APPLICATIONS OF COMBINATORIAL OPTIMIZATION ARISING FROM LARGE SCALE SURVEYS

CANDIDATE: ALESSANDRA REALE

SUPERVISOR: RENATO BRUNI

Dissertation submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
in

OPERATIONS RESEARCH

"SAPIENZA" UNIVERSITY OF ROME

Contents

Acknowledgments

I would like to express my gratitude to my supervisor, Renato Bruni, whose expertise, constant support, guidance and encouragement, added considerable value to my PhD experience. His competence and skill in Operations Research and related fields helped me in solving difficult mathematical questions arising from important practical statistical problems, allowing me to treat very large datasets and to produce a number of research papers.

I am very grateful to Fabio Sforzi, Professor of Applied Economics at the University of Parma, for his relevant comments on this thesis. His suggestions and many motivating discussions improved my knowledge in the field of Regionalization techniques.

I also express my deepest gratitude to Antonio Sassano, Professor of Operations Research at the DIAG department of the University of Rome ”Sapienza”, for having stimulated my interest in Operations Research. His support and encouragement allowed the realization of the research collaborations between the Italian National Institute of Statistics (Istat) and the University of Rome Sapienza. I thanks as well the other faculty members of the DIAG Department of the University of Rome Sapienza”, for their interest in my research.

I am grateful to my colleagues at Istat for their contribution to the development of Data Editing and Imputation system, in particular to Gianpiero Bianchi, Rosa Maria Lipsi, Giuseppina Ruocco and to the others of the MTO-E operating unit, that was charged to perform the heavy computation needed for the treatment of the Italian Census data.

During my twenty-year career at Istat, I had the opportunity to solve efficiently and accurately many statistical problems by using Operations Research techniques in the research activities that I have directed, and I would like to keep using these techniques for solving further statistical problems.

Preface

Many difficult statistical problems arising in censuses or in other large scale surveys have an underlying Combinatorial Optimization structure and can be solved with Combinatorial Optimization techniques. These techniques are often more efficient than the ad hoc solution techniques already developed in the field of Statistics. This thesis considers in detail two relevant cases of such statistical problems, and proposes solution approaches based on Combinatorial Optimization and Graph Theory. The first problem is the delineation of Functional Regions, the second one concerns the selection of the scope of a large survey, as briefly described below. The purpose of this work is therefore the innovative application of known techniques to very important and economically relevant practical problems that the "Censuses, Administrative and Statistical Registers Department" (DICA) of the Italian National Institute of Statistics (Istat), where I am senior researcher, has been dealing with.

In several economical, statistical and geographical applications, a territory must be partitioned into Functional Regions. This operation is called Functional Regionalization. Functional Regions are areas that typically exceed administrative boundaries, and they are of interest for the evaluation of the social and economical phenomena under analysis. Functional Regions are not fixed and politically delimited, but are determined only by the interactions among all the localities of a territory. In this thesis, we focus on interactions represented by the daily journey-to-work flows between localities in which people live and/or work. Functional Regionalization of a territory often turns out to be computationally difficult, because of the size (that is, the number of localities constituting the territory under study) and the nature of the journey-to-work matrix (that is, the sparsity). In this thesis, we propose an innovative approach to Functional Regionalization based on the solution of graph partition problems over an undirected graph called transitions graph, which is generated by using the journey-to-work data. In this approach, the problem is solved by recursively partitioning the transition graph by using the min cut algorithms proposed by Stoer and Wagner and Brinkmeier. This approach is applied to the determination of the Functional Regions for the Italian administrative regions.

The target population of a statistical survey, also called scope, is the set of statistical units that should be surveyed. In the case of some large surveys or censuses, the scope cannot be the set of all available units, but it must be selected from this set. Surveying each unit has a cost and brings a different portion of the whole information. In this thesis, we focus on the case of Agricultural Census. In this case, the units are farms, and we want to determine a subset of units producing the minimum total cost and safeguarding at least a certain portion of the total information, according to the coverage levels assigned by the European regulations. Uncertainty aspects also occur, because the portion of information corresponding to each unit is not perfectly known before surveying it. The basic decision aspect is to establish the inclusion criteria before surveying each unit. We propose here to solve the described problem using multidimensional binary knapsack models.

The thesis is organized as follows:

Chapter 1 resumes the main results of the research collaborations held between Istat and the Department of Computer, Control and Management Engineering (DIAG) of the Sapienza University of Rome. The subject of those collaborations was the use of Combinatorial Optimization techniques for the solution of relevant problems arising from statistical surveys.

Chapter 2 describes in detail the approach developed for the solution of problems of Functional Regionalization. We provide results on the real-world instances generated with the data of the 2001 Italian Population Census that show the effectiveness of the proposed approaches. This work is also described in [13].

Chapter 3 describes the knapsack models developed for the solution of problems of statistical units selection. We provide results on real-world instances arising from the 2010 Italian Agricultural Census that show the effectiveness of the proposed approach. This work is also described in [12].

Clearly, the techniques described in Chapters 2 and 3 by referring to Census data can be used to solve other problems of different origin but sharing the same logical characteristics. In particular, we can use the same techniques proposed in Chapter 2 to define functional regions when data refer to many types of commuting patterns, for example in the study of health service usage or for the analysis of educational workflows. The selection method proposed in Chapter 3 can also be used in the design of several sample surveys, in order to increase data quality.

Finally, Appendix A provides a brief introduction to some Combinatorial Optimization aspects and describes the main features of some basic solution techniques. The discussion is not intended to be exhaustive of the matter, but only to clarify which mathematical instruments have been used to design the approaches described in Chapters 2 and 3.

Roma, Italy, October 2015. ALESSANDRA REALE

Chapter 1

Combinatorial Optimization in Surveys

1.1 The Role and the Importance of Operations Research in Statistical Data Processing

In recent years, Operations Research techniques increased their already important role in solving statistical problems, especially in the handling of large data set, as it is the case of Census data. Their usefulness is twofold. On the one hand, more advanced optimization models are able to better fit the specific characteristics of the real problems, that often have peculiarities and traits different from the standard examples. On the other hand, faster algorithms allow to drastically reduce the computational time without loss of accuracy, thus allowing (along with the developments of computing hardware) the treatment of dataset so large as never before.

Censuses are very complex, important and expensive tasks for a National Statistic Office. They are total survey, periodically performed, for monitoring the socio-demographic, economic and agricultural topics. As in any other large-scale surveys, however, the gathered data may contain errors or missing values, due to many reasons. Nonetheless, the correct information must be discovered and published. Therefore, error detection and correction become crucial tasks. This kind of activity is generally called Data Editing and Imputation, Information Reconstruction, or also Data Cleaning.

Several problems of Data Editing and Imputation have been solved using operations research techniques. These approaches have been studied and applied to census data. During the last two decades, several models based on Integer Linear

Programming for editing and imputation problems were developed during the research collaborations between Istat and the Department of Computer, Control and Management Engineering (DIAG), Sapienza University of Rome. These results have been discussed in international conferences and published in international journals. Some of them are briefly shown in this section. The formalized research collaborations between the two organizations mentioned above have been the following:

1. Research agreement on “Modelli e algoritmi per problemi di edit and imputation” (Rep. N. 37/2001);

2. Research agreement on “Modelli e metodi per problemi di linkage e clustering di dati” (Rep. N. 30/2003);

3. Research agreement on “Individuazione di solutori open source per problemi di Programmazione Lineare Intera” (Rep. N. 45/2009);

4. Research agreement on “Nuove metodologie per il controllo a livello micro-macro dei dati e per limputazione di dati quantitativi” (Rep. N. 132/2010).

The first collaboration, held during the 14th Population and Housing Census 2001, focused on the problem of data completeness and consistency. An optimization approach was developed to handle erroneous data records and to reconstruct the corrupted information in order to obtain correct data records. This approach is based on the solution of Integer Linear Models encoding the objective of introducing minimum alterations in the erroneous data, but with constraints imposing that the erroneous records should be converted into correct records. The described approach is able to deal with both qualitative and quantitative variables, and overcomes the computational limits of the well-known and widely used Fellegi-Holt methodology ([49]), while maintaining its positive statistical feature.

A software system called DIESIS (Data Imputation and Editing System - Italian Software) was developed for the editing and imputation of hierarchical demographic data. DIESIS performs error localisation and data imputation of invalid or inconsistent responses in a general process of statistical data collection. The performance of DIESIS has been evaluated against the Canadian Nearest-neighbour Imputation Methodology (NIM) [4], by means of an evaluation study based on real data from the 1991 Italian Population Census, perturbed by introducing various amounts of artificial errors and missing values. The evaluation has been performed by computing, for each variable, accuracy indicators of preservation of individual original values as well as of preservation of the marginal distributions. DIESIS has also been used for the Editing and Imputation process for both 2001 and 2011

Italian Population Census and for the 2010 Agricultural Census. The results of this collaboration have been published in [21, 22, 89].

The second collaboration improved on the results of the first collaboration. Techniques were developed for solving data imputation problems by first stratifying the data by using nearest neighbors methods. Erroneous values are replaced by the correct values selected from statistical units named donors. A list of potential donors must be selected for each erroneous record. The donors selection is based on the use of similarity functions. For large data-sets, an accurate selection could be computationally prohibitive. Therefore, to reduce the computational burden, a clustering approach to define strata for donors selection has been applied. The donors were grouped into many subsets taking into account the similarities among the strata variables, by applying a spherical neighborhood algorithm. The results of this collaboration have been published in [8].

The third collaboration was based on the study of the performance of internationally recognized open source Integer Linear Programming solvers, compared to a reference commercial solver Cplex on real-world data having only numerical fields. The aim was to produce a stressing test environment for selecting the most appropriate open source solver for performing error localization in numerical data. The results of this collaboration have been published in [11].

The fourth collaboration lead to the development of new Editing and Imputation methods for the 2010 Agricultural Census and for the 2011 Italian Population Census. We specifically considered two kind of research problems:

- The use of micro and macro editing, that is the problem of detecting erroneous data records by considering, simultaneously, rules applied to a single record (micro edit) and rules applied to all records at the same time (macro edit). Macro edit are very useful in a number of applications, but their use generally makes the problem computationally intractable.

- The reconstruction of quantitative data that may have been corrupted, or also intentionally incorrectly answered, in cases that are particularly difficult, for example for numerical reasons.

For each of these two categories of problems, we briefly report, in the following subsections, two important practical examples and the optimization models developed for their solution during the described research collaboration. These results have been published in [9, 10].

1.1.1 Balancing problem in Agricultural Census data

When performing an Agricultural Census, data obtained from each *farm* contain *detail* information about the *cultivation area* used by that farm for each cultivation and the *number of livestock* for each type of animal. Those data may sometimes be erroneous or missing, due to a variety of reasons. In such cases, errors should be automatically detected and corrected, i.e. the information that was corrupted and lost should be "reconstructed" in order to be as similar as possible to the unknown exact value. Moreover, each farm also declares other information (called *macrodata*, information about totals): the total cultivation area and the total number of livestock, and in some cases those totals are also divided into *subtotals* by year of planting. Clearly, balancing conditions must hold between all the above microdata and the corresponding macrodata: each total (or year subtotal) must be equal to the sum of those details concerning its parts. When such conditions do not hold, data are inconsistent, and they should be changed in order to become consistent.

Records incurring in this problem are detected by checking the balancing conditions, which are called *balance edits*. However, when a balance edit is violated, the error could be either on the detail side or on the total side of the equation. The less *reliable* information should now be changed in order to restore consistency. It is generally assumed, in similar cases, that details constitute the less reliable information, since totals have already been confirmed from other sources. However, there are several ways to change the detail information in order to make it consistent, so this constitutes an optimization problem.

The models proposed for this problem will be hereinafter explained by referring to the specific case of *vineyards*. This is one of the most important cases: dozens of grapes varieties exist, and they determine type and quality of wines produced. However, the proposed models are clearly not limited to that case, but can be used for any other similar problem. Each farm could have several vine types, and each of them could have been planted in a different time period (e.g. a specific year). Denote by

I = $\{1,\ldots$, n$\}$ the set of indices of all possible vine types; with $n = 442$;

K = $\{1,\ldots$, m$\}$ the set of indices of all possible time periods; with $m = 6$.

For each farm, denote by

$a_{\mathtt{ik}}$ (real valued ≥ 0) the area of vine type i planted in period k declared by the farm, with $i \in I$ and $k \in K$;

$a_{\mathtt{i0}}$ (real valued ≥ 0) the total area of vine type i (planted during any of the periods) declared by the farm, with $i \in I$;

T_k (real valued ≥ 0) the total vine area planted in period k declared by the farm, with $k \in K$;

T (real valued ≥ 0) the total vine area owned by the farm.

In order to reconstruct the erroneous information, we need the following set of decision variables:

$x_{\mathtt{ik}}$ (real valued $\geq\ 0$, $\leq$ S) = the area of vine type i that, according to our reconstruction, has been planted in period k by the farm, with $i\ \in\ I$ and $k \in K$;

$x_{\mathtt{i0}}$ (real valued ≥ 0, $\leq$ mS) = the total area of vine type i that, according to our reconstruction, has been planted (during any of the periods) by the farm, with $i \in I$.

In other words, $x_{\mathtt{ik}}$ is the correct value for $a_{\mathtt{ik}}$. When reconstructing information for a Census, as in the case of other large-scaled surveys, it is generally assumed that the changes introduced in the data should be somehow minimized. This because, in absence of further information, being as similar as possible to the exact (unknown) data corresponds to being as similar as possible to the available (even if possibly erroneous) data. By following this minimum change paradigm, two basic alternatives exist: one is minimizing the number of changes, the other minimizing the amount of those changes.

If we need to distinguish when our reconstruction provides a result which is different form the available declaration (i.e. a change), we need the following set of binary variables:

$$y_{\mathtt{ik}} = \begin{cases} 1 & \text{if } \ x_{\mathtt{ik}} \text{ is different from } a_{\mathtt{ik}} \qquad \forall i = 1, \ldots, n \ \ \forall k = 0, \ldots, m \\ 0 & \text{otherwise} \end{cases}$$

The presence of binary variables clearly has its impact on the complexity of the model: by adding the other constraints needed for this problem, which are linear, we obtain an Integer Linear Program. Minimizing the total number of changes corresponds to the following objective function

$$\min \sum_{i=1}^{n} \sum_{k=1}^{m} y_{\mathtt{ij}} \tag{1.1}$$

When variables y are used, they should be linked to the x variables by constraints imposing that $y_{\mathtt{ik}}$ takes value 1 when $x_{\mathtt{ik}} <> a_{\mathtt{ik}}$ (using a certain numerical precision), otherwise those variables could be inconsistent. There is no need for constraints imposing $y_{\mathtt{ik}} = 0$ when $x_{\mathtt{ik}} = a_{\mathtt{ik}}$ because the objective (1.1) itself does that. Value *M* is a real number greater than all possible values of the left-hand side of the following inequalities.

$$a_{\mathtt{ik}} - x_{\mathtt{ik}} \leq My_{\mathtt{ik}} \quad \forall i = 1, \ldots, n \ \forall k = 0, \ldots, m$$

$$x_{\mathtt{ik}} - a_{\mathtt{ik}} \leq My_{\mathtt{ik}} \quad \forall i = 1, \ldots, n \ \forall k = 0, \ldots, m$$

When, on the other hand, we are interested in measuring the difference (distance) between our reconstruction $x_{\mathtt{ik}}$ and the available declaration $a_{\mathtt{ik}}$, we should consider a funcion of this difference. Several types of distance functions are available. We consider more suitable to our reconstruction problems the following three:

- The squared Euclidean distance, defined as $\sum_{i=1}^{n} \sum_{k=0}^{m} (x_{\mathtt{ik}} - a_{\mathtt{ik}})^2$;
- The Manhattan distance, defined as $\sum_{i=1}^{n} \sum_{k=0}^{m} |x_{\mathtt{ik}} - a_{\mathtt{ik}}|$;
- The Chebyshev distance, defined as $max_{\mathtt{ik}}\{|x_{\mathtt{ik}} - a_{\mathtt{ik}}|\}$.

Clearly, the structure of the optimization model that we must solve depends now on this choice. In particular, in the second case (Manhattan distance), there are absolute values in the objective. However, they can be easily linearized by introducing additional variables:

$$s_{\mathtt{ik}} \text{ (real valued} \geq 0) = the\, value\, of\ |a_{\mathtt{ik}} - x_{\mathtt{ik}}|, \quad \forall i = 1, \ldots, n \ \forall k = 0, \ldots, m$$

and linear constraints enforcing their meaning

$$s_{\mathtt{ik}} \geq a_{\mathtt{ik}} - x_{\mathtt{ik}}, \ \ s_{\mathtt{ik}} \geq x_{\mathtt{ik}} - a_{\mathtt{ik}} \quad \forall i = 1, \ldots, n \ \forall k = 0, \ldots, m$$

We can now minimize the linear function $\sum_{i=1}^{n} \sum_{k=0}^{m} s_{\mathtt{ik}}$. When adding the other constraints needed for this problem, which are linear, the problem becomes an easily solvable linear program.

In our case, however, we consider the following objective more representative of the real problem's aim the minimization of the total number of changes, and, in second place, the minimization of the amount of those changes. This because a change with respect to a value that has been deliberately declared has intrinsically a very high cost. Therefore, we prefer maintaining the maximum number of those declared values, even if this may result in a greater amount of the changes that we are forced to introduce. Consequently, the objective function becomes:

$$\min\ (M' \sum_{i=1}^{n} \sum_{k=0}^{m} y_{\mathtt{ik}} + \sum_{i=1}^{n} \sum_{k=0}^{m} s_{\mathtt{ik}})$$

where the first sums are multiplied by a numerical value M' weighting the relative importance of the first part with respect to the second one. We chose $M' = S$, so that a single change weights as much as the maximum amount of a change.
We now describe the balancing conditions that should be respected in our case. The sum of vine areas of any type planted in period k must be equal to the total vine area planted in period k (called balancing over vine types)

$$\sum_{i=1}^{n} x_{\mathtt{ik}} = T_k \quad \forall k \in K$$

The sum of the areas of vine type i planted in periods from 1 to m must be equal to the area of the same vine type planted along all the periods (called balancing over time periods)

$$x_{\mathtt{i0}} = \sum_{k=1}^{m} x_{\mathtt{ik}} \quad \forall i \in I$$

The sum of vine areas of any type planted in any period must be equal to the total vine area owned by the farm (called overall balancing)

$$\sum_{i=1}^{n} \sum_{k=1}^{m} x_{\mathtt{ik}} = T$$

The complete mixed integer linear programming model is therefore the following:

$$\left\{\begin{array}{ll}
\min(M' \sum_{i=1}^{n}\sum_{k=0}^{m} y_{\mathtt{ik}} + \sum_{i=1}^{n}\sum_{k=0}^{m} s_{\mathtt{ik}}) & \\
 & \\
\sum_{i=1}^{n} x_{\mathtt{ik}} = T_k & \forall k \in K \\
x_{\mathtt{i0}} = \sum_{k=1}^{m} x_{\mathtt{ik}} & \forall i \in I \\
\sum_{i=1}^{n}\sum_{k=1}^{m} x_{\mathtt{ik}} = T & \\
a_{\mathtt{ik}} - x_{\mathtt{ik}} \le M y_{\mathtt{ik}} & \forall i = 1,\ldots,n \;\; \forall k = 0,\ldots,m \\
x_{\mathtt{ik}} - a_{\mathtt{ik}} \le M y_{\mathtt{ik}} & \forall i = 1,\ldots,n \;\; \forall k = 0,\ldots,m \\
s_{\mathtt{ik}} \ge a_{\mathtt{ik}} - x_{\mathtt{ik}} & \forall i = 1,\ldots,n \;\; \forall k = 0,\ldots,m \\
s_{\mathtt{ik}} \ge x_{\mathtt{ik}} - a_{\mathtt{ik}} & \forall i = 1,\ldots,n \;\; \forall k = 0,\ldots,m \\
0 \le x_{\mathtt{ik}} \le S & \forall i \in I \qquad \forall k \in K \\
0 \le x_{\mathtt{i0}} \le mS & \forall i \in I \\
s_{\mathtt{ik}} \ge 0 & \forall i = 1,\ldots,n \;\; \forall k = 0,\ldots,m \\
x_{\mathtt{ik}}, s_{\mathtt{ik}} \in \mathbb{R} & \forall i = 1,\ldots,n \;\; \forall k = 0,\ldots,m \\
y_{\mathtt{ik}} \in \{0,1\} & \forall i = 1,\ldots,n \;\; \forall k = 0,\ldots,m
\end{array}\right.$$

The above model has been used by Istat in the case of the Italian Census of Agriculture 2010 to restore data consistency when total cultivation area, or total number of livestock, was not equal to the sum of the detailed values representing the parts of the above totals. The results of this approach are described in detail [10].

1.1.2 Reconstruction of Cultivation Data in Agriculture

Another relevant problem, in the described Census, concerns the development of a procedure for assigning the correct cultivations to the area for which the farm declarations are unreliable. Indeed, each *farm* specifies the cultivation area used for each *cultivation*. A classical problem is verifying the accuracy of this information, and correcting those detected as unreliable. Errors in these declarations should be detected and corrected by mathematically “guessing” the correct values, since it is clearly impossible to contact again the farm or inspect somehow the cultivations.

Farms can extend on one or more districts, and the area owned by each farm in each district is known. Therefore, the compatibility of each cultivation with each district can also be evaluated (some cultivation can grow only on specific types of soils, or need specific climatic conditions, latitude, altitude, etc.). In the specific case of vineyards, considered again as a representative example, there may be cultivation areas with missing or erroneous vineyard codes. For these areas, it is required to assign a code, according to a set of consistency constraints, taking also into account the compatibility between each district and the type of vineyard. Theoretically, the elements for solving the above problem are available, but the problem is doing this on large datasets both efficiently and in an unbiased manner.

The Italian territory is subdivided into many districts, and each farm can extend on one or more district. We denote by

I = {1,..., n} the set of all possible cultivations;

J = {1,..., m} the set of all possible districts.

Focusing on a single farm, and denoting by f its total area, all the cultivations declared by that farm are checked. Some of them verify a set of rules and conditions prepared for this aim and are therefore considered reliable, while some other do not. This may happen either because some of the declarations appear erroneous, or because there is a discordance between the total area declared and the sum of the areas declared for each cultivation. Denote by a the total farm area reliably assigned, i.e. the area for which the farm declaration are considered reliable. On the contrary, by grouping all the *unreliable declarations*, a nonempty area often remains for which the cultivation is not known. That area will be called unassigned area and denoted by u. Clearly, $f = a + u$. The central problem of our Information Reconstruction process consist now in assigning the cultivations to the mentioned unassigned area. In this Section a discrete mathematical model for this problem is proposed. For each farm, denote by

s_i (real value ≥ 0) the total area that the farm uses for cultivation i, with $i \in I$. Note that this area may span on one or more districts, and the farm does not declare, nor generally even consider, such subdivision. These values are only the ones, among all the cultivation data declared by farms, that can be considered reliable, so $\sum_i (s_i) = a$.

d_j (real value ≥ 0) the total area owned by the farm in district j, with $j \in J$. These values are not surveyed during the considered Census but are already available and are reliable.

$p_{\mathtt{ij}}$ (real value $\in [0, 1]$) the likelihood of having cultivation i in district j, with $i \in I$ and $j \in J$. Values near to 1 means high likelihood, near to 0 means very low likelihood. This values are estimated on the basis of agricultural registrations and studies, not surveyed during the considered Census.

Moreover, there are areas where specific cultivations may be used to produce foods having “controlled origin” (in Italian DOC). In particular, for the unassigned area u, it is possible to partition it into a portion that is suitable for “controlled origin” and a portion that is not suitable for that. Denote by

> C (reale value ≥ 0) the total unassigned area owned by the farm in cultivations suitable for “controlled origin”;
>
> N (real value ≥ 0) the total unassigned area owned by the farm in cultivations not suitable for“controlled origin”, so that $C + N = u$.

Those areas C and N should be assigned in order to maximize the likelihood of the assignment. Note that it is not known which district the unassigned area u is located into. On the other hand, the likelihood values depend on the districts. As a consequence, we need to locate the unassigned area u on the districts. This is apparently hard to obtain. A way of doing so is locating on the districts each of the reliable cultivation areas s_i, and then obtaining the location of u as the portion of farm area f not covered by a. In order to model the described problem, we need to introduce the following sets of decision variables:

> $x_{\mathtt{ij}}$ (real value ≥ 0) the area of cultivation i that, according to our reconstruction, is localized in district j, with $i \in I$ and $j \in J$;
>
> $v_{\mathtt{ij}}$ (real value ≥ 0) the portion of C that, according to our reconstruction, is used for cultivation i and localized in district j, with $i \in I$ and $j \in J$;
>
> $w_{\mathtt{ij}}$ (real value ≥ 0) the portion of N that, according to our reconstruction, is used for cultivation i and localized in district j, with $i \in I$ and $j \in J$.

Moreover, each of the farm unassigned areas C and N generally contains only a specific cultivation, and not a mixture of different cultivations. We therefore want to assign all C to one single type of cultivation, and not to fragment it among all the cultivations compatible with that area. A similar requirement holds for N. This requires the use of additional binary decision variables

$$y_i = \begin{cases} 1 & \text{if } C \text{ is assigned in our reconstruction to cultivation } i, \text{with } i \in I \\ 0 & \text{otherwise} \end{cases}$$

$$z_i = \begin{cases} 1 & \text{if } N \text{ is assigned in our reconstruction to cultivation } i, \text{with } i \in I \\ 0 & \text{otherwise} \end{cases}$$

Now it’s possible to formulate a mixed integer linear programming model for each farm. Cultivation assignment to areas should be done in order to maximize the likelihood. Our objective function is therefore

$$\max \sum_{i=1}^{n}\sum_{j=1}^{m} p_{\mathtt{ij}} x_{\mathtt{ij}} + \sum_{i=1}^{n}\sum_{j=1}^{m} p_{\mathtt{ij}} v_{\mathtt{ij}} + \sum_{i=1}^{n}\sum_{j=1}^{m} p_{\mathtt{ij}} w_{\mathtt{ij}}$$

This assignment should obviously verify a set of constraints. First of all, the sum of the areas assigned to the different cultivations in each district *j* must be equal to the area owned by the farm in district *j* :

$$\sum_{i=1}^{n} x_{\mathtt{ij}} + \sum_{i=1}^{n} v_{\mathtt{ij}} + \sum_{i=1}^{n} w_{\mathtt{ij}} \;=\; d_j \qquad \forall j = 1, \ldots, m$$

The sum of the areas used by the farm for cultivation *i* over all the districts must be equal to the total area used by the farm for cultivation *i*:

$$\sum_{j=1}^{m} x_{\mathtt{ij}} = s_i \qquad \forall i = 1, \ldots, n$$

The sum of the portions of C assigned to all cultivations in all districts must be equal to C. A similar condition must hold for N.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} v_{\mathtt{ij}} = C \qquad \sum_{i=1}^{n}\sum_{j=1}^{m} w_{\mathtt{ij}} = N$$

In order to connect the y variables to v, we need to impose that it is not possible assigning a portion of C to cultivation i (regardless to the district) when the corresponding variable y_i is 0. A similar condition must hold for to connect the z variables to w. Note that M is a constant value greater than all possible left-hand-side values.

$$v_{\mathtt{ij}} \le My_i \qquad \forall i = 1, \ldots, n \;\; \forall j = 1, \ldots, m$$

$$w_{\mathtt{ij}} \le Mz_i \qquad \forall i = 1, \ldots, n \;\; \forall j = 1, \ldots, m$$

The whole C must be assigned to only one cultivation. A similar condition must hold for N.

$$\sum_{i=1}^{n} y_i = 1 \qquad \sum_{i=1}^{n} z_i = 1$$

The above constraints have the effect of letting only one y (only one z) be 1, and so the previous constraints can only assigning C (respectively N) to a unique cultivation.

Finally, an assignment for C or for N cannot be accepted when the likelihood of that assignment, although being the greatest possible for the current problem, is too low. In such a case, indeed, that assignment cannot be considered reliable. For this reason we introduce, in the following constraints, two thresholds, denoted by SC and SN, respectively for the assignments made on C and N.

$$\sum_{j=1}^{m} v_{\mathtt{ij}} - \sum_{j=1}^{m} p_{\mathtt{ij}} v_{\mathtt{ij}} - SC \leq M(1 - y_i) \quad \forall i = 1, \ldots, n \tag{1.2}$$

$$\sum_{j=1}^{m} w_{\mathtt{ij}} - \sum_{j=1}^{m} p_{\mathtt{ij}} w_{\mathtt{ij}} - SN \leq M(1 - z_i) \quad \forall i = 1, \ldots, n \tag{1.3}$$

In the constraints (1.2), the assignment of C can be possible if the likelihood of assigning C to cultivation i is good (= near to 1) for the different districts where C have been located, that means y_i can assume value 1. On the other hand, when that likelihood is not good (= near to 0), that assignment is not allowed, that means y_i must be forced to value 0. Note that, if no assignment has a sufficient likelihood, those constraints cannot be satisfied and the model correctly becomes infeasible.

The above is obtained because, for assignments having good likelihood, the value of $\sum_{j=1}^{m} p_{\mathtt{ij}} v_{\mathtt{ij}}$ is only a bit smaller than the value of $\sum_{j=1}^{m} v_{\mathtt{ij}}$, and by subtracting SC the left-hand-side of the inequality (1.2) becomes smaller than or equal to 0, leaving y_i free.

When on the contrary the likelihood is not good, the value of $\sum_{j=1}^{m} p_{\mathtt{ij}} v_{\mathtt{ij}}$ is much smaller than the value of $\sum_{j=1}^{m} v_{\mathtt{ij}}$, and even subtracting SC (whose reasonable value is therefore just a fraction of $\sum_{j=1}^{m} v_{\mathtt{ij}}$, for instance one half) the left-hand-side of the inequality (1.2) becomes positive. As a consequence, $M(1 - y_i)$ must have a strictly positive value, and so y_i must have value 0.

In the constraints (1.3) the logic of the assignment of N is the same as that previously described for C in the the constraints (1.3).

On the whole, the complete mixed integer linear programming model for assigning the unassigned area of a single farm is the following:

$$\begin{cases} \max \sum_{i=1}^{n}\sum_{j=1}^{m} p_{\mathtt{ij}} x_{\mathtt{ij}} + \sum_{i=1}^{n}\sum_{j=1}^{m} p_{\mathtt{ij}} v_{\mathtt{ij}} + \sum_{i=1}^{n}\sum_{j=1}^{m} p_{\mathtt{ij}} w_{\mathtt{ij}} & \\ \sum_{i=1}^{n} x_{\mathtt{ij}} + \sum_{i=1}^{n} v_{\mathtt{ij}} + \sum_{i=1}^{n} w_{\mathtt{ij}} \ = \ d_j & \forall j = 1,\dots,m \\ \sum_{j=1}^{m} x_{\mathtt{ij}} = s_i & \forall i = 1,\dots,n \\ \sum_{i=1}^{n}\sum_{j=1}^{m} v_{\mathtt{ij}} = C \qquad \sum_{i=1}^{n}\sum_{j=1}^{m} w_{\mathtt{ij}} = N & \\ v_{\mathtt{ij}} \le M y_i & \forall i = 1,\dots,n \ \forall j = 1,\dots,m \\ w_{\mathtt{ij}} \le M z_i & \forall i = 1,\dots,n \ \forall j = 1,\dots,m \\ \sum_{i=1}^{n} y_i = 1 \qquad \sum_{i=1}^{n} z_i = 1 & \\ \sum_{j=1}^{m} v_{\mathtt{ij}} - \sum_{j=1}^{m} p_{\mathtt{ij}} v_{\mathtt{ij}} - SC \le M(1-y_i) & \forall i = 1,\dots,n \\ \sum_{j=1}^{m} w_{\mathtt{ij}} - \sum_{j=1}^{m} p_{\mathtt{ij}} w_{\mathtt{ij}} - SN \le M(1-z_i) & \forall i = 1,\dots,n \\ x_{\mathtt{ij}} \ge 0 & \forall i = 1,\dots,n \ \forall j = 1,\dots,m \\ v_{\mathtt{ij}} \ge 0 & \forall i = 1,\dots,n \ \forall j = 1,\dots,m \\ w_{\mathtt{ij}} \ge 0 & \forall i = 1,\dots,n \ \forall j = 1,\dots,m \\ x_{\mathtt{ij}} \in \mathbb{R} & \forall i = 1,\dots,n \ \forall j = 1,\dots,m \\ v_{\mathtt{ij}} \in \mathbb{R} & \forall i = 1,\dots,n \ \forall j = 1,\dots,m \\ w_{\mathtt{ij}} \in \mathbb{R} & \forall i = 1,\dots,n \ \forall j = 1,\dots,m \\ y_i \in \{0,1\} & \forall i = 1,\dots,n \\ z_i \in \{0,1\} & \forall i = 1,\dots,n \end{cases}$$

The above model has been used by Istat in the case of the Italian Agricultural Census of 2010 to restore the consistency of the cultivations data. The results of this approach are described in detail in [9].

Chapter 2

Graph Partition Approaches to Functional Regionalization

2.1 Introduction

The general problem of partitioning a territory, also known as Functional Regionalization, Territorial Districting or Territory Design, is central to several applicative tasks, such as the identification of local labour market areas, the design of school and hospital districts, political districting, sales districting, and so on, see, e.g., [43, 113]. According to [116, 30], the methods of functional regionalization can be subdivided into two main approaches.

The first one is based on statistical methods and employs numerical taxonomy principles. These methods typically use a *single-step* procedure that seeks to maximize a statistical criterion representing the objective. They include clusters analysis and specific regionalization algorithms (e.g., [93]). The single-step procedures use a single classification rule. Techniques are designed for the analysis of spatial interaction matrices. Among these methods, the most widely used are based on the Contingency tables analysis, on the Factor analysis (see, e.g., [100]) and on the Markov chains analysis (see, e.g., [20]). The last two methods also allow to identify *Nodal Regions*, such as the method proposed by Nystuen & Dacey ([106]). However, this latter argument is out of the scope of this thesis.

The *single-step* analytical methods are characterized by:

- the way in which the interaction matrix is transformed;
- the way in which localities are grouped to generate areas on the basis of the transformed matrix.

The main problems for the *single-step* algorithms are the difficulties of imposing constraints without the risk of biasing the final results, and the difficulties to establish the size of the partition because there is no objective criterion to stop the grouping process. Generally, *single-step* methods produce the best results when they are used as exploratory methods.

A particularly well-known *single-step* procedure, based on the analysis of contingency tables, is the intra-regional interaction maximization, or INTRAMAX, initially developed by Masser and Brown [93], also implemented in the software FLOWMAP developed at the University of Utrecht [130]. INTRAMAX procedure is a stepwise hierarchical algorithm based on an interaction matrix, which contains the interactions (e.g., journey-to-work flows) between all the localities constituent a given territory, including the interactions of each locality with itself. In each step, two localities are selected and merged together, producing a new locality (a region) that is the union of the two, and all the interactions involving this new locality are updated accordingly. To avoid possible cases of fusions between nonadjacent localities, a contiguity constraint has also been incorporated into the grouping procedure [93]. However, this constraint causes some loss of information on the observed flows and affects the empirical findings as showed in [115]. Interesting results of Intramax approach for identifying functional regions are described in [48, 103, 135, 94, 84].

The second approach consists of the *multi-step* procedures based on complex decision rules derived from empirical experience of the analysts. These methods do not perform any kind of processing on the interaction matrix data. The matrix is used as a source of information. Multi-step methods have been used for the identification of functional areas such as: Local Labour Market Areas (LLMAs), Travel to work areas (TTWAs), Standard Metropolitan Labour Area (SMLA). These methods are named according to the purpose of functional region that they identify.

The main characteristics of these methods involve the following aspects:

- the initial steps to identification of the functional regions are related to the selection of the focal localities;

- the final steps allow the fine tuning of regional boundaries and the validation of the obtained regions.

Multi-step methods do not require to transform the interaction matrix, but they need a set of criteria able to define self-contained regions. Unlike the single-step methods, the set of criteria which underpin a multi-step approach can be reformulated as long as they do not generate areas that fit the spatial pattern of the daily journey-to-work flows.

The TTWA method, initially proposed by Coombes and Openshaw to generate official statistical reporting areas in Britain, is the mainly adopted *multi-step* procedure [28, 29]. It is based on a traditional understanding of cities as focal points for hinterlands. The assignment of a group of localities to a TTWA is guided by the maximization of the interaction between those localities. Thus, indirectly, the interaction that crosses boundaries between TTWAs is minimized. Besides, every TTWA must reach a minimum level of self-containment and a minimum size in terms of resident occupied population. The trade-off between size and self-containment is the peculiarity of the TTWA algorithm.

A more general version of the TTWA-based algorithm was devised by Sforzi, Openshaw and Wymer to regionalize Italy in LLMAs with support from Istat, the Italian National Institute of Statistics [117]. The presence/absence of the trade-off between size and self-containment is the main difference between the TTWA-based algorithm and the LLMA-based Italian one. The choice to remove the trade-off was justified by the discretionality of the size of employment and to preserve the established threshold of self-containment, recognized as the key criterion. This algorithm was officially applied by Istat to process journey-to-work data for the functional regionalization of Italy in 1981, 1991 and 2001.

Van der Laan and Schalke [129] also develop a multi-level classification of the LLMAs identification methods. They basically distinguish between methods allowing for heterogeneity among LLMAs and methods which provide homogeneity. Then, they subdivide the homogeneous category into deductive methods, which identify at first urban centres around which the LLMAs are constructed, and inductive methods, which do not use such pre-conceived structures.

The identification of LLMAs is a very important case of functional regionalization. LLMAs are widely accepted as the most appropriate units for the statistical analysis of socio-economic phenomena, such as the detection of industrial districts ([118, 18, 24]), the evaluation of productivity advantages of local economies (see, e.g., [39]) and, of course, the measurement of the employment and unemployment rates.

A LLMAs is defined as a functional region ”where the majority of the local population seeks employment and from which the majority of local employers recruit labour” [62]. Operationally, a LLMAs consists of a group of contiguous localities defining an area in which there is a concentration of labour demand and in which workers can change jobs without changing their place of residence ([131]). In other words, it is an area where demand and supply for labour meet at high degree, and daily commuters entering the area or going out from it are only a minority.

LLMAs do not fit administrative boundaries. The boundaries of LLMAs change over time because of demographic and economic changes. These changes are reflected in the daily home-to-work flows. Therefore, LLMAs are updated periodically through the Census that collects information on such flows.

Flórez-Revuelta *et al.* present in [55, 91] a new approach to the identification of LLMAs based on evolutionary computation. The procedure is based on the maximization of a fitness function that measures the aggregate intra-region interaction under constraints of inter-region separation and minimum size. Another interesting approach to the problem is based on measures of *Modularity* (see, e.g., [45]), since there should be dense connections within each region, but only sparse connections between regions. Girvan and Newman [60] propose an algorithm that uses betweenness centrality to find community boundaries.

Modularity is used as quality index of a partition of a network into communities. It measures internal (and not external) connectivity, but it does so with reference to a randomized null model. Following this line of research, modularity has been very influential in recent community detection literature, and one can use spectral techniques to approximate it [60, 139].

Djidjev [40] shows that the problem of finding a partition maximizing the modularity of a given graph G can be reduced to a minimum weighted cut problem and suggests the use of Multilevel algorithms for graph partitioning to efficently solve this problem. However, Guimerá *et al.* [64] and Fortunato and Barthélemy [57] show that random graphs have high-modularity subsets and that exists a size scale below which modularity cannot identify communities. In particular, [57] shows that modularity optimization may fail to identify modules smaller than a given scale, which depends on the total number of the network links and on the degree of interconnectedness of the modules, even in cases where modules are unambiguously defined.

Following this line of research, Farmer *et al.* in [45] maximize the modularity of a network of commuting flows to produce a regionalization that exhibits less interactions than expected between regions. This approach should have specific advantages over existing regionalization procedures, particularly in the context of disaggregate commuting patterns of socio-economics subgroups [45].

Kropp and Schwengler in [82] propose an approach based on a modification of Nystuen and Dacey‘s dominant flow method [106] and adopt the modularity measure to assess the quality of the different delineations.

Finally, Kim et al. [80] propose an exact approach to the case of regionalization problem with a predetermined number of regions and contiguity constraints. The proposed model simultaneously determines a given number of functional centers and delimits their sphere of influence simultaneously, while explicity incorporating contiguity constraints.

In this thesis we propose and evaluate a new approach to functional regionalization problem, with specific reference to the identification of LLMAs.The problem is converted into a graph partitioning problem. The proposed approach obtains the solution by solving a sequence of minimum cut problems over an undirected graph obtained from the interactions among the localities. This graph is here called transitions graph. The procedure has been implemented in c++ and tested on real data from the Italian Census of Population 2001.

The results of the proposed approach are compared to those of the procedure officially used by the Italian National Institute of Statistics (Istat) in 1981,1991 and 2001 to define the Italian LLMAs ([107, 119, 120]). After the choice of a method highly specialized to determine the Italian LLMAs, another comparison is with the state-of-the-art procedure for general purpose graph partitioning METIS [78]. The main contribution of this work is therefore an innovative and effective approach based on Combinatorial Optimization for solving an economically important and challenging real-world problem.

The rest of this Chapter is organized as follows.

Section 2 describes in detail the Regionalization Problem related to the identication of LLMAs.

Section 3 outlines the main characteristics of two consolitaded methods for the identification of LLMAs. Firstly we describe Intramax, the most applied single-step procedure, that is considered a reference method from a statistical point of view. Secondly we describe the multi-step procedure officially used by Istat in 1981, 1991, and 2001 to define LLMAs. This algorithm is used for the comparation with our proposed procedures.

Section 4 explains the proposed procedure based on the solution of minimum cut problems.

Section 5 describes the main characteristics of the Multilevel Graph Partitioning Approach, with special emphasis on the System Metis.

Section 6 reports the experiments and the comparison on data from the 2001 Italian Population Census, along with a discussion of the empirical results.

Conclusions are given in Section 7.

2.2 The Regionalization Problem

The approach proposed for the problem of the identification of functional regions will be hereinafter explained by referring to the identification of Local Labour Markets Areas (LLMAs). This is indeed one of the most important cases of regionalization, because it has a great economic relevance and it requires to deal with a very large set of data.

However, the proposed approach is not intrinsically limited to this case, but can be used for other cases of regionalization sharing the structural characteristics described below.

In the described case, we have the set

$$A = \{a_1, \ldots, a_n\}$$

of all the localities a_i situated in a territory T, that is the geographical area under analysis. Set A is such that $\bigcup_{i=1}^{n} a_i = T$ and $a_i \cap a_j = \phi$ for $i \neq j$.

Moreover, we have an $n \times n$ matrix

$$F = \left(\begin{array}{ccc} f_{11} & \ldots & f_{1n} \\ & \ldots & \\ f_{n1} & \ldots & f_{nn} \end{array} \right)$$

of the interactions existing between all the pairs of localities. In particular, value $f_{ij} \geq 0$ is a measure of the flow of workers that reside in locality a_i and work in locality a_j, and is called *commuting flow*, or also daily journey-to-work flow. Clearly, F is not necessarily symmetric.

The identification of the LLMAs consists of a partition of the set A into subsets $R_1, \ldots, R_m$ (the functional regions) such that $R_p \cap R_q = \phi$ for $p \neq q$ and $\bigcup_{p=1}^{m} R_p = T$.

The goals of this partition may be viewed from different perspectives, but basically consist of maximizing the number of LLMAs such that the obtained regions remain statistically and economically meaningful. This means that:

(*i*) each LLMA must be sufficiently self-contained;

(*ii*) each LLMA must have a sufficient number of workers;

(*iii*) each LLMA must be composed by a set of localities;

(*iv*) each LLMA must be internally contiguous.

To impose condition (*i*) one needs to evaluate *self-containment*.

The total occupied population working in locality a_i (i.e., for short, *workers* in a_i) is $w(a_i) = \sum_{k=1}^{n} f_{ki}$. Consequently, the total number of workers in region R_p is

$$w(R_p) = \sum_{a_i \in R_p} \sum_{k=1}^{n} f_{ki}.$$

Specularly, the total occupied population residing in a_i (i.e., for short, *residents* in a_i) is $r(a_i) = \sum_{j=1}^{n} f_{ij}$. Consequently, the total number of residents in R_p is

$$r(R_p) = \sum_{a_i \in R_p} \sum_{j=1}^{n} f_{ij}.$$

Also, the total number of workers in locality a_i that reside outside of a_i (the incoming commuters) is $c^-(a_i) = \sum_{k=1,k\neq i}^{n} f_{ki}$. Consequently, the total number of incoming commuters in region R_p is

$$c^-(R_p) = \sum_{(k,i):\ a_k \notin R_p, a_i \in R_p} f_{ki}.$$

Conversely, the total number of residents in locality a_i that work outside of a_i (the outgoing commuters) is $c^+(a_i) = \sum_{j=1,j\neq i}^{n} f_{ij}$. Consequently, the total number of outgoing commuters from region R_p is

$$c^+(R_p) = \sum_{(i,j):\ a_i \in R_p, a_j \notin R_p} f_{ij}.$$

Finally, the total number of residents in region R_p that also work in R_p (the internal flow) is

$$l(R_p) = \sum_{(i,j):\ a_i, a_j \in R_p} f_{ij}.$$

Hence, value f_{ii} is also called the internal flow of locality a_i.

The *Supply-side self-containment* function for a_i is defined to evaluate the portion of people residing and working in locality a_i within the total workers in a_i, as follows:

$$sc^w(a_i) = \frac{f_{ii}}{w(a_i)}.$$

Consequently, the Supply-side self-containment for a region R_p is

$$sc^w(R_p) = \frac{l(R_p)}{w(R_p)}. \tag{2.1}$$

The *Demand-side self-containment* function for a_i is defined to evaluate the portion of people residing and working in a_i within the total residents in a_i, as follows:

$$sc^r(a_i) = \frac{f_{ii}}{r(a_i)}.$$

Consequently, the Demand-side self-containment for a region R_p is

$$sc^r(R_p) = \frac{l(R_p)}{r(R_p)}. \tag{2.2}$$

Finally, we define the *Overall self-containment* function for a_i to evaluate the portion of people residing and working in a_i within the total number of people interacting with a_i (that is, working and/or residing), as follows:

$$sc(a_i) = \frac{f_{ii}}{f_{ii} + c^-(a_i) + c^+(a_i)}.$$

Consequently, the Overall self-containment for a region R_p is

$$sc(R_p) = \frac{l(R_p)}{l(R_p) + c^-(R_p) + c^+(R_p)}. \tag{2.3}$$

Self-containment functions have been used in literature in different manners. Clearly, the first two only consider partial aspects. We select the Overall self-containment one, because in our opinion is the more coherent with the definition of LLMA. Hence, a region R_p is sufficiently self-contained, i.e., respects condition (*i*) above, when $sc(R_p) \geq c_1$, where c_1 is a threshold defined for the specific analysis. Since in any case $sc(R_p) \in [0, 1]$, possible thresholds range in $(0.5, 1]$.

We mention, however, that several authors make a different choice, and considered simultaneously the Supply-side self-containment and the Demand-side self-containment for their analysis [30].

However, note that $sc(R_p) \geq c_1$ implies both $sc^w(R_p) \geq c_1$ and $sc^r(R_p) \geq c_1$, since all the values involved in their computation are nonnegative. Indeed, given a region R_p, reaching a minimum value for the overall self-containment function $\frac{l(R_p)}{l(R_p)+c^+(R_p)+c^-(R_p)}$ guarantees that both supply- and demand-side self-containment of R_p reach that value. A simple proof is the following:

$$\min\left\{\frac{l(R_p)}{w(R_p)}, \frac{l(R_p)}{r(R_p)}\right\} = \min\left\{\frac{l(R_p)}{l(R_p) + c^-(R_p)}, \frac{l(R_p)}{l(R_p) + c^+(R_p)}\right\}$$

Since $l(R_p) \geq 0$, $c^+(R_p) \geq 0$ and $c^-(R_p) \geq 0$, we have:

$$\min\left\{\frac{l(R_p)}{l(R_p) + c^-(R_p)}, \frac{l(R_p)}{l(R_p) + c^+(R_p)}\right\} \geq \frac{l(R_p)}{l(R_p) + c^+(R_p) + c^-(R_p)}$$

Therefore, for any $c_1 \geq 0$, an overall self-containment

$$\frac{l(R_p)}{l(R_p)+c^+(R_p)+c^-(R_p)} \geq c_1 \text{ implies}$$

$$\min\left\{\frac{l(R_p)}{w(R_p)}, \frac{l(R_p)}{r(R_p)}\right\} \geq c_1.$$

A region R_p has a sufficient number of workers, i.e., respects condition (*ii*) above, when $w(R_p) \geq c_2$, where c_2 is a natural number, again depending on the specific analysis and on the size of the a_i (as an example for the present case study, $c_2 \geq 1000$).

A region R_p respects condition (*iii*) above when the number of localities composing it is such that $|R_p| \geq c_3$, where c_3 is another natural number, again depending on the specific analysis. For example, $c_3 = 2$, see also [117]. However, the proposed approach is flexible and allows setting c_3 to different natural numbers whenever the application would require it.

From the mathematical point of view, parameter c_3 could perfectly assume value 1 without affecting the operation of the proposed method. However, we do not choose such a value in our experiments, because we are interested in having LLMAs of at least 2 localities, so $c_3 = 2$. We also note that such a condition is not a novelty in the literature, but it is often required in the case of the identification of LLMAs. See for instance Istat-Irpet 1989 (p. 16) [119] among our references, where a LLMA is defined as follows: "A local labour market is an area (which includes several localities) characterized by a certain concentration of jobs, where most of the local population can find a job [...] (and the workers resident can change a job) without changing their locality of residence". As a consequence, this definition excludes the possibility that a single locality (a municipality) is considered a self-standing LLMA.

Condition (*iv*), finally, is of difficult mathematical formalization. However, several authors in the literature state that this condition should not be imposed during the generation of the LLMAs, because otherwise, at each step of a generic regionalization procedure, it would limit the possible choices, and this would very likely lead to the determination of worse LLMAs boundaries, as explained in, e.g., [26]. So, the possibility of working with non-contiguous proto-regions before producing the final LLMAs should be allowed during the generation process.

On the contrary, contiguity should be checked on the final LLMAs, and, if one of them does not respect it, one has to disassemble that LLMA and possibly merge some of its localities with other contiguous LLMAs neighboring to them.

Conditions (*i*) and (*ii*) can be slightly relaxed, in the sense that the (almost) full satisfaction of one condition is considered enough to compensate a little unsatisfaction of the other. Hence, the following condition (2.4) can subsume the two conditions (*i*) and (*ii*), where c is a threshold with a value generally ≥ 0.75 [122].

$$\left(\min\left\{\frac{sc(R_p)}{c_1},1\right\}\right)\cdot\left(\min\left\{\frac{w(R_p)}{c_2},1\right\}\right)\geq c \tag{2.4}$$

Evidently, all the above conditions are more easily respected by large LLMAs, and, indeed, one unique LLMA over the whole territory under analysis would fully satisfy them (assuming of course that the values of c_2 and c_3 are feasible for that territory). In order to avoid such kind of solutions, useless from the practical point of view, and to maximize the number of LLMAs, one usually wants that the above conditions (*i*), (*ii*) and (*iii*) are satisfied with the minimum values of self-containment, workers and areas that are able to do that, possibly relaxing conditions (*i*) and (*ii*) with condition (2.4). When this happens, and also condition (*iv*) is met, the regionalization task has been successfully performed.

Note that the practical cases of this problem are generally very computationally demanding, and all the approaches used for this problem are actually approximate procedures (unless simplifications of the problem are considered, for example by pre-assigning the number m of LLMAs that should be generated).

2.3 Two consolidated methods for the Functional Regionalization

In this Section we describe two existing approaches to solve the problem of identification of Functional Regions.

Firstly, we describe the most applied *single-step* procedure Intramax ([93]), based on the analysis of contingency tables, that is considered as a reference method from a statistical point of view ([92]) and the most successfull it single-step procedure. We do not compare this method with ours, but we refer to the comparison with TTWA algorithm described in ([116]). This research highlights that the second approach is more appropriate than Intramax method for the delimitation of the functional regions based on daily journey-to-work flows.

Secondly, we describe the *multi-step* procedure officially used by Istat in 1981, 1991 and 2001 to define the Italian LLMAs ([120]), and it will be compared with our proposed approach based on the min-cut procedure because, among the ones proposed in literature, it appears to be the most appropriate to provide a meaningful functional regionalization of Italy (see, e.g., [39, 24]). As described previously, this procedure is a more general version of the TTWA-based algorithm. It was devised by Sforzi, Openshaw and Wymer to regionalize Italy in LLMAs with support from Istat ([115]). The presence/absence of the trade-off between size and self-containment is the main difference between the TTWA-based algorithm and the LLMA-based Italian one. Take the example of the UK and Italy. In the UK the function for being a viable TTWA has changed over time.“The 75% self-containment remains the same, but the 1991-based TTWAs used a second threshold of 69.5% with an economically active population of 20,000, while the 2001-based TTWAs used a threshold of 66.67% but with an economically active population of 25,000” ([17]). In Italy, the target size value for the 2011-based TTWAs used a second threshold of 60% with an economically active population set by Istat at 10,000.

2.3.1 INTRAMAX algorithm

The INTRAMAX (intrazonal interaction maximization) algorithm was developed by Masser and Brown (1975) [93]. It is a modified version of Ward’s hierarchical aggregation procedure [132].

In this algorithm, Ward’s objective function is replaced by criteria that take into account the overall effect of interaction across group boundaries.

The INTRAMAX algorithm is a stepwise hierarchical algorithm which delimits functional areas to produce a regionalization based on the interaction matrix defined by origin-destination flows between different areas. In each step, two areas are grouped together and the interaction between them becomes an intra-zonal interaction of the new grouped area.

As stated by the authors, ”The main objective in the definition of subsystems for modeling purpose is to maximize the proportion of total interactions which takes place within the aggregation of basic data units that form the diagonal elements of the matrix, and thereby to minimize the proportion of cross-boundary movements in the system as whole”.

In practice, this objective is achieved maximizing the proportion of intrazonal interactions. The process is repeated until all areas are grouped together and all interactions becomes intra-zonal.

In details,

Let F an $n \times n$ interaction matrix

$$F = \begin{pmatrix} f_{11} & \cdots & f_{1n} \\ & \cdots & \\ f_{n1} & \cdots & f_{nn} \end{pmatrix}$$

F is a square matrix with each row representing an origin (residential area) and each column denoting a destination (workplace area). f_{ij} are the commuter flows (interactions) between origin area i and destination area j.

$f_{i.} = \sum_{j=1}^{n} f_{ij}$ is the total of interactions originating from area *i*.

$f_{.j} = \sum_{i=1}^{n} f_{ij}$ is the total of interactions into destination area *j*.

$Tr(F) = \sum_{i=1}^{n} f_{ii}$ is the total intrazonal interaction.

$f_{..} = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} = \sum_{i=1}^{n} f_{i.} = \sum_{j=1}^{n} f_{.j}$ is the total of all interactions.

The interaction matrix can be considered as a contingency table (see Table 1) where the observed value of the cell in the *i*-th row and *j*-th column is the probability

$$a_{ij} = \frac{f_{ij}}{f_{..}}$$

	$Destination$					Row total
	a_{11}				a_{1n}	O_1
		$\ldots$				
$Origin$		$\ldots$	a_{ij}			O_i
	a_{n1}				a_{nn}	O_n
Column total	D_1		D_j		D_n	1

Table 1: Contingency table of the interaction matrix.

The row total $O_i = \sum_{j=1}^{n} a_{ij}$ represents the total outflow from origin area *i* and the column total $D_j = \sum_{i=1}^{n} a_{ij}$ represents the total inflow into destination area *j*. Consequently,

$$\sum_{i=1}^{n} O_i = \sum_{j=1}^{n} D_j = 1$$

In a contingency table, the expected value a^*_{ij} , under the null hypothesis of independence for the Chi-square test, is defined as $a^*_{ij} = O_i D_j$.

The objective of the Intramax procedure is to maximize the proportion within the group interaction at each stage of the grouping process. It can be formulated in terms of the differences between the observed and the expected probabilities that are associated with their marginal totals. Therefore, the difference between observed and expected values is considered as a measure of the interaction between areas. Two areas i and j are grouped together if the objective function

$(a_{ij} - a^*_{ij}) + (a_{ji} - a^*_{ji})$ for $i \neq j$ is maximized.

Note that the objective function can only be calculated for all outflows $O_i > 0$ and for all inflows $D_j > 0$. Therefore, areas which have either no inflows or no outflows will be ignored.

In each step, two areas are grouped together and the interaction between the two areas becomes the internal interaction for the new resulting area. This new area takes the place of the two grouped areas at the next step of the analyses. So, at the last steps, all areas are grouped together into one area and all interactions become internal.

Masser and Brown (1975) had fixed contiguity constraints on the maximization process to avoid creating groups between non-contiguous areas. This choice, restricting the number of combinations to be examined at each stage of the grouping process, reduces the computational time, particularly when large data sets are involved.

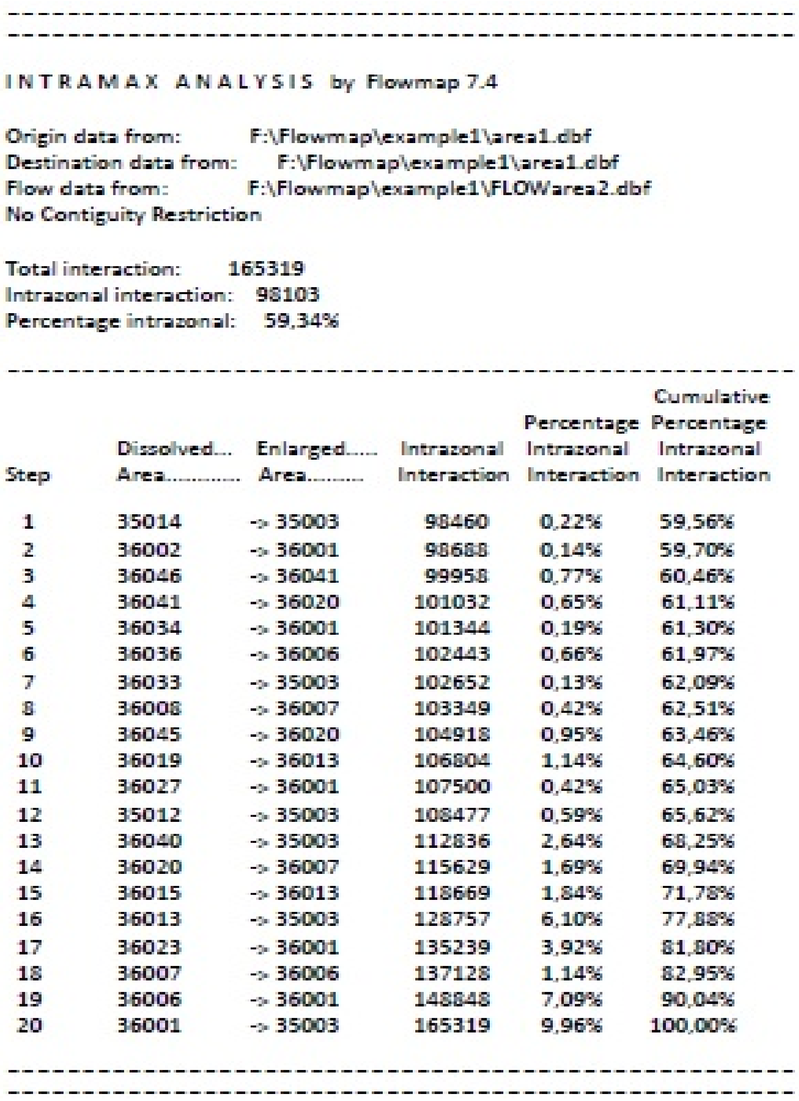
Spatial aggregation by means of the Intramax procedure is incorporated in the software called FLOWMAP. This software was developed at the Faculty of Geographical Sciences in Utrecht University, the Netherlands [130].

In FLOWMAP, contiguity constraints may or not be introduced. These constraints take the form c_{ij}= 1 if area *i* and area *j* are contiguous, c_{ij}= 0 otherwise.

We use an illustrative example to describe the main outputs of an Intramax Analysis by FLOWMAP. This example considers only flows within and between a set of twenty Municipalities. An identification code (Istat Municipality code) has been assigned to each area.

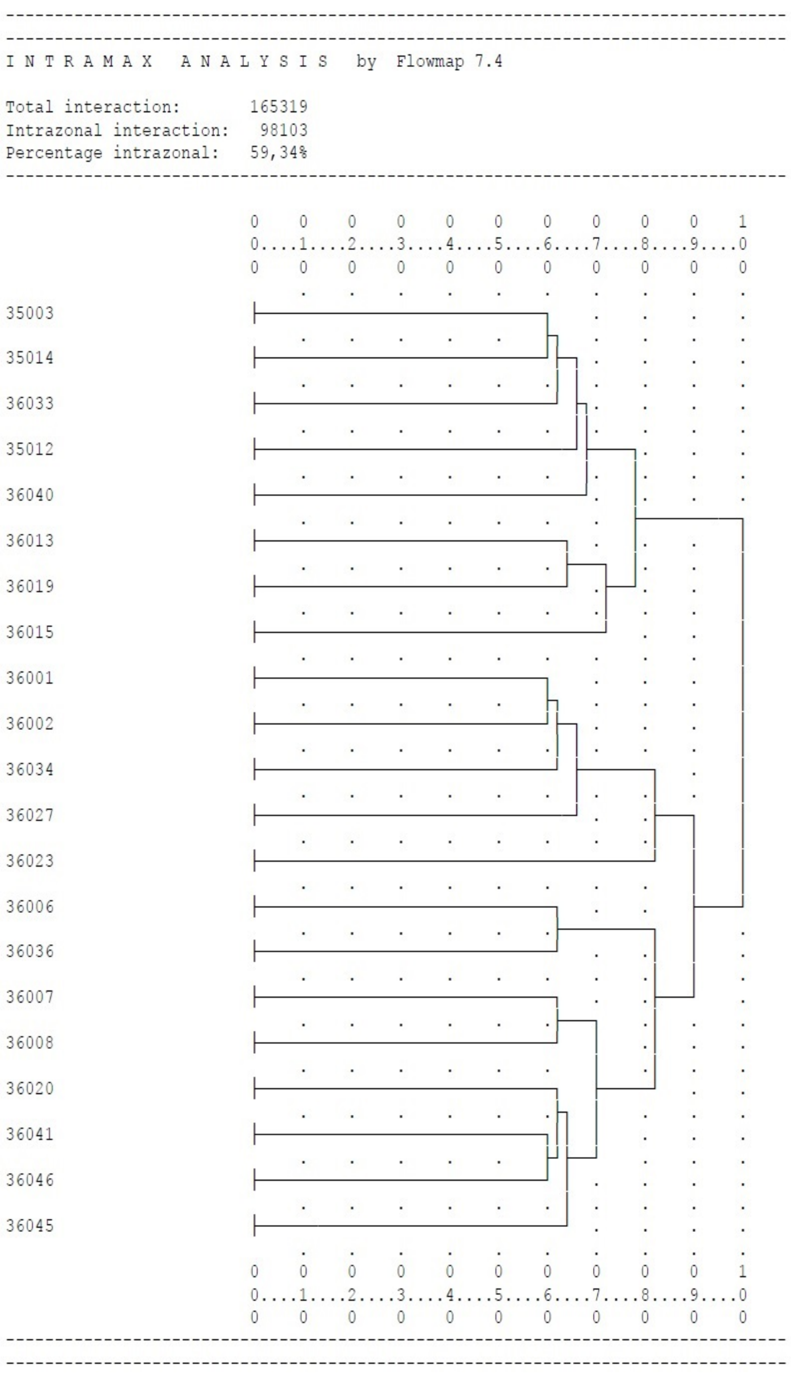
The results of Intramax Analysis for this set of commuting flows are showed in the next two Figures. Particularly, Figure 2.1 shows a part of the fusion report. It describes the aggregation history of the clustering procedure.

The header of the Intramax output denotes the data files which were used. The total number of interactions is$f_{..}$= 165,319, while $Tr(F)$= 98,103 is the total intrazonal interaction. The initial percentage intrazonal interaction is 59.34%. For each step, the algorithm merges two areas which maximize the proportion of the intrazonal interactions, so each area in the first column is grouped with the corresponding area from the second column. The total intrazonal interaction after this merge is shown in the third column. The forth column shows the percentage increase which will occur after this merge. The last column represents the cumulative percentage of intrazonal interaction after the merge. In the first step, area 35014 is added to area 35003, in the second one area 36002 is added to area 36001, then area 36046 is added to area 36041, and so on. The cumulative intrazonal interaction is gradually increasing. After 20 steps all areas are merged into a single area and the cumulative intrazonal interaction becomes 100%.



```
----------------------------------------------------
----------------------------------------------------

INTRAMAX ANALYSIS  by  Flowmap 7.4

Origin data from:          F:\Flowmap\example1\area1.dbf
Destination data from:      F:\Flowmap\example1\area1.dbf
Flow data from:            F:\Flowmap\example1\FLOWarea2.dbf
No Contiguity Restriction

Total interaction:      165319
Intrazonal interaction:    98103
Percentage intrazonal:     59,34%

----------------------------------------------------
                                                       Cumulative
                                           Percentage  Percentage
         Dissolved...  Enlarged.....  Intrazonal   Intrazonal   Intrazonal
Step     Area..........  Area.........  Interaction  Interaction  Interaction

  1      35014        -> 35003         98460      0,22%      59,56%
  2      36002        -> 36001         98688      0,14%      59,70%
  3      36046        -> 36041         99958      0,77%      60,46%
  4      36041        -> 36020        101032      0,65%      61,11%
  5      36034        -> 36001        101344      0,19%      61,30%
  6      36036        -> 36006        102443      0,66%      61,97%
  7      36033        -> 35003        102652      0,13%      62,09%
  8      36008        -> 36007        103349      0,42%      62,51%
  9      36045        -> 36020        104918      0,95%      63,46%
 10      36019        -> 36013        106804      1,14%      64,60%
 11      36027        -> 36001        107500      0,42%      65,03%
 12      35012        -> 35003        108477      0,59%      65,62%
 13      36040        -> 35003        112836      2,64%      68,25%
 14      36020        -> 36007        115629      1,69%      69,94%
 15      36015        -> 36013        118669      1,84%      71,78%
 16      36013        -> 35003        128757      6,10%      77,88%
 17      36023        -> 36001        135239      3,92%      81,80%
 18      36007        -> 36006        137128      1,14%      82,95%
 19      36006        -> 36001        148848      7,09%      90,04%
 20      36001        -> 35003        165319      9,96%     100,00%

----------------------------------------------------
----------------------------------------------------
```

Figure 2.1: Results of Intramax Analysis by FLOWMAP - Fusion report.



```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
I N T R A M A X   A N A L Y S I S   by  Flowmap 7.4

Total interaction:       165319
Intrazonal interaction:   98103
Percentage intrazonal:   59,34%
--------------------------------------------------------------------------------
```

Figure 2.2: Results of Intramax Analysis by FLOWMAP - Dendrogram.

In Figure 2.2, a dendrogram shows which areas have been grouped. It indicates the groups created at each steps referring to the total volume of interaction within groups.

A comparative evaluation of some single-step or multi-step methods was already carried out in Italy in the 80s, using data referred to the regional urban system of Tuscany ([116]).

The interactions data are represented by the daily journey-to-work flows between Tuscany Municipalities (287 localities), collected by the supplementary General Population Census of 1971 conducted by the Italian Region of Tuscany in cooperation with Istat. Figure 2.3 shows the spatial pattern of the daily journey-to-work flows.

INTRAMAX method was applied to the above cited interaction data, taking into account the contiguity constraints of municipalities. The results, shown in Figure 2.4, are referred to the thirtieth step of the grouping process. The stopping criterion has been chosen subjectively. Figure 2.5 shows the results of the application TTWA method to the same data by applying a minimum threshold of supply and demand self-containment, at least equal to 0.75.

2.3.2 The Italian LLMA-based algorithm

We describe here the procedure that has been officially used by Istat for the definition the Italian LLMAs. It is an agglomerative multi-stage heuristic constituted by five principal phases summarized below:

(1) Identification of potential LLMAs focal points;

(2) Amalgamation of potential LLMAs focal points;

(3) Expansion of focal points into proto LLMAs;

(4) Identification of potential LLMAs;

(5) Optimization of LLMAs boundaries.

This procedure is described in detail in [120]. The main criteria and functions used for identification of LLMAs are defined below.

The *Job ratio* function for the area a_i is defined as follows:

$$jr(a_i) = \frac{c^-(a_i)}{c^+(a_i)}$$

where numerator indicate the total incoming flows for area a_i related to workers residents in other areas, while the denominator indicate the total outcoming flows

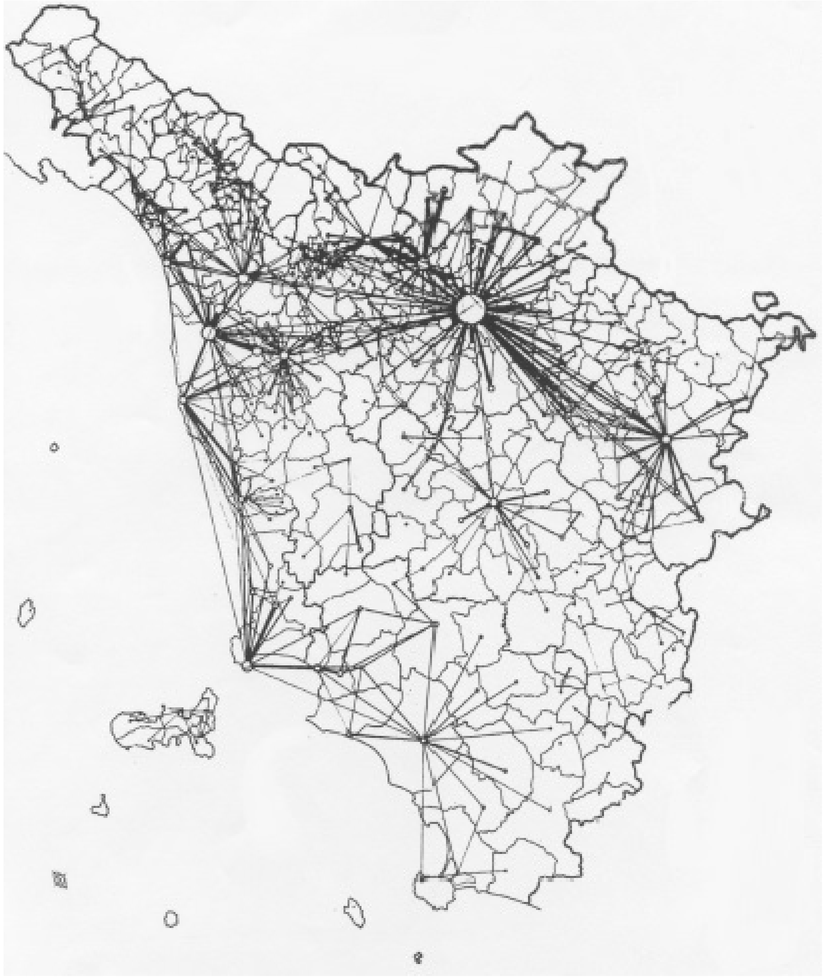


Figure 2.3: Spatial pattern of the daily journey-to-work flows in Tuscany, 1971. (Source: Sforzi 1985)

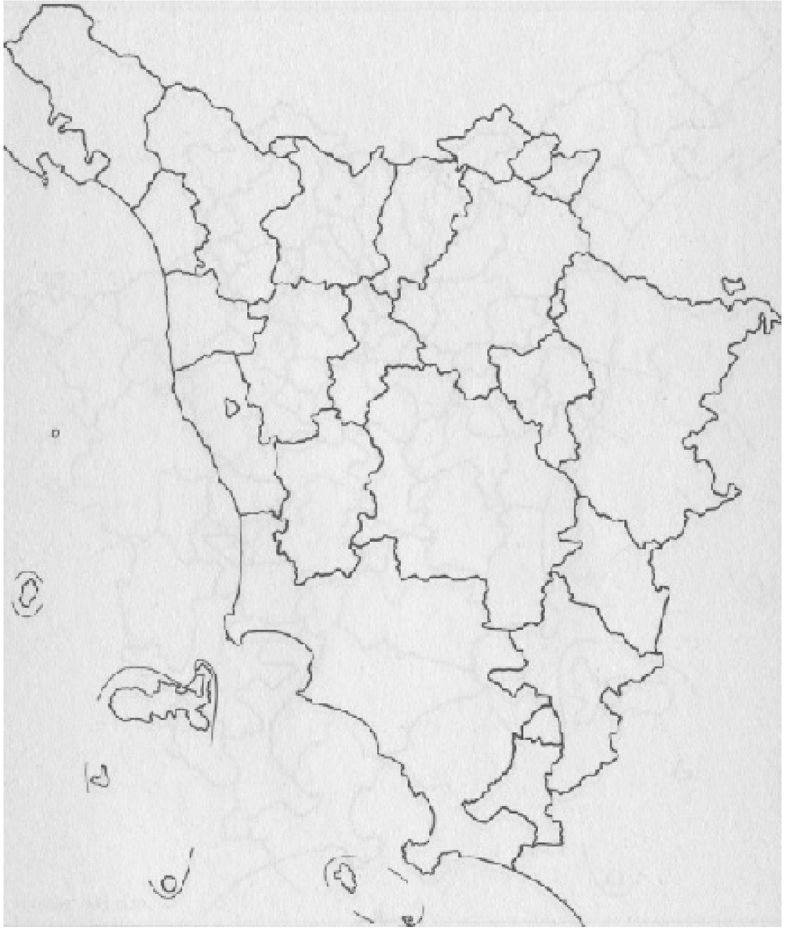


Figure 2.4: Functional Regionalization of Tuscany via INTRAMAX. (Source: Sforzi 1985)

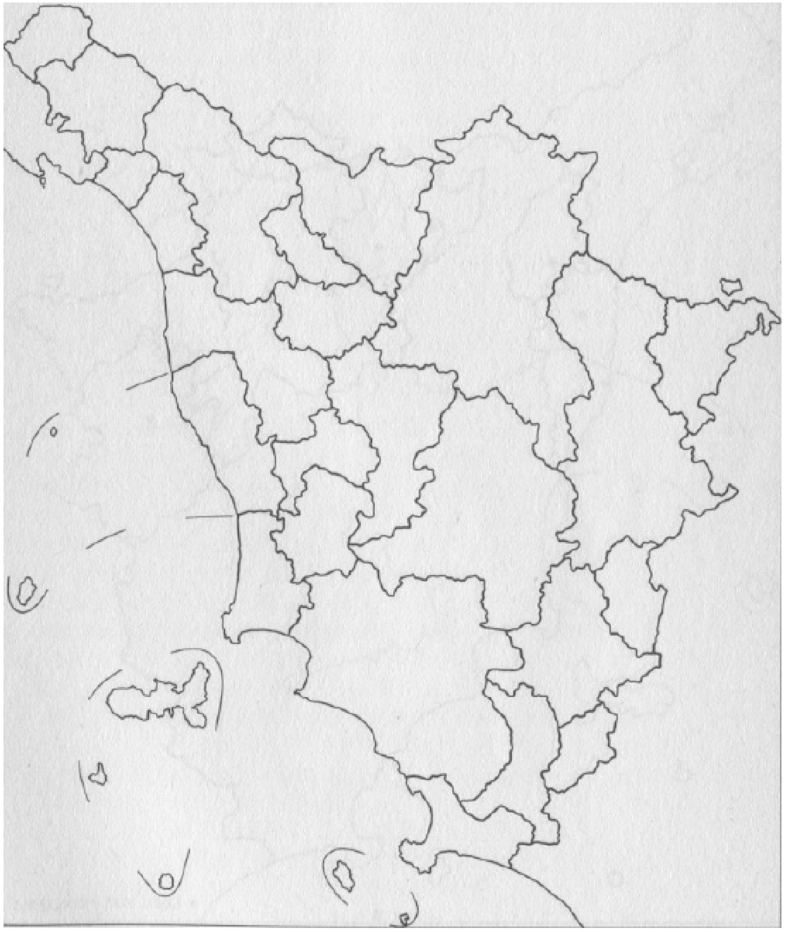


Figure 2.5: Functional Regionalization of Tuscany via TTWA. (Source: Sforzi 1985)

related to the economically active residents for a_i having the workplace in other areas.

The *Supply-side self-containment* function for the area a_i, used to evaluate the portion of people residing and working in locality a_i within the total workers in a_i, is defined as follows:

$$sc^w(a_i) = \frac{f_{ii}}{w(a_i)}.$$

The *Demand-side self-containment* function for the area a_i, used to evaluate the portion of people residing and working in a_i within the total residents in a_i, is defined as follows:

$$sc^r(a_i) = \frac{f_{ii}}{r(a_i)}$$

(1) Identification of potential LLMAs focal points

Obiective: Select from the set of the Municipalities $M \subset A$ those could be used as focal points for building LLMAs detected on the basis of job ratio and supply-side self-containment functions.

For each candidate locality $a_i \in M$, it evaluates $sc^w(a_i)$ and $jr(a_i)$ and chooses the localities that have values in the top 20% for either of the two measures.

(2) Amalgamation of potential LLMAs focal points

Obiective: Amalgamate focal points that exhibit a high degree of interaction.

In this phase are only considers links between the focal point themselves.
For *focus* a_i , having a high degree of interaction with another focus a_j means that are verified all the following conditions:

- for the area obtained merging a_i and a_j either supply-side or demand-side self-containment must be less than 0.5;

- a_i receives at least 10% of the flows coming out of a_j, that is
  $f_{ji} \geq 0.1(c^+(a_j) + f_{jj})$;

- a_j receives at least 1% of flow coming out of a_i, that is
  $f_{ij} \geq 0.01(c^+(a_i) + f_{ii})$;

Focal point are sorted in descending order by their values of incoming flows $c^-(a_i)$ and each is considered in turn, starting from the first *focus* in the ranking.
If some a_i has $\min\{sc^w(a_i), sc^r(a_i)\} \geq 0.5$ then it is not merged whit other *foci* and considered in the next phase.

Else, if some a_i has $\min\{sc^w(a_i), sc^r(a_i)\} < 0.5$, then it is merged, if exists, with the area a_j having a high degree of interaction with it and that maximizes the following *weighted interaction index*, provided that this exceeds 0.002,

$$\frac{f_{ji}^2}{(f_{jj} + c^+(a_j)(f_{ii} + c^-(a_i))} + \frac{f_{ij}^2}{(f_{ii} + c^+(a_i)(f_{jj} + c^-(a_j))}. \tag{2.5}$$

The new combined locality replaces both a_i and a_j and is considered as a *focus*. This process continues until no more of such amalgamations can be done. At the end of this phase we obtain a small number of localities containing a single municipality or a few of those.

(3) Expansion of focal points into proto LLMAs

Obiective: Expand amalgamated foci to form proto LLMAs by allocating the *foci* themselves and also between foci and non-*foci* areas. All areas are sorted in descending order of the following function, where t_1=t_2=0.75, t_3=1000.

$$F(a_i) = \left(\min\left\{\frac{sc^w(a_i)}{t_1}, \frac{sc^r(a_i)}{t_2}, 1\right\}\right) \cdot \left(\min\left\{\frac{w(a_i)}{t_3}, 1\right\}\right) \tag{2.6}$$

If some a_i has $F(a_i) \geq 0.75$ then it is not merged whit other areas and considered in the next phase.

Else, if some a_i has $F(a_i) < 0.75$, then it is merged in a P_k with another locality a_j such that: $F(a_j) < 0.75$ and a_i receives at least 10% of the flows coming out of a_j and a_j is the one that, when merged to a_i, maximizes the above $F(P_k)$. Any locality a_i such that $F(a_i) \geq 0.75$ is a proto-LLMA by itself.

(4) Identification of potential LLMAs

Obiective: Allocate remaining non-foci area units to proto-TTWA. Iteratively dismember proto-TTWA that do not meet the objective value and reallocate component area units.

At first, the algorithm iteratively dismember group of areas with $F(P_k) < 0.75$, in order to reallocate its localities. After this, the set of localities not yet allocated to proto-LLMAs is sorted in decreasing order of in-flow $c^-(a_i)$, and each of them is joined with the proto-LLMA with which it shows the strongest connection, i.e., the one maximizing (2.5). Only localities without in-flows and out-flows are left isolated. Then, iteratively, proto-LLMAs are again checked and those having $F(P_k) < 0.75$ are dismembered and its constituent localities are once again joined to the remaining proto-LLMA just like at the beginning of phase 4. The phase continues until there are only proto-LLMAs with $F(P_k) \geq 0.75$ or isolated localities.

(5) Optimization of LLMAs boundaries

Obiective: Allocate localities of proto-LLMAs having not contiguos localities.

Finally, the algorithm checks if each proto-LLMA is contiguous. Those that are localities not contiguous are dismembered and each of its constituent localities is joined to the contiguous proto-LLMA that maximizes (2.5), in a fashion similar to phase (4).

Figure 2.6, Figure 2.7 and Figure 2.8 show the functional regionalization of Italy in 1981, 1991 and 2001. There were 955 LLMAs in 1981, 784 in 1991 and 686 in 2001.

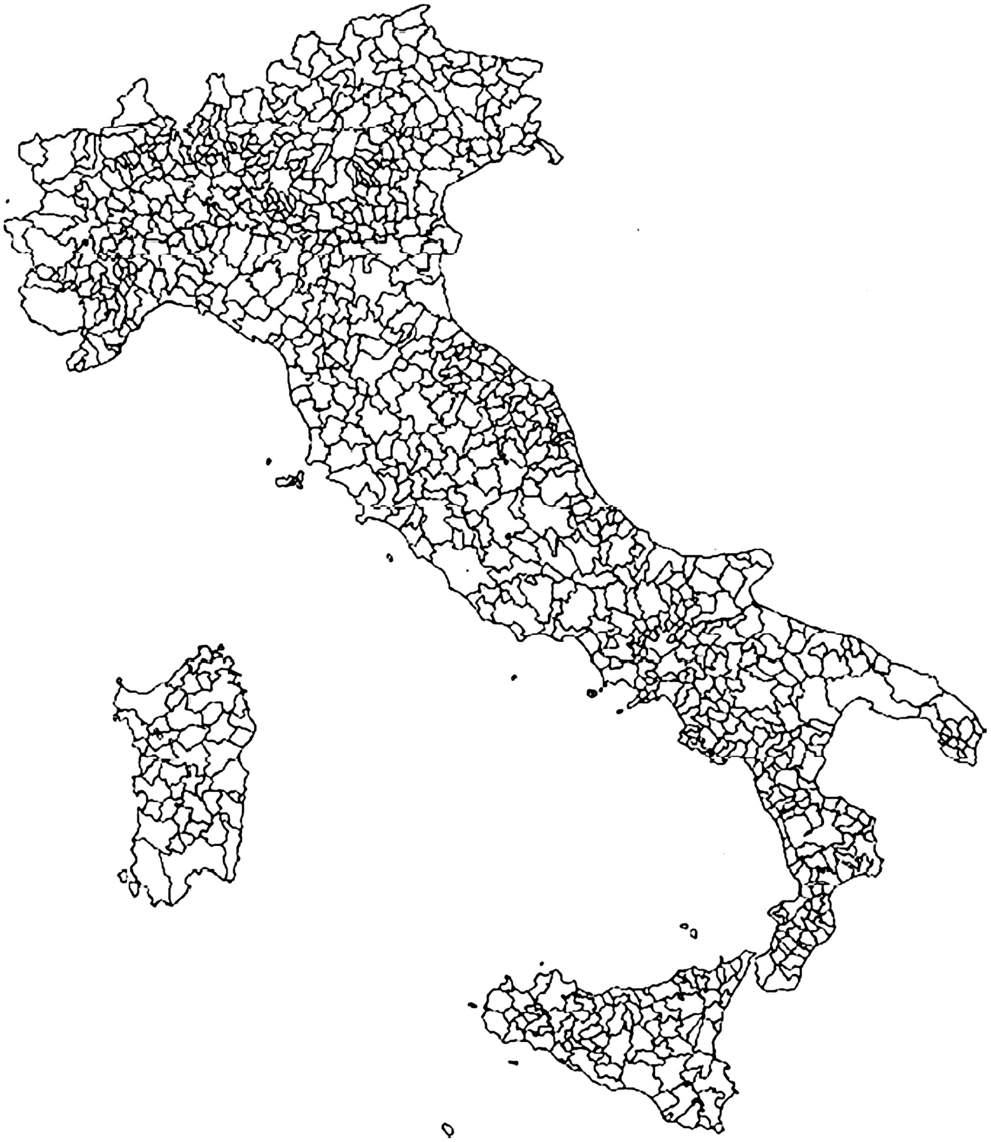


Figure 2.6: Local labour market areas, 1981 (Source: ISTAT-IRPET 1989).

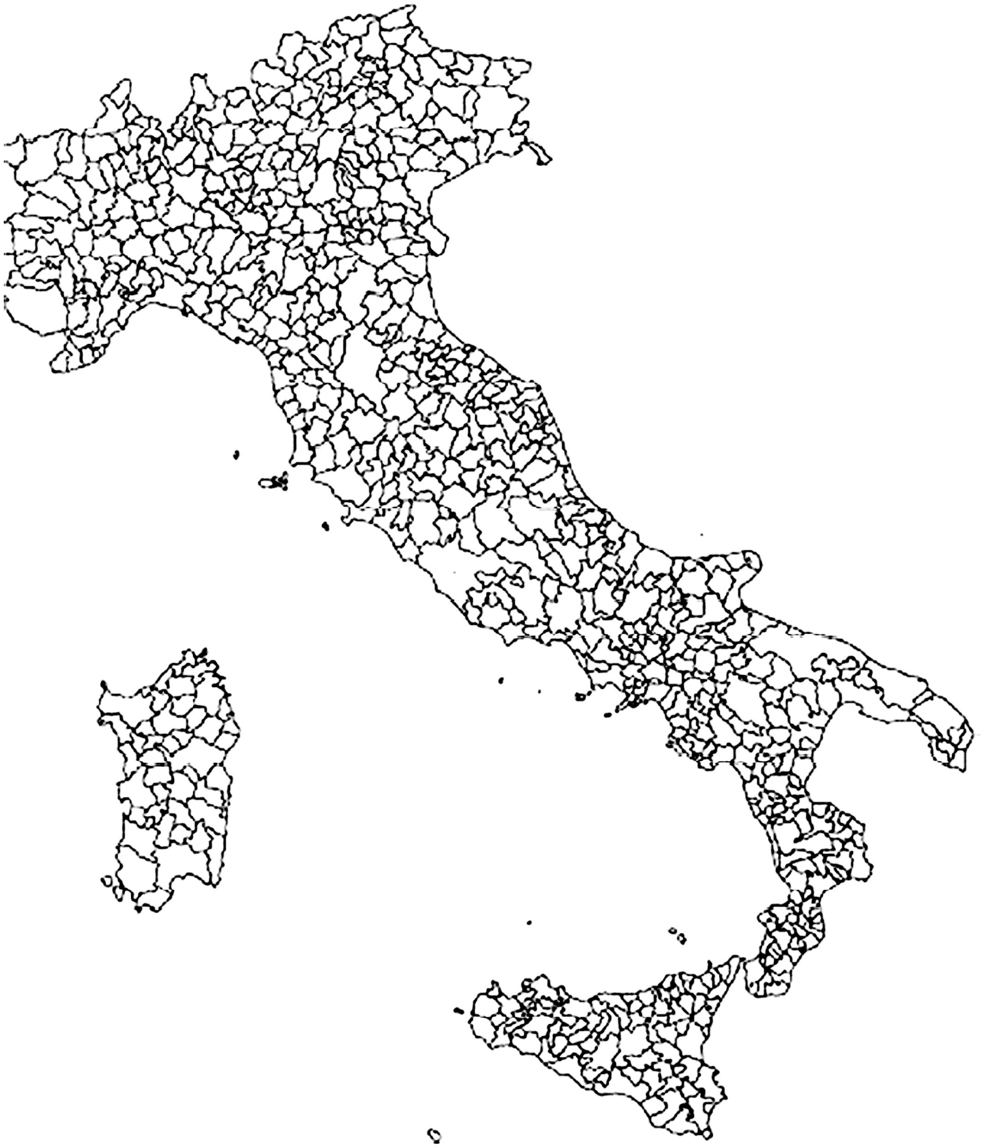


Figure 2.7: Local labour market areas, 1991 (Source: ISTAT 1997).

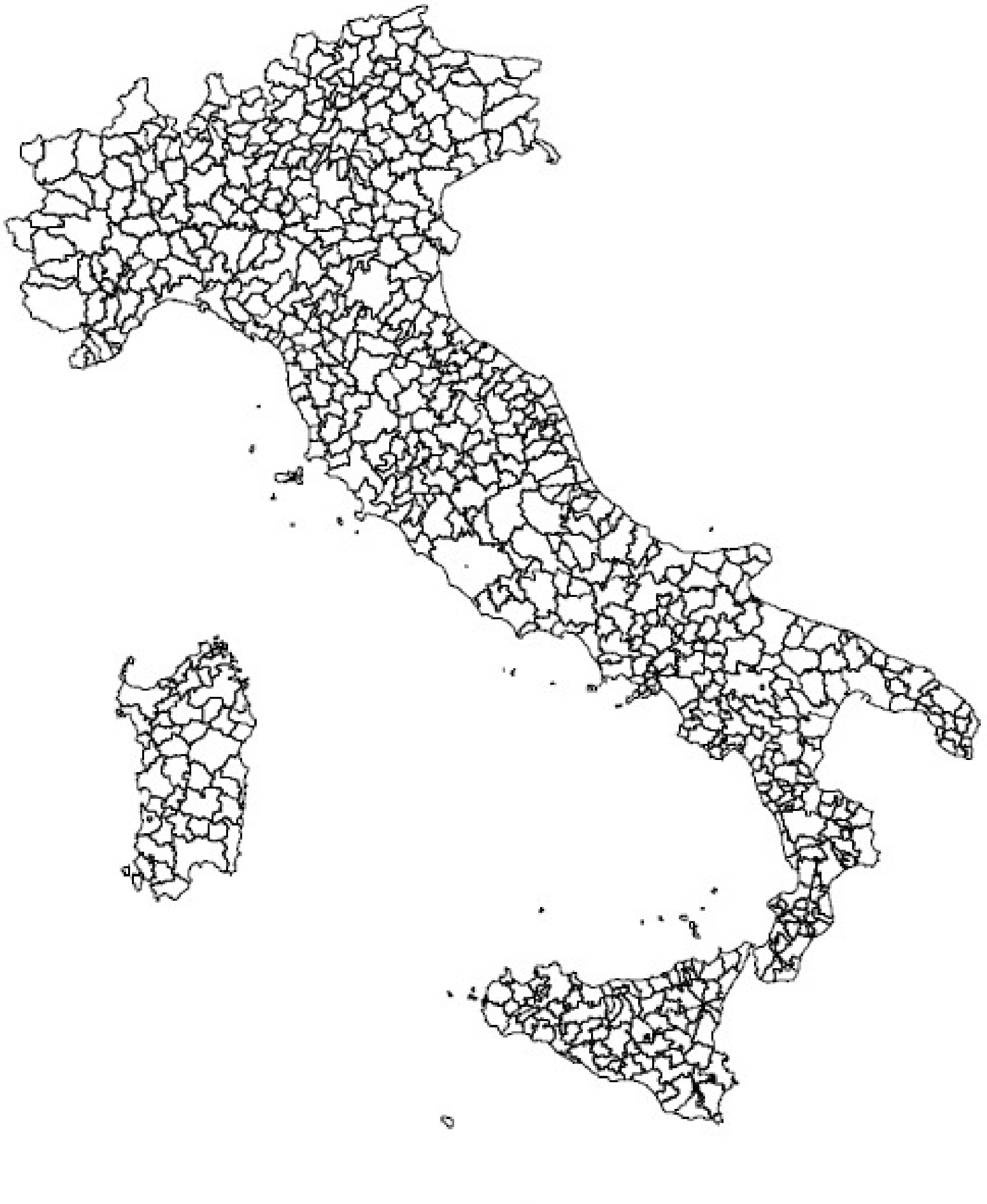


Figure 2.8: Local labour market areas, 2001 (Source: ISTAT 2005).

2.4 The Proposed Min-Cut Approach

By viewing the set A as the set of the vertices V of a graph, and the set of the values of F as the weights of a set of edges E' connecting all the pairs of vertices in V, the problem evidently becomes a type of *graph partitioning* problem over a complete graph $G' = (V, E')$.

However, differently from the standard cases, G' contains also loops, i.e., arcs of the type (i, i), going from i to i itself and corresponding to the mentioned internal flows. Since F is not symmetric, G' will be a directed graph.

Graph partitioning problems have been extensively studied (see [14] for references), since they have applications in many areas, e.g., clustering, detection of cliques in social, pathological and biological networks, programs mapping onto parallel architectures, image segmentation, numerical analysis, VLSI design.

Typically, graph partitioning problems fall under the category of NP-hard problems, and practical solutions algorithms are based on heuristics (see, e.g., [104]).

One widely used approach is the so-called multilevel one. Multilevel Partition algorithms iteratively reduce the size of the graph by collapsing vertices and edges, partition the smaller graph, then map back and refine this partition on the original graph. A good example of this approach is implemented in the software METIS [78]. We analyze the usability of similar approaches for solving the regionalization problem in the following Section 2.6.

However, we note that expression (2.3) for the computation of sc, has the following property: both numerator and denominator are constituted of sums of commuting flows, and f_{ij} is contained in one of those sums if and only if also f_{ji} is contained in it.

Therefore, we consider an undirected complete graph $G = (V, E)$ having for each arc $(i, j) \in E$ a weight

$$w(i, j) = f_{ij} + f_{ji}$$

In the framework of our study, to ignore the directionality of flows is not a negligence. The focus of our study does not require attention to the directionality of flows, because LLMAs are neither hierarchical functional areas, as in Fusco and Caglioni ([58]), nor nodal regions, as in Brown and Holmes ([20]). Therefore, what is important in our case is the total interaction between each pair of units. Graph G can be used instead of G' for the computation of (2.3) without any loss of information.

The advantage is that G has a considerably smaller number of arcs. We also observe that a partition solving our regionalization problem disconnects G in such a way that the arcs that are removed constitute a set that should have *small* total sum of the $w(i,j)$ flows and that certainly would not include loops. Therefore, when searching for such a set of arcs, we can remove all loops from G.

We obtain in this manner an undirected graph G, that we call *transitions graph*, having $\frac{n^2-n}{2}$ arcs, instead of the directed graph G' having n^2 arcs. Graph G does not contain the whole information of the commuting flows. Nevertheless, each solution to the original problem is obtainable as a partition of G, since loops would never be cut.

On the other hand, the internal flows are needed for the evaluation of self-containment. Therefore, our algorithm includes a validation step, during which the partitions obtained on G are checked for the satisfaction of condition (2.4) by computing sc as in (2.3), that is, by considering also the internal flows and the weights of the cutted edges.

As showed in the previous section 2.2, given a generic LLMA, an Overall self-containment $\geq c_1$ implies that both its Supply- and Demand-side self-containment are $\geq c_1$. As a consequence, we can guarantee that the partitions of G produced by our algorithm respect the conditions described in the section 2.2.

Note also that the internal flows generally represent the larger values in matrix F, and some of them may be order of magnitude larger than all non-internal flows. So, the restriction of the partitioning problem to G also improves the numerical condition of the problem, that may indeed be originally ill-conditioned.

We now describe the recursive partitioning procedure that we apply. We denote the set of vertices of a generic graph G as $V(G)$ or simply by V when there is no ambiguity.

A *cut* in G is a partition $(S,\bar{S})$ of V, a *cutset* is the set of arcs connecting S and $\bar{S}$ in G. We define the weight of cut $(S,\bar{S})$ as

$$W(S,\bar{S}) = \sum_{i \in S,\ j \in \bar{S}} w(i,j).$$

Similarly, we define for any two sets A, B of vertices of G, the weight

$$W(A:B) = \sum_{i \in A, j \in B} w(i,j)$$

A minimum weight cut, also called for brevity minimum cut or min-cut, of an undirected graph with edge weights is a set of edges with minimum sum of weights, such that its removal would cause the graph to become disconnected.

The total weight of the edges in a minimum cut of G is denoted by λ_G and called, as said before, $edge-connectivity$ of G (see, e.g., [38]).
The transitions graph G is partitioned in order to obtain the subsets of vertices $(S_1,\ldots,S_m)$ corresponding to the LLMAs $(R_1,\ldots,R_m)$ by finding cuts with minimum weight.

We also need to define an operation, called *contraction*, or equivalently *join* or *merge*, of two or more vertices of a graph G, that produces a graph with less vertices, as follows.

Given a graph G, the contraction of two vertices ν and μ produces a new graph $G/\nu \sim \mu$, where ν and μ are replaced by a new vertex $[\nu] = [\mu]$, and the weights of the edges (v, ν) and (v, μ), for any generic vertex $v \neq \nu, \mu$, are summed, i.e., $w(v, [\nu]) = w(v, \nu) + w(v, \mu)$.

Figure 2.9 shows an illustrative example of the contraction of two vertices.

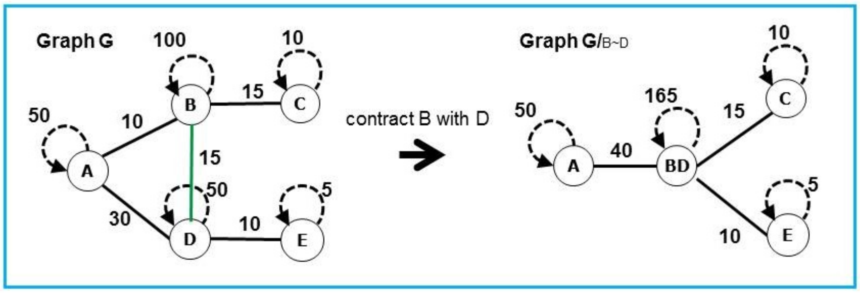


Figure 2.9: Contraction of a pair of vertices

The contraction of a set of vertices is the repeated contraction of its pairs of vertices. The contraction of a (sub-)graph is the contraction of the set of all its vertices.

Contraction algorithms rely on the following theorem, relating the *edge-connectivity* of a graph with the one of a quotient graph ([19]):

Theorem 1 (Thm 2.1 of [125])**.** *Let ν and μ be two vertices of an undirected weighted graph $G = (V, E)$. Then the edge-connectivity of G is the minimum of the weights of a minimum ν-μ-cut and the edge-connectivity of the graph $G/\nu \sim \mu$, obtained by the contraction of ν and μ, i.e.*

$$\lambda_G = min\ (\ \lambda_G\ (\nu,\ \mu)\ ,\ \lambda_{G/\nu\sim\mu}\ )$$

Proof. We differentiate two cases. Either each minimum cut of G separates ν and μ , then $\lambda_G = \lambda_G\ (\nu,\ \mu)$ and $\lambda_{G/\nu\sim\mu} > \lambda_G$, or there exists at least one minimum cut not separating ν and μ and hence induces a minimum cut of $G/\nu \sim \mu$, leading to $\lambda_G = \lambda_G\ (\nu,\ \mu) \leq \lambda_G\ (\nu,\ \mu)$.

□

We say that a (sub-)graph G is *feasible* when it satisfies conditions (2.4) and $|V(G)| \geq c_3$. We say that we *split* a (sub-)graph G when we remove from it all the arcs of a cutset. We also say that a (sub-)graph G is *unsplittable* when it is feasible but it has values of self-containment, workers and localities such that any further splitting of G will produce subgraphs that are not feasible.

Since our aim is to partition the transitions graph at the maximum extent, we try to obtain a partition corresponding to subgraphs that are all unsplittable. Therefore the proposed approach does not require the value *m* (cardinality of the partition) to be specifed *a priori*.

In the procedure below we use a list of *open problems* L to store all the subgraphs of G that have not yet been identified as unsplittable, and so will undergo the cutting operation. We also use a list of *closed problems* T to store the subgraphs G_p that have been recognized as unsplittable.

Procedure for the generation of LLMAs

Input An undirected complete graph $G(V, E)$ with n vertices associated with the n localities a_i, and edge weights $w(i, j) = f_{ij} + f_{ji} \geq 0$.

Output A partition of V into $(S_1, \dots, S_m)$ such that $|S_p| \geq c_3$ for $p = 1, \dots, m$, and that the corresponding $(R_1, \dots, R_m)$ respect condition (2.4) and are contiguous. Value m is not fixed in advance.

Initialization:
Remove from E all the edges with $w(i, j) = 0$
Identify the connected components of G, call them $G_0, \dots, G_c$ and insert them into the list of open problems L
The list of closed problems T is empty

Iteration:

Cut If L is empty: break the iteration and goto **Contiguity_enforcement**
If L is not empty: extract (sub-)graph G_h from L
Apply procedure **MinCut** to G_h to obtain the minimum weight cut $(S, \bar{S})_h$
Remove the corresponding cutset obtaining subgraphs G_{h+1} and G_{h+2}

Valid Check if G_{h+1}, G_{h+2} satisfy condition (2.4) and have at least c_3 localities each

Case 1 If both G_{h+1} and G_{h+2} satisfy these conditions: insert G_{h+1} and G_{h+2} in L and repeat the **Iteration**

Case 2 If neither G_{h+1} nor G_{h+2} satisfy these conditions: G_h is unsplittable
Insert G_h in T and repeat the **Iteration**

Case 3 If only one of them, say w.l.o.g. G_{h+1}, satisfies these conditions:
Contract G_{h+2} into a single vertex ν,
Contract ν with the vertex $\mu = \underset{v \in V(G_{h+1})}{\operatorname{argmax}} \{w(\nu, v)\}$
Add the obtained $G_{h+3} = G_{h+1}/\nu \sim \mu$ to L and repeat the **Iteration**

Contiguity_enforcement:

Split For each $G_p \in T$, consider the set $S_p = V(G_p)$. If S_p is non-contiguous, split it in its contiguous parts $S_{p1} \ldots S_{pk}$

Join Join each S_{pi} with the set $S_q = \underset{S:\ \text{co.}\wedge\text{ne.}}{\operatorname{argmax}} \{W(S : S_{pi})\}$, where
co. = contiguous and ne. = neighboring to S_{pi}
Return $(S_1, \ldots, S_m)$

During the initialization step, we remove all zero edges from G in order to further reduce the size of the problem. In case this operation disconnects G, we simply work independently on each of its connected components. Quite often, however, G remains connected. We use a depth-first search algorithm for finding the connected components in linear time (see, e.g., [70]).

After the initialization step, we repeatedly apply the iteration step, that means we split each (sub-)graph G_h contained in L by removing the edges corresponding to the minimum weight cut. The procedure for finding such cut is described below.

If the subgraphs G_{h+1} and G_{h+2} obtained in this way are still feasible, they may be even further splittable. Therefore, we insert them in L so that they will undergo a new cutting operation. On the contrary, if G_{h+1} and G_{h+2} are not feasible, this means that G_h is unsplittable, ans so G_h is inserted in T.

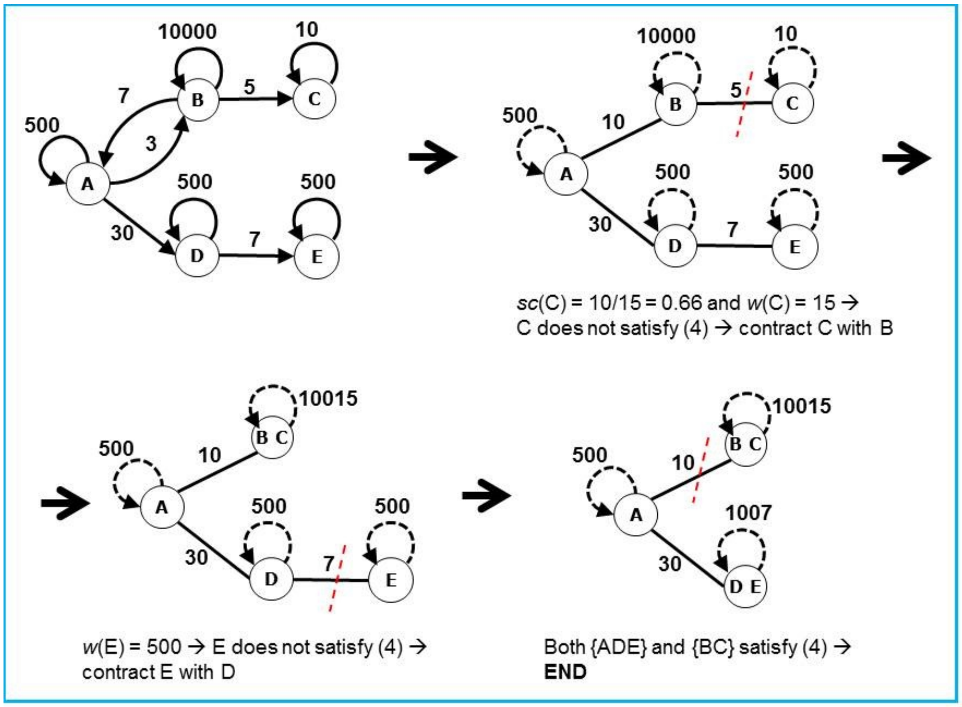
Finally, when exactly one of G_{h+1} and G_{h+2} is feasible, the infeasible one represents a set of localities that cannot remain alone but are strongly interconnected. Therefore, we contract them into a single vertex, and join it to the vertex of the feasible subgraph that maximizes the interaction.

Since the graph obtained by this operation may be even further splittable, we insert it in L. The procedure stops when L becomes empty, that means all the generated subgraphs are unsplittable. When this happens, T contains those m unsplittable subgraphs whose sets of vertices $(S_1, \ldots, S_m)$ constitute the wanted partition.

The sets of localities, called proto-LLMAs $(R_1, \ldots, R_m)$ corresponding to such sets must now be checked for geographical contiguity. Each non-contiguous R_p is split in its contiguous parts $R_{p1} \ldots R_{pk}$. Then, each part R_{pi} is joined with a contiguos neighboring proto-LLMA R_q such that the total weights of the arcs connecting R_{pi} and R_q is maximum.

The described procedure guarantees that the generated LLMAs satisfy conditions (2.4), (iii) and (iv), (or other conditions that may be imposed in the validation step). The number m of generated LLMAs is not guaranteed to be maximum, thought it is generally large enought.

Figure 2.10 shows an illustrative example in order to clarify the proto-LLMAs detection. We assume c = 0.75, c_1 = 0.75, c_2 = 1000, c_3 = 2. The evolution of the algorithm can be described as follows: we initially convert the directed graph into the transitions graph; then we cut edge BC; since C does not satisfy conditions (2.4) we contract it to B; then we cut edge DE; since E does not satisfy (2.4) we contract it to D; finally we cut edge A(BC) and we obtain two feasible subgraphs. therefore, the final proto-LLMAs are ADE and BC.



sc(C) = 10/15 = 0.66 and *w*(C) = 15 →
C does not satisfy (4) → contract C with B

w(E) = 500 → E does not satisfy (4) →
contract E with D

Both {ADE} and {BC} satisfy (4) →
END

Figure 2.10: Detection of proto-LLMAs

To compute minimum cut on undirected graphs with nonnegative real edge weights there exist in the literature many methods. One group of algorithms is based on the well-known result of Ford and Fulkerson [56] regarding the duality of maximum s-t-flows and minimum s-t-cuts for arbitrary vertices s and t. Following this approach, Hao and Orlin [66] shown an algorithm to solve all the necessary max-flow problems in time asymptotically equal to one max-flow computation, requiring $O(|V| \times |E| \log(|V|^2/|E|))$ steps.

Nagamochi and Ibaraki [101] described an algorithm without using maximum flows. Instead, they construct spanning forests and iteratively contract edges with high weights. This lead to an asymptotic runtime of $O(|V| \times |E| + |V|^2 \log |V|)$ on undirected graphs with nonnegative real edge weights. Their approach was refined in [125] by Stoer and Wagner, by replacing the construction of spanning forests with the construction of Maximum Adjacency (MA) order, and by Brinkmeier [19] by contracting more than one pair of vertices if possible by introducing an alternative data structure called *priority queues with threshold*.

Particularly, the vertices of a graph $G = (V, E)$, are arranged in a MA order, if, for each vertex v_i, with $i > 1$, the sum of the weights from v_i to all preceding vertices $v_1, \ldots, v_{i-1}$ is maximal among all vertices v_k with $k \geq i$ [101, 125, 102, 19].

A Maximum Adjacency (MA) Order is defined as follows.

Definition 2.1 (Maximum Adjacency order)**.** *Let $G = (V, E)$ be an undirected weighted graph. An order $v_1, v_2, \ldots, v_n$ on the vertices of G is a Maximum Adjacency order, if*

$$w(v_1, v_2, \ldots, v_{i-1}; v_i) := \sum_{j=1}^{i-1} w(v_i, v_j) \geq \sum_{j=1}^{i-1} w(v_k, v_j) \geq =: w(v_1, v_2, \ldots, v_{i-1}; v_k)$$

for all $k \geq i$.

The values $w(v_1, v_2, \ldots, v_{i-1}; v_i)$ are called *adjacencies*.

Lemma 2.2 (Lemma 3.1 of [125])**.** *For each MA order $(v_1, v_2, \ldots, v_n)$ of the undirected weighted graph $G = (V, E)$, the cut $(\{v_1, v_2, \ldots, v_{n-1}\}, \{v_n\})$ is a minimum $v_n - v_{n-1}$-cut.*

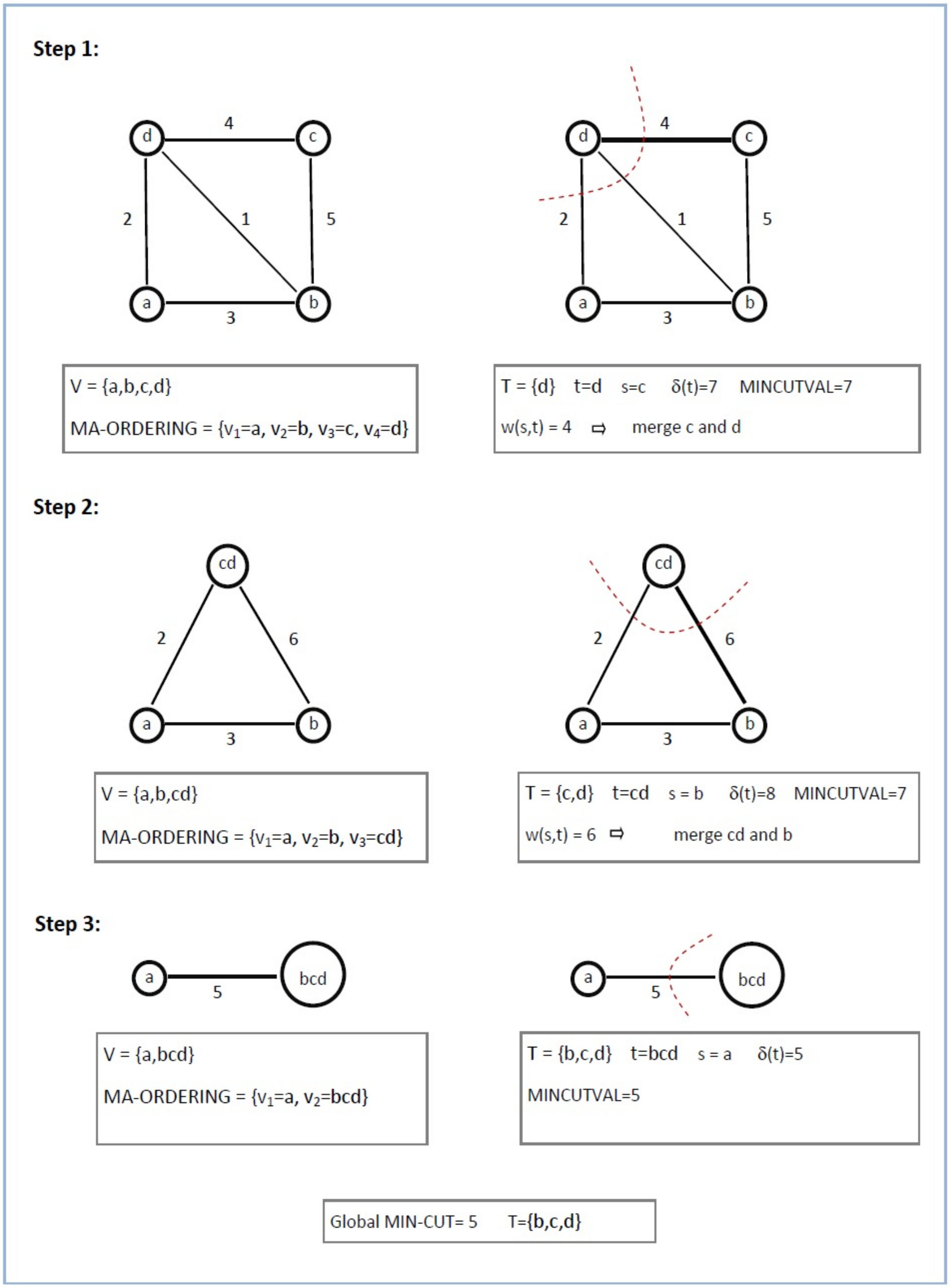
Therefore, in an MA order the degree of v_n is equal to the weight of a minimum $v_n - v_{n-1}$-cut in G.

Incorporating these improvements, the algorithm obtains an asymptotic runtime of $O(|V|^2 \lambda_G)$ for undirected graphs with nonnegative integer weights, as it is the case of our transitions graph, and this is the algorithm that we apply.

The procedure for finding the minimum weight cut is described below.

```
Procedure MinCut

Input  An undirected connected $G(V, E)$ with edge weights $w(i, j) \geq 0$
Output A cut $(S, \bar{S})$ in $G$ with minimum weight

Initialization:
Chose any vertex $\in V$ and call it $v_1$
Let $n = |V|$. Let $S = \{v_1\}$

for i = 2 to n:
    Let $v_i$ be the vertex corresponding to $\underset{v \in V \setminus S}{\operatorname{argmax}} W(S : \{v\})$
    Let $S := S \cup v_i$
endfor

if n = 2: return the cut $(\{v_1\}, \{v_n\})$
else: return the minimum cut between $(\{v_1, \ldots, v_{n-1}\}, \{v_n\})$
and MinCut($G/v_{n-1} \sim v_n$)
```

The algorithm, as shown in Figure 2.11 starts with any vertex, say v_1, and builds an ordering of the vertices by always adding to the set S of the selected vertices the vertex whose total weight to S is maximized. This provides an MA ordering. After this, the cut induced by the last vertex in the ordering is considered, as well as the cuts obtained by recursively applying the procedure to the graph obtained by contracting the last two vertices. The minimum among the cuts obtained during these recursions is the global minimum weight cut of the graph.



V = {a,b,c,d}

MA-ORDERING = {v_1=a, v_2=b, v_3=c, v_4=d}

T = {d} t=d s=c δ(t)=7 MINCUTVAL=7

w(s,t) = 4 ⇨ merge c and d

V = {a,b,cd}

MA-ORDERING = {v_1=a, v_2=b, v_3=cd}

T = {c,d} t=cd s = b δ(t)=8 MINCUTVAL=7

w(s,t) = 6 ⇨ merge cd and b

V = {a,bcd}

MA-ORDERING = {v_1=a, v_2=bcd}

T = {b,c,d} t=bcd s = a δ(t)=5

MINCUTVAL=5

Global MIN-CUT= 5 T={b,c,d}

Figure 2.11: Global min-cut using Maximum Adjacency Order.

2.5 K-way Graph Partitioning Problem

The K-way graph partitioning problem has extensive applications in many scientific and engineering areas. It consists in partitioning the vertices of a graph into$K$ subsets with specific properties based on size or cardinality constraints.

The K-way graph partitioning problem is defined as follows:

Given a graph $G = (V, E)$ with nonnegative real edge weights, the problem ask for a partition P_k of V with K subsets of vertices $P = \{V_1, \ldots, V_k\}$ such that:

1) $\bigcup_{i=1}^{K} V_i = V;$

2) $V_i \cap V_j = \phi$ for $i \neq j;$

3) a balance constraints (optional) demand that for all $i \in \{1, \ldots, k\}$:
$|V_i| \leq Lmax = \frac{|V|}{K}(1 + \alpha)$, for some *imbalance parameter* $\alpha \geq 0$;

4) the edge cut of the partition is minimized.

Generally, graph partitioning problems are NP-complete. Solutions to this problem are often derived using heuristics and approximation algorithms. They are a trade-off between run-time and solution quality. The two main classes of graph partition methods are $global$ and $local$.

Particularly, global approaches take in account the properties of the entire graph. A well knows global method is Spectral partitioning, which derives the partition from the Laplacian Spectrum of the graph [95, 96]. This method uses the eigenvector of the second smallest eigenvalue of the Laplacian matrix to find a small separator of a graph. Among all eigenvalues of the Laplacian Spectrum of graph one of the most popular is the second smallest, called by Fiedler [52, 53] the algebraic connectivity of a graph. Its importance is due to the fact that it is a good parameter to measure how a graph is connected.

The spectral method of graph partitioning was proposed by Donath and Hoffman (1972) [41, 42], who has first suggested using the eigenvectors of adjacency matrices of graph to find partitions.

Fiedler investigated the properties of the second smallest eigenvalue and the corresponding eigenvector of the Laplacian of a graph with its connectivity and suggested partitioning by splitting vertices according to their value in the corresponding eigenvector.

Spectral method for graph partitioning have been known to be robust but computationally expensive. Therefore, many heuristic algorithms was proposed in literature for computing vertex separator from the second eigenvector of the Laplacian [112].

Other approaches to computing vertex separators in sparse matrix have been considered by several researchers. Local methods as the *Kernighan-Lin* alghoritm [79] and the *Fiduccia-Mattheyses* algorithm [51] are among the most widely used. They are iterative min-cut heuristic that find optimal partition maintaining a desired balance based on size of the subsets.

Kernighan-Lin algorithm starts with an initial bipartition of the graph. In each iteration, it searches for a subset of vertices from each part of the graph such that swapping them leads to a partition with a smaller edge-cut. Each iteration takes $O(|E|\log(||E|))$). Fiduccia-Mattheyses algorithm is an improvement on the original Kernighan-Lin algorithm. It reduces complexity to $O(|E|)$ by using appropriate data structures.

In the 1990s a number of researchers have proposed Multilevel Graph partitioning schemes [23, 67, 5, 78]. These algorithms, in order to improve the computational time, reduce the size of the graph by collapsing vertices and edges, partition the smallest graph in the sequence and, finally, project the coarse partition back through the sequence of graphs improving it with a local refinement algorithm.

This approach provides a good compromise between the run-time (complexity) and the quality solution. In fact, Software packages based on this approach as Chaco [68], Metis [74] and Scotch [110] have been extremely successful. Today, too, for large problems the partitioning itself must be done in parallel. The most important parallel software packages for Graph partitioning as ParMetis [75], mt-Metis[83], PT-Scotch [111], and Zoltan [37] use variations of the Multilevel Graph partitioning algorithm.

In the next Section, the main phases of the Multilevel Graph Partitioning schemes and the main characteristics of software package METIS have been described. After the choice of a method highly specialized to determine the Italian LLMAs, another comparison has been made with the state-of-the-art procedure for general purpose graph partitioning METIS [78], not yet applied in the functional regionalization context.

Therefore, METIS is selected for the second comparison with the min-cut method and used in the following approach proposed based on a Multilevel graph partition approach.

2.5.1 Main characteristics of the Multilevel Graph Partitioning Schemes

The Multilevel partitioning scheme has a very simple basic structure having three main steps [78] summarized below:

(1) Coarsening

Obiective: construct a sequence of increasingly coarse approximations of the originary graph.

A sequence of successively smaller graphs is constructed. The original graph G_0 is used to generate a series of coarser graphs $G_1, G_2, \ldots, G_N$, such that $|V_0| > |V_1| >, \ldots, |V_N|$. The size of the graph G_0 is reduced by collapsing vertices and edges. The purpose is to create a small graph G_N, such as its bisection is not significantly worse than the bisection directly obtained to G_0.

(2) Initial partitioning

Obiective: partitioning of the smallest coarser graph.

An high-quality partition (i.e., small edge-cut) of the coarse graph G_N is computed. Many different algorithms can be used without significantly affecting the overall runtime and the partition quality.

(3) Uncoarsening and refinement

Obiective: projection of the coarse partition back throughand and reduction of the edge-cut.

In sequence of graphs derived, the initial partition is improved with a local refinement algorithm. This is done by firstly projecting the partition of G_{i+1} to G_i, followed by partitioning refinement whose goal is to reduce the edge-cut by moving vertices among the partitions.

Figure 2.12 shows the main phases of the Multilevel Graph Partitioning schemes. Dashed lines referred to the partitioning projected from the coarse graph, while solid shaded lines referred to refined partitioning.

The multilevel partitioning algorithms find reasonably good partitions in a reasonable computational time. The overall effectiveness of the multilevel strategy depends on the selected algorithms to identify the set of all vertices that will be contracted during the coarsening phase and the partitioning refinement phase. This selection also depends on the type of data to be processed and the objectives to be achieved.

In our application, we have used METIS. It is a general purpose graph partitioner, not specifically designed for identification of Functional Regions, but aiming at finding high quality partitions for several problems. Particularly, METIS

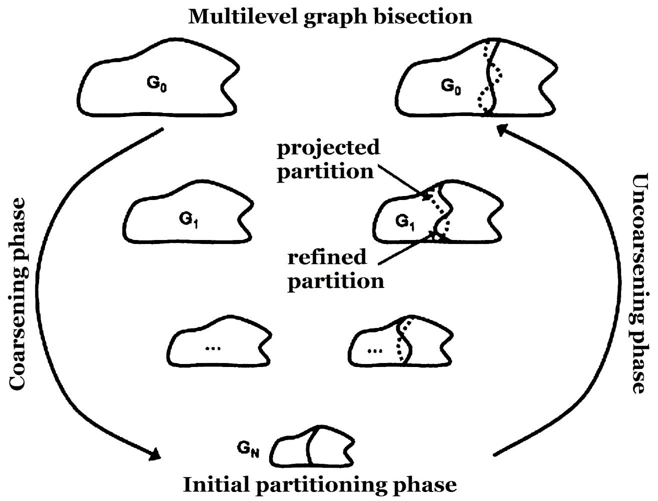


Figure 2.12: Main phases of the Multilevel Graph Partitioning schemes

is a software package for partitioning large graphs. The algorithms implemented by METIS are based on the multilevel graph partitioning schemes described in ([73, 76, 77]) . This Software package is copyrighted by the regents of the University of Minnesota.

The hMETIS Multilevel Partitioning Scheme is composed by four phases, described below:

(1) Coarsening. During this phase METIS uses algorithms that make it easier to find a good-quality partition at the coarsest graph. During this phase, a sequence of successively smaller graphs is constructed. The purpose of coarsening is to create a small graph, such that a good bisection of the small graph is not significantly worse than the bisection directly obtained for the original graph. The group of vertices, that are contracted together to form single vertices in the next level coarse graph, can be selected in different ways. METIS implements various grouping schemes (also called matching schemes). A detailed description of some of these schemes can be found in [73].

(2) Initial Partitioning phase. During this phase a bisection of the coarsened graph is computed. Since this graph has a very small number of vertices (usually less than 100 vertices) many different algorithms can be used without significantly affecting the overall runtime and quality of the algorithm. METIS uses multiple random bisections followed by the Fiduccia-Mattheyses ([51]) refinement algorithm.

(3) Uncoarsening and refinement phase. During this phase, the partitioning of the coarsest graph is used to obtain a partitioning for the finer graph. This is done by successively projecting the partitioning to the next level finer graph and using a partitioning refinement algorithm to reduce the cut and thus improve the quality of the partitioning. Since the next level finer graph has more degrees of freedom, such refinement algorithms tend to improve the quality.METIS implements a variety of algorithms that are based on the Fiduccia-Mattheyses algorithm. The details of some of these schemes can be found in [73].

(4) V-Cycle Refinement (optional). Duringthis phase, METIS focuses primarily on the portion of the graph that is close to the partition boundary. These highly tuned algorithms allow METIS to quickly produce good-quality partitions for a large variety of graphs. The idea behind this refinement algorithm is to use the power of the multilevel paradigm to further improve the quality of a bisection. The V-cycle refinement algorithm consists of two phases, namely a coarsening and an uncoarsening phase. The coarsening phase preserves the initial partitioning that is input to the algorithm. We will refer to this as restricted coarsening scheme. In this scheme, vertices that belong only to one of the two partitions are merged to form the vertices of the coarse graphs correspond to. As a result, the original bisection is preserved throughout the coarsening process, and becomes the initial partition from which we start performing refinement during the uncoarsening phase. The uncoarsening phase of the V-cycle refinement algorithm is identical to the uncoarsening phase of the multilevel graph partitioning algorithm described earlier. It moves vertices between partitions as long as such moves improve the quality of the bisection. Note that the various coarse representations of the original graph, allow refinement to further improve the quality as it helps it climb out of local minima.

Recently, similar techniques have been rediscovered to find *network communities* in real-world networks. Methods used for complex network community detection provide an approach to the identification of LLMAs. An analysis of the characteristics of the main heuristic used for network community detection is carried out by Leskovec *et al.* [87]. They explore a range of such methods in order to compare them and to understand their relative performance and the systematic biases in the networks community they identify. They evaluate several common objective functions that are used to formalize the notion of a network community, and they examine several different classes of approximation algorithms that aim to optimize such objective functions. In particular, they compare structural properties of network community extracted by two methods based on two completely different computational paradigms: a spectral based graph partitioning method called Local Spectral [1] and METIS [78]. The latter is an effective graph partitioner used for finding low-conductance cuts that may be followed by MQI [85], an exact flow-based technique for obtaining the lowest conductance cut whose small side is contained in one of the two half-graphs produced by METIS.

Analyzing the network community profile (NCP) of a large number of communities, they highlight that METIS is generally better than Local Spectral at the nominal task of finding cuts with low conductance, although some of METIS clusters may be internally disconnected.

In this thesis we evaluated also a regionalization approach to find Functional Regions based on a Multilevel graph partition approach via Metis. It is important to consider that the structure of the regionalization problem differs fundamentally from graph portioning problem, because the size of the subsets (regions) are unknown in advance and subset are subject of self-containment constraints.

2.6 Experimental Results

The procedure described in Section 4 was implemented in C++ and tested by generating the LLMAs for all the Italian administrative regions. The commuting flows considered for this test were gathered via the 2001 Italian Population Census. This Census collects data about inter-municipal commuting flows. Thus, municipalities are the localities constituting our basic units of data. The target was to meet conditions (2.4), (iii) and (iv), with $c = 0.75, c_1 = 0.75, c_2 = 1000, c_3 = 2$. The experiments were conducted on a PC Intel Pentium CPU 3.10 Ghz with 4Gb of RAM under MS windows7 64 bit Operating System.

We first compare our algorithm with the reference algorithm based on [120] and described in Section 2.3.2 For each Italian administrative region, Table 1 reports the number of LLMAs generated by our algorithm based on minimum cuts; their average value of Overall self-containment (mean sc); their minimum value of Demand- and Supply-side self-containment (min sc^r, min sc^w); time required for processing the whole administrative region; the number of LLMAs generated by the reference algorithm; all the values already provided for the first algorithm. Conditions (2.4), (iii) and (iv) are always satisfied by the two approaches. Times are in minutes and are computed on the same machine. Although both methods were successful in generating feasible LLMAs, our approach showed to be able to generate a number of LLMAs that is (often considerably) larger than the reference algorithm in all cases. In addition, the values of self-containment of the LLMAs generated by our approach are closer (from above) to the self-containment threshold than those provided by the reference algorithm. These are positive features for a regionalization algorithm, as explained in Section 2. Hence, the proposed method provides LLMAs with better statistical quality. Times needed by the proposed algorithm are generally much shorter than those of the reference algorithm. While the total running time of our procedure, for all Italian administrative regions, is 8 hours and 22 minutes, the total running time of the reference procedure is 16 hours.

	Our Algorithm					Reference Algorithm				
Region	LLMAs	mean sc	min sc^r	min sc^w	time	LLMAs	mean sc	min sc^r	min sc^w	time
Piemonte	36	0.69	0.67	0.71	90	35	0.70	0.71	0.75	239
V. d'Aosta	4	0.71	0.73	0.73	4	3	0.83	0.89	0.86	21
Lombardia	55	0.65	0.63	0.68	320	54	0.67	0.66	0.76	452
Trentino AA	33	0.66	0.61	0.72	4	28	0.74	0.64	0.76	16
Veneto	37	0.66	0.65	0.70	19	35	0.70	0.65	0.76	34
Friuli VG	10	0.68	0.65	0.75	3	6	0.75	0.65	0.78	7
Liguria	17	0.68	0.62	0.72	3	16	0.72	0.64	0.77	13
Emilia Rom.	45	0.68	0.63	0.74	4	41	0.68	0.64	0.76	10
Toscana	47	0.69	0.68	0.69	3	40	0.73	0.73	0.76	7
Umbria	15	0.74	0.66	0.78	1	14	0.73	0.68	0.77	5
Marche	34	0.66	0.67	0.73	2	29	0.69	0.67	0.76	7
Lazio	21	0.70	0.62	0.80	7	18	0.71	0.63	0.80	23
Abruzzo	22	0.68	0.64	0.64	4	20	0.70	0.66	0.76	15
Molise	13	0.69	0.66	0.72	1	9	0.76	0.72	0.80	5
Campania	54	0.66	0.66	0.68	16	49	0.67	0.62	0.75	32
Puglia	48	0.68	0.65	0.67	3	34	0.71	0.68	0.75	8
Basilicata	21	0.71	0.67	0.78	1	18	0.72	0.72	0.76	5
Calabria	60	0.67	0.64	0.69	7	47	0.72	0.66	0.79	25
Sicilia	66	0.69	0.63	0.70	5	48	0.74	0.70	0.76	17
Sardegna	44	0.69	0.65	0.69	5	38	0.72	0.69	0.76	18

Table 2.1: Comparison of our and reference algorithm on all Italian administrative regions

Then, we compare our algorithm with the state-of-the-art graph partitioner METIS [78]. METIS takes in input the number of clusters (the LLMAs in our case) that should be produced. This number was assigned to the same number of LLMAs produced by our algorithm in order to make a fair comparison of the regionalization quality. For each Italian administrative region, Table 2 reports the number of vertices (localities) and of edges (linkages) of the transitions graph, and an analysis of the self-containment of the LLMAs produced by the two algorithms, by giving their minimum and maximum values, their average and their variance. METIS produces LLMAs that often are not enough self-contained (they often do not satisfy conditions (2.4) and (iii)), and the self-containement values of the LLMAs generated by METIS are much more variable than those of the LLMAs generated by our algorithm. Indeed, some lack of internal cluster connection is quite intrinsic in METIS approach. Therefore, the proposed algorithm appears a better option for performing a similar functional regionalization, although METIS is a faster graph partitioner.

				Our Algorithm				METIS Algorithm			
Region	$\mid V \mid$	$\mid E \mid$	LLMAs	min sc	max sc	mean sc	var sc	min sc	max sc	mean sc	var sc
Piemonte	1206	35903	36	0.57	0.89	0.69	0.007	0.26	0.85	0.61	0.033
V. d'Aosta	74	1021	4	0.64	0.89	0.71	0.010	0.58	0.85	0.70	0.017
Lombardia	1546	91625	55	0.56	0.87	0.65	0.006	0.27	0.89	0.55	0.023
Trentino AA	339	5826	33	0.57	0.95	0.66	0.006	0.21	0.89	0.61	0.034
Friuli VG	219	7168	37	0.57	0.82	0.66	0.006	0.31	0.89	0.61	0.020
Veneto	581	24926	10	0.57	0.87	0.68	0.009	0.27	0.90	0.65	0.035
Liguria	235	3523	17	0.57	0.95	0.68	0.017	0.34	0.95	0.64	0.040
Emilia Rom.	341	9529	45	0.57	0.85	0.68	0.006	0.29	0.85	0.58	0.024
Toscana	287	7462	47	0.58	0.91	0.69	0.006	0.30	0.92	0.63	0.025
Umbria	92	1321	15	0.58	0.87	0.74	0.008	0.39	0.90	0.67	0.032
Marche	246	5586	34	0.56	0.84	0.66	0.005	0.30	0.95	0.60	0.023
Lazio	378	8729	21	0.56	0.92	0.70	0.007	0.27	0.89	0.59	0.036
Abruzzo	305	5615	22	0.56	0.87	0.68	0.010	0.32	0.86	0.60	0.033
Molise	136	1785	13	0.59	0.82	0.69	0.006	0.38	0.86	0.61	0.023
Campania	551	20590	54	0.56	0.80	0.66	0.004	0.33	0.84	0.58	0.017
Puglia	258	7372	48	0.56	0.83	0.68	0.005	0.36	0.89	0.60	0.020
Basilicata	131	1958	21	0.61	0.86	0.71	0.004	0.39	0.90	0.68	0.019
Calabria	409	8247	60	0.57	0.87	0.67	0.004	0.30	0.88	0.62	0.015
Sicilia	390	9189	66	0.57	0.93	0.69	0.007	0.33	0.89	0.65	0.019
Sardegna	377	7686	44	0.56	0.88	0.69	0.008	0.31	0.90	0.62	0.028

Table 2.2: Comparison of our algorithm and METIS on all Italian administrative regions

2.7 Conclusions

We proposed an innovative multi-step approach to the problem of the generation of LLMAs by using techniques of Combinatorial Optimization. This procedure is based on the iterative partitioning of the transitions graph, which represents the interaction among the localities of the territory under analysis. The proposed procedure works at the formal level, hence it can be used for other problems of different origin but sharing the same structure. Since the arising minimum cut problems can be solved to optimality in extremely short times by using state-of-the-art min-cut algorithms, the procedure is able to generate LLMAs in large real-world networks such as the Italian administrative regions in times that are very reasonable and much shorter than the official reference method used for comparison. The statistical quality of the partitions generated by the proposed method is generally better than that obtained by using the reference method, and clearly better than that obtained by using a general purpose graph partitioner not specifically designed for this task.

Chapter 3

A Combinatorial Optimization Approach to the Selection of Statistical Units

3.1 Introduction

In a statistical survey, the target population (*scope*) is the subset of statistical *units* that should be surveyed. In many large surveys, the scope cannot be the list of all possible units, because otherwise the cost or the complexity would be prohibitive. In other words, the scope should be selected, also on the basis of economic assessment. In particular, in the case of Agricultural Censuses, slightly differently from a more traditional approach, the survey population excludes the farms that are too small or economically irrelevant.

Over the time, the European (EU) legislation has introduced specific restrictions to the census scope, in order to reduce survey costs and promote data comparability among countries.

In the past, for the Italian Agricultural Censuses of 1981,1991 and 2001, the census scope was determined ex-post (before transmitting national data to Eurostat and after exhaustive data collection of all units), by applying the parameters defined at European level. Specifically, up to 2007, the census scope according to EU criteria included all farms (also the exclusively forested or zoo-technique ones) with at least one hectare of Utilized agricultural Area (UA), or the farms having at least a fixed amount of products commercialized in the reference year of the survey.

In the last round of agricultural census (2010), for determining the target population, the Community Regulation (EC, 2007) has identified a group of compulsory criteria, based exclusively on physical thresholds, replacing the previous criteria. The European Regulation introduced the use of a single physical threshold grid, related to the main cultivations and animal breeding, so that surveyed farms represent at least 98% of the UA and at least 98% of livestock units. If the suggested thresholds were not sufficient to guarantee the minimum national coverage settled, each Member Country could have adopted alternative minimum physical thresholds.

In Italy, the adoption of the EU thresholds would have resulted in different degrees of coverage among the regional areas, mainly for particular specialized productions (e.g. flowers, olive plantations). The analysis of the regional coverage levels has highlighted the need to include in the scope some of the units that, due to their size, would have been excluded by the recommended thresholds. Therefore, lower minimum levels of UA were identified and applied to the Italian farms with less than one hectare of UA, thus increasing the total national coverage too.

A main problem, in similar cases, is establishing criteria or somehow tracing boundaries for dividing the units that should be included or not. Theoretically, the problem can be described as follows: there is a very large set of statistical units (e.g. farms, companies, etc.) belonging to the target population. Surveying each of them has a *cost* and represents a different portion of the whole statistical *information* (e.g. the state of agriculture, the industrial production, etc.). The *coverage constraints* require to select only a subset of the whole statistical information, taking into account the minimum total cost too.

An important additional challenge is that the information for identifying eligible units is not perfectly known before surveying it. This often happens because the units may have been surveyed only during a previous Census, typically held several years before, and the information is integrated using administrative data not properly updated, so their characteristics may have changed in the meanwhile. It therefore needs to establish *reliable inclusion criteria* on the basis of the available data describing the unit, sometimes outdated.

This problem is often called Scope Selection, or statistical Universe Selection, see e.g. [44, 46], and evidently contains a Combinatorial Optimization structure, with the optimal solution being one of the feasible subsets of the ground set of all the units. Defining a scope has connections with the general statistical task of Population Definition, see e.g. [63, 123]. The selection of statistical units is also considered in [50]. Other optimization models arising from the treatment of agricultural data are in [9, 10, 98].

The problem of selecting units from a list can also be viewed as a particular case of the wider problem of quota sampling, for which several approaches and techniques of purposive sampling have been proposed, see also [81, 97, 126]. After the selection problem has been solved, the coverage levels should be checked, or in other words the risk of undercoverage [7] must be evaluated.

While similar problems are usually solved by means of a variety of ad hoc techniques, whose features typically depend on the specific application, we propose a more general approach, based on the use of a binary linear model and solved by means of Combinatorial Optimization techniques.

This innovative approach to the problem overcomes the key features of other methods, and moreover allows taking advantage of effective algorithms already developed in this latter field. More precisely, by using binary variables associated with the above mentioned units, the described selection problem is here modelled as a multidimensional binary knapsack problem (see e.g. [104]).

Since those models may reach in many cases very large dimensions, a separated procedure [2, 65, 140] is also needed, to assess the coverage level at the end of each iteration. A solution to the above knapsack model will be referred to as an Optimal Selection.

However, due to the above described uncertainty aspects, not just one but a sequence of Optimal Selection problems must be solved. More specifically, in order to develop *inclusion criteria* based on thresholds, we need to evaluate the safety margins, related to the risk of undercoverage for different inclusion thresholds.

The procedure has been implemented in c++ and tested, in cooperation with the Italian National Institute of Statistics (Istat), on real data from the 5th Agricultural census. The results are very encouraging, both from the computational and the statistical point of view. The main contribution of this work is, therefore, an innovative and effective approach based on Combinatorial Optimization for solving a challenging large-sized and economically important real-world problem.

The work is organized as follows.

Section 2 describes the basic model proposed for the selection of an optimal subset of statistical units and the techniques to improve the formulation and therefore solving the overall model by means of a Branch&Cut approach. Section 3 explains how the solution of the Optimal Selection problem, under different conditions, leads to the determination of the inclusion criteria based on thresholds. In Section 4, we provide extensive results on real-world data from the Italian Agricultural Census. Finally, in Section 5 the conclusions and in Section 6 a future case study.

This work has been published in [12].

3.2 Solving the Optimal Selection Problem

The model proposed for the above problem will be hereinafter explained by referring to the specific case of the Agricultural Census. This is probably the most important case, because it has a great economic relevance and a very large dimension. Moreover, in the case of EU countries, gathered information must be published and provided to the EU level, where they constitute a basis for assigning financial resources, for planning production, and for several other economical European policies. However, the proposed model is not intrinsically limited to that case, but can be used for similar cases of Scope Selection problems.

In the Agricultural Census, there is a very large list $U = \{u_1, \ldots, u_n\}$ of all the existing statistical units that could be surveyed. Each unit u_i represents a farm, which is described by the areas used for every cultivation (plus other data not relevant for this work). Units have therefore the following structure:

$$u_i = \{a_{i1}, \ldots, a_{im}, a_{i\mathrm{T}}\} \qquad \text{for } \ i = 1, \ldots, n$$

where a_{ij} is the area that farm i uses for cultivation j, with $i = 1, \ldots, n$ and $j = 1, \ldots, m$, and $a_{i\mathrm{T}}$ is the total area of farm i used for cultivations, technically called Utilized agricultural Area (UA).

Unfortunately, at least for the majority of the cases, the data in the list U are those that were surveyed during the last census, often held several years before, or were obtained by other possible sources that may still be outdated. Hence, the available data may very well be different from the current farm situation. In particular, the single cultivation areas a_{ij} may easily have changed, while total cultivation area $a_{i\mathrm{T}}$ of the farm is more stable.

For every cultivation j that we are interested in, including some types of livestock, a certain *coverage level* $q_j \geq 0$ and ≤ 1 is required, with $j = 1, \ldots, m$. Value q_j represents the minimum portion of the total area of cultivation j that must be surveyed: if this total area is $\sum_{i=1}^{n} a_{ij}$, we need to survey at least an area $q_j \sum_{i=1}^{n} a_{ij}$ of cultivation j (e.g. survey at least 0.8 of the total cultivation of oranges, at least 0.5 of the total cultivation of apples, etc.). A required coverage level q_T for the total cultivation areas is also given.

For the case of EU countries, coverage levels are generally assigned by European regulations, e.g. [44]. The set of coverage levels is denoted by

$$\{q_1, \ldots, q_m, q_\mathrm{T}\}$$

Surveying unit u_i has a cost w_i (evaluated either in terms of expense, or complexity, or other) and produce, for each cultivation j, an amount of statistical information that, in absence of further elements, is estimated being equal to the available value of the cultivation area a_{ij}. By defining the cost of a set of units to be the sum of their individual costs w_i, we want to choose a subset $S \subseteq U$ of units producing the minimum total cost for being surveyed and respecting the above defined $m + 1$ coverage levels.

In order to represent whether to include or not a unit, we introduce a set of binary decision variables $\{x_i\}$, with $i = 1, \ldots, n$, such that

$$x_i = \begin{cases} 1 & \text{if unit } u_i \text{ is excluded from the scope;} \\ 0 & \text{if unit } u_i \text{ is included in the scope.} \end{cases}$$

Now, cost minimization can be expressed by maximizing the total cost of the units that we do not survey (i.e. the saving).

Respecting the coverage levels, on the other hand, can be expressed by imposing that the area that is excluded cannot be more than the maximum area we are allowed to exclude. This latter condition should be imposed both for each cultivation and for the total area. The described Optimal Selection problem can be modeled as the following *multidimentional binary knapsack* problem.

$$\begin{cases} \max & \sum_{i=1}^{n} w_i x_i \\ \text{s.t.} & \sum_{i=1}^{n} a_{i1}\, x_i \leq (1-q_1) \sum_{i=1}^{n} a_{i1} \\ & \ldots \\ & \sum_{i=1}^{n} a_{im}\, x_i \leq (1-q_m) \sum_{i=1}^{n} a_{im} \\ & \sum_{i=1}^{n} a_{i\mathrm{T}} x_i \leq (1-q_{\mathrm{T}}) \sum_{i=1}^{n} a_{i\mathrm{T}} \\ & x_i \in \{0,1\} \end{cases} \tag{3.1}$$

Multidimentional binary knapsack is a well-known combinatorial optimization problem [104]; in its optimization version it is NP-hard. Note that a complementary choice for the meaning of the x_i variables (1 if unit u_i is included in the scope, 0 otherwise), that may appear more straightforward, would have led to a *multidimensional packing problem* [104], that has the same complexity level. However, the proposed modeling choice will have less variables at 1, with consequent computational advantages.

Model (3.1) has a number of variables equal to the number n of units in list U and a number of constraints equal to the number of coverage levels $m+1$, so it may reach in practical cases a very large dimension. Therefore, solving such an integer linear program by means of a simple Branch&Bound approach can be excessively time consuming (see e.g. [59]), and we prefer using a Branch&Cut approach to the solution of model (3.1).

In order to explain this approach, we denote by $\mathbf{0}_n$ and $\mathbf{1}_n$ the vectors of zeros and ones having n elements, and we define the matrix A of cultivation areas and the vector e of admissible exclusions, as follows.

$$A = \begin{pmatrix} a_{11} & \dots & a_{n1} \\ & \dots & \\ a_{1m} & \dots & a_{nm} \\ a_{1\mathrm{T}} & \dots & a_{n\mathrm{T}} \end{pmatrix} \qquad e = \begin{pmatrix} (1-q_1)\ \sum_{i=1}^{n} a_{i1} \\ \dots \\ (1-q_m)\ \sum_{i=1}^{n} a_{im} \\ (1-q_\mathrm{T})\ \sum_{i=1}^{n} a_{i\mathrm{T}} \end{pmatrix}$$

We can now represent the polytope P defined by the linear relaxation of model (3.1), as follows.

$$P = \{x \in I\!R^n : \ Ax \le e, \ \mathbf{0}_n \le x \le \mathbf{1}_n\}$$

Effective and practically fast algorithms are available for finding solutions of a linear formulation (e.g. [33], see also [34, 35, 99]). Even though P is a linear formulation of the Optimal Selection problem, it will in general be quite different from the optimal formulation of the problem $K = conv\{x^{(f)}\}$, that is the convex hull of all the feasible binary solutions $\{x^{(f)}\}$ to the Optimal Selection problem. So, although solving the linear problem on K would solve (3.1) to optimality, (see e.g. [104]), solving the same on P will not suffice.

Even though there is no explicit analytic expression of K, and in any case it would contain a number of constraints that is exponential in n, several procedures for improving a formulation like P have been proposed in literature, see e.g. [90]. A main one is by using the so called *covering* inequalities. Given a single knapsack constraint $\sum_{i=1}^{n} a_{ij}\ x_i \le e_j$, a set $C \subseteq \{1, \dots, n\}$ is a *cover* if

$$\sum_{i \in C} a_{ij} > e_j \tag{3.2}$$

In addition, the cover C is said to be *minimal* if C loses property (3.2) as soon as any of its element is removed. Given a cover C, we can write a simple valid inequality expressing that not all variables x_i , for $i \in C$, can be simultaneously one.

Proposition 2.1. *Let $C \subseteq \{1, \dots, n\}$ be a cover. The cover inequality*

$$\sum_{i \in C} x_i \le |C| - 1 \tag{3.3}$$

is valid for K. Moreover, if C is minimal, then the inequality defines a facet of $conv(K_C)$, where $K_C = K \cap \{x : x_i = 0 \text{ for } i \notin C\}$ ([2, 65, 140]).

Therefore, inequality (3.3) can be added to model (3.1) for improving the formulation given by its linear relaxation. The meaning is that we simply cannot exclude from the survey a whole set C of units having an area that is too big.

Example 2.2. *Consider the following set of points given by a binary knapsack:*

$$B = \{x \in \{0,1\}^6 : 5x_1 + 5x_2 + 5x_3 + 5x_4 + 3x_5 + 8x_6 \leq 17\}$$

$C = \{1, 2, 3, 4\}$ is a minimal cover for K and the corresponding cover inequality $x_1+x_2+x_3+x_4 \leq 3$ is valid for the optimal formulation of any linear optimization problem over B.

So far, one may in principle generate all inequalities in the form (3.3) and add them to model (3.1). However, this is often impracticable, since the number of those inequalities can be too large. In practical cases, a constraint generation approach within a Branch&Cut scheme constitutes an effective technique for practically solving the problem (see e.g. [104]). This means that model (3.1) is solved by iteratively solving a number of easier linear problems, where the k-th of them is denoted by $LP^{(k)}$ and constituted as follows

$$\begin{cases} \max c'x \\ D^{(k)}x \leq d^{(k)} \\ \mathbf{0}_n \leq x \leq \mathbf{1}_n \\ x \in I\!R^n \end{cases} \tag{3.4}$$

To obtain $LP^{(k)}$ we only need the constraint matrix $D^{(k)}$ and the right-hand side vector $d^{(k)}$, while c is the vector composed by the costs c_i for $i = 1, \ldots n$. The general scheme follows, composed by: Initialization (**Init**), Solution (**Solve**), Bounding (**Bound**), Updating (**Upd**), Separation (**Sep**), and Branching (**Bran**).

Branch&Cut scheme for problem (3.1)

Input An instance of multidimensional knapsack problem (3.1), defined by an $(m+1) \times n$ matrix A, an n-vector c and an $(m+1)$-vector e

Output The global optimal binary solution $x^\star$ to (3.1)

Init $LP^{(0)}$ is the linear relaxation of (3.1): $D^{(0)} := A$ and $d^{(0)} := e$. Insert $LP^{(0)}$ in the list of open problems L that was empty. Current lower bound $LB := -\infty$ and current binary solution x^o is undefined.

Solve If L is not empty, extract problem $LP^{(k)}$ from L and solve it to optimality, finding the variable values $\bar{x}^{(k)}$ and the corresponding objective value $\bar{z}^{(k)}$. If L is empty, STOP, current binary solution x^o is the global optimal solution $x^\star$ to (3.1).

Bound If $\bar{z}^{(k)} \leq LB$, problem $LP^{(k)}$ cannot improve the current binary solution, cancel $LP^{(k)}$, update $k := k + 1$ and go to step **Solve**.

Upd If all components of $\bar{x}^{(k)}$ are binary, update current binary solution $x^o := \bar{x}^{(k)}$ and $LB := \bar{z}^{(k)}$, update $k := k + 1$ and go to step **Solve**

Sep Solve the *separation problem*: find a valid inequality $d'x \leq d_0$ for cutting away $\bar{x}^{(k)}$, or conclude that such inequality does not exist. If the valid inequality $d'x \leq d_0$ is obtained, add it to the formulation $LP^{(k+1)} := LP^{(k)} \cap \{x : d'x \leq d_0\}$, insert $LP^{(k+1)}$ in L, update $k := k + 1$ and go to step **Solve**.

Bran Otherwise, do a standard branching on one of the non-binary components $\bar{x}_i^{(k)}$: generate $LP^{(k+1)} := LP^{(k)} \cap \{x : x_i \leq \lfloor \bar{x}_i^{(k)} \rfloor\}$ and $LP^{(k+2)} := LP^{(k)} \cap \{x : x_i \geq \lceil \bar{x}_i^{(k)} \rceil\}$, insert $LP^{(k+1)}$and $LP^{(k+2)}$ in L, update $k := k+2$ and go to step **Solve**.

The *separation problem*, essential element of the above scheme, is defined as follows:

Definition 2.3. *Let $\bar{x} \in R^n$ be a given point and K be a given polytope. The separation problem is to either prove that $\bar{x} \in K$ or find an inequality, called* cut *or* cutting plane, *that is valid for K but cuts $\bar{x}$ away from K.*

Finding such a valid inequality means in practice determining a vector d and a number d_0 such that $d'x \leq d_0$ for all $x \in K$ but not for $x = \bar{x}$. We look for a cut in the form of a cover inequality of one of the knapsack constraints in (3.1).

Denote by c the incidence vector of the generic subset C of $\{1,\ldots,n\}$, i.e. the binary n-vector whose i-th element is 1 if $i \in C$, 0 otherwise. By recalling that $a_{ij} \in R$ and ≥ 0, set C is a cover for the j-th knapsack constraint of problem (3.1) if and only if its incidence vector c satisfies the following condition

$$\sum_{i=1}^{n} a_{ij}c_i > (1-q_j)\sum_{i=1}^{n} a_{ij}$$

that, by introducing $\epsilon > 0$ equal to the minimum possible difference in the a_{ij} values, can be rewritten as

$$\sum_{i=1}^{n} a_{ij}c_i \geq [(1-q_j)\sum_{i=1}^{n} a_{ij}] + \epsilon \tag{3.5}$$

Among all possible covers, we want a cover C such that the components of $\bar{x}$ corresponding to elements of C sum to a value $> |C|-1$, that means $\bar{x}$ can be cut away by the cutting plane generated by C. Cover C represents in practice a set of units that cannot be simultaneously excluded from the survey, but are actually excluded in solution $\bar{x}$. The condition of cutting away $\bar{x}$ can be expressed as follows:

$$\begin{aligned}&\sum_{i\in C}\bar{x}_i > |C|-1 \Rightarrow \sum_{i=1}^{n}\bar{x}_i c_i > \sum_{i=1}^{n} c_i - 1 \Rightarrow\\ &\Rightarrow \sum_{i=1}^{n}(\bar{x}_i - 1)c_i > -1 \Rightarrow \sum_{i=1}^{n}(1-\bar{x}_i)c_i < 1\end{aligned} \tag{3.6}$$

Putting together condition (3.5) and (3.6) we have the following optimization problem encoding our separation procedure for the j-th knapsack constraint of problem (3.1).

$$\left\{\begin{array}{ll} \min & \sum_{i=1}^{n}(1-\bar{x}_i)c_i \\ s.t. & \sum_{i=1}^{n} a_{ij}c_i \geq [(1-q_j)\sum_{i=1}^{n} a_{ij}] + \epsilon \\ & c_i \in \{0,1\} \end{array}\right. \tag{3.7}$$

When model (3.7) is solved to optimality, we obtain vector $c^\star$ and the corresponding objective value $v^\star = \sum_{i=1}^{n}(1-\bar{x}_i)c_i^\star$. If $v^\star$ is < 1, there exists a covering inequality that is valid for K but cuts away $\bar{x}$ from K, and $c^\star$ is the incidence vector of the cover C generating that cover inequality. On the contrary, if $v^\star$ is ≥ 1, we cannot obtain from the j-th knapsack constraint a covering inequalities cutting away $\bar{x}$. In this latter case, we must try obtaining it from another one of the knapsack constraints of (3.1). If such a covering inequality cannot be obtained after all knapsack constraints have been tested, it does not exist.

Therefore, solving several problems in the form (3.7) may be needed. Although (3.7) is a binary problem with n variables, it can be solved quite easily by observing the following. Its optimal solution $c^\star$ actually represent the constraint that is more violated by $\bar{x}$, since we minimize the objective. However, there may be also other constraints that are valid for K and violated by $\bar{x}$. Indeed, any feasible solution $\hat{c}$ of of (3.7) such that the corresponding objective value $\hat{v}$ is < 1, even if sub-optimal, is the incidence vector of a cover $\hat{C}$ whose cover inequality cuts away $\bar{x}$ from K. Therefore, we may accept those kind of solutions, and search them with the following procedure. Recall that every variable c_i has a *cost* given by $(1 - \bar{x}_i)$ and a *value* a_{ij}. We denote by RHS the right-hand side of the only constraint in (3.7).

Greedy Algorithm for the solution of problem (3.7)

Input An instance of separation problem (3.7), defined by the cost n-vector $(1-\bar{x})$, the values n-vector a_j and a value RHS.

Output A binary feasible (and possibly optimal) solution $\hat{c}$ to (3.7)

1 Order by increasing cost/value ratio the indices of the binary variables.

2 Following the above greedy order, put $\hat{c}_i = 1$ until the left-had side of the constraint becomes $\geq RHS$ (i.e. we have a feasible solution), and $\hat{c}_h = 0$ for all the rest of the indices. If the value of this solution is $\hat{v} < 1$, there exists a covering inequality that is valid for K but cuts away $\bar{x}$ from K, and $\hat{c}$ is the incidence vector of the cover generating that.

The above heuristic solution could be evaluated by using the lower bound given by the solution $r^\star$ of the linear relaxation of problem (3.7): we put $r_i = 1$ until the left-had side of the constraint remains $\leq RHS$ (i.e. we have a maximal infeasible solution), then $r_{(i+1)} = (RHS - LHS)/a_{(i+1)j}$, and finally $r_h = 0$ for all the rest of the indices. If the objective value corresponding to $r^\star$ is $v_r^\star \geq 1$, the value $v^\star$ of the integer solution of (3.7) is $\geq v_r^\star$, so we know that the j-th knapsack constraint cannot provide a covering inequalities cutting away $\bar{x}$. On the contrary, when $v_r^\star < 1$ but $\hat{v} \geq 1$, a covering inequalities cutting away $\bar{x}$ may exist, but was not found by the procedure. Nonetheless, we try obtaining it from another one of the knapsack constraints of (3.1), and if none of them can provide it, we simply branch following the above Branch&Cut scheme.

Finally, when a cover C is obtained, we can improve the covering inequality by computing its *extension* $E(C)$. This is done by adding to C all $k \in \{1, \ldots, n\}$ such that $a_{kj} \geq a_{ij}$ for all $i \in C$. So, given the incidence vector $\bar{c}$ of the cover C and the incidence vector $\bar{e}$ of its extension $E(C)$, the cut to be added becomes

$$\sum_{i=1}^{n} \bar{e}_i x_i \leq \sum_{i=1}^{n} \bar{c}_i - 1$$

Now, the described Branch&Cut scheme is complete and can be successfully applied for reaching an optimal solution to problem (3.1).

3.3 Determining Reliable Inclusion Criteria

Since data are uncertain, an optimal solution $x^\star$ of model (1) cannot guarantee providing a set of units really respecting the required coverage levels. Indeed, as a trivial example, if the real cultivation areas of some of the selected farms have become smaller than what described by the available data a_{ij}, the risk of *undercoverage* (i.e. failing the required coverage levels q_j) is present. Hence,we need to distinguish between:

- solving the *Optimal Selection* problem, that is solving to optimality model (1);
- solving the *Scope Selection* problem, that is finding the set of units that we use as scope in practice.

For solving the Scope Selection problem, we need to determine a priori *inclusion criteria* for selecting the set of units respecting the coverage levels. A priori means here criteria that, for each unit u_i, could be checked *before* surveying u_i. A basic and mostly adopted criterion is using *thresholds*. Given a threshold value t_j, for each unit u_i one could determine whether to survey it or not: we survey u_i if $a_{ij} \geq t_j$, we do not survey it otherwise. Since in the analyzed case the total utilized area (UA) is the more reliable among the available informations, it was preferred to establish a threshold t only on that value.

The coverage levels, initially required by EU [44] and assigned for the whole Nation, were modified and slightly increased so as to determine more specific coverage levels assigned for each Region. Those new levels were determined by experts of the field according to specific features of the different regions whose description goes beyond the aim of this work. The final established regional coverage levels, for Citrus plantations, Fruit trees cultivation, Olive cultivations, Arable land, Vineyard cultivation and UA, are reported in Table 1.

Table 3.1: Regional coverage levels

Region	Citrus	Fruit	Olive	Arable land	Vineyard	UA
Piemonte	-	98.5	90.7	99.5	98.7	99.2
Valle d'Aosta	-	81.1	-	84.0	83.9	98.6
Lombardia	96.3	93.2	88.3	99.7	-	99.4
Trentino Alto Adige	-	99.3	66.1	95.8	97.8	98.8
Veneto	-	97.4	95.4	99.3	98.7	98.3
Friuli-Venezia Giulia	-	98.0	88.8	98.5	99.1	98.4
Liguria	68.4	84.8	89.5	92.6	82.0	92.7
Emilia-Romagna	-	98.7	91.6	99.6	99.5	99.4
Toscana	68.4	95.0	97.5	99.1	98.4	98.3
Umbria	-	94.8	96.9	98.8	97.1	98.5
Marche	80.3	94.2	94.3	99.1	98.6	98.8
Lazio	68.3	97.4	92.8	98.5	94.6	97.0
Abruzzo	85.1	94.6	96.0	98.2	99.1	98.5
Molise	-	96.3	96.2	99.1	97.2	98.7
Campania	82.4	97.2	95.3	96.8	94.5	96.7
Puglia	98.6	97.4	97.6	98.7	99.4	98.4
Basilicata	96.5	96.3	95.6	98.9	95.7	98.6
Calabria	98.0	97.6	97.1	96.3	95.0	97.3
Sicilia	97.4	97.0	94.6	97.8	99.2	97.6
Sardegna	93.6	93.9	95.8	99.4	97.4	99.3

In order to satisfy the above regional coverage levels, one may in general consider different options. A first one could be solving the regionals Optimal Selection problems, in order to determine, for each region, a selected set of units $U^\star$. After this, use $U^\star$ to determine the value a_{iT} of UA corresponding to the smallest included farm of the region, then compute how reliable that value is, possibly modify it for having a safety margin, and use it as inclusion threshold (at the regional level).

Another alternative would be fixing, according to some predetermined decision, a number of threshold values on the total utilized area (UA), and solve the regional Optimal Selection problems for each of them. Given a threshold t, denote by U_t the set $\{u_i \in U : a_{i\mathrm{T}} \geq t\}$ and by $U_t^\star$ the result of the Optimal Selection on U_t (the set of units $u_i \in U_t$ having $x_i = 0$). Define $R_t = U_t - U_t^\star$ the set of units that satisfy threshold t but are not in the solution of the Optimal Selection (i.e. the set of units $u_i \in U_t$ having $x_i = 1$). Sets U_t and R_t can now be used for computing statistical indicators that evaluate the safety margin with respect to the risk of undercoverage obtained by using t as inclusion threshold.

After this, the threshold value $t^\star$ that, among the predetermined ones, corresponds to the best compromise between risk estimation and list reduction for the region, is selected and adopted as inclusion criterion (again at the regional level). The solution to the Scope Selection problem would therefore be $U_{t^\star}$ This last option was preferred in the analyzed case, because it could provide more robustness in the procedure, in the sense of making it more stable and less prone to the changes that may have occurred in the data.

We now describe the statistical indicators that were built by experts of the field for evaluating the safety margin from the risk of undercoverage corresponding to each threshold t.

A basic index number is the percentage of farms taken in addition to $U_t^\star$ when using threshold t, denoted by $\beta(t)$ and computed as follows.

$$\beta(t) = 100\frac{|R_t|}{|U_t|}$$

The larger the value of $\beta(t)$, the more threshold t is able to provide a set U_t that is bigger than the minimum set respecting the coverage levels, and consequently the more secure is the selection by using threshold t.

Another index number, for each cultivation j, is the percentage of cultivation area taken as safety margin when using threshold t, denoted by $\gamma_j(t)$ and computed as follows.

$$\gamma_j(t) = 100 \, \frac{\displaystyle\sum_{u_i \in R_t} a_{ij}}{\displaystyle\sum_{u_i \in U_t} a_{ij}}$$

Again, the larger the value of $\gamma_j(t)$, the more threshold t is able to provide an area $\sum_{u_i \in U_t} a_{ij}$ that is bigger than the minimum area respecting the coverage levels, and consequently the more secure is the selection by using threshold t. Clearly, the higher the values of t, the smaller the above safety margins become, but the larger the savings in the survey are. Therefore, we need to chose the higher t still having acceptable values for the above indicators, so as to obtain the maximum savings with negligible risk.

Given a set of farms U_t, define $S_j(U_t) = \{u_i \in U_t : a_{ij} > 0\}$ to be the subset of farms in U_t having cultivation j. Denote now by $\mu\{S_j(U_t)\}$ the average area of cultivation j over set $S_j(U_t)$. A third indicator, for each cultivation j, is the average number of units in R_t that are needed to obtain a portion of information that is equivalent to the portion of information given by an average unit of $U_t^\star$, or, in other words, the average number of units in R_t needed to replace a unit in $U_t^\star$ (for example in case the latter one does not exist anymore). That will be denoted by $\omega_j(t)$ and computed as follows.

$$\omega_j(t) = \frac{\mu\{S_j(U_t^\star)\}}{\mu\{S_j(U_t) - S_j(U_t^\star)\}}$$

The $\omega(t)$ is clearly ≥ 1, and the smaller the values, the more robust is the choice of threshold t.

3.4 Experimental Results

The described procedure was implemented in C++ and has been tested for the treatment of data from the Italian Agricultural Census of 2010. The experiments were conducted on a 16 cores server having 128Gb of RAM under Linux Operating System. The linear relaxations (3.4) are solved by means of the open source solver Clp (Coin-or linear programming, available from https://projects.coin-or.org/Clp), which is a good implementation of primal and dual simplex and barrier methods, written in C++ by a research group headed by Dr. John J. Forrest, from the IBM Watson Research Center, within a joint project among IBM, Maximal and Schneider called COIN-OR (COmputational INfrastructure for Operations Research, http://www.coin-or.org/index.html). This solver was selected because it appeared the most suitable open source LP solver in a previous study [11].

The predetermined threshold levels on the total utilized area (UA) were 0.0 (meaning all is included); 0.1; 0.2; 0.3; 0.4 hectares.

Table 3.2 reports, as an example of the results, the detail of this analysis for one Italian region (Marche). Evidently, when increasing the inclusion threshold t, the cardinality of U_t decreases consistently (less farms satisfy that threshold). On the other hand, the cardinality of $U_t^\star$ increases (the choices become limited to farms satisfying that threshold), but this happens very slowly. Consequently, the differences between U_t and $U_t^\star$ tend to become smaller, and the described β, γ tend to decrease. Therefore, when increasing t, there is a trade-off between the cost and complexity reduction caused by the decreasing of $|U_t|$ and the rise in the risk of undercoverage evaluated by the β, γ. Acceptable values for the β, γ indicators were considered those respectively above 10%, 0.5%. On the contrary, values for the ω indicator should be as small as possible. Hence, for the case of Marche region, the best compromise is $t = 0.4$, corresponding to a very acceptable risk level but producing considerable savings: 66,536 - 60,309 = 6,254 farms.

Table 3.2: Results of the procedure applied to the Marche Region

Thershold	$\lvert U_t \rvert$	$\lvert U_t^\star \rvert$	β	γ_{vineyard}	ω_{vineyard}	γ_{olive}	ω_{olive}	γ_T
0.0	66,563	50,051	24.81	3.50	6.10	7.51	3.45	1.35
0.1	65,438	50,051	23.51	3.50	6.10	7.51	3.45	1.35
0.2	64,374	50,051	22.25	3.47	5.95	7.38	3.31	1.34
0.3	62,474	50,051	19.86	3.32	5.61	6.68	3.11	1.29
0.4	60,309	50,057	17.01	3.00	5.27	5.41	3.07	1.20

Figure 3.1 shows the geographical distribution of the threshold values $t^\star$ selected as best compromise between risk estimation and list reduction. Table 3.3 reports, for each Italian region: the number of all existing statistical units $|U|$; the number of units selected when solving model model (3.1) to optimality $|U^\star|$; the value of threshold $t^\star$ selected; the number $|U_t|$ of existing statistical units above threshold t; the number $|U_t^\star|$ of units selected from $|U_t|$ when solving model (3.1) to optimality; computational time in seconds for the overall treatment of the region, including solving 5 Optimal Selection problems and evaluating the described indicators (Time All); computational time in seconds for solving to optimality the single Optimal Selection problem (3.1) corresponding to $t^\star$.

Table 3.4 reports, for each Italian Region, the values of some of the described statistical indicators corresponding to the selected threshold $t^\star$. In particular, we report $\beta, \gamma_{\text{vineyard}}, \omega_{\text{vineyard}}, \gamma_{\text{olive}}, \omega_{\text{olive}}, \gamma_{\text{T}}$. Vineyard and olive were selected because they are particularly important, being the two most typical Italian cultivations and subject to several EU regulations.

Table 3.5, finally, summarizes the Italian situation. It reports, for the case of threshold = 0 (all U) and for the threshold $t^\star$ reported for each region in Table 3.3, the total number of farms, their total utilized area UA, the number of farm obtained for the optimal solution, and their total UA.

When using $t^\star$ as inclusion criterion, we have a reduction in the number of farms of 206,679, corresponding to a saving of 7.97%, and a reduction in the area of about 50,948 hectares, corresponding to a loss of only 0.39% of the total information and a negligible risk of failing the required coverage levels.

Note, finally, that, if the cultivation data were updated and reliable, the set $U_0^\star$ could have been surveyed directly, with a reduction in this case of 950,506 farms,

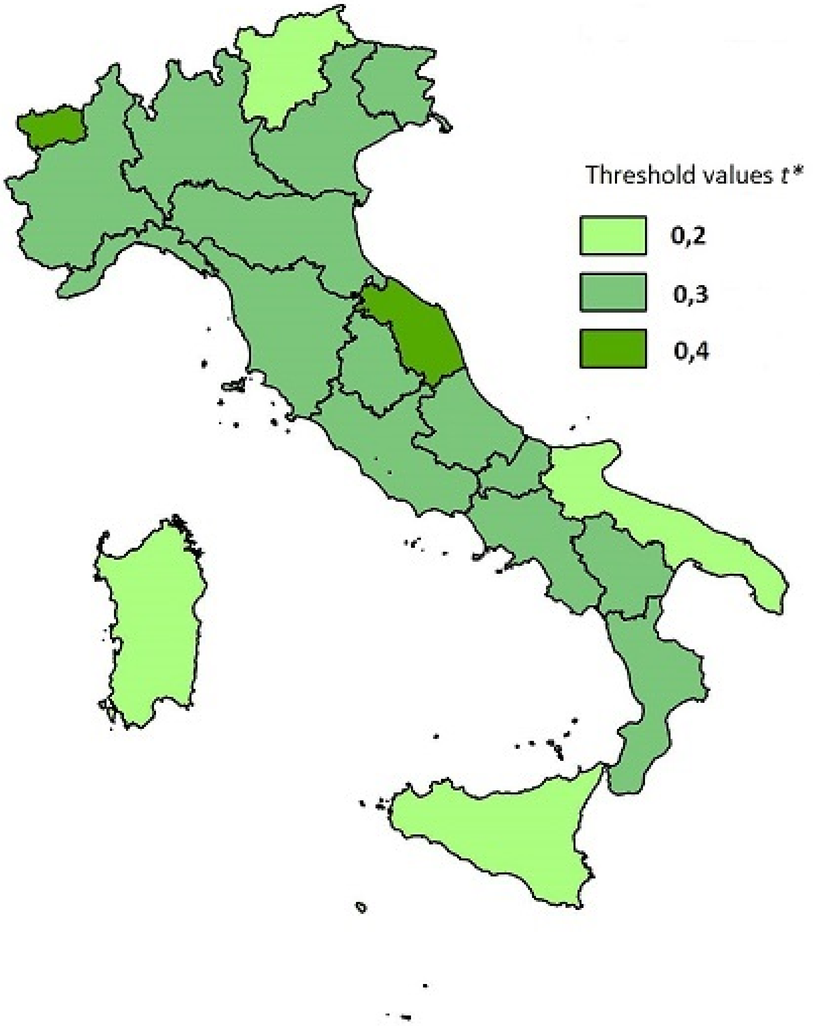


Figure 3.1: Geographic distribution of the thresholds $t^{\star}$.

Table 3.3: Results of the procedure applied to all Italian Regions

Region	$\|U\|$	$\|U^\star\|$	$t^\star$	$\|U_t\|$	$\|U_t^\star\|$	Time All	Time $t^\star$
Piemonte	120,965	78,651	0.3	103,347	78,651	605.2	116.0
Valle d'Aosta	6,595	4,050	0.4	5,441	4,051	33.5	7.6
Lombardia	74,867	56,949	0.3	69,890	56,949	374.5	73.8
Trentino Alto Adige	61,253	33,804	0.2	51,816	33,804	306.8	61.2
Veneto	191,085	118,204	0.3	176,251	118,204	955.2	186.0
Friuli-Venezia Giulia	34,963	25,455	0.3	32,953	25,455	175.6	34.1
Liguria	44,266	21,654	0.3	34,167	21,654	221.0	43.2
Emilia-Romagna	107,888	89,468	0.3	103,744	89.468	539.5	104.8
Toscana	139,872	77,823	0.3	119,788	77,823	699.0	136.8
Umbria	57,153	36,538	0.3	51,772	36,538	286.3	57.2
Marche	66,563	50,051	0.4	60,309	50,057	333.0	65.6
Lazio	214,666	123,026	0.3	189,906	123,026	1,073.3	210.6
Abruzzo	82,833	58,478	0.3	78,036	58,478	414.2	81.8
Molise	33,973	25,285	0.3	31,955	25,285	170.8	36.2
Campania	248,932	143,319	0.3	216,635	143,318	1,245.0	246.0
Puglia	352,510	229,118	0.2	348,380	229,118	1,763.0	346.6
Basilicata	81,922	58,460	0.3	76,307	58,460	410.5	82.1
Calabria	196,484	113,719	0.3	173,866	113,719	982.2	193.4
Sicilia	365,346	223,912	0.2	355,038	223,912	1,827.6	358.4
Sardegna	112,689	76,355	0.2	108,545	76,355	563.5	108.6

corresponding to a saving of 36.63%, and a reduction in the area of about 204,848 hectares, corresponding to a loss of only 1.55% of the total information with the guarantee of respecting the coverage levels.

3.5 Conclusions

We proposed an innovative approach to the Scope Selection problem based on Combinatorial Optimization. The proposed multidimensional knapsack model can be solved to optimality in short times by means of a Branch&Cut algorithm based on the generation of cover inequalities.

The procedure has been implemented and tested on the real-world case of an Agricultural Census. By solving the Optimal Selection problem in different condi-

Table 3.4: Values of the indicators corresponding to the selected thresholds

Region	β	$\gamma_{vine.}$	$\omega_{vine.}$	γ_{olive}	ω_{olive}	γ_{T}
Piemonte	23.90	4.42	7.35	10.02	3.32	1.15
Valle d'Aosta	25.55	11.01	2.62	-	-	1.15
Lombardia	18.52	4.19	8.23	10.53	4.22	0.67
Trentino Alto Adige	34.76	6.59	7.37	33.88	2.43	1.71
Veneto	32.93	5.12	7.07	8.08	3.72	3.64
Friuli-Venezia Giulia	22.75	2.90	8.13	12.50	5.24	1.77
Liguria	36.62	16.25	2.94	12.24	4.09	7.18
Emilia-Romagna	13.76	3.03	5.25	9.86	2.83	0.74
Toscana	35.03	4.36	9.37	5.97	6.92	2.31
Umbria	29.43	5.26	6.02	6.29	5.44	1.98
Marche	17.02	3.00	5.27	5.41	3.07	1.20
Lazio	35.22	8.18	4.49	9.88	4.29	3.98
Abruzzo	25.06	4.19	5.35	7.20	3.82	2.16
Molise	20.87	4.42	4.27	6.47	3.30	1.52
Campania	33.84	8.47	3.45	8.34	3.94	5.22
Puglia	34.23	5.29	5.59	7.00	6.42	3.56
Basilicata	23.39	5.54	3.53	7.24	3.55	1.58
Calabria	34.59	7.79	3.98	6.83	6.38	4.53
Sicilia	36.93	5.11	7.33	9.48	5.16	3.81
Sardegna	29.66	6.48	4.64	7.96	5.08	1.09

Table 3.5: Aggregate results at the National level

Threshold	$\lvert U_t\rvert$	Total UA for U_t	$\lvert U_t^\star\rvert$	Total UA for $U_t^\star$
0.0	2,594,825	13,206,296.76	1,644,319	13,001,448.97
$t^\star$	2,388,146	13,155,349.09	1,644,326	13,003,198.84

tions, statistical indicators for the determination of reliable inclusion criteria based on thresholds are computed.

The proposed approach allows to considerably reduce costs and complexity of the survey while ignoring only a very small portion of the whole information that can be surveyed.

The risk of failing the required coverage levels, i.e., the risk that such ignored portion was larger than the maximum admissible portion we are authorized to ignore, is negligible.

3.6 A future case study

The optimal selection approach proposed can be adopted to solve other issues of different origin, but presenting the same logical characteristics. For instance, the selection of a subset of municipalities for the sample surveys to check the quality and the coverage of the Italian Statistical Farm Register. This frame, currently under development in Istat, is a key tool in agricultural statistics, particularly for drawing representative samples of farms in the intercensal period.

The Farm Register is the result of the integration of multiple sources, as the 2010 Agricultural Census, AGEA (Italian Agency for Disbursements in Agriculture), Agricultural Incomes, ASIA (Italian Statistical Records of Active Companies), Regional Archives, CCIA (Chambers of Commerce), VAT, Italian Land Registry. These sources contain information on administrative units that could be identified as agricultural holdings. The need to assess the accuracy and reliability of some information related to the main crops and livestock could lead to survey a sample of units from the register.

The design of a sample survey, involving a large number of decision makers, and the need to ensure high quality operational standards, could cause to deal with strict organizational constraints. These restrictions may lead to the choice of alternative methods for the selection of the units to be collected, thus limiting the representativeness of the selected sample only to particular territorial levels. Sometimes, under additional financial budget limits, it is necessary to use complex sampling designs to increase the efficiency and to overcome the constraints imposed on the sample size. In this case, a survey of high organizational flexibility could be more suitable, in order to meet the requirements of the different statistical offices involved and to obtain high quality standards.

As an example, the design of a two-stage sample could be adopted for the evaluation of the coverage and the quality of the Farm Register. The survey strategy to achieve this goal should satisfy both the need to collect the information for assessing the eligibility of administrative units to agricultural holdings and the need to manage the organizational and financial constraints.

The survey design could be based fundamentally on a two-stage design: The 1st stage: logical choice of a subset of municipalities, selected according to the number of firms and regional agricultural specializations. The 2nd stage: selection of a sample according to a complex probabilistic design, based on methods of calibration and balancing [36].

The municipalities to be collected could be chosen according to the data from the Farm Register and the information provided by the Regions, responsible for the local implementation of agricultural policy. The trade-off between the survey costs and data quality can be solved by selecting from a set of units (municipalities), an optimal subset of statistical units (having minimum cardinality, i.e. a minimum number of elements), which ensures the achievement of fixed coverage levels. This problem could be formulated using the proposed optimal selection approach described in 3.2 section.

In brief, the procedure to select the farms to include in the sample can be summarized as follows:

1- a list of potentially selectable provinces according to some criteria is defined in each region.

2- For each provincial population, by applying the described algorithm, the minimum subset of municipalities is determined, to meet the pre-arranged coverage constraints, concerning the most relevant cultivations at provincial level.

3- Subsequently, for each municipality and for the main crops corresponding to the provincial specialities, the minimum subset of units to be collected to meet the predetermined coverage levels is selected.

Over the time, the suggested method could support the fulfilment of an agricultural statistical system resulting from the integration of administrative sources and sample surveys, thus limiting the respondent burden and the costs to gather and update the available information.

Appendix A

Appendix A

A.1 The basic Structure of Combinatorial Optimization Problems

Many problems of both practical and theoretical origin concern with the choice of a “best” solution among a set of “feasible” solutions, and so they are called optimization problems. Optimization problems may assume very different shapes, and can be classified according to several of their features. However, one very basic subdivision is in the two following categories: those whose solutions are coded with real variables and those whose solutions are coded with discrete variables. Among the latter, we can find the class of the problems of combinatorial optimization, that have a pronounced combinatorial or discrete structure [27]. In these cases, we are looking for an object from a countable set. This object is usually an integer, a subset, a permutation or a graph structure [16]. The set of such problems, together with the methodologies used for their solution, constitute the field called Combinatorial Optimization (CO), that is usually considered as a discipline belonging to applied mathematics and theoretical computer science. CO has an increasing importance because of the large number of important practical problems that can be formulated and solved in this manner.

Combinatorial Optimization has an increasing importance because of the large number of important practical problems that can be formulated and solved in this manner. CO problems may have several different connotations. However, when they have a linear objective function, as it happens very frequently in practice, they all share an underlying mathematical structure defined below.

Definition 1.1. A combinatorial optimization problem with linear objective is defined by

- a finite set $B = \{b_1, \dots, b_n\}$ called ground set;
- a collection of feasible solutions $\mathcal{F} = \{F_1, \dots, F_m\}$, each of which is a subset of the ground set $F_j \subseteq B$;
- a cost vector $c = \{c_1, \dots, c_n\}$ providing the cost of each ground element;

and consists in finding a feasible solution $F^\star$ such that

$$c(F^\star) = \sum_{i\colon b_i \in F^\star} c_i$$

is minimum.

We recall that maximizing an objective function f is the same as minimizing $-f$, hence it is possible to address minimization problems without loss of generality. By denoting a set by means of its incidence vector, that is a vector having 1 in correspondence of each element of the set and 0 otherwise, a feasible solution F_j may be represented by a binary vector $x \in \{0, 1\}^n$. This allows to express every CO problems by using Integer (or Binary) Linear Programming.

A.2 Notions of Integer Linear Programming

Integer and mixed integer programming are subsets of the broader field of mathematical programming. Mathematical programming formulations use a set of *decision variables*, which represent actions or decisions that can be taken in the system being modeled. One then attempts to optimize (either in the minimization or maximization sense) an *objective*, that is a function of these variables which maps each possible set of decisions into a single score that assesses the quality of the solution. The limitations of the system are included as a set of *constraints*, which are usually stated by restricting functions of the decision variables to be equal to, not more than, or not less than, a certain numerical value. Another type of constraint can simply restrict the set of values to which a variable might be assigned.

Several applications involve decisions that are *discrete* (e.g., to which hospital an emergency patient should be assigned), while some other decisions are *continuous* in nature (e.g., determining the dosage of fluids to be administered to a patient). When a problem contains only continuous variables and linear objective and constraints, the problem is called *linear programming*. When on the contrary a problem contains only discrete variables and linear objective and constraints, the problem is called *integer linear programming*. Note that a parallelism can be traced between integer linear programming and combinatorial optimization with linear objective function [104]. When a problem contains both types of variables and linear objective and constraints, the problem is called *mixed integer linear programming*.

While discrete variables may appear easy to handle, the number of combinations of their values is usually huge, and so complete enumeration techniques have important implications on processing time. As the problem size increases, complete enumeration approaches are not computationally viable. Computer speedups, however impressive, are simply no match for exponential enumeration problems. Therefore, more efficient techniques are required to solve problems containing discrete variables. Those techniques do not explicitly examine every possible combination of discrete solutions, but instead examine a subset of possible solutions, and use optimization theory to prove that no other solution can be better than the best one found. This type of technique is referred to as implicit enumeration.

Linear Programming Linear Programming problems (LP, also called "linear programs") use a set of decision variables, which are the unknown quantities or decisions that are to be optimized. In the context of linear and mixed integer programming problems, the function that assesses the quality of the solution, called the "objective function", should be a linear function of the decision variables. An LP will either minimize or maximize the value of the objective function. Finally, the decisions that must be made are subject to certain requirements and restrictions of a system. We enforce these restrictions by including a set of constraints in the model. Each constraint requires that a linear function of the decision variables is either equal to, not less than, or not more than, a scalar value. A common condition simply states that each decision variable must be nonnegative. In fact, all linear programming problems can be transformed into an equivalent minimization problem with nonnegative variables and equality constraints [6].

A solution that satisfies all constraints is called a *feasible solution*. Feasible solutions that achieve the best objective function value (according to whether one is minimizing or maximizing) are called *optimal solutions*. Sometimes no feasible solution exists, and the optimization problem itself is called *infeasible*. On the other hand, some feasible LP problems have no optimal solution, because it is possible to achieve infinitely good objective function values with feasible solutions. Such problems are called *unbounded.*

Thus, suppose we denote $x_1, \dots, x_n$ to be our set of decision variables. Linear programming problems take on the form:

$$\begin{array}{ll} \text{min or max} & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ & \\ \text{subject to} & a_{\mathtt{11}}x_1 + a_{\mathtt{12}}x_2 + \cdots + a_{\mathtt{1n}}x_n \;\; (\leq, =, \text{or} \geq) \;\; b_1 \\ & a_{\mathtt{21}}x_1 + a_{\mathtt{22}}x_2 + \cdots + a_{\mathtt{2n}}x_n \;\; (\leq, =, \text{or} \geq) \;\; b_2 \\ & \dots \\ & a_{\mathtt{m1}}x_1 + a_{\mathtt{m2}}x_2 + \cdots + a_{\mathtt{mn}}x_n \;\; (\leq, =, \text{or} \geq) \;\; b_m \\ & \\ & x_j \geq 0 \quad \forall j = 1, \dots, n \end{array} \tag{A.1}$$

Values $c_j, \forall j = 1, \dots, n$, are referred to as objective coefficients, and are often associated with the costs associated with their corresponding decisions in minimization problems, or the revenue generated from the corresponding decisions in maximization problems. The values $b_1, \dots, b_m$ are the right-hand-side values of the constraints, and often represent amounts of available resources (especially for $\leq$ constraints) or requirements (especially for $\geq$ constraints). The a_{ij}-values thus typically denote how much of resource/requirement *i* is consumed/satisfied by decision *j*. Note that nonlinear terms are not allowed in the model, prohibiting for instance the multiplication of two decision variables, the maximum of several variables, or the absolute value of a variable.

Any maximization (minimization) problem can be converted into a minimization (maximization) problem by multiplying the coefficients of the objective function by -1.

$$\max \sum_{j=1}^{n} c_j x_j = -\min \sum_{j=1}^{n} -c_j x_j$$

Moreover, each linear programming problem in generic form can be transformed into an equivalent problem in *canonical* form:

$$\begin{array}{lll} \min & \sum_{j=1}^{n} c_j x_n & \\ & & \\ \textit{subject to} & \sum_{j=1}^{n} a_{\mathtt{ij}} x_j \geq b_i & \forall i = 1 \dots m \\ & & \\ & x_j \geq 0 & \forall j = 1, \dots, n \end{array} \tag{A.2}$$

This canonical form can be expressed in a compact notation as follows.

$$\begin{array}{ll} \min & c^T x \\ & Ax \geq b \\ & x \in \mathrm{I\!R}^{\mathrm{n}} \end{array} \tag{A.3}$$

where x represents the vector of variable (to be determined), c e b are vector of coefficients, A is a (known) matrix of coefficients. The inequalities $Ax \geq b$ are constraints which specify a convex politope over which the objective function is to be optimized. Linear programming problems can be converted into canonical form as follows:

- For each variable x_j, add the equality constraint $x_j = x_j^+ - x_j^-$ and the inequalities $x_j^+ \geq 0$ and $x_j^- \geq 0$.
- Replace any equality constraint $\sum_j a_{\mathtt{ij}} x_j = b_i$ with two inequality constraints $\sum_j a_{\mathtt{ij}} x_j \geq b_i$ and $\sum_j a_{\mathtt{ij}} x_j \leq b_i$.
- Replace any constraint $\sum_j a_{\mathtt{ij}} x_j \leq b_i$ with the equivalent constraint $\sum_j -a_{\mathtt{ij}} x_j \geq -b_i$.

Another useful format for linear programming problems is *standard* form, which is expressed as:

$$\begin{array}{ll} \min & c^T x \\ subject\ to & Ax = b \\ and & x \geq 0 \end{array} \tag{A.4}$$

Note that a LP not in standard form can be converted to standard form by eliminating inequalities by introducing slack and/or surplus variables and replacing variables that are not sign-constrained with the difference of two sign-constrained variables.

Mixed Integer Linear Programming When some of the variables are restricted to take integer values, the problem becomes a Mixed Integer Linear Programming one (MILP, also called “mixed integer linear programs”). When variables are restricted to take on either 0 or 1 values the term “integer” is replaced with “0-1” or “binary”. All that was specified for the case of linear programming holds, *mutatis mutandis*, for the mixed integer case. Typically, modeling MILP requires the definition of a set of decision variables, that represent choices that must be optimized in the system, and the statement of an objective function and constraints (see also [136]).

It is very common, though, to recognize during model construction that the initial set of decision variables defined for the model are inadequate. Often, decision variables that seem to be implied consequences of other actions must also be defined. The addition of new variables after an unsuccessful attempt at formulating constraints and objectives is the “loop” in the process. The correct definition of decision variables can be especially complicated in modeling with integer variables. If one is allowed to use binary variables in a formulation, it is possible to represent yes-or-no decisions, enforce if-then statements, and even permit some sorts of nonlinearity in the model (which can be transformed to an equivalent mixed integer linear program).

Some common tips and tricks in modeling with integer variables are:

1. *Integrality of quantities*. In staffing and purchasing decisions, it is often impossible to take fractional actions. One cannot hire, for instance, 6.5 new staff members, or purchase 1.3 hospital beds. The most obvious use of integer variables thus arises in requesting integer amounts of quantities that can only be ordered in integer amounts. In general, the optimal solution of an integer program need not be a rounded-off version of an optimal solution to a linear program.

2. *If-then statements*. Consider two continuous (i.e., possibly fractional) variables, x and y, defined so that $0 \leq x \leq 10$ and $0 \leq y \leq 10$. Suppose we wish to make a statement that if $x > 4$, then $y \leq 6$. On the surface, since no integer quantities are requested, it does not appear that integer variables will be necessary. However, the general form of linear programs as given in equations (A.1) does not permit if-then statements like the one above. Instead, if-then statements can be enforced with the aid of a binary variable, z. We wish to make $z = 1$ if $x > 4$ (note that we make no claims on z if $x \leq 4$). This can be accomplished by adding the constraint:

   $$x \leq 4 + 6z \tag{A.5}$$

   since the event that $x > 4$ implies that $z = 1$ (even if $z = 1$, the largest value for x is 10, which now makes a constraint of the form x is 10 unnecessary). If $z = 1$, then we must also require that $y \leq 6$. This is achieved by reducing the upper bound of 10 on y to 6 if z is equal to 1 as follows:

   $$y \leq 10 + 4z \tag{A.6}$$

   where once again, the bound constraint $y \leq 10$ may now be omitted. In general, suppose we wish to make the following statement: “if $q_1 x_1 + \cdots +$

$q_n x_n > Q$, then $r_1 x_1 + \cdots + r_n x_n \leq R$". The following conditions should be included in the model:

$$q_1 x_1 + \cdots + q_n x_n \leq q + M'z \tag{A.7}$$

$$r_1 x_1 + \cdots + r_n x_n \leq M'' - (M'' - R)z \tag{A.8}$$

$$z \; binary \tag{A.9}$$

where M' and M'' are "sufficiently large" constants. These values should be just large enough to not add unintentional restrictions to the model. For instance, we are not attempting to place any hard restriction on the quantity $q_1 x_1 + \cdots + q_n x_n$ (written conveniently as $q^T x$ in vector form). If $z = 1$, the upper bound on $q^T x$ is $Q + M'$, and hence M' must be large enough so that even if constraint (A.7) is removed from the model, $q^T x$ would still never be more than $Q + M'$. Likewise, if $z = 0$, a large enough value of M'' must be chosen in (A.8) such that $r^T x$ could never be more than M'' even without the restriction (A.8). It is worth noting that assigning arbitrarily large values for M' and M'' is not recommended.

3. *Enforce at least k out of p restrictions.* This situation is similar to if-then constraints in the way we model such restrictions. For a simple example, suppose we have nonnegative variables $x_1, \ldots, x_n$, and wish to require that at least three of these variables take on values of 5 or more. Then we can define binary variables $z_1, \ldots, z_n$, such that if $z_j = 1$, then $x_j \geq 5, \forall j = 1, \ldots, n$. This simple if-then constraint can easily be modeled by employing the following constraints:

$$x_j \geq 5z_j \;\; \forall j = 1, \ldots, n \tag{A.10}$$

Clearly, if $z_j = 1$, then $x_j \geq 5$. If $z_j = 0$, it is still possible for $x_j \geq 5$, but no such restrictions are enforced. It is necessary to guarantee that three variables take on values of 5 or more, and so the following "k-out-of-p" constraint is added:

$$z_1 + \cdots + z_n = 3 \tag{A.11}$$

Again, this constraint does not state that exactly three variables will be at least 5, but rather that at least three variables are guaranteed to be at least 5. This same trick can be used to enforce the condition that at least k out of p sets of constraints are satisfied, and so on, often by using *M*-values as introduced in the point on if-then constraints.

4. *Non linear product terms.* In some circumstances, nonlinear terms can be transformed into linear terms by the use of linear constraints. First, note that if x_j is a binary variable, then $x_j = x_j^q$ for any positive constant q. After that substitution is made, suppose that we have a nonlinear term of the form $x_1 \cdot x_2 \cdots x_k \cdot y$, where $x_1, \ldots, x_k$ are binary variables and $0 \leq y \leq u$ is another variable, either continuous or integer. That is, all but perhaps one of the terms is a binary variable. First, replace the nonlinear term with a single continuous variable, w. Using the if-then concept expressed above, note that if x_j equals zero for any $j \in \{1, ..., k\}$, then w equals zero as well. Also, note that w can never be more than the upper bound, u, on the y-variable. Hence, we obtain the constraints

$$w \leq u x_j \quad \forall j = 1, \ldots, k \tag{A.12}$$

Of course, to guarantee that w equals zero in case any x_j-variable equals to zero, we must also state a non-negativity constraint:

$$w \geq 0 \tag{A.13}$$

Now, suppose that all $x_1 = \cdots = x_k = 1$. In this case, it is necessary to add constraints that enforce the condition that $w = y$. Regardless of the x-variable values, w cannot be more than y, and so we state the constraint:

$$w \leq y \tag{A.14}$$

However, in order to get the constraint "$w \geq y$ if $x_1 = \cdots = x_k = 1$," we include the constraint:

$$w \geq u(x_1 + \cdots + x_k - k) + y \tag{A.15}$$

If each x-variable equals to 1, then (A.15) states that $w \geq y$, which along with (A.14) guarantees that $w = y$. On the other hand, if at least one $x_j = 0,\ j \in \{1, \ldots, k\}$, then the term $u(x_1 + \cdots + x_k - k)$ is not more than $-u$, and the right-hand-side of (A.15) is not positive; hence, (A.15) allows w to take on the correct value of zero (as would be enforced by (A.12) and (A.13)). As a final note, observe that even if y is an integer variable, we need not insist that w is an integer variable as well, since (A.12) - (A.15) guarantee that $w = x_1 \cdots x_k \cdot y$, which must be an integer given integer x- and y-values.

A.3 Exact Solution Techniques

Often, there are alternative ways of modeling optimization problems as MILP. There sometimes exist trade-offs in these different modeling approaches. Some models may be smaller (in terms of the number of constraints and variables required), but may be more difficult to solve than larger models. It is important to understand the basics of MILP solution algorithms in order to understand the key principles in MILP modeling. To illustrate the branch-and-bound process, we consider the following example MILP:

$$\begin{array}{ll} \min & 4x_1 + 6x_2 \\ s.t. & 2x_1 + 2x_2 \geq 5 \\ & x_1 - x_2 \leq 1 \\ & x_1, x_2 \geq 0 \; and \; integer \end{array} \tag{A.16}$$

A *relaxation* of an MILP is a problem such that (a) any solution to the MILP corresponds to a feasible solution to the relaxed problem, and (b) each solution to the MILP has an objective function value greater than or equal to that of the corresponding solution to the relaxed problem. The most commonly used relaxation for an MILP is its *LP relaxation*, which is identical to the MILP with the exception that variable integrality restrictions are eliminated. Clearly, any integer-feasible solution to the MILP is also a solution to its LP relaxation, with matching objective function values.

When describing the branch-and-bound algorithm for MILP, it is helpful to know how LP is solved. See [6, 69, 114, 138] for an explanation of linear programming theory and methodology. Graphically, Figure A.1 illustrates the feasible region (set of all feasible solutions) to the LP relaxation of formulation (A.16). The point (1.75, 0.75), is the optimal solution to the LP relaxation, and has an objective function value of 11.5. In general, the optimal solution to the LP is not supposed to be unique, and so it is possible that different MILP solutions exist with an identical objective function to the optimal LP solution. The important result is that a lower bound on the optimal MILP solution is obtained from the LP relaxation. No solution to the MILP can be found with an objective function value less than 11.5.

Of course, the solution (1.75, 0.75) is not a feasible solution to (A.16). All feasible solutions have the trait that either $x_1 \leq 1$ or $x_1 \geq 2$. In fact, the problem (A.16) can be splitted into two subproblems: one in which $x_1 \leq 1$ (called region 1), and one in which $x_1 \geq 2$ (called region 2). All solutions to the original MILP are contained in exactly one of these two new subproblems. This process is called *branching*, and we could have also branched on x_2 instead, by requiring that either $x_2 \leq 0$ or $x_2 \geq 1$.

The feasible regions of the two new subproblems are depicted in Figure A.2. When $x_1 \leq 1$, the optimal solution is (1, 1.5) with objective function value 13. When $x_1 \geq 2$, the optimal solution is (2, 1) with objective function value 14. In the $x_1 \leq 1$ region, the lower bound is 13. In the $x_1 \geq 2$ region, though, the best solution happens to be an integer solution. Therefore, the best integer solution in the $x_1 \geq 2$ region has an objective function value of 14; there is no need to further search that region. This region is said to be fathomed by integrality. We store the solution (2, 1), and call it *incumbent solution*. If no better solution is found, it will become our optimal solution.

At this point, there is one *active* region (or "active node" in the context of branch-and-bound trees), which is region 1. An active region is one that has not been branched on, and that must still be explored, because there is a possibility that it contains a solution better than the incumbent solution. The initial region is not active, because we have branched on it. Region 2 is not active since the best integer solution has been found in that region. Region 1, however, is still active and must be explored. The lower bound over this region is 13; thus, the optimal solution to the entire problem must have an objective function value somewhere between 13 and 14 (inclusive). We recursively divide region 1, in which $x_1 \leq 1$. Since the optimal solution in this region was (1, 1.5), we branch by creating two new subproblems: one in which both $x_1 \leq 1$ and $x_2 \leq 1$ (called region 3), and one in which both $x_1 \leq 1$ and $x_2 \geq 2$ (called region 4). Once again, all integer solutions in region 1 are contained in either region 3 or region 4.

However, note that region 3 is empty, because the stipulation that both x_1 and x_2 are no more than 1 makes it impossible to satisfy (A.16). There are therefore no integer solutions in this region either, and so we stop searching region 3. This region is said to be fathomed by infeasibility. The optimal solution to region 4‘s linear relaxation is (0.5, 2), with objective function value 14. However, our incumbent solution has an objective function value of 14. We have not found the best integer solution in region 4, but we know that the best solution in region 4 will not improve the incumbent solution we have found. Thus, we are not interested in any integer feasible solution in region 4, and we stop searching that region. (An alternative optimal integer solution can exist in that region, but we are not seeking to find all optimal solutions, just one.) Region 4 is said to be fathomed by bound.

Figure A.3 depicts a tree representation of this search process, which is called the "branch-and-bound tree". Each node of the tree represents a feasible region. Now, there are no more regions to be examined (no more active nodes), and the algorithm terminates with the incumbent solution, (2, 1), as an optimal solution.

A formal description of the branch-and-bound algorithm for minimization problems is given as follows.

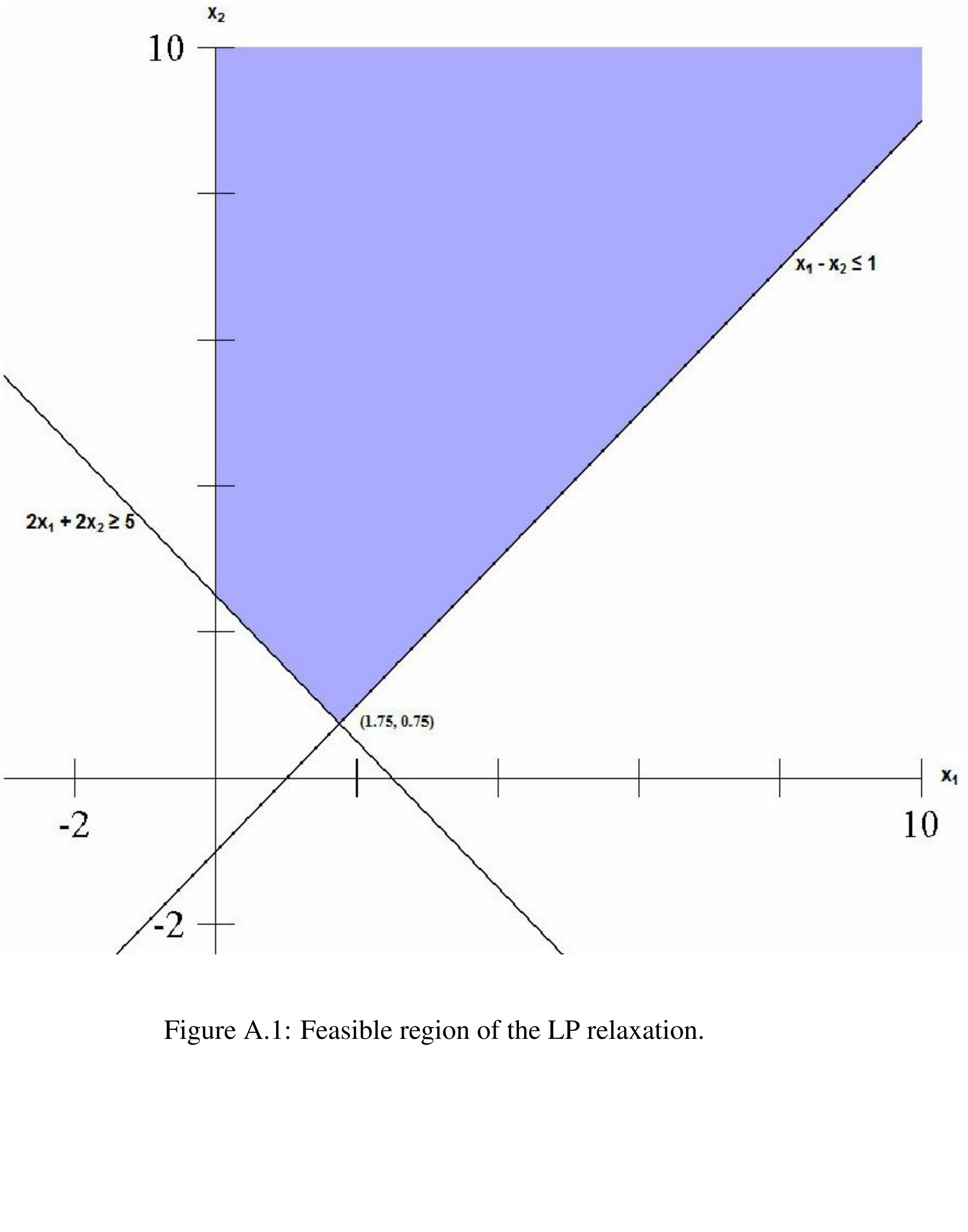


Figure A.1: Feasible region of the LP relaxation.

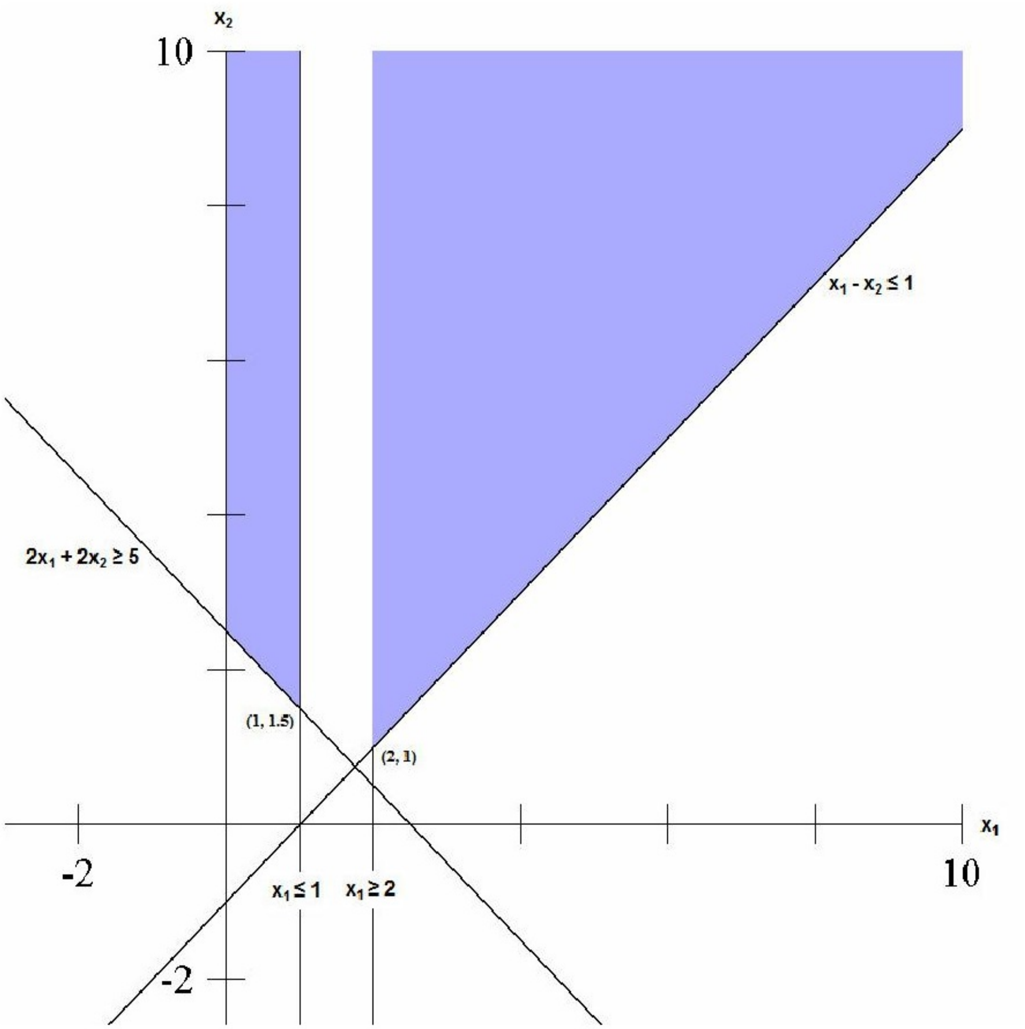


Figure A.2: Feasible regions of the subproblems.

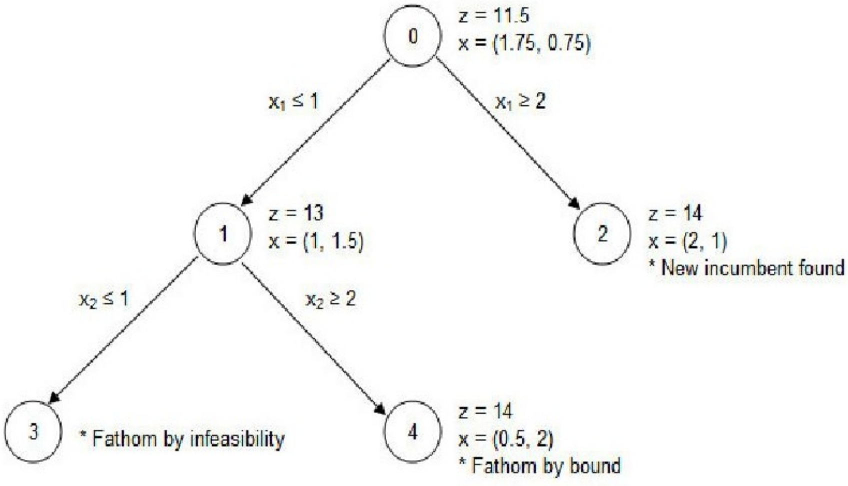


Figure A.3: Branch-and-Bound tree.

Step 0 Set the incumbent objective $v = \infty$ (assuming that no initial feasible integer solution is available). Set the active node count $k = 1$ and denote the original problem as an "active" node. Go to Step 1.

Step 1 If $k = 0$, then stop: the incumbent solution is an optimal solution. (If there is no incumbent, i.e., $v = \infty$, then the original problem has no integer solution.) Else, if $k > 1$, go to Step 2.

Step 2 Choose any active node, and call it the "current" node. Solve the LP relaxation of the current node, and make it inactive. If there is no feasible solution, then go to Step 3. If the solution to the current node has objective value $z^* \geq v$, then go to Step 4. Else, if the solution is all integer (and $z^* < v$), then go to Step 5. Otherwise, go to Step 6.

Step 3 Fathom by infeasibility. Decrease *k* by 1 and return to Step 1.

Step 4 Fathom by bound. Decrease *k* by 1 and return to Step 1.

Step 5 Fathom by integrality. Replace the incumbent solution with the solution to the current node. Set $v = z^*$, decrease k by 1, and return to Step 1.

Step 6 Branch on the current node. Select any variable that is fractional in the LP solution to the current node. Denote this variable as x_s and denote its value in the optimal solution as *f*. Create two new active nodes: one by adding the constraint $x_s \leq |f|$ to the current node, and the other by adding $x_s \geq |f|$ to the current node. Add 1 to k (two new active nodes, minus one due to branching on the current node) and return to Step 1.

Note that in Step 0, a *heuristic procedure* could be executed to quickly obtain a good-quality solution to the MILP with no guarantees on its optimality. This solution would then become our initial incumbent solution, and could possibly help conserve branch-and-bound memory requirements by increasing the rate at which active nodes are fathomed in Step 4. In Step 2, we may have several choices of active nodes on which to branch, and in Step 6, we may have several choices on which variable to perform the branching operation. There has been much empirical research designed to establish good general rules to make these choices, and these rules are implemented in commercial solvers. However, for specific types of formulations, one can often improve the efficiency of the branch-and-bound algorithm by experimenting with node selection and variable branching rules.

The best-case scenario in solving a problem by branch-and-bound is that the original node yields an optimal LP solution that happens to be integer, and the algorithm terminates immediately. Indeed, in (A.16), by simply adding the constraint $x_1 + x_2 \geq 3$ and solve the LP relaxation, we would obtain the optimal solution (2, 1) immediately.

Thus, a classical way to reduce the presence of fractional solutions is to find *valid inequalities*, which do not cut off any integer solutions, but do cut off some fractional solutions. A *cutting plane* is a valid inequality that removes the optimal LP relaxation solution from the feasible region. The *cutting plane* method is an umbrella term for optimization methods which iteratively refine a feasible set or objective function by means of linear inequalities. Such procedures are generally used to find integer solutions to integer and mixed integer linear programming problems, and may be used also to solve other general optimization problems.

The theory of linear programming dictates that under mild assumptions (if the linear program has an optimal solution, and if the feasible region does not contain a line), one can always find a vertex that is optimal. The obtained optimal solution is tested for being integer. If it is not, there is guaranteed to exist a linear inequality that separates this LP relaxation solution from the *convex hull* of the set of integer solutions. Finding such an inequality is known as the *separation problem*, and such an inequality is a *cut*. A cut can be added to the relaxed linear program. Then, the current non-integer solution is no longer feasible to the relaxation. This process is repeated until an optimal integer solution is found.

In theory, MILP can be solved without branching either by (a) including enough valid inequalities before solving the LP relaxation, so that the LP relaxation provides an integer solution, or (b) looping between solving the LP relaxation, adding a cutting plane, and re-solving the LP relaxation, until the LP relaxation yields an integer solution.

However, using these approaches by themselves may suffer from numerical instability problems or require the solution of intractable problems. Therefore, the most effective implementations often use a combination of valid inequalities added a *priori* to the model, after which branch-and-bound is executed, with cutting planes periodically added to the nodes of the branch-and-bound tree. This approach is called "branch-and-cut". Valid inequality and cutting-plane approaches can either be generic or problem-specific. Clearly, the second approach needs a problem-by-problem analysis, but can provide very efficient solution techniques. A large amount of research work on these subjects during the last 50 years lead to the development of many different cut types. Many classical cutting plane approaches are described in greater detail for instance in [104].

References

[1] Andersen R., Chung F., Lang K.: Local graph partitioning using PageRank vectors. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 475-486 (2006).

[2] Balas E.: Facets of the Knapsack Polytope, *Mathematical Programming* 8, 146-164 (1975).

[3] Ball R.M.: The use and definition of travel-to-work areas In Great Britain: some problems. *Regional Studies* 14(2), 125-139 (1980).

[4] Bankier M.: Experienced with the New Imputation Methodology used in the 1996 Canadian Census with extension for future Censuses. In: Proceedings of the Workshop on Data Editing, UN/ECE, Italy (1999).

[5] Barnard S.T., and Simon H.D.: Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems, *Concurrency: Practice and Experience* 6(2), 101-117 (1994).

[6] Bazaraa M.S., Jarvis J.J., and Sherali H.D.: *Linear Programming and Network Flows*. John Wiley & Sons, second edition, New York (1990).

[7] Bell R.M., Cohen M.L.: *Coverage measurement in the 2010 Census - Panel on Correlation Bias and Coverage Measurement in the 2010 Decennial Census*, The National Academic Press: Washington, DC (2008).

[8] Bianchi G., Bruni R., Nucara R., Reale A.: Data Clustering for Improving the Selection of Donors for Data Imputation, In: Proceedings of CLADAG2005, Parma, Italy (2005).

[9] Bianchi G., Bruni R., Reale A.: Information Reconstruction via Discrete Optimization for Agricultural Census Data, *Applied Mathematical Sciences* 6(125), 6241-6251 (2012).

[10] Bianchi G., Bruni R., Reale A.: Balancing of Agricultural Census Data by Using Discrete Optimization, *Optimization Letters* 8, 1553-1565 (2014).

[11] Bianchi G., Bruni R., Reale A.: Open Source Integer Linear Programming Solvers for Error Localization in Numerical Data. In N. Torelli, F. Pesarin, A. Bar-Hen (Eds.), *Advances in Theoretical and Applied Statistics*, Springer-Verlag, New York (2012).

[12] Bianchi G., Bruni R., Reale A.: A Combinatorial Optimization Approach to the Selection of Statistical Units. *Journal of Industrial & Management Optimization (JIMO)*12(2), 515-527 (2016).

[13] Bianchi G., Bruni R., Reale A., Sforzi F.: A Min-Cut Approach to Functional Regionalization, with a Case Study of the Italian Local Labour Market Areas. *Optimization Letters*. DOI: 10.1007/s11590-015-0980-6, ISSN:1862-4472, e-ISSN: 1862-4480 (2015).

[14] Bichot C.E., Siarry P.: *Graph Partitioning*. ISTE - John Wiley, London (2011).

[15] Blum C., Puchinger J., Raidl G. R. , and Roli A.: Hybrid metaheuristics in combinatorial optimization: A survey, *Applied Soft Computing* 11(6), 4135-4151 (2011).

[16] Blum C. and Roli A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys (CSUR)* 35(3), 268-308 (2003).

[17] Bond S. and Coombes M.: 2001-based Travel-To-Work Areas Methodology. Technical report, CURD Working Paper (2007).

[18] Boix R., Trulln J.: Industrial Districts, Innovation and I-district Effect: Territory or Industrial Specialization?, *European Planning Studies*, 18(10), 1707-1729 (2010).

[19] Brinkmeier M.: A Simple and Fast Min-Cut Algorithm, *Theory of Computing Systems* 41(2), 369-380 (2007).

[20] Brown L., Holmes J.: The delimitation of functional regions, nodal regions, and hierarchies by functional distance approaches, *Journal of Regional Science* 11(1), 57-72 (1971).

[21] Bruni R., Reale A., Torelli R.: Optimization Techniques for Edit Validation and data Imputation. In: Proceedings of the Statistics Canada Symposium 2001 *Achieving Data Quality in Statistical Agency: a Methodological Perspective*. XVIIIth International Symposium on Methodological Issues (2001).

[22] Bruni R.: Discrete Models for Data Imputation. *Discrete Applied Mathematics* 144, 59-69 (2004).

[23] Bui T. and Jones C.: A heuristic for reducing fill in sparse matrix factorization. In: Proceedings of the 6th SIAM Conf. Parallel Processing for Scientific Computing, 445-452 (1993).

[24] Canello J., Pavone P.: Mapping the Multifaceted Patterns of Industrial Districts: A New Empirical Procedure with Application to Italian Data, *Regional Studies* http://dx.doi.org/10.1080/00343404.2015.1011611 (2015).

[25] Casado-Diaz J.M.: Local Labour Market Areas in Spain: A Case Study, *Regional Studies* 34(9), 843-856 (2000).

[26] Casado-Diaz J.M., Coombes M.G.: The delineation of 21st century local labour market areas: a critical review and research agenda. Boletín de la Asociación de Geógrafos Españoles 57, 7-32 (2011).

[27] Christofides N.: Combinatorial optimization. In A Wiley-Interscience Publication, Based on a series of lectures, given at the Summer School in Combinatorial Optimization, held in Sogesta, Italy, May 30th-June 11th, 1977, edited by Nicos Christofides,1, John Wiley, Chichester (1979).

[28] Coombes M.G., Openshaw S.: The Use and Definition of Travel-to-Work Areas in Great Britain: Some Comments, *Regional Studies* 16(2), 141-149 (1982).

[29] Coombes M.G., Green A.E., Openshaw S.: An efficient algorithm to generate official statistical reporting areas: the case of the 1984 Travel-to-Work Areas revision in Britain, *Journal of the Operational Research Society* 37, 943-953 (1986).

[30] Coombes M.G.: Defining boundaries from syntetic data, *Environment and Planning* 32, 1499-1518, (2000).

[31] Coombes M.G.: Study on employment zones. Document E/LOC/20. Luxembourg: Eurostat (1992).

[32] Dahmann D.C., Fitzsimmons J.D., (eds.): Metropolitan and nonmetropolitan areas: New approaches to geographical definition. Bureau of the Census Working Paper 12, Washington: Bureau of the Census (1995).

[33] Dantzig G.B.: *Linear Programming and Extensions*. Report R-366-PR, RAND Corporation, USA (1963).

[34] Dantzig G.B., Thapa M.N.: *Linear programming 1: Introduction.* Springer-Verlag, Berlin (1997).

[35] Dantzig G.B., Thapa M.N.: *Linear Programming 2: Theory and Extensions.* Springer-Verlag, Berlin (2003).

[36] Deville J.C., Till Y. : Efficient Balanced Sampling: the Cube Method, *Biometrika*, 91, 893-912 (2004).

[37] Devine K., Boman E., Heaphy R., Hendrickson B. , and Vaughan C.: Zoltan data management services for parallel dynamic applications. *Computing in Science and Engineering*, 4(2), 90-97 (2002).

[38] Diestel R.: *Graph Theory*, 4th edition, Springer-Verlag, Berlin (2010).

[39] Di Giacinto V., Gomellini M., Micucci G., Pagnini M.: Mapping local productivity advantages in Italy: industrial districts, cities or both?, *Journal of Economic Geography* 14 (2), 365-394 (2014).

[40] Djidjev H. N.: A scalable multilevel algorithm for graph clustering and community structure detection. In *Algorithms and models for the web-graph*, Springer-Verlag, Berlin, 117-128 (2008).

[41] Donath W. E. and Hoffman A. J.: Algorithms for Partitioning of Graphs and Computer Logic Based on Eigenvectors of Connection Matrices, IBM Technical Disclosure Bulletin, 15(3): 938-944 (1972).

[42] Donath W. E. and Hoffman A. J.: Lower Bounds for the Partitioning of Graphs, *IBM Journal of Research and Development*, 17(5):420-425 (1973).

[43] Duque J.C., Ramos R., Suriach, J.: Supervised regionalization methods: a survey. *International Regional Science Review* 30, 195-220 (2007).

[44] European Parliament, *Regulation of the European Parliament* N. 1166, (2008).

[45] Farmer C.J.Q., Steward Fotheringam A.: Network-based functional regions, *Environment and Planning* 43, 2723-2741 (2011).

[46] Food and Agriculture Organization of the United Nations (FAO), *A system of integrated agricultural censuses and surveys*. Vol.1 - World Programme for the Census of Agriculture 2010. FAO Statistical Development Series (2005).

[47] Fearne A., Ivarez-Coque J.M.G., Mercedes T.L.U., Garca S.. Innovative firms and the urban/rural divide: the case of agro-food system, *Management Decision*, 51(6), 1293-1310 (2013).

[48] Feldman O., Simmonds D., Troll N. and Tsang F.: Creation of a system of functional areas for England and Wales and for Scotland. Paper presented at the 52th Annual Conference of the North American Regional Science Association, Las Vegas, NV. (2005).

[49] Fellegi I. P. and Holt D.: A Systematic Approach to Automatic Edit and Imputation, *Journal of the American Statistical Association.* 71, 17-35, (1976).

[50] Ferri M., Piccioni M.: Optimal selection of statistical units: An approach via simulated annealing, *Computational Statistics & Data Analysis* 13 , 47-61 (1992).

[51] Fiduccia C. M. and Mattheyses R. M.: A linear time heuristic for improving network partitions. In: Proceedings of 19th IEEE Design Automation Conference, 175-181 (1982).

[52] Fiedler M.: Algebraic Connectivity of Graphs, *Czechoslovak Math. J.* 23, 298-305 (1973).

[53] Fiedler M.: Eigenvectors of acyclic matrices, *Czechoslovak Math. J.* 25, 607-618 (1975).

[54] Fischer M.M.: Regional taxonomy: A comparison of some hierarchic and non-hierarchic strategies, *Regional Science and Urban Economics* 10, 503-537 (1980).

[55] Flórez-Revuelta F., Casado-Díaz J.M., Martínez-Bernabeu L.: An evolutionary approach to the delineation of functional areas based on travel-to-work flows, *International Journal of Automation and Computing* 5(1), 10-21 (2008).

[56] Ford L.R., Fulkerson D.R., Maximal flow through a network, *Canadian Journal of Mathematics* 8, 399-404 (1956).

[57] Fortunato S., Barthélemy M.: Resolution limit in community detection. In: Proceedings of the National Academy of Sciences of the United States of America, 104(1), 36-41 (2007).

[58] Fusco, G., Caglioni, M.: Hierarchical clustering through spatial interaction data. The case of commuting flows in South-Eastern France. Lecture Notes in Computer Science 6782, 135-151 (2011).

[59] Garey M.R., Johnson D.S.: *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman: San Francisco, CA (1979).

[60] Girvan M., Newman M.E.J.: Community structure in social and biological networks. In: Proceedings of the National Academy of Sciences of the United States of America 99(12), 7821-7826 (2002).

[61] Giusti A., Grassini L.: Cluster analysis of census data using the symbolic data approach, *Advances in Data Analysis and Classification* 2 (2), 163-176 (2008).

[62] Goodman J.F.B.: The definition and analysis of local labour markets: some empirical problems, *British Journal of Industrial Relations* 8, 179-186 (1970).

[63] Groves R.M., Fowler F.J.Jr., Couper M.P., Lepkowski J.M., Singer E., Tourangeau R.: *Survey Methodology* (Wiley Series in Survey Methodology), John Wiley & Sons Inc.: Hoboken, NJ (2009).

[64] Guimerá R., Sales-Pardo M., Amaral, L.: Modularity from fluctuations in random graphs and complex networks, *Physical Review* E, 70:025101 (2004).

[65] Hammer P.L., Johnson E.L., Peled U.N.: Facets of regular 0-1 polytopes, *Mathematical Programming* 8, 179-206 (1975).

[66] Hao J.X., Orlin J.B.: A Faster Algorithm for Finding the Minimum Cut in a Directed Graph, *Journal of Algorithms* 17(3), 424-446 (1994).

[67] Hendrickson B. and Leland R.: A Multilevel Algorithm for Partitioning Graphs, Tech. report SAND93-1301, Sandia National Laboratories, Albuquerque, NM (1993).

[68] Hendrickson B. and Leland R.: The chaco user's guide, version 1.0. Technical Report SAND93-2339, Sandia National Laboratories (1993).

[69] Hillier F.S. and Lieberman G.J.: *Introduction to Operations Research*, McGraw-Hill, New York, NY, 8th edition (2005).

[70] Hopcroft J., Tarjan R.: Efficient algorithms for graph manipulation. *Communications of the ACM* 16(6), 372-378 (1973).

[71] ISTAT: I sistemi locali del lavoro 2011. Nota metodologica (2015). Retrieved from http://www.istat.it/it/files/2014/12/nota-metodologica_SLL2011_rev20150205.pdf

[72] Jourdan L., Basseur M., and Talbi E.G.: Hybridizing exact methods and metaheuristics: A taxonomy, *European Journal of Operational Research* 199(3), 620-629 (2009).

[73] Karypis G., Aggarwal R., Kumar V. and Shekhar S.: Multilevel hypergraph partitioning: Application in vlsi domain. In: Proceedings of the Design and Automation Conference (1997).

[74] Karypis G. and Kumar V.: METIS: Unstructured graph partitioning and sparse matrix ordering system. Technical report, Dept. Computer Science, University of Minnesota (1995). http://www.cs.umn.edu/ karypis/metis.

[75] Karypis G. and Kumar V.: Parmetis: Parallel graph partitioning and sparse matrix ordering library. Technical Report 97-060, Dept. Computer Science, University of Minnesota (1997). http://www.cs.umn.edu/ metis.

[76] Karypis G. and Kumar V.: hMetis: A Hypergraph Partitioning Package. Version 1.5.3. Technical Report, Dept. Computer Science, University of Minnesota (1998). http://www.cs.umn.edu/ karypis.

[77] Karypis G. and Kumar V.: Multilevel k-way hipergraph partitioning. Technical Report TR 98-036, Dept. Computer Science, University of Minnesota (1998). http://www.cs.umn.edu/ karypis.

[78] Karypis G. and Kumar V.: A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing* 20(1) 359-392 (1998).

[79] Kernighan B. W. and Lin S.: An efficient heuristic procedure for partitioning graphs, *Bell System Technical Journal* 49(2), 291307 (1970).

[80] Kim, H., Chun, Y., Kim, K.: Delimitation of Functional Regions Using a p-Regions Problem Approach, *International Regional Science Review* doi:10.1177/0160017613484929 (2013).

[81] Kremers W.K.: Completeness and Unbiased Estimation for Sum-Quota Sampling, *Journal of the American Statistical Association* 81, 1070-1073 (1986).

[82] Kropp P., Schwengler B.: A Three-Step Method for Delineating Functional Labour Market Regions. Draft version. Paper preparated for the Regional Studies Association European Conference. Izmir, Turkey, 16-18 June (2014)

[83] La Salle D. and Karypis G.: Multi-Threaded Graph Partitioning. In: Proceedings of 27th IEEE International Parallel and Distributed Processing Symposium, (2013).

[84] Landré M. and Håkansson J.: Rule versus interaction function: evaluating regional aggregations of commuting flows in Sweden, *European Journal of Transport and Infrastructure Research* 13(1),1-19 (2013).

[85] Lang K., Rao S.: A flow-based method for improving the expansion or conductance of graph cuts. In: Proceedings of the 10th International IPCO Conference on Integer Programming and Combinatorial Optimization, 325-337 (2004).

[86] Lazzeretti L., Capone F.: Spatial Spillovers and Employment Dynamics in Local Tourist Systems in Italy (19912001), *European Planning Studies* 17 (11), 1665-1683 (2009).

[87] Leskovec J., Lang K.J., Mahoney M.W.: Network Empirical Comparison of Algorithms for Community Detection. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, April 26-30, AACM New York (2010).

[88] Liu J.W.H.: The minimum degree ordering with constraints, *SIAM J. Sci. Stat. Comput.*, 10(5), 1136-1145 (1989).

[89] Manzari A., Reale A. : Towards a new system for edit and imputation of the 2001 Italian Population Census data: A comparison with the Canadian Nearest-neighbour Imputation methodology. In: Proceedings, International Association of Survey Statisticians, 634-655. (2002).

[90] Marchand H., Martin A., Weismantel R., Wolsey L.: Cutting planes in integer and mixed integer programming, *Discrete Applied Mathematics* 123, 397-446 (2002).

[91] Martínez-Bernabeu, L., Flórez-Revuelta, F., Casado-Díaz, J.M.: Grouping genetic operators for the delineation of functional areas based on spatial interaction, *Expert Systems with Applications* 39, 6754-6766 (2012).

[92] Martini M., Vittadini G.: Analisi del mercato del lavoro: metodi per la costruzione di aree funzionali. In *Aree funzionali per il mercato del lavoro*, Lazio Ricerche, Note e Studi dell'IRSPEL, 1, 16-28 (1985).

[93] Masser I., Brown P.J.B.: Hierarchical aggregation procedures for interaction data, *Environment and Planning* 7, 509-523 (1975).

[94] Miha K., Lisec A. , and Drobne S.: Methods for delineation of functional regions using data on commuters. In: Proceedings of the 13-th AGILE International Conference on Geographic Information Science, Guimares, Portugal (2010).

[95] Mohar B.: *The Laplacian Spectrum of Graph*, 871-898, John Wiley, New York (1991).

[96] Mohar B. and Poljak S.: Eigenvalues in combinatorial optimization. Department of Mathematics, University of Ljubiana, Slovenia (1992).

[97] Moser C.A., Quota Sampling, *Journal of the Royal Statistical Society* 115, 411-423 (1952).

[98] Mucherino A., Papajorgji P., Pardalos P.M.: *Data Mining in Agriculture*. Springer-Verlag, New York (2009).

[99] Murty K.G.: *Linear programming*. John Wiley & Sons Inc.: New York (1983).

[100] Nader G.A.: The delineation of a hierarchy of nodal regions by means of higher-order factor analysis, *Regional Studies* 15(6), 475-492 (1981).

[101] Nagamochi H., Ibaraki T.: Computing edge-connectivity in multigraphs and capacitated graphs, *SIAM Journal of Discrete Mathematics* 5, 54-66 (1992).

[102] Nagamochi H., Ibaraki T.: Graph connectivity and its augmentation: applications of MA orderings, *Discrete Applied Mathematics* 123(1-3), 447-472 (2002).

[103] Nel J.H., Krygsmany S.C. and De Jong T.: The identification of possible future provincial boundaries for South Africa based on an analysis of journey-to-work data. ORiON: *The Journal of ORSSA* 24(2), 131-156 (2008).

[104] Nemhauser G.L. and Wolsey L.A.: *Integer and Combinatorial Optimization*, J. Wiley & Sons, Inc., New York (1999).

[105] Newman M.E. and Girvan M.: Finding and evaluating community structure in networks, Physical Review E 69, 026113, (2004).

[106] Nystuen J.D., Dacey M.F.: A Graph Theory Interpretation of Nodal Regions. In: Papers and Proceedings of the Regional Science Association 7(1), 29-42(1961).

[107] Orasi A., Sforzi F.: I sistemi locali del lavoro. ISTAT research report published online: http://dawinci.istat.it/daWinci/jsp/md/download/sll_comunicato.pdf. (2005).

[108] Papadimitriou C. H.: *Computational complexity*. John Wiley & Sons Ltd., New York (1994).

[109] Papps K.L., Newell J.O.: Identifying Functional Labour Market Areas in New Zealand: A Reconnaissance Study Using Travel-to-Work Data. Discussion Papers 443, Institute for the Study of Labor (IZA), Bonn (2002).

[110] Pelligrini F.: *SCOTCH 3.4 user's guide*. Research Rep. RR-1264-01, LaBRI (2001).

[111] Pelligrini F.: *PT-SCOTCH 5.1 user's guide*. Research Rep., LaBRI (2008).

[112] Pothen A., Simon H. and Liou K.P.: Partitioning sparse matrices with eigenvectors of graphs, *SIAM Journal on Matrix Analysis and Applications*, 11(3), 430-452 (1990).

[113] Ricca F., Scozzari A., Simeone B.: Political Districting: from classical models to recent approaches, *Annals of Operations Research* 204, 271-299 (2013).

[114] Schrijver A.: *Theory of Linear and Integer Programming*. John Wiley, New York (1986).

[115] Sforzi, F., Openshaw, S., Wymer, C.: La delimitazione di sistemi spaziali sub-regionali: scopi, algoritmi, applicazioni, 3rd AISRe Annual Confer., Venezia, 10-12 November (1982).

[116] Sforzi F.: ”La regionalizzazione dei flussi come base per la pianificazione dei trasporti: alcune valutazioni empiriche delle principali tecniche”. In: A. Reggiani, a cura di, *Territorio e trasporti. Modelli matematici per l'analisi e la pianificazione*, Franco Angeli, Milano, 188-213 (1985).

[117] Sforzi F. (ed.): I mercati locali del lavoro in Italia . ISTAT-IRPET, Seminario su: Identificazione di sistemi territoriali. Analisi della struttura sociale e produttiva in Italia, Roma 3-4 December (1986).

[118] Sforzi F.: The geography of industrial districts in Italy. In E. Goodman, J. Bamford with P. Saynor, eds., *Small Firms and Industrial Districts in Italy*, Routledge, London, 153-173 (1989).

[119] Sforzi F. (ed.): *I Mercati Locali Del Lavoro in Italia.* Franco Angeli, Milano (1989).

[120] Sforzi F., Openshaw S., Wymer C.: La procedura di identificazione dei sistemi locali del lavoro. In: F. Sforzi, a cura di, *I sistemi locali del lavoro 1991.* 235-247, ISTAT, Roma (1997).

[121] Sforzi F.: From Administrative Spatial Units to Local Labour Market Areas. Some Remarks on the Unit of Investigation of Regional Economics with Particular Reference to the Applied Research in Italy. In: Fernández Vázquez, E., Rubiera Morollón (eds.) *Defining the Spatial Scale in Modern Regional Analysis*, 3-21, Springer-Verlag, Berlin (2012)

[122] Smart M.W.: Labour market areas: uses and definition, *Progress in Planning* 2(4), 239-353 (1974)

[123] Smith T.M.F.: Populations and Selection: Limitations of Statistics, *Journal of the Royal Statistical Society.* Series A (Statistics in Society) 156, 144-166 (1993).

[124] Spence N.A., Taylor P.J.: Quantitative methods in regional taxonomy, *Progress in Geography* 2, 1-64 (1970).

[125] Stoer M., Wagner F.: A simple min-cut algorithm, *Journal of the ACM*, 44(4), 585-591 (1997).

[126] Sudman S.: Probability Sampling with quotas, *Journal of the American Statistical Association* 61, 749-771 (1966).

[127] Sui X., Nguyen D., Burtscher M., Pingali K.: Parallel Graph Partioning on Multicore Architectures. In: Proceedings of the 23rd international conference on Languages and compilers for parallel computing. Pages 246-260 . Springer-Verlag Berlin, Heidelberg (2011).

[128] Spence, N.A., Taylor, P.J.: Quantitative methods in regional taxonomy, *Progress in Geography* 2, 1-64 (1970)

[129] van der Laan L., Schalke R.: Reality versus Policy: The Delineation and Testing of Local Labour Market and Spatial Policy Areas, *European Planning Studies* 9(2), (2001).

[130] van der Zwan J., van der Wel R., de Jong T., Floor H.: Flowmap 7.2 Manual, Faculty of Geosciences, University of Utrecht (2005).

[131] War Manpower Commission: Directory of Important Labor Market Areas, Washington, G.P.O. (1945).

[132] Ward J.M.: Hierarchical Grouping to Optimize an Objective Function, *Journal of the American Statistical Association* 58, 236-244 (1963).

[133] Watts M.: Local Labour Markets in New South Wales: Fact or Fiction?, Centre of Full Employment and Equity, Working Paper No. 04-12 (2004).

[134] White S., Smyth P.A.: Spectral clustering approach to finding communities in graphs. In: Proceedings of the 5th SIAM International Conference on Data Mining, 76-84 (2005).

[135] William M. and Watts M.: Identifying functional regions in Australia using hierarchical aggregation techniques, *Geographical Research* 48.1, 24-41 (2010).

[136] Williams H.P.: *Model Building in Mathematical Programming*. J.Wiley, Chichester, (1993).

[137] Williamson D. P. and Shmoys D. B.: *The design of approximation algorithms*. Cambridge University Press, (2011).

[138] Winston W.L. and Venkataramanan M.: *Introduction to Mathematical Programming: Applications and Algorithms. Volume 1*. Duxbury Press (fourth edition), Belmont, CA, (2002).

[139] White S., Smyth P.A.: Spectral clustering approach to finding communities in graphs. In: Proceedings of the 5th SIAM International Conference on Data Mining, 76-84 (2005).

[140] Wolsey L.A.: Faces for a linear inequality in 0-1 variables, *Mathematical Programming* 8, 165-178 (1975).