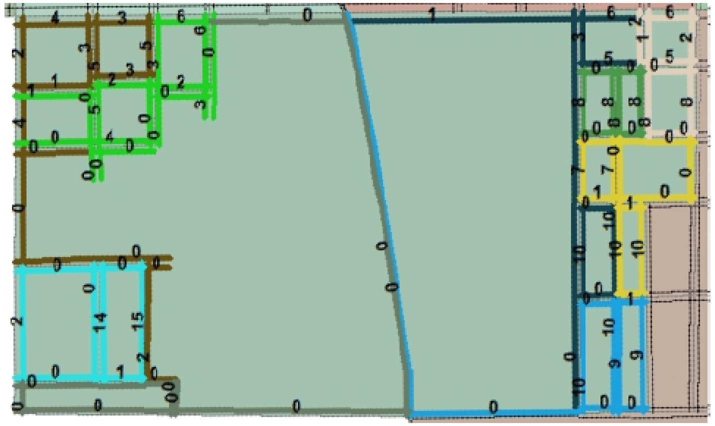
- Context and problem definition
  - Populational census in Argentina
  - The segmentation problem
- Resolution strategy
- Results
- Conclusions

Results

Segmented tracks examples:

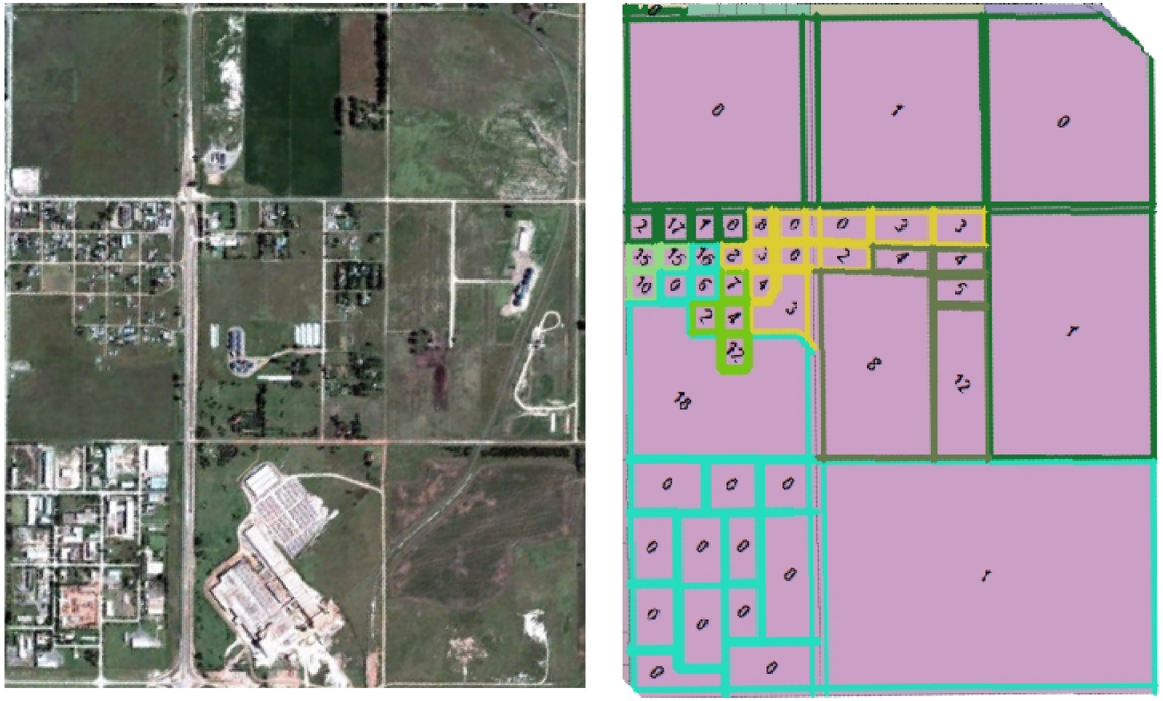






Results

Segmented tracks examples:



Results

- In the previous census (2001) an attempt to develope an automatic tool was made (greedy), having no success.
  - Segmentation was done manually.
  - 25 operators, double shift, for 30 días in a row (around 6000 manhours).
  - There were 15% less census tracks.

Results

- In the previous census (2001) an attempt to develope an automatic tool was made (greedy), having no success.
  - Segmentation was done manually.
  - 25 operators, double shift, for 30 días in a row (around 6000 manhours).
  - There were 15% less census tracks.
- In this census (2010) our tool was applied.
  - 96% of the tracks were solved in approximately 320 hours of processing (e.g., less than a day in a cluster with 15 PCs).
  - Homogeneous segmentation and uniform criteria (versus manual segmentation which strongly depends. on each operator).

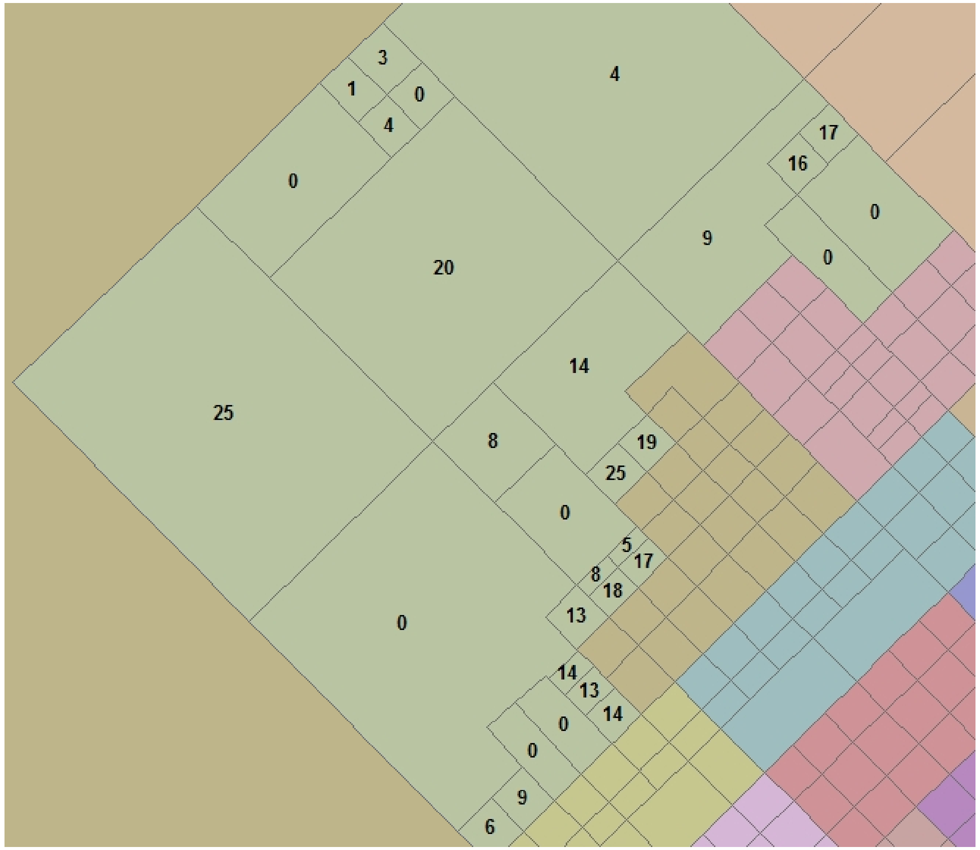
Results

- Segment generation is done in a few seconds (worst cases around 2 minutes).
- Over 99% of the IP models could be solved in a few seconds.
  - Linear relaxation too tight (!).
  - The first feasible solution found is often optimal.
- In very few tracks, time limit is reached with suboptimal solution (which is taken as the track solution).

Results

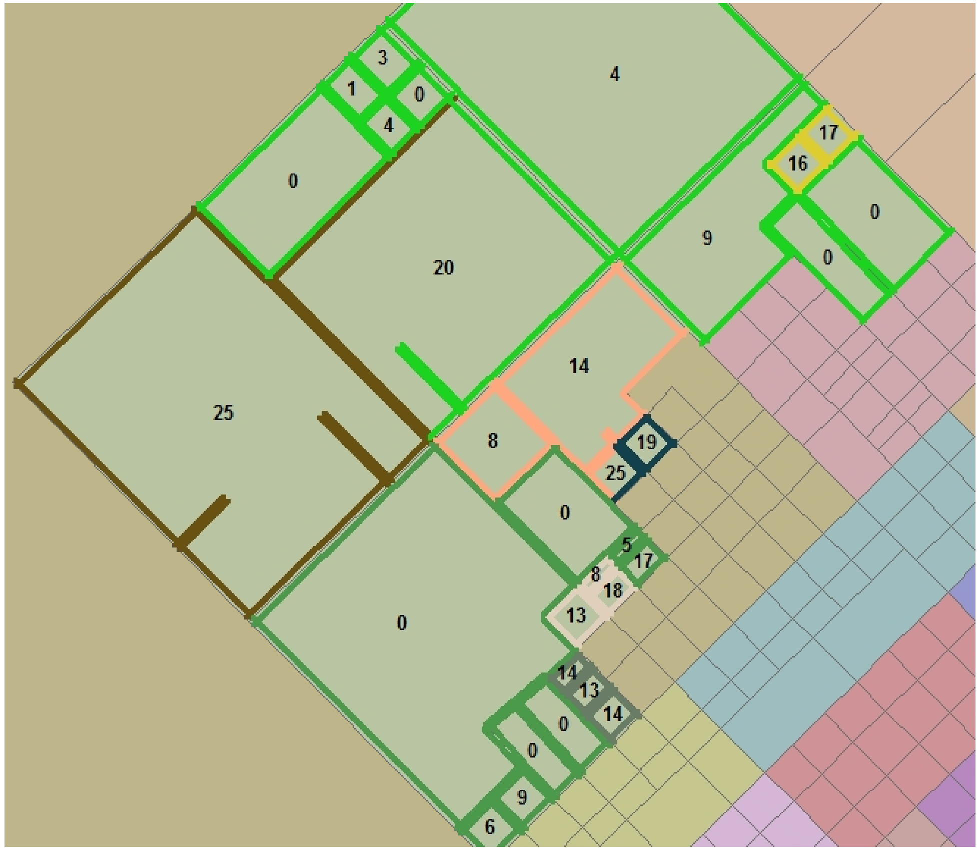
- Near 600 tracks can not be automatically solved.
- A few high populated blocks, surrounded by rural blocks.
- These tracks were solved using our tool by relaxing constraints or, in the worst case, manually.

Results



- Near 600 tracks can not be automatically solved.
- A few high populated blocks, surrounded by rural blocks.
- These tracks were solved using our tool by relaxing constraints or, in the worst case, manually.

Results



- Near 600 tracks can not be automatically solved.
- A few high populated blocks, surrounded by rural blocks.
- These tracks were solved using our tool by relaxing constraints or, in the worst case, manually.

Outline

- Context and problem definition
  - Populational census in Argentina
  - The segmentation problem
- Resolution strategy
- Results
- Conclusions

Conclusions

- The segmentation problem in the Buenos Aires Province could be solved in time (we only had 2 months for all the project).
  - The sequential segment generation helped to follow the preference order in the obtained solutions.
  - The track classification in three classes allowed to properly handle almost every instance.
- Data processing using a geographical information system was crucial for the development of our tool.

Thank you!