Assignment 1: Facility Location *

Emanuel Tomé †, Joana Almeida ‡

University of Porto, 1^{st} of April 2021

1 Introduction

The objective of the present assignment is to determine ten different cities in continental Portugal in which is it optimal to store ten units consisting of extremely heavy firefighting equipment. Constrained, first, by a lower bound regarding the population of each potential city where to allocate the units, and second, also by a minimum distance to abide by between any two selected cities. Therefore, the problem we are hereby considering is the *uncapacitated facility location problem*, of a static nature, hereinafter denoted as UFLP.

In general, *facility location problems*, popularised by Alfred Weber in 1909 [1], can be of a variety of types, but mainly consist of a classical optimisation problem for determining the best locations for a set of facilities, as well as the supply flows of a product/service from such facilities to the interested parties, at minimum activation cost plus transportation cost/distance. Furthermore, *uncapacitated* ones also presuppose the absence of natural restrictions to the quantity that can be supplied by each facility, or, in this case, implying that the firefighting equipment has virtually no constraints to the number of times it can serve its use.

Adding to the above, when firefighting is in concern, to be able to reach every city from a determined location as quickly as possible is imperative. In other words, the objective of the optimisation problem in question ought to be the minimisation of the distance from a facility to the farthest city to be serviced by it. Such objective is hereby attained through the use of the algorithms k-Center [2, 3] and its variant, k-Cover with binary search [4, 5]. Additionally, it was used a powerful modern solver for integer optimisation, Gurobi.

2 Question 1: Mathematical optimisation model

The UFLP herein can be stated as follows: given a set of N cities, find the subset of locations to receive the 10 (k) available units of heavy firefighting equipment, knowing that only cities with at least 10000 inhabitants (M) are candidates to receive the units, due to the logistics of maintaining the equipment. The subset of cities should be chosen in a way so that each city is reached as quickly as possible (i.e., the maximum distance between a city with the equipment and a city without the equipment should be minimised). The distance between cities are approximated by the L1 norm (also called *Manhantan distance*), which can be computed from:

$$d_{ij} = \frac{2\pi R(|lat_i - lat_j| + |lng_i - lng_j|)}{360}$$

where lat_i and lat_j are the latitude of cities i and j, respectively, lng_i and lng_j are the longitude of cities i and j, respectively, and R the earth radius in kilometers. The **problem data** is synthesised in Table 1.

To solve the UFLP, two approaches can be followed: the k-Center and the k-Cover with binary search. Both of them have been implemented, and the obtained results are detailed described in the next sections.

2.1 k-Center approach

In the k-Center approach, the objective function to be minimised is z, which represents the maximum distance of any selected location to receive the units to the farthest city it serves. This optimisation problem has two binary variables: x_{ij} is 1 when the demand of city i is met by city j and 0 otherwise; and y_j is 1 if the city j has the heavy firefighting equipment, that is, if it is one of the k chosen cities to receive the units, or 0 otherwise. Therefore, the objective z have to be greater or equal to the maximum distance between the cities j equipped

*This work was submitted on the framework of the course Data-Driven Decision Making

†Emanuel Tomé is a student at the Faculty of Sciences of the University of Porto. Currently, he is enrolled in the 1^{st} year of the Master's in Data Science (e-mail: up200702634@edu.fc.up.pt).

‡Joana Almeida is a student at the Faculty of Economics of the University of Porto. Currently, she is enrolled in the 1^{st} year of the Master's in Finance (e-mail: up201901225@edu.fep.up.pt).

Table 1: Problem data.

N		Set of cities
lat_i	$i \in N$	Cities' latitude
lng_i	$i \in N$	Cities' longitude
pop_i	$i \in N$	Cities' population
R		Earth radius (km)
$d_{ij} = \frac{2\pi R(\lvert lat_i - lat_j \rvert + \lvert lng_i - lng_j \rvert)}{360}$	$i, j \in N$	Distance from city i to city j (km)
$k = 10$		Number of available units
$M = 10000$		Minimum Population

with a unit to the cities i that they respectively serve (in other words, $\sum_{j=1}^{N} d_{ij}x_{ij} \leq z$, for $i = 1, ..., N$). Moreover, each and every city $i \in N$ ought to be served, but only by one, and one only activated location $j \in N$ (in other words, $\sum_{j=1}^{N} x_{ij} = 1$). Additionally, the subset of total activated locations must be equal to the k equipment units to be allocated ($\sum_{j=1}^{N} y_j = k$). As such, a city j serves any city i only if it belongs to the subset of activated locations (in other words, $x_{ij} \leq y_j$, for $i = 1, ..., N;\ j = 1, ..., N$). Adding to this, as previously stated, each location can only be activated if its population P_j is equal or higher than M.

Thus, to summarise, using the k-Center method the variables and formulation are as follows:

Variables:

x_{ij}	$i \in N, j \in N$	1 when the demand of city i is met by city j, 0 otherwise
y_j	$j \in N$	1 if the city j has the heavy equipment for firefighting, 0 otherwise

Formulation:

$$\text{minimise: } z \tag{1}$$

$$\text{subject to: } \sum_{j=1}^{N} x_{ij} = 1 \qquad \text{for } i = 1, ..., N \tag{2}$$

$$\sum_{j=1}^{N} y_j = k \tag{3}$$

$$x_{ij} \leq y_j \qquad \text{for } i = 1, ..., N;\ j = 1, ..., N \tag{4}$$

$$\sum_{j=1}^{N} d_{ij}x_{ij} \leq z \qquad \text{for } i = 1, ..., N \tag{5}$$

$$P_j \geq M \times y_i \qquad \text{for } j = 1, ..., N \tag{6}$$

$$x_{ij} \in \{0, 1\} \qquad \text{for } i = 1, ..., N;\ j = 1, ..., N \tag{7}$$

$$y_j \in \{0, 1\} \qquad \text{for } j = 1, ..., N \tag{8}$$

The AMPL source code for the k-Center approach can be found in section A.1 of the Appendix A. The setback of this approach being the min-max objective function, a problem for which mathematical solvers can become weak in the case of large-sized problems. Should it be deemed necessary to avoid the min-max objective, a variant of the k-Center approach can be used, which formulation will be addressed in the next section.

2.2 k-Cover approach

As already stated, the same problem can also be tackled using a different formulation and a binary search process, through the k-Cover approach. This is a variant of the k-Center which allows to avoid the min-max objective [6].

Considering a graph $G_\theta = (V, E_\theta)$ constituted by a set of edges, whose distances from a city to a selected location does not exceed the threshold value θ (i.e., edges $E_\theta = \{\{i, j\} \in E : d_{ij} \leq \theta\}$), and given a subset $S \subset V$ of the vertex set, S is called a *cover* if every vertex $i \in V$ is adjacent to at least one of the vertices in S [6]. The optimum value of the k-Cover problem is less than or equal to θ if there is a cover with cardinality $|S| = k$ on graph G_θ.

This method considers two binary variables: y_j that is 1 if the facility j is open and 0 otherwise; and z_i that is 1 if the vertex i is adjacent to no vertex in S and 0 otherwise. Denoting $[a_{ij}]$ as the incidence matrix of G_θ, whose element a_{ij} is equal to 1 if vertices i and j are adjacent and 0 otherwise (in other words, $a_{ij} = (d_{ij} <= \theta)$). To

determine whether the graph G_θ has a cover $|S| = k$, we should minimise the sum of all z_i ($\sum_{i=1}^{N} z_i$) subjected to the restrictions $\sum_{j=1}^{N} a_{ij} \times y_j + z_i \geq 1$ for $i = 1, ..., N$, $\sum_{j=1}^{N} y_j = k$ and $P_j \geq M \times y_i$ for $j = 1, ..., N$.

For a given value of θ there are two possibilities: the optimal objective value is zero or greater than zero. If the objective is zero, all cities i are covered within a distance θ by the k locations with the equipment units. In that case, one attempts to reduce θ and check if all cities remain covered. If the objective is greater than zero, at least one city is farther than θ to a location with an unit of the equipment for firefighting. In that case, one should increase θ. This process is repeated until the bound for θ are close enough, in a process called *binary search* [6].

To summarise, using the k-Cover method the variables and formulation are as follows:

Variables:

y_j	$j \in N$	1 if the facility j is open, 0 otherwise
z_i	$i \in N$	1 if vertex i is adjacent to no vertex in S, 0 otherwise.

Formulation:

$$\text{minimise:} \quad \sum_{i=1}^{N} z_i \tag{9}$$

$$\text{subject to:} \quad \sum_{j=1}^{N} a_{ij} \times y_j + z_i \geq 1 \qquad \text{for } i = 1, ..., N \tag{10}$$

$$\sum_{j=1}^{N} y_j = k \tag{11}$$

$$P_j \geq M \times y_i \qquad \text{for } j = 1, ..., N \tag{12}$$

$$y_j \in \{0, 1\} \qquad \text{for } j = 1, ..., N \tag{13}$$

$$z_i \in \{0, 1\} \qquad \text{for } i = 1, ..., N \tag{14}$$

The AMPL and Python source code for this approach can be found in section A.2 of the Appendix A.

3 Question 2: Solution of the problem

In Table 2 the solutions obtained through the k-Center and the k-Cover methods are presented. Slightly different solutions were obtained using both approaches, although the maximum distance between a city with a equipment and a served city is about 95.05623 km for both (see Table 2). This means that at least two different solutions are possible for this problem. It can also be concluded from Table 2 that the total number of cities that each city equipped with the firefighting units serves is similar for both solutions, and that the city furthest from a city equipped with a firefighting unit is in both cases connected to the city of Trofa. Regarding the set of locations of the optimal solutions, 7 out of 10 locations are the same in both approaches. It should be mentioned also that the number of locations that each unit serves is not balanced. For instance, while Trofa serves 142 locations, Bragança just serves 6 locations. However, this is due to the fact that Trofa and Moita (or Pinhal Novo in the k-Cover solution) are located in the two regions of Portugal Continental with higher population density (and because of that, those regions are more represented in the set of cities).

Table 2: Number of cities served and maximum distance for each city with the equipment.

k-Center			k-Cover		
Facility	No. served cities	Max dist (km)	Facility	No. served cities	Max dist (km)
Beja	7	82.119793	Beja	8	82.119793
Bragança	6	88.178813	Bragança	6	88.178813
Trofa	142	95.056229	Olhão	10	65.587308
Quarteira	19	89.903449	Pinhal Novo	94	92.641072
Moita	91	86.564261	Trofa	142	95.056229
Portalegre	10	84.611675	Lagos	10	60.263288
Lousã	28	89.004993	Portalegre	10	84.611675
Santo André 2	7	84.797371	Lousã	28	89.004993
Cartaxo	43	94.982841	Cartaxo	45	94.982841
Moimenta da Beira	26	91.391239	Moimenta da Beira	26	91.391239

The representation of the obtained solutions can be seen in Figure 1, Appendix B. However, it should be noticed that the edges presented in the maps were recomputed after the optimal solutions were found. Namely, for the k-Center approach it was found that not every city was being serviced by the closest activated location, although the distance was smaller than the objective of 95.056229km (see Figure 2 (a), Appendix B). Regarding the k-Cover, the solution provided at the end of the binary search allowed several cities to be serviced by more than one selected location (see Figure 2 (b), Appendix B). This happens because those cities are located in a distance smaller that the objective (θ) to more than one location with the equipment.

The computation time of the k-Center approach (in a Virtual Machine with an Intel Core i7-8565U CPU @ 1.80GHz, 14.6GB of RAM running Ubuntu 20.04.2 LTS) was about 72.4 seconds (median value of five runs). On the other hand, for the k-Cover method, the running time was about 3.36 seconds (median value of five runs in the same virtual machine and for a tolerance $\delta = 1 \times 10^{-6}$). This means that, as expected, the k-Cover approach solves the same problem more than 21.5 times faster than the k-Center.

4 Question 3: New constrain

The additional constraint imposes that the distance between any two locations with the equipment must be of 120 km or more, which can be written as:

$$\text{new constrain: } d_{ij} \geq 120 \times y_i \times y_j \qquad \text{for } i = 1, ..., N, j = 1, ..., N \text{ and } i \neq j \tag{15}$$

which can be translated in AMPL as follows:

```
newrestr {i in N, j in N: i<>j}: d[i,j]>=120*y[i]*y[j];
```

However, this is a nonlinear equation which increases the computation time. A linear alternative is:

$$\text{new constrain: } d_{ij} \geq 120 \times y_j \quad if\ y_i = 1 \qquad \text{for } i = 1, ..., N, j = 1, ..., N \text{ and } i \neq j \tag{16}$$

which can be translated in AMPL as follows:

```
newrestr {i in N, j in N: i<>j}: y[i] == 1 ==> d[i,j]>=120*y[j];
```

Once again, different solutions were obtained for the k-Center and the k-Cover methods, although for both a maximum distance of about 97.225645 km (see Table 3) was reached. Therefore, the optimum maximum distance increased by 2.169415 km when compared to the solution obtained in section 3, meaning that previously there were locations with the equipment distanced below the new bound of 120 km. Regarding the set of locations of the optimal solutions, 8 out of 10 locations are the same in both approaches. In Figure 3 (Appendix B) are represented in the map of Portugal the obtained solutions for both approaches.

Table 3: Number of cities served and maximum distance for each city with the equipment.

k-Center			k-Cover		
Facility	No. served cities	Max distance (km)	Facility	No. served cities	Max distance (km)
Monsanto	37	97.225645	Monsanto	38	97.225645
Portimão	13	62.423808	Portimão	13	62.423808
Beja	10	96.254912	Beja	10	96.254912
Bragança	6	88.178813	Bragança	6	88.178813
Trofa	142	95.056229	Bougado	142	96.201539
Moita	103	92.774506	Portalegre	10	84.611675
Portalegre	10	84.611675	V. R. Sto. António	7	76.746847
V. R. Sto. António	7	76.746847	Lousã	25	89.004993
Lousã	25	89.004993	Sto. A. Charneca	102	93.224846
Moimenta da Beira	26	91.391239	Moimenta da Beira	26	91.391239

The computation time of the k-Center approach (in a Virtual Machine with an Intel Core i7-8565U CPU @ 1.80GHz, 14.6GB of RAM running Ubuntu 20.04.2 LTS) was about 102.0 seconds (median value of five runs). For the k-Cover approach, the running time was about 41.4 seconds (median value of five runs in the same virtual machine and for a tolerance $\delta = 1 \times 10^{-7}$). This means that, once again, the k-Cover approach solves the same problem more than 2.46 times faster than the k-Center.

5 Conclusions

The *uncapacitated facility location problem* - in this case, given by the need to find a given set (k=10) of locations in which it is optimal to store 10 units of heavy firefighting equipment, in order to serve every city in

Continental Portugal - for which the total time criterion - needed to take the equipment to the cities it is meant to serve - is, obviously, of superior importance, and thus the maximum distance between any location and the farthest city should be minimised, can be solved through the use of the k-Center method or its variant k-Cover with binary search.

Theoretically, both algorithms must provide the same optimum objective solution, but several different optimal sets of selected locations for that same optimum are allowed. This is consistent with the obtained results, which was an optimum maximum distance between a selected location and the farthest city it serves of 95.05623 km for the first problem and 97.225645 km for the second. Regarding the set of optimal solutions, in the first problem 7 out of 10 locations were the same in both approaches, and in the second problem were 8 out of 10, each of which with very similar number of served cities.

When another constraint was added to the problem, imposing a minimum distance between any two locations with the equipment must be 120 km or more, the optimum maximum distance increased by 2.169415 km, meaning that previously there were locations with the equipment distanced below the new bound of 120 km.

One problem found using the k-Center approach, spotted when creating the figures on Appendix B, was that not every city was being serviced by the closest activated location. To solve this, the edges were recomputed before plotting them in the map of Portugal.

On the other hand, a problem with the edges was also found using the k-Cover: the solution provided at the end of the binary search allowed several cities to be serviced by more than one selected location. This happens because those cities are located in a distance smaller that the objective (θ) to more than one location with the equipment. For this reason, the edges were also recomputed in order to get the representations in Appendix B.

Lastly, it is important to mention that the k-Cover with binary search allows for a better relative computational time than that required by the k-Center. This happens because the k-Cover approach has a formulation that avoids the min-max objective. For the problem herein solved, the k-Cover ran 21.5 times faster for the first problem and 2.46 times faster for the second problem. This, in the real world and with larger sized problems, plays a crucial role when deciding between both methods.

References

[1] Vasco Mota. "Solving the uncapacitated facility location problem using fusing moves and its application in computer vision problems". PhD thesis. Universidade de Coimbra, 2015.

[2] S. L. Hakimi. "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph". In: *Operations Research* 12.3 (1964), pp. 450–459. ISSN: 0030364X, 15265463. URL: http://www.jstor.org/stable/168125.

[3] S. L. Hakimi. "Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems". In: *Operations Research* 13.3 (1965), pp. 462–475. DOI: 10.1287/opre.13.3.462. eprint: https://doi.org/10.1287/opre.13.3.462. URL: https://doi.org/10.1287/opre.13.3.462.

[4] G.Y. Handler and P.B. Mirchandani. *Location on Networks: Theory and Algorithms*. MIT Press series in signal processing, optimization, and control. MIT Press, 1979. ISBN: 9780262080903. URL: https://books.google.pt/books?id=3JxgAAAAMAAJ.

[5] Zvi Drezner. "The p-cover problem". In: *European Journal of Operational Research* 26.2 (1986), pp. 312–313. ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(86)90196-7. URL: https://www.sciencedirect.com/science/article/pii/0377221786901967.

[6] João Pedro Pedroso et al. *Mathematical Optimization Documentation - Release 1*. 2019.

Appendixes

A Source code

A.1 k-Center

AMPL model:

```
set N;   # set of cities
param lat{N};   # cities' latitude
param lng{N};   # cities' longitude
param pop{N};   # cities' population
param k :=10;  # Number of available units
param M := 10; # Minimum population (in thousands of people)
```

```

data assignment01.dat;

# constants
param R := 6371.009;   # earth radius
param pop_thou{i in N} := floor(pop[i] / 1000);
param pi := 3.14159265359;

# distance
param d {i in N, j in N} := 2 * pi * R * (abs(lat[i]-lat[j]) + abs(lng[i]-lng[j]))/360;

# main variables
var x {N, N} binary;
var y {N} binary;
var z >= 0;

minimize maxdist: z;

subject to
Serve {i in N}: sum {j in N} x[i,j] = 1;
Kfacil: sum {j in N} y[j] = k;
Activate {i in N, j in N}: x[i,j] <= y[j];
MinZ {i in N}: sum {j in N} d[i,j] * x[i,j] <= z;
minpop {j in N}: pop_thou[j] >= M * y[j];

solve;
var time:= _solve_elapsed_time;

printf "\nElapsed seconds for most recent solve command %f\n\n", time;
printf "Chosen cities and their population:\n";
printf {j in N: y[j] == 1}: "%s %g\n",  j, pop[j];
```

A.2 k-Cover

AMPL model:

```
set N;   # set of cities
param lat{N};   # cities' latitude
param lng{N};   # cities' longitude
param pop{N};   # cities' population
param k :=10;  # Number of available units
param M := 10; # Minimum population (in thousands of people)

param theta;

data assignment01.dat;

# constants
param R := 6371.009;   # earth radius
param pop_thou{i in N} := floor(pop[i] / 1000);
param pi := 3.14159265359;

# distance
param d {i in N, j in N} := 2 * pi * R * (abs(lat[i]-lat[j]) + abs(lng[i]-lng[j]))/360;

# main variables
var y {N} binary;
var z {N} binary;


minimize cost: sum {i in N} z[i];

subject to
Serve {i in N}: sum {j in N: d[i,j] <= theta} y[j] + z[i] >= 1;
Kfacil: sum {j in N} y[j] = k;
minpop {j in N}: pop_thou[j] >= M * y[j];
```

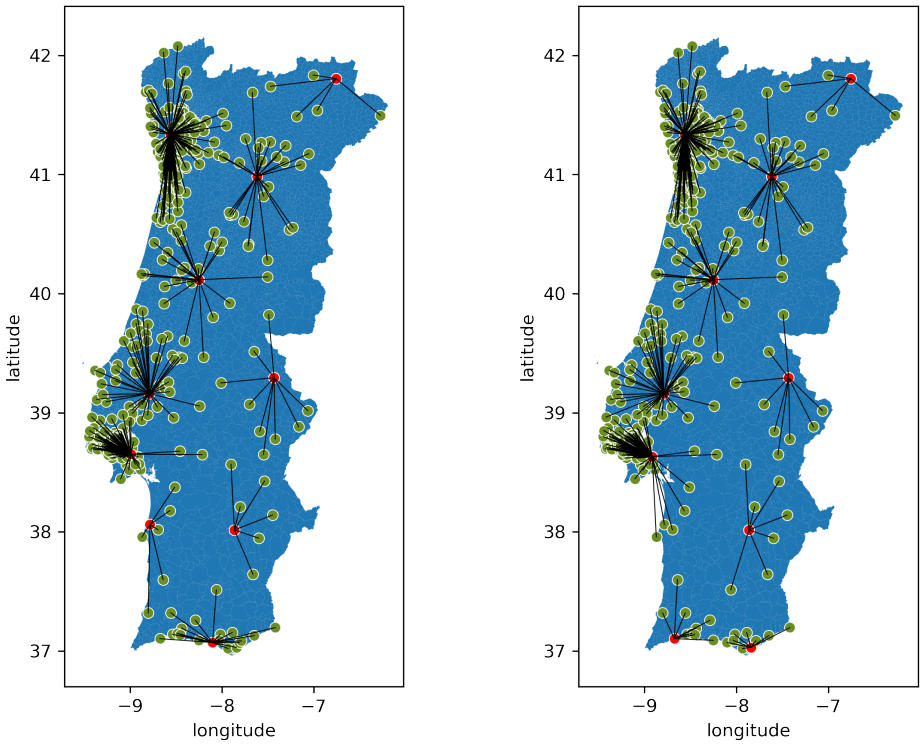
Python code for binary search:

```
# Set AMPL location and solver
ampl2 = AMPL()
ampl2.option['solver'] ='gurobi'

# Read the model
print("[INFO] Reading model...")
ampl2.read("assignment01_kcover.mod")
```

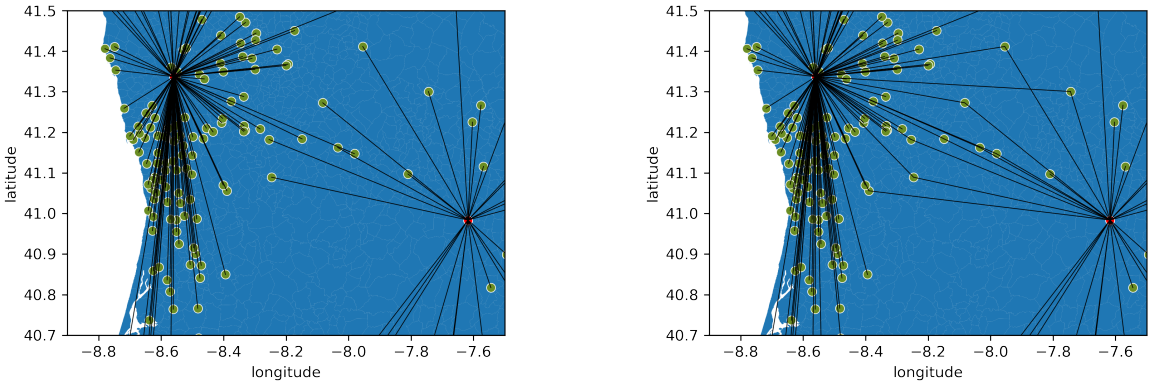
```
8
9  # data from AMPL model
10 print("[INFO] Reading data from the model...")
11 N = ampl2.getSet('N').getValues().toPandas()
12 d = ampl2.getParameter('d').getValues().toPandas()
13
14
15 print("[INFO] Seting tolerance, LB and UB...")
16 delta = 1.e-6 # tolerance
17 LB = 0
18 UB = max(d['d'])#max(d[i,j] for (i,j) in d)
19
20 # Solving
21 print("[INFO] Solving....\n")
22 start_time = time.time()
23 while UB-LB > delta:
24     theta = (UB+LB) / 2
25     ampl2.param['theta'] = theta
26     ampl2.solve()
27     cost = ampl2.obj['cost']#.getObjective('cost')#.obj['cost']
28     if cost.value() < delta:
29         UB = theta
30     else: # infeasibility > 0:
31         LB = theta
32
33 end_time = time.time()
34
35 print('\n[INFO] Execution time: {:.3f} seconds'.format(end_time-start_time))
36 print('[INFO] Objective is equal to {:.0f}'.format(cost.value()))
37
38 # Facilities
39 y_ = ampl2.var['y']
40 facilities = [j for j in N.index if y_[j].value()>0.5]
41
42 # Facilities and population
43 pop_ = ampl2.getParameter('pop')
44 fac_population = [(i,pop_[i]) for i in facilities]
45 fac_population
```

B Maps of Portugal with the obtained solutions



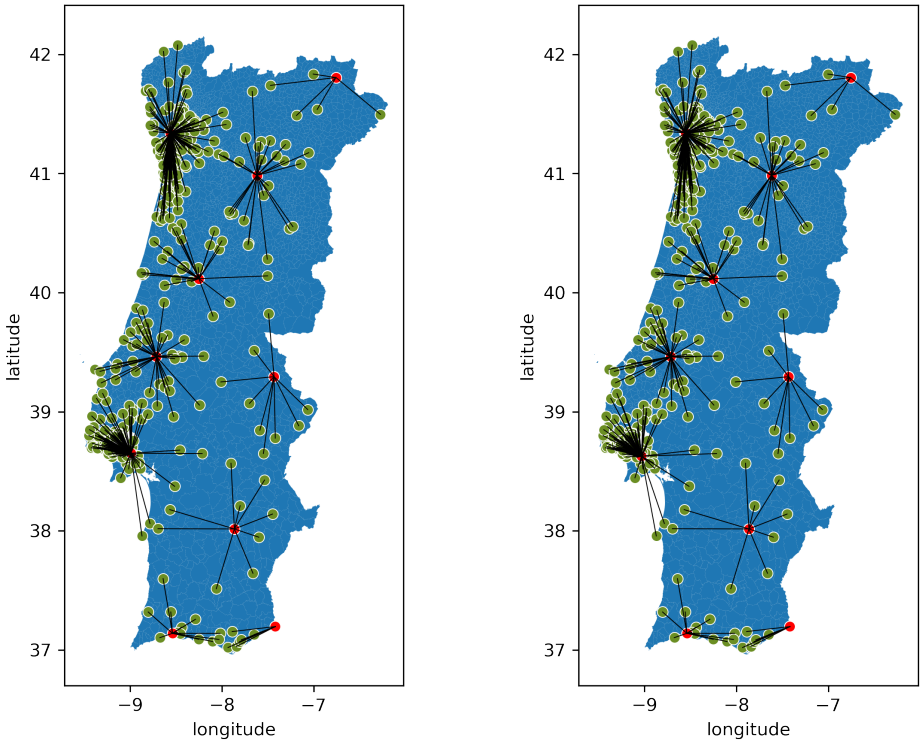
(a) k-Center (b) k-Cover

Figure 1: Solution obtained for the initial problem.



(a) k-Center (b) k-Cover

Figure 2: Detail of the obtained edges from AMPL.



(a) k-Center (b) k-Cover

Figure 3: Solution obtained with the restriction from Question 3.