Optimizing French Regions

Tristan Cazenave[1], Anne Epaulard[2,3], Claire Bernard[3]

[1] PSL-Université Paris-Dauphine, LAMSADE CNRS UMR 7243, Paris, France
Tristan.Cazenave@dauphine.fr

[2] PSL-Université Paris-Dauphine, LEDa, Paris, France
Anne.Epaulard@dauphine.fr

[3] France Stratégie, Paris, France

Abstract : France is in the process of reorganizing its regions. We propose a heuristic search algorithm that optimizes french regions using geographical, economical and political heuristics. The corresponding admissible heuristics are explained and integrated into the search algorithm.

1 Introduction

When we started our work, there were 22 regions in metropolitan France. They were diverse in size, either measured by GDP, population or territory (see Figure 1). The design of borders between regions was based on a mix of history, geography, central government voluntarism and political forces at local and central levels.

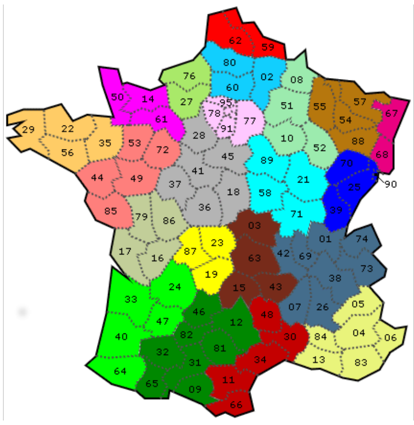

Figure 1: The 22 regions before the reorganization

By law of July 5th 1972 they are supposed to intervene in transport infrastructures and public transport, country and town planning, economic development, high schools and lifelong learning, research and culture.

There are (at least) two lower jurisdictional levels in France: departments (96 in Metropolitan France) and communes (36552 in Metropolitan France). The number of departments by region varied from 2 to 8.

In many occasions since 1972, the question of the number of regions has been debated and few proposals have been put forward to reduce their number and to redesign their borders. Usually these new designs are based on the same set of ingredients that was initially used, but they don't lead to the same designs, depending on the weights given to each ingredients. What we intend to do in this article is to highlight the different options for redesigning the regions' borders and show how different weights or arbitrary choices lead to different maps. The list of ingredients we are going to use to redesign regions is going to be based on the lessons we can draw from the economic literature. To these different ingredients (there will be 4 of them) we will assign weights and show how different weights lead to different borders. But ingredients and their weights are not the only variables that play a role in the final design of the map. Our methodology is the following. We start by defining a function of ingredients and weights that sums up the trade-off between small homogenous territories and their costs. We then try to find the set of regional borders that minimizes this function. A region is defined by the addition of at least 2 contiguous departments (of which there are 96). This is done by using a search algorithm. In addition, as we will see, the presence of one ingredient (travel time to the regional capital) makes it necessary to have a list of regional capitals. From all this optimization we get a set of optimal regional borders than can be used for policy decisions.

The paper is organized in five remaining sections: section 2 presents the ingredients from the economic literature, section 3 gives the function to minimize, section 4 explains the corresponding admissible heuristics, section 5 the search alorithm and section 6 the experimental results.

2 Selecting a set of ingredients from the economic literature

The question we are interested in is how to divide a country into regions and set the jurisdictional boundaries based on economic rationale. To our knowledge this precise question has never been studied from an economic point of view. But there are three fields of economics from which we can draw to help set our problem: fiscal federalism, optimal size of countries, new geographic economy. The literature on fiscal federalism takes jurisdictional boundaries as given and discusses what type of functions and instruments should be given to each jurisdictional level (states in federation, regions, counties, cities...) and potential transfers between states or regions. There are rarely any discussion regarding where precisely boundaries should be drawn. In his Essay on Fiscal Federalism, William Oates (Oates, 1999) recognizes that if one were to start from an empty map of the United states and try to lay out a rational set of levels of government and borders for the jurisdictions at each level of government “One thing seems clear: such a system of jurisdictions would bear little resemblance to our existing map”. Particularly he notes that from the perspective of public good management rivers, that historically have been used to separates states, are the worse borders one can imagine since two independent jurisdictions have to share the management of the common public good. Yet, the fiscal federalism literature is useful from another point of view: it clearly sets the reason why there could be an advantage of local governments compared to the central government: the provision of goods and services whose consumption is limited to the area under the local government allows these goods and services to be tailored to local preferences and circumstances, thus increasing economic welfare compared to the situation were these public goods and services are provided uniformly across the country. This tells us that the main reason for the very existence of local government is that they have a better knowledge of preferences of individuals and production costs within their constituencies and are thus better at providing some (not all) public goods and services. On the opposite, the advantage of central government lies in the size of the economy / population it can tax to provide public goods and services that do not depend on local preference (defense, security, judicial system, macroeconomic stabilization) at a lower cost for taxpayers because of economies of scale and/or economies of scope. In addition, within a given country, social safety net may be more efficiently handed at the central level.

The second branch we can draw from to think about the division of a country in a set of regions, is the theoretical and empirical literature interested in the link between size and economies efficiency of countries. In their important book *The Size of Nations* Alesina and Spolaore start from the basic trade-off identified by the fiscal federalism literature between the benefit of size and the heterogeneity of preference over public goods. From an empirical point of view, it is difficult to find a link between size of nations and economic performance (Rose, 2006), but it is true however that smaller countries tend to have bigger government possibly due to the fact that they don’t benefit from economies of scale and scopes and/or to the fact that being smaller they need to be more open to international trade and thus get more term-of-trade shocks that in turn translate into bigger government to mitigate these shocks (Rodrik, 1998).

The third branch of literature that could help answer our question is the new geographic economy.

The economic literature from different fields is consistent with the fact that local governments’ actions can improve welfare when preferences are more homogenous over their territory than over the whole country and/or when local governments are better informed of demand for and production costs of goods and services provided by the government. For our problem this would imply to have regional design based on distance and preference. Applying this two criteria (information network and preferences) would lead to regions within France that are very different in population and GDP. This might not be a problem per se. However because of fixed costs and economies of scale very small regions that would be derived on information network and preferences will bear higher costs either per capita or by unit of GDP. Conversely, regional borders based on equalizing either population or GDP across region might lead to designed regions with very heterogeneous preferences where it will be difficult to satisfy the needs of inhabitants.

3 The function to minimize

The function we want to minimize mainly measures the loss of efficiency within regions. The total loss function is the weighted average of four components.

We use time to travel by car from a department to the regional capital city as an indication of the ability of local government to get information about demand, preferences and production costs for goods and services that are provided by local government. A higher travel time between a department and the regional capital city reduces the informational advantage of the local government. We suppose this loss of efficiency to be quadratic with travel time, and for each region we compute the population weighted average of the square time to travel from department capital city to regional capital city. The distances are normalized and the variance is set to 1.

To represent political preferences, we use the results of the second round of presidential elections of 2007 and 2012 for which data are available on data.gouv.fr. For a possible region (a set of department) we calculate the average result of the two political parties (they add up to one). We then measure the heterogeneity of preference within the region by computing the weighted sum over departments of this region of $(1.0 - \%\ of\ people\ voting\ as\ the\ region\ majority)^2$, with weight being the weight of the population in the department region. The higher this number is the more heterogeneous the political preferences within the region.

The third ingredient is the number of people traveling regularly from one department to another one. This value is normalized between 0.0 and 1.0. A value of 1.0 means no people travel between the two departments.

The fourth ingredient is the amount of financial exchanges between pairs of departments. This value is also normalized between 0.0 and 1.0, a value of 1.0 meaning no exchange. Data used to measure daily commuting and financial links between pairs of department are presented in (Amabile *et al.*, 2015).

For the transports and the finance heuristics the sum of the values for pairs of departments in the same region is computed and divided by the number of ordered pairs.

The loss function can be written as:

$$C_d \times \sum_{r \in regions} \sum_{d \in r} \frac{population\ of\ d}{average\ population} \times dist^2 + C_m \times \sum_{r \in regions} \sum_{d \in r} \frac{population\ of\ d}{average\ population} \times (1.0 - majority)^2 + C_t \times \sum_{r \in regions} \sum_{d, d_1 \in r} \frac{transports[d][d_1]}{nb\ ordered\ pairs\ in\ r} + C_f \times \sum_{r \in regions} \sum_{d, d_1 \in r} \frac{finance[d][d_1]}{nb\ ordered\ pairs\ in\ r}$$

4 Admissible heuristics

The overall admissible heuristic used by the algorithm is a combination of four different heuristics. Each heuristic is associated to a weight and the weighted sum of the four heuristics gives the overall heuristic.

4.1 Strings and liberties

As an analogy to the game of Go (Bouzy & Cazenave, 2001; Müller, 2002) we will name strings the connected components of departments assigned to the same capital. A liberty of a string is a department neighbor of the string that is yet unassigned to a capital. The number of liberties of a string is the number of different unassigned neighbor departments.

If a string has zero liberties it means the region of the string is fully enclosed and that the exact value of the loss function for the region can be computed.

4.2 Distances

In order to have an admissible heuristic for distances to the regional capital we add the square of the distance to the closest regional capital when the department is not yet assigned.

When a department is assigned we add the square of the distance to its capital.

The resulting value is multiplied by the population of the department divided by the average population of departments in order to normalize it and have a value proportional to the number of inhabitants of the department.

4.3 Votes

For each department which is not yet assigned we add the minimum between $(1.0 - right)^2$ and $(1.0 - left)^2$.

For a completely enclosed region we add $\sum(1.0 - majority)^2$.

When a region is not completely enclosed we add the minimum between $\sum(1.0 - right)^2$ and $\sum(1.0 - left)^2$.

The weights for the left and right percentages are multiplied by the number of inhabitants of the department divided by the average number of inhabitants of a department.

The exact code used to compute the votes heuristic is given in Algorithm 1.

4.4 Transports

If a department is unassigned or if it is part of a region that still has liberties, the Transports heuristic cannot be computed since a region must be complete to take into account all the links to other departments of the region. Therefore we use the smallest possible value for the contribution of this department to the heuristic. To that end we take the smallest possible value between the department and all other possible departments. If a department is assigned to a capital and the other is assigned to another capital then the link is not taken into account. As the smallest possible value corresponds to the most important transport link of the department it already gives some important information to the search algorithm.

When a regions is completely enclosed we can compute the exact contribution of the region to the Transports heuristic. We sum all the transports values between each pair of departments and we divide the sum by the number of pairs in the region.

The Transports heuristic is given in Algorithm 2.

4.5 Finance

The Finance heuristic is very similar to the Transports heuristic except that instead of using the transports matrix, the finance matrix is used.

5 Search algorithm

The search algorithm is a combination of IDA* (Korf, 1985), optimization algorithms and Constraint Satisfaction heuristics (Bacchus & van Run, 1995; Gent *et al.*, 1996; Sadeh & Fox, 1996).

5.1 Contiguity

The first constraint on the regions is that they have to be contiguous. When considering a new state, if the algorithm detects that contiguity is impossible to reach, it stops search and returns a failure.

This is the case for example if a string has no liberties and if there is another string associated to the same capital.

Another constraint we have enforced is that each region has to contain at least two departments.

5.2 Fail-first heuristic

In order to try all the possible assignments of departements to capitals one has to select a department and try all its possible assignments to capitals. The choice of the next department to assign is important to optimize so as to reduce the search time. The key heuristic we use for choosing the next department to assign is the fail-first heuristic. It chooses the department that is the most constrained for its assignment.

The principle of choosing the most constrained department enables the search algorithm to fail fast if it is in a state that cannot be assigned. Instead of spending a lot of time in useless computations since a department cannot be assigned, it will see it rapidly and avoid useless searches.

Algorithm 1 The Votes heuristic

```
Votes ()
sum ← 0
for s in strings do
  if number of liberties of s = 0 then
    right ← 0
    for d in departments of s do
      right ← right + population[d] × right[d]
    end for
    right ← right/ population of s
    majority ← 0
    for d in departments of s do
      if right > 0.5 then
        majority ← (population of d / average population) ×(1.0 − right[d])^2
      else
        majority ← (population of d / average population) ×(1.0 − left[d])^2
      end if
    end for
    sum ← sum + majority
  else
    right ← 0
    left ← 0
    for d in departments of s do
      right ← (population of d / average population) ×(1.0 − right[d])^2
      left ← (population of d / average population) ×(1.0 − left[d])^2
    end for
    if right > left then
      sum ← sum + left
    else
      sum ← sum + right
    end if
  end if
end for
for d in unassigned departments do
  if right[d] > 0.5 then
    sum ← sum+ (population of d / average population) ×(1.0 − right[d])^2
  else
    sum ← sum+ (population of d / average population) ×(1.0 − left[d])^2
  end if
end for
return sum
```

Algorithm 2 The Transports heuristic

```
Transports ()
sum ← 0
for s in strings do
  if number of liberties of s = 0 then
    nbLinks ← 0
    trans ← 0
    for d in departments of s do
      for d1 in departments of s do
        if d1 ≠ d then
          if region[d1] = region[d] then
            trans ← trans + transports[d][d1]
            nbLinks ← nbLinks + 1
          end if
        end if
      end for
    end for
    if nbLinks > 0 then
      sum ← sum + trans/nbLinks
    end if
  end if
end for
for d in departments not in an enclosed region do
  t ← ∞
  for d1 in departments do
    if d1 ≠ d then
      if transports[d][d1] < t then
        t ← transports[d][d1]
      end if
    end if
  end for
  sum ← sum + t
end for
return sum
```

The choice of the most constrained department is given in Algorithm 4. It looks for unassigned departments and if it finds one with only one possible assignment it selects it. In the case where all departments have more than one possible assignment, it selects the department that has the smallest associated estimated search tree size. The computation of the estimated tree size is given in Algorithm 3. It uses the current threshold of the search and the admissible heuristic to estimate the complexity of the search after the assignment. The coeffTreeSize constant (set to 32.0 in our implementation) is elevated to the power of the threshold minus the heuristic value of the node to give the estimated tree size.

Algorithm 3 The algorithm finding the maximum number of possible assignments

maxAssignment (dept, treeSize)
if at least one unassigned neighbor department **then**
  $oneLiberty \leftarrow false$
end if
for c in possible capitals **do**
  if number of liberties of string(c) = 0 **then**
    $possible \leftarrow false$
  end if
  if not $oneLiberty$ **then**
    $noNeighborRegion \leftarrow true$
    for n in neighbors of dept **do**
      if region(n) = c **then**
        $noNeighborRegion \leftarrow false$
      end if
    end for
    if $noNeighborRegion$ **then**
      $possible \leftarrow false$
    end if
  end if
  if $possible$ **then**
    assign dept to c
    $h \leftarrow heuristic()$
    if $h < threshold$ **then**
      add c as possible capital
      $treeSize \leftarrow treeSize + pow(coeffTreeSize, (threshold - h))$
    end if
    unassign dept
  end if
end for
return number of possible assignments

Algorithm 4 The algorithm for choosing the most constrained department to assign

deptFailFirst ()
$mini \leftarrow \infty$
for i in unassigned departments **do**
  $n \leftarrow$ maxAssignment (i, treeSize)
  if $n \leq 1$ **then**
    return i
  end if
  if $treeSize < mini$ **then**
    $mini \leftarrow treeSize$
    $d \leftarrow i$
  end if
end for
return d

Table 1: Various combinations of weights and of the fail first heuristic.

C_d	C_m	C_t	C_f	$fail$	$nodes$	$time$
1.0	0.0	0.0	0.0	yes	499	1.77 s.
1.0	0.0	0.0	0.0	no	985	0.03 s.
0.5	0.5	0.0	0.0	yes	1,854	9.62 s.
0.5	0.5	0.0	0.0	no	$> 10^8$	>1 h.
0.25	0.25	0.25	0.25	yes	834	1.96 s.

5.3 Succeed-first heuristic

When the department to assign is chosen, the order in which the possible assignments are searched is important. The order we use follows the succeed-first heuristic. We start with the closest capital and continue with other capitals in order from the second closest to the farthest one. This enables the search to focus early on the most promising assigments, leaving the unlikely ones for later.

5.4 Search

The search algorithm is given in Algorithm 5. As the threshold and the heuristic estimates are real numbers it is slightly different from the usual IDA* algorithm that increments an integer threshold at each step. In our implementation the threshold is multiplied by a constant (1.2 gives good results) after each failure of the algorithm. It means that the algorithm will continue searching for better solutions even when it has found a solution below the threshold. This enables it to find the solution with the smallest evaluation. At the start of the algorithm the threshold is set to the value of the overall admissible heuristic. Exponential deepening A* has already been investigated by (Sharon *et al.*, 2014).

In the beginning of Algorithm 5 a test to check if the contiguity constraint is verified is done. If a region is not contiguous the algorithm returns failure (i.e. false). If the region is enclosed and has only one department the algorithm also fails. Then the usual test of IDA* that fails if the admissible heuristic is greater than the threshold is performed. Then the most constrained department is chosen. If all departments are assigned, the map is evaluated and memorized, else the different possible assignments of the department are tried.

6 Experimental Results

We have run the algorithm for different combinations of weights and the results are given in Table 1. The fail first heuristic is activated or not as indicated in the fail column. We see that computing the optimal assignment using only the distances to the capital is easily solved by the algorithm. Without the fail first heuristic it solved the problem faster but in more nodes.

We can also see that the fail-first heuristic can be very useful in some cases such as $C_d = C_m = 0.5$.

Figure 2 gives an example of an assignment found by the algorithm and Figure 3 gives an assignment where departments that can change regions according to the different weights used for the ingredients are marked with a point.

7 Conclusion

We have presented admissible heuristics for four economic, political and geographical measurements that are used to optimize the assignment of departments to regional capitals in France.

An original search algorithm finding exact solutions to the problem has been described. It optimizes real valued objectives and uses effective search heuristics.

Many simulations for different combinations of regional capitals have been performed with the algorithm. It has enabled to draw conclusions on the frequently occurring configurations of regions and on the assignment of border departments. For example a rule that has been observed is that if Montpellier is not a regional capital then Gard (30) and Herault (34) are assigned to Marseille.

Helping political deciders with AI is a new, exciting and promising application of heuristic search. More generally there are multiple possible uses of AI in economy and social sciences. Economic problems

Algorithm 5 The search algorithm

```
boolean search ()
moves ← possible moves
for s in strings do
  if number of liberties of s = 0 then
    if number of strings for region of s > 1 then
      return false
    end if
    if number of departments for region of s = 1 then
      return false
    end if
  end if
end for
if heuristic() > threshold then
  return false
end if
dept ← deptFailFirst()
if all departments are assigned then
  eval ← heuristic()
  if eval < threshold then
    threshold ← eval
    memorize state
  end if
  return true
end if
found ← false
for cap in capitals do
  assign dept to cap
  if search () then
    found ← true
  end if
  unassign dept
end for
return found
```

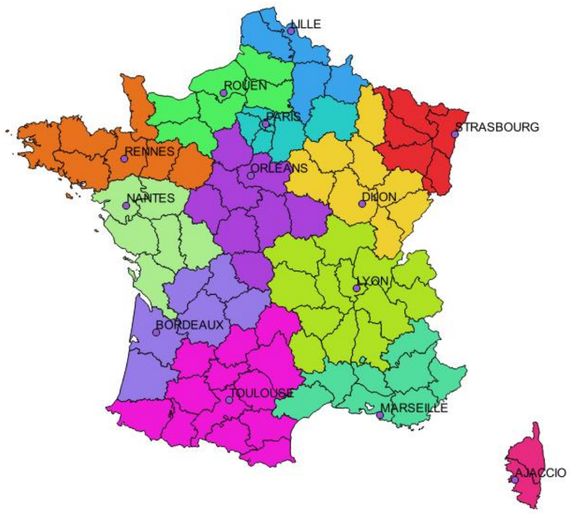

Figure 2: Example of an assignment found by the algorithm

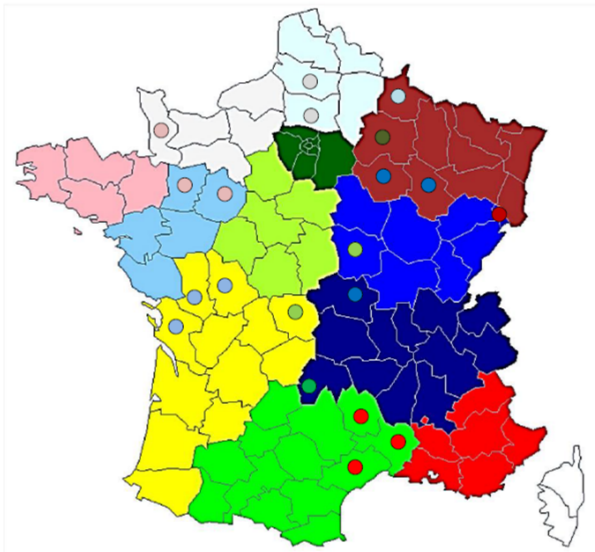

Figure 3: A possible assignment with points on departments that can change of region

can often be described with a loss function to optimize and many possible choices to search. Designing admissible heuristics to solve these problems exactly and using the associated search algorithm to optimize the choices will be addressed in future work.

References

AMABILE A., BERNARD C. & EPAULARD A. (2015). Une évaluation de la cohérence économique interne des régions. *France Stratégie, Document de travail*.

BACCHUS F. & VAN RUN P. (1995). Dynamic variable ordering in CSPs. In *Principles and Practice of Constraint ProgrammingCP'95*, p. 258–275: Springer.

BOUZY B. & CAZENAVE T. (2001). Computer Go: an AI oriented survey. *Artificial Intelligence*, **132**(1), 39–103.

GENT I. P., MACINTYRE E., PRESSER P., SMITH B. M. & WALSH T. (1996). An empirical study of dynamic variable ordering heuristics for the constraint satisfaction problem. In *Principles and Practice of Constraint ProgrammingCP96*, p. 179–193: Springer.

KORF R. E. (1985). Iterative-deepening-A*: An optimal admissible tree search. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, August 1985*, p. 1034–1036.

MÜLLER M. (2002). Computer go. *Artificial Intelligence*, **134**(1), 145–179.

OATES W. E. (1999). An essay on fiscal federalism. *Journal of Economic Literature*, **37**, 1120–1149.

RODRIK D. (1998). Why do more open economies have bigger governments? *Journal of Political economy*, **106**, 997–1032.

ROSE A. K. (2006). Size really doesn't matter: In search of a national scale effect. *Journal of the Japanese and International Economies*, **20**(4), 482–507.

SADEH N. & FOX M. S. (1996). Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem. *Artificial intelligence*, **86**(1), 1–41.

SHARON G., FELNER A. & STURTEVANT N. R. (2014). Exponential deepening a* for real-time agent-centered search. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, p. 871–877.