REGION DIVISION

ISAAC MEZA

In this short note, we aim to explain the *division problem* and how we worked out the solution.

1. The problem

The objective is to have a *fair* division of 1404 zip-codes into 60 regions, such that the resulting areas are connected and disjoint. First we need to define what we mean for a partition to be *fair*.

Definition 1.1. *Let S be a finite set and $S = \{S_1, \dots, S_n\}$ be a partition. We say that $\{S_1, \dots, S_n\}$ is f-fair if*

$$f(S_i) = f(S_j) \qquad \forall\, i, j = 1, \dots, n$$

for f a set function.

It is important to note that f-fair partition not always exists, in that case we look for the best f-fair approximation, i.e. a partition that minimizes the variance

$$\min_{\{S_1,\dots,S_n\}} \frac{1}{n} \sum_i |f(S_i) - \mu|^2$$

where $\mu := \frac{1}{n} \sum f(S_i)$.

Thus, the problem we will try to solve is

$$\min_{\{S_1,\dots,S_n\}} \frac{1}{n} \sum_i |f(S_i) - \mu|^2 \quad (1)$$

$$\text{s.t.}$$

$$S_i \text{ is connected}$$

Note that the problem is NP, so a heuristic needs to be considered to approach a solution.

2. Proposed algorithm

In order to make the problem more tractable, we impose some structure to the function f. Let $f : 2^X \longmapsto \mathbb{R}$ satisfy

(i) Monotonicity : $|A| \geq |B| \Longrightarrow f(A) \geq f(B)$

(ii) Sub-additive : $A \subseteq B \Longrightarrow f(A) \leq f(B)$

As already pointed, 1 is an NP problem. We consider the following to be a good approximation[1] of problem 1.

We map each point of $S \subset \mathbb{R}^2$ with a point in $\mathbb{R}$, in such a way that closeness is preserved. The injection is then $S' := \{1, \dots, |S|\}$. The problem is then

$$\min_{c_1,\dots,c_{n-1}} \frac{1}{n} \sum_i |f(S_i') - \mu\prime|^2 \quad (2)$$

[1] Moreover, attempts to solve problem 1 'directly' did not achieve good solutions. The heuristics of this attempts were variants of k-means and fair-clustering.

where

$$S'_i = \begin{cases} [1, c_1) & \text{if} \quad i = 1 \\ [c_i, c_{i+1}) & \text{if} \quad i = 2, \ldots, n-1 \\ [c_{n-1}, |S|] & \text{if} \quad i = n \end{cases}$$

and $\mu\prime = \frac{1}{n} \sum f(S'_i)$.

The following routines are followed in order to find a solution to 2.

As in may optimization problems, the solution may be sensitive to the initial point chosen. The first procedure (1) chooses this initial point.

Algorithm 1: Initial solution

input : $S' = \{1, \ldots, |S|\}$
output: $\sigma^2, \{c_i\}$

$c_1 \leftarrow \lfloor (|S|/n) \rfloor$
for $i \leftarrow 2$ **to** $n-1$ **do**
  $c_i \leftarrow c_{i-1} + \lfloor (|S|/n) \rfloor$
  while $f(S'_i) > 2f(S'_{i-1})$ **do**
    $c_{i-1} \leftarrow c_{i-1} + 1$
    $c_i \leftarrow c_i - 1$
$\sigma^2 \leftarrow \mathrm{var}\{f(S'_i)\}$

Procedure 2 balances individually each 'cut' c_i, while algorithms 3 (4) move simultaneously all the cuts to the right (left) from the minimum (maximum) f-evaluated component of the partition.

Algorithm 2: Greedy variance minimization

input : $\sigma^2, \{c_i\}$
output: $\sigma^2, \{c_i\}$

for $i \leftarrow 1$ **to** $n-1$ **do**
  $\sigma_0^2 \leftarrow \infty$
  while $\sigma^2 < \sigma_0^2$ **do**
    $\sigma_0^2 \leftarrow \sigma^2$
    $(c_i^0) \leftarrow (c_i)$
    if $f(S'_i) < f(S'_{i+1})$ **then**
      $c_i \leftarrow c_i + 1$
    else
      $c_i \leftarrow c_i - 1$
    $\sigma^2 \leftarrow \mathrm{var}\{f(S'_i)\}$
  $\sigma^2 \leftarrow \sigma_0^2$
  $(c_i) \leftarrow (c_i^0)$

Finally, routine 5 splits the largest component (in terms of f) and merges the smallest component with one of its neighbours.

Algorithm 3: Push to the right

input : $\sigma^2, \{c_i\}$
output: $\sigma^2, \{c_i\}$

$\sigma_0^2 \leftarrow \infty$
$m \leftarrow \min_i\{f(S_i')\}$
while $\sigma^2 < \sigma_0^2$ **do**
    $\sigma_0^2 \leftarrow \sigma^2$
    $(c_i^0) \leftarrow (c_i)$
    $(c_i)_{i=m}^{n-1} \leftarrow (c_i)_{i=m}^{n-1} + 1$
    $\sigma^2 \leftarrow \mathrm{var}\{f(S_i')\}$
$\sigma^2 \leftarrow \sigma_0^2$
$(c_i) \leftarrow (c_i^0)$

Algorithm 4: Push to the left

input : $\sigma^2, \{c_i\}$
output: $\sigma^2, \{c_i\}$

$\sigma_0^2 \leftarrow \infty$
$M \leftarrow \max_i\{f(S_i')\} - 1$
while $\sigma^2 < \sigma_0^2$ **do**
    $\sigma_0^2 \leftarrow \sigma^2$
    $(c_i^0) \leftarrow (c_i)$
    $(c_i)_{i=1}^{M} \leftarrow (c_i)_{i=1}^{M} + 1$
    $\sigma^2 \leftarrow \mathrm{var}\{f(S_i')\}$
$\sigma^2 \leftarrow \sigma_0^2$
$(c_i) \leftarrow (c_i^0)$

3. Practical division

We have geocoded the addresses of all defendants of all the lawsuits the JLCA received during a year. Then, we take S to be the collection of all zip-codes that have at least one recorded lawsuit[2]. In practice we consider f to be a measure of the backlog (in notifications) of each area. Therefore, we are considering divisions that 'equates' the backlog for each area. We understand the backlog as inflow - outflow, where the inflow measures the proportion of cases each area needs to notify and the outflow is the time it takes to visit each point.

Specifically, the inflow is a constant proportion (5%) of all addresses that fall in an area, while the outflow is the time it takes a notifier to complete a Hamiltonian path of this 5% (starting and ending at the JLCA)[3]. As we randomly choose 5% we run 100 replications.

The route programmed to be followed by each notifier is simply achieved by the *greedy algorithm* - choose the next closest point.

The first step toward the solution is to create a map from $S \subset \mathbb{R}^2$ to $S' := \{1, \ldots, |S|\}$. We achieved this by producing a greedy-route (from the centroid of the zip-codes) starting from the JLCA. This gave us the order of the zip-codes visited which can be seen in figure 1

Applying the algorithm described in section 2, we come up with the figure 2.

[2]Because at the end we would like to have a partition of all the 1404 zip-codes that form the CDMX, those zip-codes not included are assigned to their nearest region

[3]Note that this function satisfies the properties required in section 2

Algorithm 5: Merge & Split

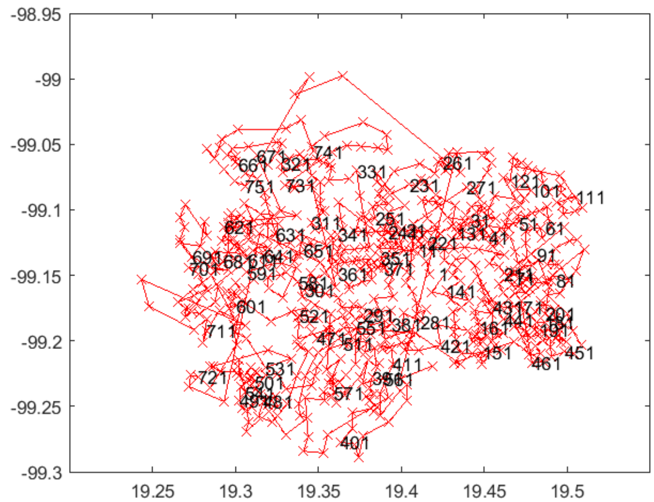
input : $\sigma^2, \{c_i\}$
output: $\sigma^2, \{c_i\}$

$\sigma_0^2 \leftarrow \infty$
while $\sigma^2 < \sigma_0^2$ **do**
  $\sigma_0^2 \leftarrow \sigma^2$
  $(c_i^0) \leftarrow (c_i)$
  $m \leftarrow \min_i\{f(S_i')\}$
  $M \leftarrow \max_i\{f(S_i')\} - 1$
  # Merge
  if $m < M$ **then**
    $(c_i)_m^{M-1} \leftarrow (c_i)_{m+1}^M$
  else
    $(c_i)_{M+1}^m \leftarrow (c_i)_m^{M-1}$
  # Split
  if $M = 1$ **then**
    $c_M \leftarrow \lfloor c_M/2 \rfloor$
  else
    if $M < |S|$ **then**
      $c_M \leftarrow \lfloor \frac{c_{M+1}+c_M}{2} \rfloor$
    else
      $c_M \leftarrow \lfloor \frac{|S|+c_M}{2} \rfloor$
  $vr_0 \leftarrow \infty$
  $vr \leftarrow \mathrm{var}\{f(S_M), f(S_{M+1})\}$
  while $vr < vr_0$ **do**
    $vr_0 \leftarrow vr$
    if $f(S_i') < f(S_{i+1}')$ **then**
      $c_i \leftarrow c_i + 1$
    else
      $c_i \leftarrow c_i - 1$
    $vr \leftarrow \mathrm{var}\{f(S_M), f(S_{M+1})\}$
  $\sigma^2 \leftarrow \mathrm{var}\{f(S_i')\}$
$\sigma^2 \leftarrow \sigma_0^2$
$(c_i) \leftarrow (c_i^0)$

Figure 3, shows the resulting distribution for some statistics, which includes the backlog. The objective function is then the variance of the latter - **51**.

Complementing this histograms, the heat-maps of figure 5 display the spatial statistics of the backlog, as well as a decomposition of its inflow and outflow.

Figure 1. Snake path



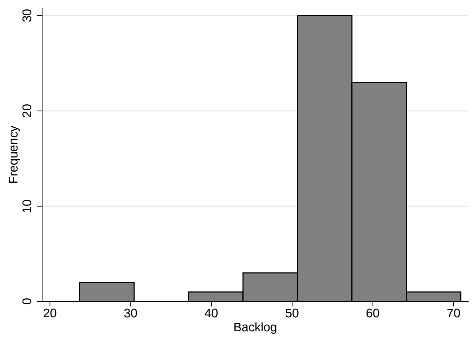
Notes: Greedy route, visiting all zip-codes starting from the JLCA.

Figure 2. Clusters

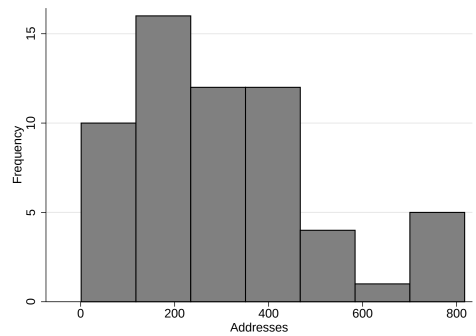
Notes: The figure shows the final clusters. Some clusters share the same color.

Figure 3. Histograms

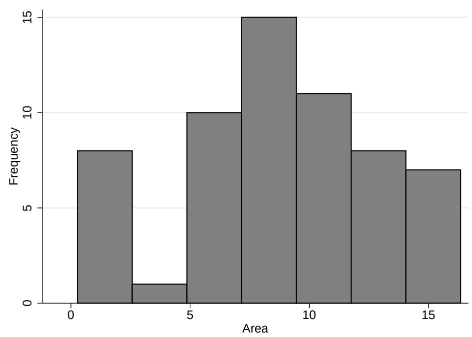
(a) Backlog



(b) Addresses - Lawsuits



(c) Area



(d) # Zip-Codes

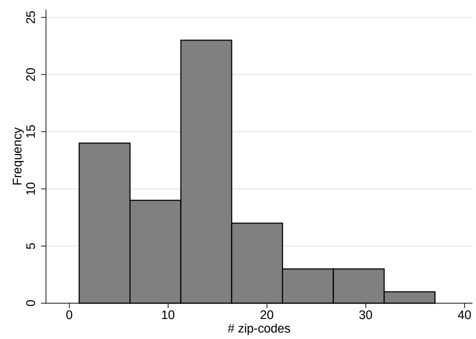
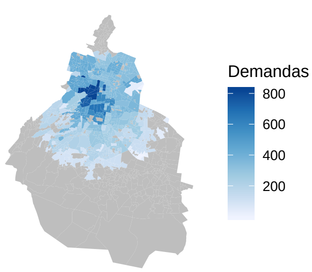
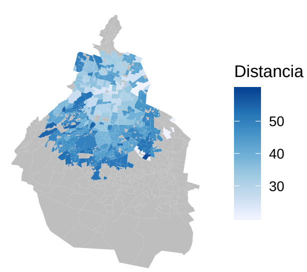


Figure 5. Heat maps

(a) Inflow



(b) Outflow



(c) Backlog