Programa de doctorado "Matemáticas"

PhD Dissertation

MATHEMATICAL OPTIMIZATION FOR THE VISUALIZATION OF COMPLEX DATASETS

Author

Vanesa Guerrero Lozano

Supervisors

Prof. Dr. *Emilio Carrizosa Priego*

Prof. Dr. *Dolores Romero Morales*

March 27, 2017

A mis padres.

Agradecimientos

Agradezco a mis directores, Emilio y Dolores, por enseñarme a disfrutar de este trabajo y dejarme aprender de ellos. Gracias a los dos por sacar tiempo para resolver dudas, hacer correcciones y contestar correos a cualquier hora. Emilio, gracias por confiar en mí y ofrecerme tantas oportunidades, por ser paciente y compresivo en lo académico y lo personal. Tenías razón, al final merece la pena subir los escalones, aunque cueste, porque la vista resulta espectacular. Te agradezco enormemente que me dieras la oportunidad de trabajar con Dolores. Loli, eres una mujer ejemplar. Gracias por descubrirme Copenhage, con bici incluída y cafés deliciosos de por medio, por abrirme las puertas de tu casa y tratarme como una más de tu familia. Que sigan los ratitos de charla tanto de matemáticas como de la vida.

A Manolo Barco, por hacer que cruzara las puertas de esta facultad hace ya casi diez años. Gloria y Ana, por tantas horas de estudio y de sonrisas. PhD Mérida, Manu, por ti esta tesis existe. Empezamos a la vez y terminaremos a la vez, como no podría ser de otro modo. Aún estando tan lejos, siempre estás cerca amigo mío.

Compañeros del IMUS, el camino se hace más llevadero cuando lo compartes con gente tan maravillosa como vosotros. Por los ratos de charlas que nunca terminan, por las comidas, los cafés con y sin dulces, las cenas a domicilio y las salidas improvisadas. A mis compañeros de la L2 por hacer los días de trabajo más amenos y, especialmente a vosotras, Alba, Marina y Asun. Gracias por cada momento, los que han sido buenos como los que no lo fueron tanto, de todos hemos aprendido. Alba, gracias por estar siempre disponible para todo; Marina, eres la alegría de este pasillo; Asun, compañera de viajes e historias, te dejo el relevo. No os olvidéis: *"que lo que el IMUS ha unido, no lo separen las postdocs"*.

A esos amigos de toda la vida que siempre tienen una palabra de aliento en el momento adecuado y son capaces de darte el empujoncito que necesitas. Sara, Rocío, Elena, Cristina, Ana, seguiremos luchando por nuestros sueños, pero siempre de la mano. María Luisa, por ser mi medio clarinete, ahora y siempre.

Juanlu, mi compañero de ilusiones y de vida, gracias por estar conmigo y apoyarme en cada paso, por aguantar los malos ratos y disfrutar juntos de los buenos. Gracias por hacerme feliz.

Gracias a mi familia, por estar siempre cerca aún estando lejos. Abuela Pilar, gracias por quererme a pies juntos; abuela Lucía, por seguir haciéndolo pese a todo. Abuelos, sé que me seguís cuidando desde allí. Tita Carmen y tita Pili, por considerarme vuestra niña por mucho que pasen los años. Padrinos, por estar tan pendientes mí aún viviendo a cientos de kilómetros. Primos, no es fácil pero se puede llegar a la meta.

Y, por supuesto, gracias a mis padres por educarme e inculcarme desde pequeña que es necesario esforzarse y trabajar para conseguir lo que quieres. Papá, me alegras la vida. Mamá, creces y yo contigo. Os quiero.

Gracias, a todos...

Resumen

Esta tesis se centra en desarrollar nuevos modelos y algoritmos basados en la Optimización Matemática que ayuden a comprender estructuras de datos complejas frecuentes en el área de Visualización de la Información. Las metodologías propuestas fusionan conceptos de Análisis de Datos Multivariantes y de Optimización Matemática, aunando las matemáticas teóricas con problemas reales.

Como se analiza en el Capítulo 1, una adecuada visualización de los datos ayuda a mejorar la interpretabilidad de los fenómenos desconocidos que describen, así como la toma de decisiones. Concretamente, esta tesis se centra en visualizar datos que involucran distribuciones de frecuencias y relaciones de proximidad, pudiendo incluso ambas variar a lo largo del tiempo. Se proponen diferentes herramientas para visualizar dicha información, basadas tanto en la Optimización (No) Lineal Entera Mixta como en la optimización de funciones Diferencia de Convexas. Además, metodologías como la Búsqueda por Entornos Grandes y el Algoritmo DCA permiten el desarrollo de mateheurísticas para resolver dichos modelos.

Concretamente, el Capítulo 2 trata el problema de visualizar simultáneamente una distribución de frequencias y una relación de adyacencias en un conjunto de individuos. Esta información se representa a través de un mapa rectangular, es decir, una subdivisión de un rectángulo en porciones rectangulares, de manera que las áreas de estas porciones representen las frecuencias y las adyacencias entre las porciones representen las adyacencias entre los individuos. Este problema de visualización se formula con la ayuda de la Optimización Lineal Entera Mixta. Además, se propone una mateheurística basada en este modelo como método de resolución.

En el Capítulo 3 se generaliza el modelo presentado en el capítulo anterior, construyendo una herramienta que permite visualizar simultáneamente una distribución de frecuencias y una relación de disimilaridades. Dicha visualización se realiza mediante la partición de un rectángulo en porciones rectangulares a trozos de manera que el área de las porciones refleje la distribución de frecuencias y las distancias entre las mismas las disimilaridades. Se plantea un modelo No Lineal Entero Mixto para este problema de visualización, que es resuelto a través de una mateheurística basada en la Búsqueda por Entornos Grandes.

En contraposición a los capítulos anteriores, en los que se busca una partición de la región de visualización, el Capítulo 4 trata el problema de representar una distribución de frecuencias y una relación de disimilaridad sobre un conjunto de individuos, sin forzar a que haya que recubrir dicha región de visualización. En este modelo de visualización los individuos son representados como cuerpos convexos cuyas áreas son proporcionales a las frecuencias dadas. El objetivo es determinar la localización de dichos cuerpos convexos dentro de la región de visualización. Para resolver este problema, que generaliza el tradicional Escalado Multidimensional, se utilizan técnicas de optimización basadas en funciones Diferencia de Convexas.

En el Capítulo 5, se extiende el modelo desarrollado en el capítulo anterior para el caso en el que los datos son dinámicos, es decir, las frecuencias y disimilaridades se observan a lo largo de varios instantes de tiempo. Se emplean técnicas de optimización de funciones Diferencias de Convexas así como Optimización Cuadrática Binaria No Convexa para la resolución del modelo.

Todas las metodologías propuestas han sido testadas en datos reales.

Finalmente, el Capítulo 6 contiene las conclusiones a esta tesis, así como futuras líneas de investigación.

Abstract

This PhD dissertation focuses on developing new Mathematical Optimization models and solution approaches which help to gain insight into complex data structures arising in Information Visualization. The approaches developed in this thesis merge concepts from Multivariate Data Analysis and Mathematical Optimization, bridging theoretical mathematics with real life problems.

The usefulness of Information Visualization lies with its power to improve interpretability and decision making from the unknown phenomena described by raw data, as fully discussed in Chapter 1. In particular, datasets involving frequency distributions and proximity relations, which even might vary over the time, are the ones studied in this thesis. Frameworks to visualize such enclosed information, which make use of Mixed Integer (Non)linear Programming and Difference of Convex tools, are formally proposed. Algorithmic approaches such as Large Neighborhood Search or Difference of Convex Algorithm enable us to develop matheuristics to handle such models.

More specifically, Chapter 2 addresses the problem of visualizing a frequency distribution and an adjacency relation attached to a set of individuals. This information is represented using a rectangular map, i.e., a subdivision of a rectangle into rectangular portions so that their areas reflect the frequencies, and the adjacencies between portions represent the adjacencies between the individuals. The visualization problem is formulated as a Mixed Integer Linear Programming model, and a matheuristic that has this model at its heart is proposed.

Chapter 3 generalizes the model presented in the previous chapter by developing a visualization framework which handles simultaneously the representation of a frequency distribution and a dissimilarity relation. This framework consists of a partition of a given rectangle into piecewise rectangular portions so that the areas of the regions represent the frequencies and the distances between them represent the dissimilarities. This visualization problem is formally stated as a Mixed Integer Nonlinear Programming model, which is solved by means of a matheuristic based on Large Neighborhood Search.

Contrary to previous chapters in which a partition of the visualization region is sought, Chapter 4 addresses the problem of visualizing a set of individuals, which has attached a dissimilarity measure and a frequency distribution, without necessarily cov-

ering the visualization region. In this visualization problem individuals are depicted as convex bodies whose areas are proportional to the given frequencies. The aim is to determine the location of the convex bodies in the visualization region. In order to solve this problem, which generalizes the standard Multidimensional Scaling, Difference of Convex tools are used.

In Chapter 5, the model stated in the previous chapter is extended to the dynamic case, namely considering that frequencies and dissimilarities are observed along a set of time periods. The solution approach combines Difference of Convex techniques with Nonconvex Quadratic Binary Optimization.

All the approaches presented are tested in real datasets.

Finally, Chapter 6 closes this thesis with general conclusions and future lines of research.

Contents

Chapter 1

Introduction

In the Big Data era, Information Visualization arises as a powerful discipline to deal with knowledge discovery in complex datasets, (Keim et al., 2008; Liu et al., 2014; Wolfe, 2013). Extracting knowledge from data, such as global underlying patterns or unusual behaviors, has become a crucial task for analysts to build models and improve decision making in many areas, such as Health Care (Bertsimas et al., 2016; Ustun and Rudin, 2016), Risk Management (Baesens et al., 2003; Van Vlasselaer et al., 2016) or Data Mining (Bao and Datta, 2014). In order to bring to the forefront those hidden structures, visualizing data by means of suitable graphic tools is a widespread exploratory technique in Information Visualization and, in general, in Analytics. Nevertheless, the increase in data complexity has made the classic visualization techniques obsolete, and more sophisticated frameworks are thus needed (Chen and Zhang, 2014).

In this context, Mathematical Optimization plays an important role, both developing new models and algorithmic approaches to adapt existent visualization tools as well as creating new frameworks which cope with nowadays requirements, (Abbiw-Jackson et al., 2006; Dzemyda et al., 2013; Liu et al., 2014; Olafsson et al., 2008; Strehl and Ghosh, 2003; von Landesberger et al., 2016). Due to its versatility, Mathematical Optimization allows to design (new or improved) models and solution approaches (heuristics, matheuristics and exact methods) which gain insight into specific datasets' features. For instance, one can capture proximity or numerical magnitudes of data entries, such as frequency distributions, as well as categorical attributes by means of aesthetic criteria, such as colors, hues or geometric shapes and layouts. Supervised Learning (Carrizosa and Romero Morales, 2013; Witten and Tibshirani, 2011) and visualization and dimensionality reduction techniques, such as Principal Component Analysis or Multidimensional Scaling, fit into this category due to their flexibility in incorporating extra requirements to enhance interpretability, such as sparsity (Carrizosa and Guerrero, 2014a,b).

The aim of this thesis is to develop new Mathematical Optimization models and solution approaches to build visualization frameworks for complex datasets which arise in Information Visualization. In particular, datasets involving frequency distributions and proximity relations, which even might vary over time, are the ones considered in this PhD dissertation. Dealing with data visualization from an optimization perspective allows analysts to enhance interpretability and help decision making (Carrizosa et al., 2010; Freitas, 2013; Martens et al., 2007).

1.1 The aim of Information Visualization

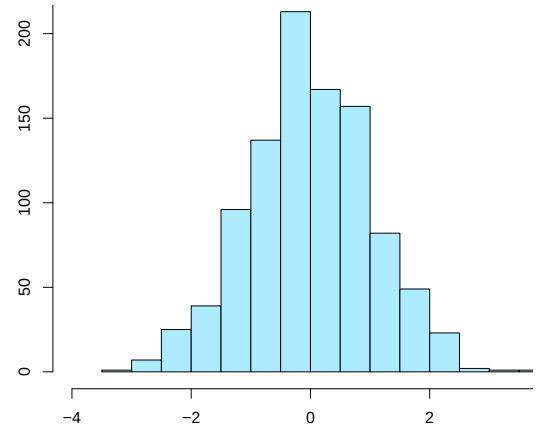
The usefulness of Information Visualization lies with its power to enhance interpretability and decision making from the unknown phenomena described by raw data. Adequately visualizing such data helps, for instance, to make the correct assumptions involved in statistical models or data mining algorithms, as well as to obtain meaningful

presentations of the analysis' outputs. The challenge is to identify the most suitable tool to make the visualization effective for the user's purpose (Keim et al., 2008). From the simplest to the most complex dataset, its features are encoded by means of, e.g. layouts, shapes, colors and hues (Benbasat and Dexter, 1985; Hall et al., 2016; Shmueli et al., 2016). Therefore, the different degrees of complexity in data endows visualization tools with different levels of sophistication, (Dzemyda et al., 2013; Heer et al., 2010; Tufte, 1986).

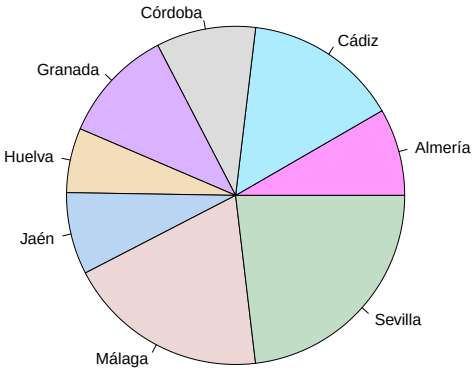
One of the simplest structures of datasets consists of a list of entries with a frequency distribution attached. Examples of these datasets are found in any real-life context: in opinion polls, a set of political parties characterized by their number of voters; in finance, the assets forming a market portfolio distributed by their proportions; or in text mining, a list of words and its frequency in a text. If the aim of the analyst is to study the distribution of data, for instance, for statistical hypothesis testing, the preferred option may be a *histogram* (Tufte, 1986), i.e. a figure which plots a bar for each entry, whose area represents the corresponding frequency. However, if the purpose of the visualization is to give an idea about the relative sizes of the individuals involved, namely relative frequencies, the straightforward approach is to split a given visualization region into portions in such a way that their areas represent the relative frequencies (Spence and Lewandowsky, 1991). The well-known *pie chart* and *fan chart* fit into this category. More creative representations have been specifically designed in some fields. This is the case of text visualization, for which *word clouds* (Viégas and Wattenberg, 2008) have become a baseline to depict the frequency of most used words in, for instance, speeches or Twitter's trending topics. In a word cloud, the list of words, scaled by their frequencies, are written forming a compact layout. This idea can be generalized to any dataset involving frequencies, since the words could be the labels attached to the data entries. Figure 1.1 (a) – 1.1 (d) show examples of a histogram, a pie chart, a fan chart and a word cloud, respectively.

The complexity of a dataset structure increases due to different factors, and thus its visualization becomes more challenging. For instance, data may involve hierarchies, either by their own nature, as in counties, states and countries, or as a result of a grouping procedure, such as a Cluster Analysis. Combining these hierarchies with the faithful representation of a distribution of frequencies has been done, among others, by means of *treemaps* (Shneiderman, 1992) and *circle-packing layouts* (Wang et al., 2006), see Figures 1.2 (a) and 1.2 (b), respectively.

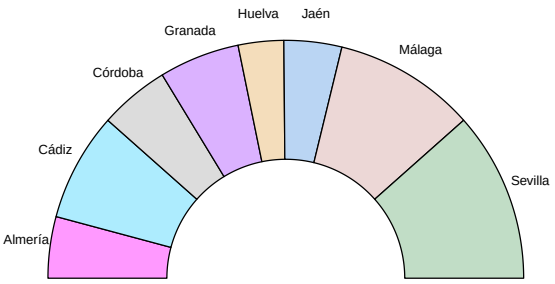
The higher the dimension of a dataset is, the more difficult is to engrave its enclosed information on a picture (Dzemyda et al., 2013). When dealing with multivariate data, namely a list of entries for which p variables have been measured, and thus multiple frequency distributions exist, a *scatter plot* (Tufte, 1986) is the straightforward visualization tool to get a first guess. In a scatter plot each observation is depicted as a dot



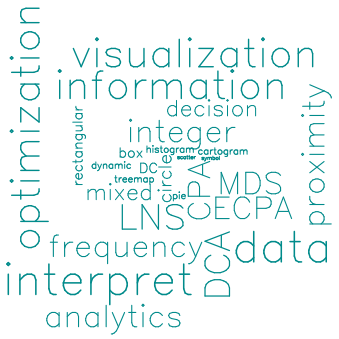
(a) Histogram



(b) Pie Chart

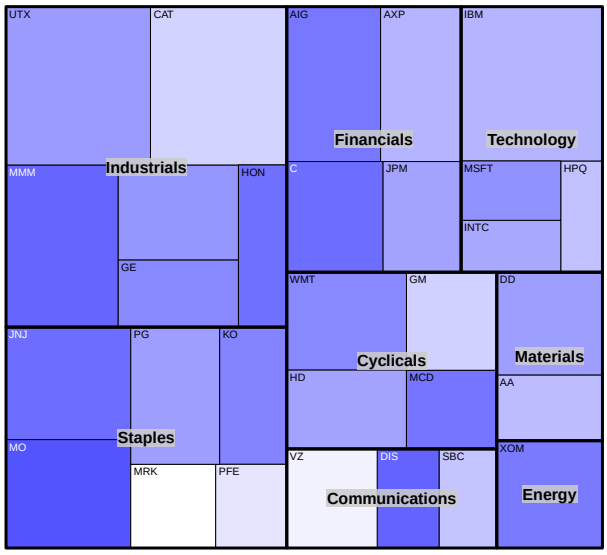


(c) Fan Chart

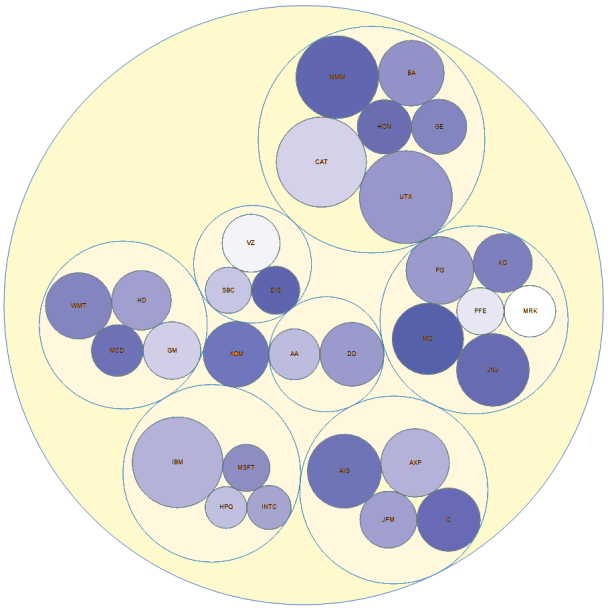


(d) Word cloud

Figure 1.1: Visualization of frequencies.



(a) Treemap



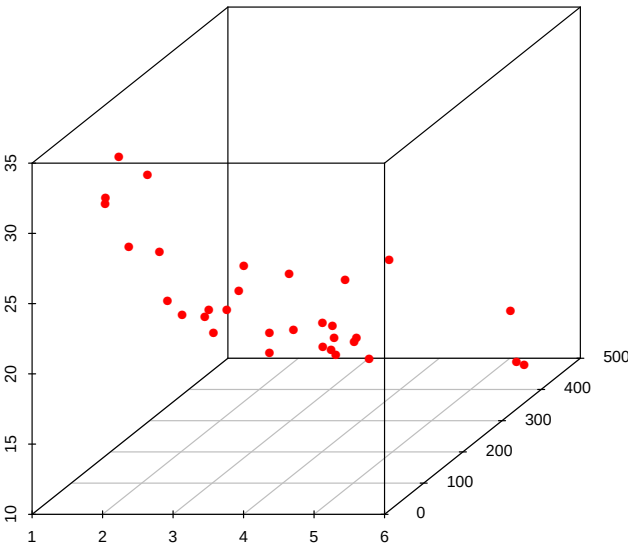
(b) Circle-packing layout

Figure 1.2: Visualization of frequencies and hierarchies.

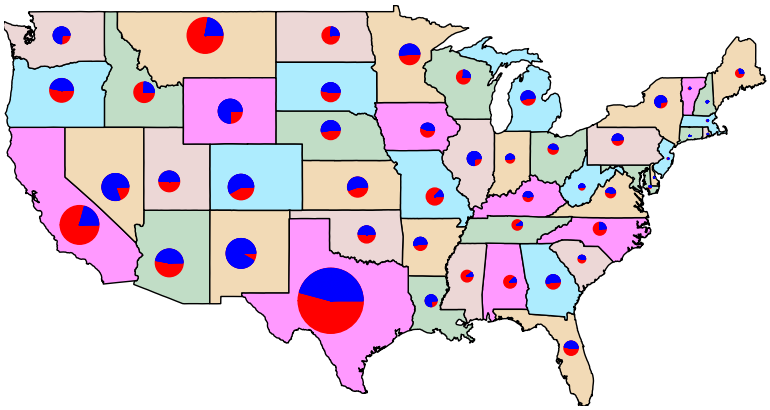
in the p-dimensional Euclidean space, in which each Cartesian edge represents a variable, see Figure 1.3 (a) for a scatter plot with $p = 3$. Since visualizing a p-dimensional space with $p > 3$ is meaningless for any human eye the 2 or 3-dimensional projections are usually studied. Hence, other approaches have been proposed to visualize complex multivariate datasets, including the standard Principal Component Analysis (Pearson, 1901), Multidimensional Scaling (Kruskal, 1964; Torgerson, 1958), as well as more sophisticated tools such as *proportional symbol maps*, (Cabello et al., 2010). In a proportional symbol map, geometric objects, such as discs or rectangles, are placed on a set of locations (usually geographical). Those geometric objects are scaled to one of the frequency distributions involved in the multivariate dataset. Indeed, those geometric objects might be visualization frameworks themselves, which depict e.g. another distribution of frequencies or the results of certain analysis. However, it is unclear what is the most suitable scale to depict such symbols (Cano et al., 2013; Kunigami et al., 2014). Figure 1.3 (b) shows a proportional symbol map considering as locations the 48 U.S. contiguous states in which scaled pie charts are located over each state.

According to Cottam et al. (2012), data which vary over time is one of the most relevant data to be depicted, since most datasets representing a real phenomenon are intrinsically dynamic. The usual approach is to consider a collection of snapshots, which depicts the desired information in each time period. Once the decision about how the dynamic dataset is represented in each snapshot has been made, mainly depending on the data structure, the challenge lies on how to show such collection in a clear and understandable manner: *small multiples*, namely showing each snapshot independently, and *animations* are the most common ways (Archambault and Purchase, 2016). However, abrupt changes in the layout or shapes go in detriment of easy identification of individuals in future periods. Therefore, achieving smoothness in the transition between consecutive snapshots becomes crucial to enhance interpretability.

An important piece of information in Data Science consists of the knowledge of relations that a set of entities may have. Being able to visually identify how close (or far) two individuals are helps to find groups in data and thus, eases the interpretability in terms of vicinity (Kaufman and Rousseeuw, 2009). This type of analysis is very frequent in, for instance, e-commerce to recommend the customer new products based on customers' previous purchases, and social networks to detect and analyze specific communities (Fortunato, 2010). To perform an analysis like this, a proximity measure is to be introduced. There exist different ways of measuring proximities, mainly depending on the application. Adjacencies arise as a first attempt to identify (un)related individuals, whose visualization have been usually managed by *Graph Drawing* techniques (Battista et al., 1999). The individuals are represented as nodes, and there is an edge between two individuals if they are related. See Figure 1.4 (a) for an example of such a graph. Other approaches measure dependency between variables, such as correlations. In general,



(a) Scatter plot



(b) Proportional symbol map

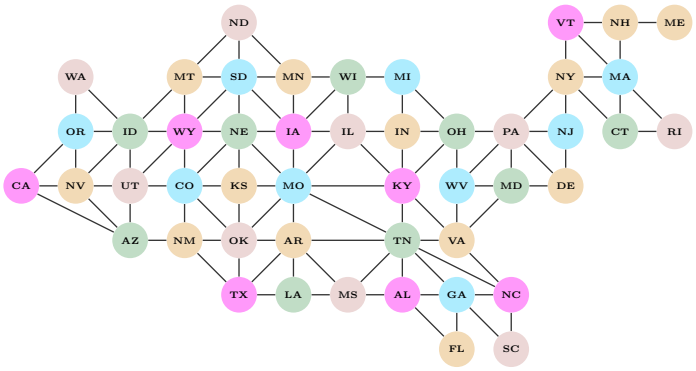
Figure 1.3: Visualization of multivariate data.

computing spatial distances, such the Euclidean or Manhattan distances, between each pair of individuals yields also a straightforward way of measuring proximities. Those numbers are frequently visualized in a matrix structure, in which color saturation shows the intensity of the proximity, called *adjaceny matrix* (Heer et al., 2010). This way the closeness of pairs of individuals by means of colors' intensities could be easily compared. For an example of an adjacency matrix visualization see Figure 1.4 (b). Geographical distances are a specific example of distances that Information Visualization usually has to deal with. Whereas standard representation methods are sufficient to correctly depict them, if an extra piece of information is to be included, such as population rates or vote intention in each region, more sophisticated tools are necessary. In order to give answer to this combination of visualizing frequency distributions and geographical distances, *cartograms* are the usual approach (Tobler, 2004). In a cartogram, each region is scaled in such a way that its area represents a frequency, whereas the relative positions between regions are maintained. There exist approaches in which regions are depicted with relatively simple shapes, usually rectangles (Heilmann et al., 2004) or circles (Dorling, 1996), as the rectangular cartogram shown in Figure 1.4 (c) for the U.S., where the states have been scaled by their land areas (Panse, 2016). Nevertheless, there exist other possibilities rather than correlations or distances, which consist of a set of positive coefficients that become small (close to zero) when two individuals are close to each other and that are large when they are different. These coefficients are called dissimilarities (Kaufman and Rousseeuw, 2009) and their faithful representation has been usually handled through a well-studied technique in Information Visualization: Multidimensional Scaling (MDS) (Kruskal, 1964; Torgerson, 1958). MDS is reviewed in Section 1.2 due to its importance in the methods developed in this thesis.

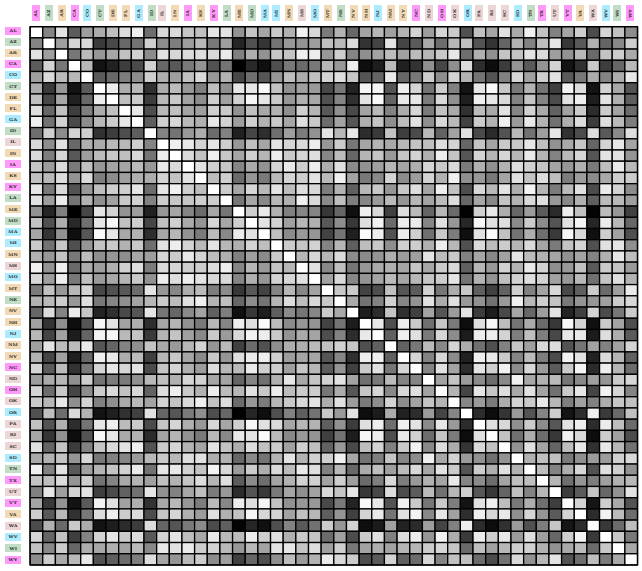
In conclusion, Information Visualization has to deal with data whose complexity may come from different sources: multivariate information, hierarchies, proximities or time-dependent features. Although specific frameworks have been designed for some of them, there exist issues which deserve further attention. That is the case of representing simultaneously a distribution of relative frequencies and adjacencies or dissimilarities, studied in Chapters 2 and 3, respectively. Moreover, combining the visualization of dissimilarities and frequencies, which both may or may not vary over time, is a timely topic, studied in Chapters 4 and 5.

1.2 Multidimensional Scaling

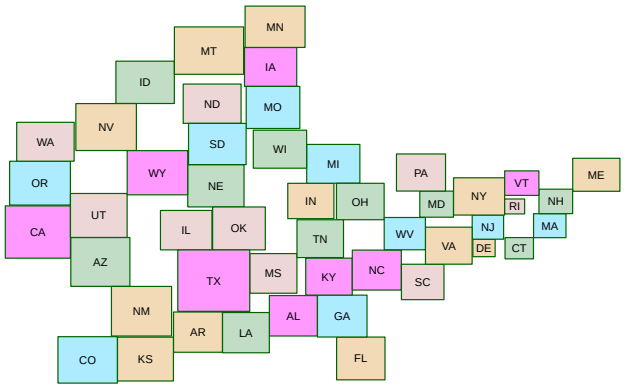
Multidimensional Scaling (MDS) (Borg and Groenen, 2005; Cox and Cox, 2000; Kruskal, 1964; Torgerson, 1958) comprises a family of techniques for the analysis of proximities among a set of individuals, where the term individual encompasses either people, countries, foodstuffs, etc. The aim of MDS is to model the proximity data as



(a) Graph Drawing



(b) Adjacency matrix



(c) Cartogram

Figure 1.4: Visualization of proximities.

distances among points located in a low dimensional space, where each point represents an individual. In general, measures of proximity are twofold: *dissimilarities*, which measure how different two individuals are, and *similarities*, which quantify how much they are alike. Since both concepts are dual, in what follows the discussion about MDS is made for dissimilarities without loss of generality.

In general, a measure of dissimilarity between a pair of individuals is a nonnegative real number, which is close to zero if they are alike, and which becomes large when they are different. Hartigan (1967) gives twelve possible dissimilarity structures, listed by Cox and Cox (2000) cf. Chapter 1, which have different levels of generality. In this thesis, dissimilarity is a function that for each pair of individual v_i and v_j, their dissimilarity δ_{ij} verifies:

(i) $\delta_{ij} \geq$, for all i, j,

(ii) $\delta_{ii} = 0$, for all i,

(iii) $\delta_{ij} = \delta_{ji}$, for all i, j.

Straightforward examples of dissimilarity measures satisfying (i)–(iii) are distances, such as the Euclidean or the Manhattan distances. Nevertheless, distances have stronger assumptions since, contrary to dissimilarities, they must satisfy the triangle inequality. Correlations between variables attached to the individuals are also a common approach to measure the proximity between them. In order to transform correlations into dissimilarities satisfying (i)–(iii) a monotone transformation is usually applied to make the coefficients nonnegative and transform high correlations into small dissimilarities. Indeed, an analogous transformation converts an adjacency relation into a dissimilarity.

Therefore, given a set of individuals $V = \{v_1, \ldots, v_N\}$, which have attached a dissimilarity measure satisfying (i)–(iii), MDS seeks an embedding from V to $\mathbb{R}^n$ such that the dissimilarities are preserved. There exist two usual ways in which this goal is pursued: *metric* and *nonmetric* MDS. Whereas metric MDS models (Torgerson, 1958) attempt to find points $\boldsymbol{x}_i \in \mathbb{R}^n$ whose pairwise distances are approximately equal to an algebraic transformation of the dissimilarities ($\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \approx f(\delta_{ij})$, where $\|\cdot\|$ denotes a norm and $f : \mathbb{R} \longrightarrow \mathbb{R}$ is a continuous parametric monotone function), nonmetric MDS (Kruskal, 1964) aims to represent only the ordinal properties of the dissimilarities (if $\delta_{ij} < \delta_{kl}$, then $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \|\boldsymbol{x}_k - \boldsymbol{x}_l\|$). The metric approach will be the one followed in this thesis.

The usual strategy to handle a metric MDS, hereafter called simply MDS, is to state the problem as a continuous optimization problem, in which the least square error incurred when approximating the transformed dissimilarities by the distances between

the points in the embedding is minimized, yielding

$$\min_{\substack{\boldsymbol{x_i}\in\mathbb{R}^n \\ i=1,\dots,N}} \sum_{v_i,v_j\in V} \left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - f(\delta_{ij})\right)^2, \tag{1.1}$$

where the function f is usually taken as the identity or a linear function, and $\|\cdot\|$ as the Euclidean or Manhattan norm (Hubert et al., 1992; Leung and Lau, 2004). In general, the objective function in (1.1) is known as *stress*. There exist some variants, such as the normalized stress and the weighted stress, and related but different approaches, such as the classic scaling and the s-stress (Borg and Groenen, 2005).

The global optimization problem in (1.1) is known to be multimodal (Trosset and Mathar, 1997; Žilinskas and Podlipskytė, 2003). Indeed, there exists multiplicity of the optimal solution since any isometric transformation of the points yields the same objective value in (1.1). Thus, local search algorithms based on gradients or subgradient might get stuck in local optima, and several strategies have been proposed in the literature to find its global optimum. For instance, the *SMACOF* algorithm, initially proposed by De Leeuw (1977) and then refined by Groenen (1993), is based on an iterative majorization procedure; *ALSCAL* (Takane et al., 1977) is an alternating procedure in two phases, an optimal scaling phase and a model estimation phase; De Leeuw and Heiser (1977) and Le Thi and Pham Dinh (2001) develop approaches based on majorization and Difference of Convex (DC) optimization techniques, and Žilinskas and Žilinskas (2009) design a branch and bound algorithm for MDS with the Manhattan distance.

MDS plays a very important role in the visualization of complex datasets thanks to its capability of reducing the dimensionality to $\mathbb{R}^2$ or $\mathbb{R}^3$ (Buja et al., 2008). Whereas the classic approach stated in (1.1) is able to deal only with dissimilarity data, some work has been done to customize this technique to datasets with additional features, since the Mathematical Optimization model in (1.1) may incorporate extra requirements by means of constraints and terms in the objective function (Chen and Buja, 2009; De Leeuw and Heiser, 1980). This is the case of some graph drawing techniques, which locate nodes by means of an MDS (Battista et al., 1999; Dörk et al., 2012). In a graph visualization context, MDS aims to reproduce the adjacencies' structure while the number of edge crossings is reduced or a certain layout is imposed. For hierarchical data, some approaches attempt to preserve a proximity measure among the levels in the hierarchies and make use of MDS (Duarte et al., 2014; Nocaj and Brandes, 2012). Proportional symbol maps make use of geographical locations to depict extra information given as frequencies (Cano et al., 2013; Kunigami et al., 2014). Indeed, using MDS seems a sensible way to locate symbols to visualize a general dissimilarities structure jointly with frequencies. To do this, a first approach might consist of executing an MDS, and then replacing points by the symbols (say, discs or rectangles) centered at the MDS

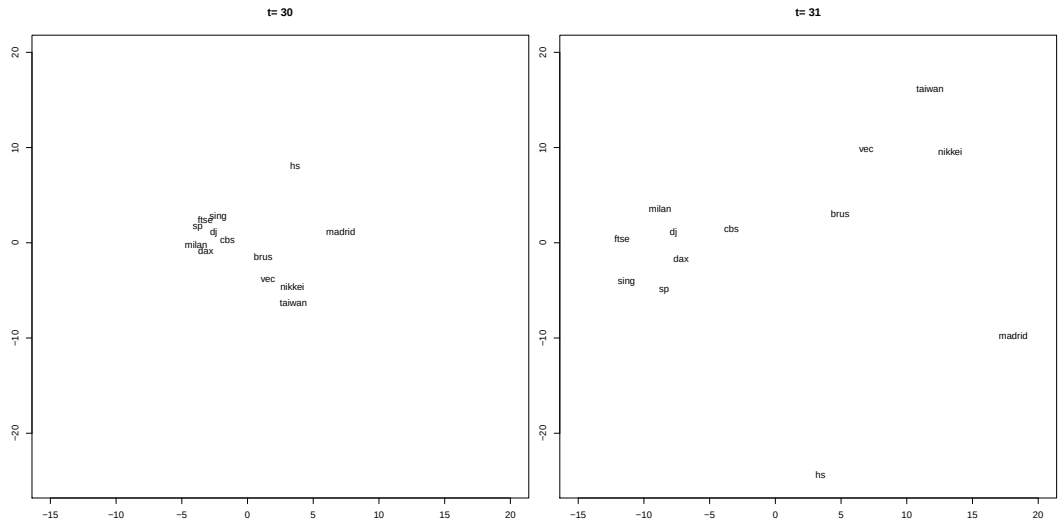


Figure 1.5: Illustration of MDS applied to dynamic dissimilarities coming from the stock markets dataset (Groenen and Franses, 2000).

points, and whose area is proportional to the frequency value. Nevertheless, the scale chosen by the user to (proportionally) depict such information may yield either too small objects or excess of overlapping between them. Thus, the changes in the perception of distances, induced by these (a posteriori) depicted objects, might yield misleading conclusions about the proximity between the individuals (Liu et al., 2013). MDS has been also customized to visualize dynamic data (Groenen and Franses, 2000; Xu et al., 2013). A straightforward approach to visualize dynamic dissimilarities, observed along T time periods, would consist of executing T independent MDS, one per period. Nevertheless, this approach might yield difficult-to-interpret visualizations, especially when the dissimilarities change abruptly in consecutive periods or, since MDS results are invariant under rotations and reflections, the snapshots may turn upside-down. In order to illustrate this statement, let us consider the dataset consisting of $N = 13$ stock markets, studied along $T = 200$ time periods. Dissimilarities are measured through the correlation between stock markets indices as studied in Groenen and Franses (2000). The plots for time periods $t = 30$ and $t = 31$, obtained with the `isoMDS()` function in R, (R Core Team, 2016), are given in Figure 1.5. As the reader can observe, a visual effort is required to read from snapshot $t = 30$ to $t = 31$, since stock markets appear rotated and further from each other.

Summarizing, MDS is a Mathematical Optimization based visualization framework which is broadly used in Information Visualization thanks to its dimensionality reduction purpose. MDS will be at the heart of the models developed in this thesis to help in the visualization of complex datasets.

1.3 Contributions of this thesis

This thesis is devoted to design new Mathematical Optimization based models and solution approaches to visualize complex datasets. Mixed Integer (Non)linear Programming and Difference of Convex techniques are at the heart of these models, which are successfully tested in real-world datasets. In what follows we briefly describe the problems treated and the reasons which make them interesting for the Information Visualization and the Mathematical Optimization communities.

In Chapter 2, based on the paper Carrizosa et al. (2015a), we address the problem of visualizing a frequency distribution and an adjacency relation attached to a set of individuals. We represent this information using a rectangular map, i.e., a subdivision of a rectangle into rectangular portions so that each portion is associated with one individual, their areas reflect the frequencies, and the adjacencies between portions represent the adjacencies between the individuals. This approach extends the classic pie and fan charts, in which the only freedom in the location of the portions is the choice of a permutation. Due to the impossibility of satisfying both area and adjacency requirements, our aim is to fit as well as possible the areas, while representing as many adjacent individuals as adjacent rectangular portions as possible and adding as few false adjacencies, i.e., adjacencies between rectangular portions corresponding to non-adjacent individuals, as possible. We formulate this visualization problem as a Mixed Integer Linear Programming (MILP) model. We propose a matheuristic that has this MILP model and a constrained Multidimensional Scaling approach at its heart. Our experimental results demonstrate that our matheuristic provides rectangular maps with a good fit in both the frequency distribution and the adjacency relation. The methodology proposed in Chapter 2 is applied to three real-world datasets.

Chapter 3, which is based on the paper Carrizosa et al. (2017c), deals with the rigidity of the shapes used to represent individuals in Chapter 2, namely rectangles, and the specific nature of the proximities considered, namely adjacencies. We address the problem of visualizing a frequency distribution and a dissimilarity measure attached to a set of individuals. This information is depicted by means of a Space-filling Box-connected Map, which consists of a partition of a rectangle into so-called box-connected rectangles, in such a way that the area of the box-connected rectangles reflect the frequencies and the distances between them represent the dissimilarities. The construction of a Space-filling Box-connected Map is formally stated as an MILP, in which the violation of the correct representation of frequencies and dissimilarities is minimized. A matheuristic based on Large Neighborhood Search is proposed to solve this model. The methodology proposed in Chapter 3 is applied to three real-world datasets.

In both Chapters 2 and 3, the impossibility to represent both the frequency distribution and the proximity measure accurately is stated. Thus, in Chapter 4, which is

based on the paper Carrizosa et al. (2015b), we relax the space-filling constraint. This way a visualization framework is obtained which faithfully represents the frequencies by means of the areas of convex bodies, whereas it trades off the correct representation of the dissimilarities as the distances between those objects and their spread in the visualization region. This problem, which extends the standard Multidimensional Scaling, is written as a global optimization problem whose objective is the Difference of two Convex functions (DC). Suitable DC decompositions allow us to use the Difference of Convex Algorithm (DCA) in a very efficient way. Our algorithmic approach is used to visualize two real-world datasets.

As we stated in Section 1.1, visualizing dynamic datasets is crucial for enhancing interpretability and decision making. In Chapter 5, based on the papers Carrizosa et al. (2017a) and Carrizosa et al. (2017b), we extend the methodology developed in Chapter 4 to the dynamic case. Therefore, we develop a new framework to visualize datasets consisting of individuals observed along different time periods. These individuals have attached a time-dependent frequency distribution and a dissimilarity measure, which may vary over time as well. DCA and Nonconvex Quadratic Binary Optimization techniques are combined to solve the proposed Mathematical Optimization model. This way a visualization framework is obtained which, besides the criteria in the static model presented in Chapter 4, it includes also the preservation of the mental map. Our procedure is successfully tested on dynamic geographical and linguistic datasets.

Finally, in Chapter 6 some conclusions and future lines of research are briefly discussed.

Chapter 2

Visualizing frequencies and adjacencies as Rectangular Maps: A Mixed Integer Linear Programming approach

In this chapter we address the problem of visualizing a dataset involving a frequency distribution and a proximity relation, measured by adjacencies. This information is presented using a rectangular map, i.e., a subdivision of a rectangle into rectangular portions so that each portion is associated with one individual in the dataset, their areas reflect the frequencies, and the adjacencies between portions represent the adjacencies between the individuals. We formulate this visualization problem as a Mixed Integer Linear Programming (MILP) model. We propose a matheuristic that has this MILP model at its heart and a Multidimensional Scaling based model as surrogate to determine the relative positions of the rectangular portions. Our experimental results demonstrate that our matheuristic provides rectangular maps with a good fit in both the frequency distribution and the adjacency relation.

2.1 Introduction

A natural and frequent task in Information Visualization is to depict a set of individuals $V = \{v_1, \ldots, v_N\}$, to which there is attached a frequency distribution, $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_N)$, such that $\sum_{i=1}^{N} \omega_i = 1$ and $\omega_i \geq 0$, $i = 1, \ldots, N$ (Spence and Lewandowsky, 1991). Market share, vote intention or population rates, just to name a few, are usual examples. As we noted in Section 1.1, besides histograms or word clouds, a common approach to visualize frequencies is to consider a bounded region of the plane and to subdivide it into portions $\mathbf{P} = (P_1, \ldots, P_N)$ of common shape whose areas represent the frequencies. Well-known visualization tools for this kind of data are the classic pie or fan charts, Figures 1.1 (b) and 1.1 (c) respectively, and rectangular maps (Baudel and Broeksema, 2012; Heilmann et al., 2004), see Figure 2.1. In this kind of representations, holes are not allowed, thus, receiving the name of planar space-filling visualization maps.

A planar space-filling map to visualize the frequencies attached to individuals in a bounded set Ω of the plane can be found by constructing the portions of the desired area and putting them together to fill Ω. This is straightforward in the case of the pie or fan charts: for a permutation $\sigma(1), \sigma(2), \ldots, \sigma(N)$ of the indices $1, 2, \ldots, N$, portions of areas proportional to $\omega_{\sigma(1)}, \omega_{\sigma(2)}, \ldots, \omega_{\sigma(N)}$ are placed sequentially in Ω. The only freedom in such planar space-filling visualization maps is thus the choice of the permutation, which can be made according to different seriation criteria as exposed by Hahsler (2017). For the case of rectangular maps, where Ω is the unit square and portions are rectangles, the same approach, illustrated in Figure 2.1 (left), can be used, where the rectangular portions go all the way from North to South (or, by rotation, from West to East, for instance). While pie and fan charts only admit different sequential arrangements, rectangular maps allow more freedom than the choice of a permutation, as illustrated in Figure 2.1 (right). The flexible layout offered by rectangular maps is

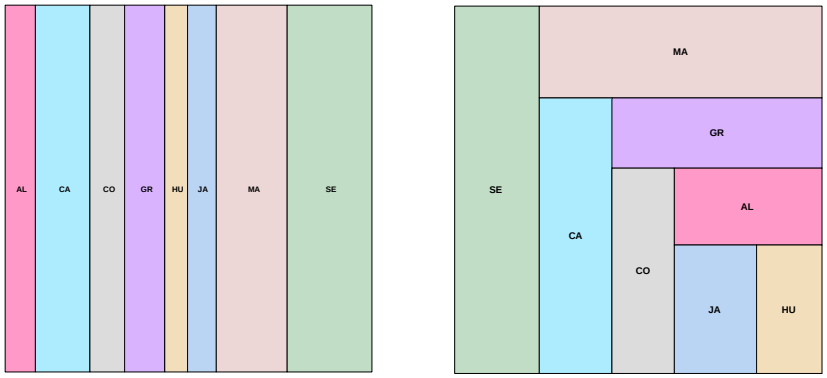


Figure 2.1: Examples of rectangular maps.

also desirable when, in addition to frequencies, we are interested in visualizing proximity, measured by adjacencies, which is the subject of this chapter.

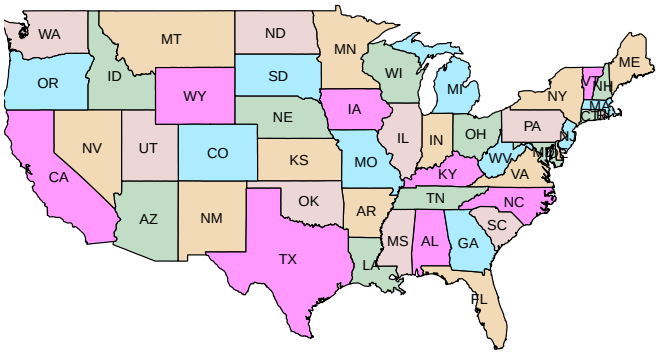
The nature of the adjacency relation can be diverse, a classical example being geographical adjacencies. As discussed in Section 1.1, cartograms have been the usual visualization framework to simultaneously depict geographical adjacencies and frequencies. Specifically, rectangular cartograms have served to analysts as easy-to-interpret graphics thanks to the use of simple shapes, namely rectangles (Buchin et al., 2012; Eppstein et al., 2012; Heilmann et al., 2004; Kreveld and Speckmann, 2007; Raisz, 1934). The approaches developed in the literature to obtain rectangular cartograms take advantage of the geographical relative positions of the individuals (countries, cities, etc.), and thus their approaches cannot be directly extended to more general data structures. Figure 1.4 (c) depicts a rectangular cartogram for the geographical area of the states in the U.S. built using the Recmap package in R (Panse, 2016). There exist approaches which build space-filling rectangular cartograms as well. This is the case of grid maps (Eppstein et al., 2015), which represent as accurately as possible the adjacencies present in a geographical dataset by assigning exactly one cell of the grid to each individual, although frequencies are not taken into account. Figure 2.2 (b) depicts the grid map built for the 48 contiguous states in the U.S. by Eppstein et al. (2015) cf. Figure 6-L_2^2, representing 56 adjacencies of the 105 present in the actual map, see Figure 2.2 (a). With the methodology described in Section 2.3, we are able to represent 63 adjacencies of the 105 present in Figure 2.2 (a), see Figure 2.2 (c). In this chapter, our goal is to propose a mathematical optimization formulation and a suitable solution approach to build rectangular maps to visualize the frequency distribution $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_N)$ and the proximity between the individuals, measured by an adjacency matrix $E = (e_{ij})$. As far as we are aware, this is a novel problem in the literature.

Throughout this chapter, the weighted graph $G = (V, E, \boldsymbol{\omega})$ will model the set V of

individuals, attached with the binary relation (adjacency) E and the frequency distribution $\boldsymbol{\omega}$. Similarly, we denote by $G^{\mathbf{P}} = (V, E^{\mathbf{P}}, \boldsymbol{\omega}^{\mathbf{P}})$, the weighted graph associated with the rectangular map, denoted by $\mathbf{P} = (P_1, \ldots, P_N)$. The binary relation in $G^{\mathbf{P}}$ is defined as follows, $(v_i, v_j) \in E^{\mathbf{P}}$ if portions P_i and P_j are adjacent, i.e., their borders intersect in more than one point, while for the node weights $\boldsymbol{\omega}^{\mathbf{P}}$, $\omega_i^{\mathbf{P}}$ is equal to the area of the rectangle P_i. In general, one cannot guarantee the existence of a rectangular map that satisfies area and adjacency requirements on the rectangles, i.e., $\boldsymbol{\omega}^{\mathbf{P}} = \boldsymbol{\omega}$ and $E^{\mathbf{P}} = E$ (Kreveld and Speckmann, 2007). This is especially the case when the graph G to be represented is not planar. Alam et al. (2013) and Biedl and Genç (2005) deal with complexity results on rectangular maps. Due to this impossibility, we seek to represent as many adjacent individuals as adjacent rectangles as possible, and to have as low as possible both the number of rectangular adjacent portions corresponding to non-adjacent individuals and the total deviation of the areas of the portions from the frequencies. This optimization problem is very hard. The computational burden might be strongly reduced if additional information could be added to reduce the number of possible layouts. This is done in rectangular cartograms by imposing each rectangle to contain a point, which is usually chosen as the centroid of the geographical region (Duarte et al., 2014; Heilmann et al., 2004; Wood and Dykes, 2008). In this chapter, we develop a new tool based on Multidimensional Scaling (MDS) (Kruskal, 1964; Torgerson, 1958) to find such a set of points, having valuable information about the adjacencies and the frequencies, which can be applied to any type of individuals, i.e., not only for geographical data, as long as they have a dissimilarity measure attached to them (Carrizosa et al., 2007).

Although not focused on Information Visualization, the case in which there are no frequencies (weights) attached to the individuals, and the graph G is planar, has been studied in the literature and it has many applications, for instance, in Very Large Scale Integration circuits design (Anjos and Liers, 2012; Tani et al., 1991). The usual approach there is to find a rectangular dual of a planar graph, which consists of a subdivision of the unit square in such a way that each vertex (individual) corresponds to a different rectangle in the subdivision and, if v_i and v_j are linked, then the corresponding portions P_i and P_j are adjacent in the subdivision. Biedl and Genç (2005), de Berg et al. (2009) and Koźmiński and Kinnen (1985) study some characterizations of planar graphs that admit a rectangular dual. Rectangular duals are also related with Facility Layout (Anjos and Vieira, 2016; Jankovits et al., 2011; Sherali et al., 2003), whose aim is to find a layout which minimizes the flow between a set of facilities of given areas, and Graph Drawing, (Dörk et al., 2012; Klimenta and Brandes, 2013; Owen-Smith et al., 2002; Tamassia, 2013). These frameworks use very ad-hoc approaches and either disregard the proper representation of adjacencies, frequencies, or are not space-filling.

In this chapter, the problem of building rectangular maps which simultaneously



(a) The U.S. map

WA	MT	ND	MN	WI	NY	VT	ME
OR	ID	SD	IA	MI	PA	NE	MA
NV	WY	NE	IL	IN	OH	CT	RI
UT	CO	KS	MO	KY	WV	MD	NJ
CA	NM	OK	AR	TN	SC	VA	DE
AZ	TX	LA	MS	AL	GA	FL	NC

(b) Grid map for the U.S. built in Eppstein et al. (2015)

WA	ID	ND	MN	WI	NJ	VT	NH
OR	MT	SD	IA	MI	PA	NY	MA
NV	WY	NE	IL	IN	OH	CT	RI
UT	CO	KS	MO	KY	WV	MD	DE
AZ	NM	OK	AR	TN	VA	NC	SC
CA	TX	LA	MS	AL	GA	FL	ME

(c) Grid map for the U.S. built with the ECPA methodology in Section 2.3

Figure 2.2: Visualizations for the U.S.

optimizes the fit in the adjacencies and areas for weighted graphs G, not necessarily planar, is modeled by means of Mathematical Optimization. We consider the unit square Ω split into K rows and L columns, each cell representing thus a $100/(K \times L)\%$ of the total area of Ω, yielding the so-called (K, L)-*rectangular maps*. This grid structure, also proposed by e.g. Abbiw-Jackson et al. (2006), Eppstein et al. (2015), Fried et al. (2015), Liu et al. (2015) and Strong and Gong (2014), allows us to easily measure areas, and simplifies the notion of adjacency, since two portions are adjacent if they touch in, at least, one cell.

We formulate the problem of building (K, L)-rectangular maps as a Mixed Integer Linear Program (MILP). However, such MILP is a difficult problem and thus there is a need for developing a sophisticated matheuristic solution approach to find good (K, L)-rectangular maps. To do so, first, we introduce the concept of *locating cells*, which reduce

the number of possible layouts by fixing the relative positions between the rectangles, and, as will be seen in our numerical experience, they speed up the computation of the (K, L)-rectangular maps. Second, we design a tailored MDS to choose these locating cells by taking into account the adjacencies and area deviations measures. This MDS can handle *any* set of individuals with frequencies and an adjacency relations attached, and not necessarily of geographic nature, as is the case for rectangular cartograms (Raisz, 1934).

The remainder of the chapter is structured as follows. In Section 2.2 we introduce the optimization model to build (K, L)-rectangular maps and formulate it as an MILP. In Section 2.3 we present an algorithm to compute (K, L)-rectangular maps. Section 2.4 is the experimental section. Section 2.5 concludes the chapter with a summary.

2.2 The Mathematical Optimization model

Given a set of individuals $V = \{v_1, \ldots, v_N\}$, a (K, L)-rectangular map has associated a weighted graph $G^{\mathbf{P}} = (V, E^{\mathbf{P}}, \boldsymbol{\omega}^{\mathbf{P}})$, in which $(v_i, v_j) \in E^{\mathbf{P}}$ if portions P_i and P_j are adjacent, i.e., they touch in at least one cell, and $\boldsymbol{\omega}^{\mathbf{P}}$ denotes the rectangles' areas. An ideal (K, L)-rectangular map representation of a given graph $G = (V, E, \boldsymbol{\omega})$ should satisfy the following conditions:

(C1) The portions in $\mathbf{P} = (P_1, \ldots, P_N)$ form a partition of $\Omega = [0, 1] \times [0, 1]$.

(C2) P_i is a rectangle made up of a collection of cells of the (K, L)-grid in which Ω is divided, $i = 1, \ldots, N$.

(C3) $E^{\mathbf{P}} = E$

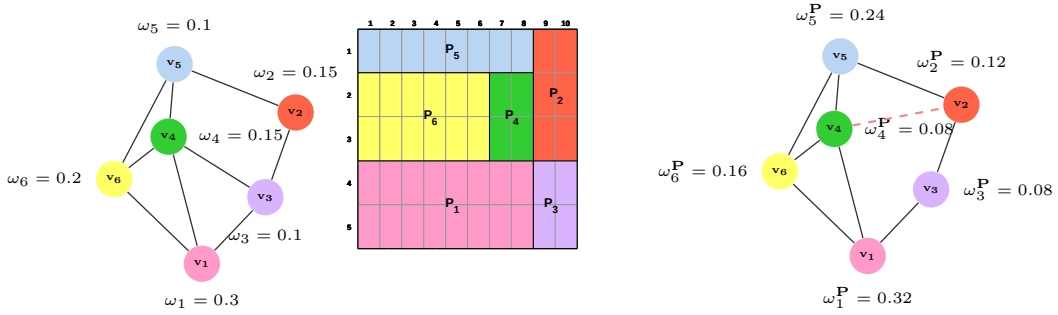
(C4) $\omega_i^{\mathbf{P}} = \omega_i$, namely $\dfrac{1}{K \times L}|P_i| = \omega_i$, where $|P_i|$ denotes the number of cells in P_i, $i = 1, \ldots, N$.

Constructing (K, L)-rectangular maps which satisfy conditions (C1) and (C2) is straightforward. One simply needs to allocate cells belonging to the same portion forming rectangles, as in Figure 2.1 (left). However, including conditions (C3) and (C4) as hard requirements may make the problem infeasible (Kreveld and Speckmann, 2007). Thus, we model conditions (C3) and (C4) as soft constraints, and consider their violation, combined through a scaling vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$, $\lambda_t \geq 0,\ t = 1, 2, 3$, as the objective to be optimized. This yields the $\boldsymbol{\lambda}$-Rectangular Map model $(RM)_{\boldsymbol{\lambda}}$, stated as

$$
\begin{array}{ll}
\max & \lambda_1 |E \cap E^{\mathbf{P}}| - \lambda_2 |\overline{E} \cap E^{\mathbf{P}}| - \lambda_3 \sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| \\
\text{s.t.} & \mathbf{P} = (P_1, \ldots, P_N) \text{ satisfying (C1), (C2).}
\end{array}
\qquad (RM)_{\boldsymbol{\lambda}}
$$

On one hand, the resemblance between E and $E^{\mathbf{P}}$, i.e. (C3), is modeled by means of the cardinality of the sets $E \cap E^{\mathbf{P}}$ and $\overline{E} \cap E^{\mathbf{P}}$ weighed through parameters λ_1 and λ_2, respectively, where $\overline{E}$ denotes the complement of E. This way, the number of adjacencies in E that are also in the (K, L)-rectangular map and those that are not in E but do appear in the map are counted. On the other hand, the condition (C4) is stated as the sum of the deviations from the frequencies in $\boldsymbol{\omega}$ to the area of the rectangles in $\boldsymbol{\omega}^{\mathbf{P}}$ weighed by parameter λ_3. Thus, different values of $\boldsymbol{\lambda}$ yield different (K, L)-rectangular maps, highlighting the different aspects involved.

Figure 2.3 illustrates the concept of (K, L)-rectangular map, using as G the weighted graph plotted in Figure 2.3 (a), where $N = 6$, $|E| = 9$ and $\boldsymbol{\omega} = (0.3, 0.15, 0.1, 0.15, 0.1, 0.2)$. Figure 2.3 (b) represents G as a $(5, 10)$-rectangular map, where the $K = 5$ rows are numbered from top to bottom and the $L = 10$ columns from left to right. We may observe that 8 out of the 9 true adjacencies, i.e., the adjacencies in E, are reproduced by $E^{\mathbf{P}}$, which are shown as solid edges in the graph in Figure 2.3 (c). There is only one true adjacency missing in $E^{\mathbf{P}}$: v_3 and v_4 are adjacent in G but their associated rectangles P_3 and P_4 are not in the $(5, 10)$-rectangular map. (Note that if two cells touch only in a corner, they are not considered adjacent.) The $(5, 10)$-rectangular map adds a false adjacency, i.e., an adjacency which was not in E, which is drawn as a dashed edge in Figure 2.3 (c): v_2 and v_4 are not adjacent in G but P_2 and P_4 are in the $(5, 10)$-rectangular map. Finally, and with respect to the weights, the $(5, 10)$-rectangular map approximates them. For instance, v_4 has a weight equal to $\omega_4 = 0.15$, while the area of P_4 is equal to $4/50 = 0.08$.



(a) $G = (V, E, \boldsymbol{\omega})$ (b) $(5, 10)$-rectangular map of G (c) $G^{\mathbf{P}} = (V, E^{\mathbf{P}}, \boldsymbol{\omega}^{\mathbf{P}})$: $E \cap E^{\mathbf{P}}$ solid line, $\overline{E} \cap E^{\mathbf{P}}$ dashed line.

Figure 2.3: A $(5, 10)$-rectangular map for G; $|E \cap E^{\mathbf{P}}| = 8$, $|\overline{E} \cap E^{\mathbf{P}}| = 1$, $\sum_{i=1}^{6} |\omega_i^{\mathbf{P}} - \omega_i| = 0.24$.

In what follows, Problem $(RM)_{\boldsymbol{\lambda}}$ is stated as an MILP. Note that indices i and j are used for portions, r for rows of the grid and s for columns.

2.2.1 Decision variables

Let x_{irs} be binary variables which control whether the cell (r, s) belongs to the portion P_i or not, defined as

$$x_{irs} = \begin{cases} 1 & \text{if cell } (r,s) \text{ belongs to portion } P_i \\ 0 & \text{otherwise.} \end{cases}$$

Thanks to these variables, a portion P_i can be expressed as $P_i(x) = \{(r, s) : x_{irs} = 1,\ r = 1, \ldots, K,\ s = 1, \ldots, L\}$.

In order to model adjacencies between portions $P_i(x)$ and $P_j(x)$, binary variables z_{ij} are defined as

$$z_{ij} = \begin{cases} 1 & \text{if portion } P_i(x) \text{ is adjacent to portion } P_j(x) \\ 0 & \text{otherwise.} \end{cases}$$

Observe that x and z-variables are closely related: if $P_i(x)$ and $P_j(x)$ are two adjacent portions, then $z_{ij} = 1$ and $x_{irs} = x_{jr's'} = 1$, where (r', s') is either equal to $(r-1, s)$ or $(r+1, s)$ or $(r, s+1)$ or $(r, s-1)$.

The variables u^l_{ijrs} indicate whether portions i and j are adjacent at cell (r, s) from above, below, to the right or to the left, respectively. Thus,

$$u^1_{ijrs} = \begin{cases} 1 & \text{if portion } P_j(x) \text{ is adjacent to portion } P_i(x) \text{ at cell } (r,s) \text{ from above} \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we can define u^2_{ijrs}, u^3_{ijrs}, and u^4_{ijrs}, which indicate if portions $P_i(x)$ and $P_j(x)$ are adjacent from below, to the left or to the right, respectively. Observe that also x and u-variables are closely related, since $u^1_{ijrs} = x_{irs} \cdot x_{jr-1s}$, $u^2_{ijrs} = x_{irs} \cdot x_{ir+1s}$, $u^3_{ijrs} = x_{irs} \cdot x_{jrs+1}$ and $u^4_{ijrs} = x_{irs} \cdot x_{jrs-1}$.

Finally, φ_i and ψ_i are positive real variables to linearize the area deviation $|\omega^{\mathbf{P}}_i - \omega_i|$, i.e., $|\omega^{\mathbf{P}}_i - \omega_i| = \varphi_i + \psi_i$ and $\omega^{\mathbf{P}}_i - \omega_i = \varphi_i - \psi_i$.

We illustrate these variables using the $(5, 10)$-rectangular map in Figure 2.3 (b). For instance, rectangle P_4 has four cells defined by $x_{427} = x_{428} = x_{437} = x_{438} = 1$. Moreover, P_4 has four adjacent rectangles: P_1, P_2, P_5 and P_6. Thus, $z_{41} = z_{42} = z_{45} = z_{46} = 1$ and $u^1_{4527} = u^1_{4528} = u^2_{4627} = u^2_{4637} = u^3_{4238} = u^3_{4228} = u^4_{4137} = u^4_{4138} = 1$. The remaining binary variables of the form x_{4rs}, z_{4j} and u^l_{4jrs} are zero. Finally, $\varphi_4 = 0$ and $\psi_4 = 0.07$.

2.2.2 Objective function

Because of the definition of the variables, it is straightforward to see that the objective function in Problem $(RM)_{\boldsymbol{\lambda}}$ (written in maximization form) is,

$$\lambda_1 \sum_{\substack{i,j=1\ldots N \\ (i,j)\in E}} z_{ij} - \lambda_2 \sum_{\substack{i,j=1\ldots N \\ (i,j)\in \overline{E}}} z_{ij} - \lambda_3 \sum_{i=1,\ldots,N} (\varphi_i + \psi_i), \tag{2.1}$$

for fixed scaling nonzero vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$, $\lambda_t \geq 0,\ t = 1, 2, 3$.

2.2.3 Constraints

We now write the constraints in Problem $(RM)_{\boldsymbol{\lambda}}$ using the decision variables above, and give a brief explanation of each group of constraints.

$$\sum_{i=1,\ldots,N} x_{irs} = 1, \qquad r = 1,\ldots,K,\ s = 1,\ldots,L, \tag{2.2}$$

$$\sum_{\substack{r=1,\ldots,K \\ s=1,\ldots,L}} x_{irs} \geq 1, \qquad i = 1,\ldots,N, \tag{2.3}$$

$$\sum_{\substack{\min\{r,r'\}\leq r''\leq \max\{r,r'\} \\ \min\{s,s'\}\leq s''\leq \max\{s,s'\}}} x_{i\,r''\,s''} \geq (|r-r'|+1)\cdot(|s-s'|+1)\cdot(x_{irs}+x_{ir's'}-1), \qquad \begin{array}{l} i = 1,\ldots,N, \\ r, r' = 1,\ldots,K, \\ s, s' = 1,\ldots,L, \end{array} \tag{2.4}$$

$$\sum_{\substack{r=2,\ldots,K \\ s=1,\ldots,L}} u^1_{ijrs} + \sum_{\substack{r=1,\ldots,K-1 \\ s=1,\ldots,L}} u^2_{ijrs} + \sum_{\substack{r=1,\ldots,K \\ s=1,\ldots,L-1}} u^3_{ijrs} + \sum_{\substack{r=1,\ldots,K \\ s=2,\ldots,L}} u^4_{ijrs} \geq z_{ij}, \qquad i,j = 1,\ldots,N,\ i \neq j, \tag{2.5}$$

$$x_{irs} + x_{j\,r-1\,s} \leq z_{ij} + 1, \qquad i,j = 1,\ldots,N,\ i \neq j, r = 2,\ldots,K,\ s = 1,\ldots,L, \tag{2.6}$$

$$x_{irs} + x_{j\,r+1\,s} \leq z_{ij} + 1, \qquad i,j = 1,\ldots,N,\ i \neq j, r = 1,\ldots,K-1,\ s = 1,\ldots,L, \tag{2.7}$$

$$x_{irs} + x_{j\,r\,s+1} \leq z_{ij} + 1, \qquad i,j = 1,\ldots,N,\ i \neq j, r = 1,\ldots,K,\ s = 1,\ldots,L-1, \tag{2.8}$$

$$x_{irs} + x_{j\,r\,s-1} \leq z_{ij} + 1, \qquad i,j = 1,\ldots,N,\ i \neq j, r = 1,\ldots,K,\ s = 2,\ldots,L, \tag{2.9}$$

$$u^1_{ijrs} \leq x_{irs}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 2,\ldots,K,\ s = 1,\ldots,L, \tag{2.10}$$

$$u^1_{ijrs} \leq x_{j\,r-1\,s}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 2,\ldots,K,\ s = 1,\ldots,L, \tag{2.11}$$

$$x_{irs} + x_{j\,r-1\,s} \leq u^1_{ijrs} + 1, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 2,\ldots,K,\ s = 1,\ldots,L, \tag{2.12}$$

$$u^2_{ijrs} \leq x_{irs}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K-1,\ s = 1,\ldots,L, \tag{2.13}$$

$$u^2_{ijrs} \leq x_{j\,r+1\,s}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K-1,\ s = 1,\ldots,L, \tag{2.14}$$

$$x_{irs} + x_{j\,r+1\,s} \leq u^2_{ijrs} + 1, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K-1,\ s = 1,\ldots,L, \tag{2.15}$$

$$u^3_{ijrs} \leq x_{irs}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K,\ s = 1,\ldots,L-1, \tag{2.16}$$

$$u^3_{ijrs} \leq x_{j\,r\,s+1}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K,\ s = 1,\ldots,L-1, \tag{2.17}$$

$$x_{irs} + x_{j\,r\,s+1} \leq u^3_{ijrs} + 1, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K,\ s = 1,\ldots,L-1, \tag{2.18}$$

$$u^4_{ijrs} \leq x_{irs}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K,\ s = 2,\ldots,L, \tag{2.19}$$

$$u^4_{ijrs} \leq x_{j\,r\,s-1}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K,\ s = 2,\ldots,L, \tag{2.20}$$

$$x_{irs} + x_{j\,r\,s-1} \leq u^4_{ijrs} + 1, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K,\ s = 2,\ldots,L, \tag{2.21}$$

$$\frac{1}{KL} \sum_{\substack{r=1,\ldots,K \\ s=1,\ldots,L}} x_{irs} - \omega_i = \varphi_i - \psi_i, \qquad i = 1,\ldots,N, \tag{2.22}$$

$$x_{irs},\ z_{ij},\ u^l_{ijrs} \in \{0,1\}, \qquad i,j = 1,\ldots,N,\ i \neq j,\ r = 1,\ldots,K,\ s = 1,\ldots,L,\ l = 1,\ldots,4, \tag{2.23}$$

$$\varphi_i, \psi_i \geq 0, \qquad i = 1,\ldots,N. \tag{2.24}$$

Firstly, note that condition (C1) is satisfied thanks to the definition of the x-variables and constraint (2.2), which forces that every cell must belong to exactly one portion, and thus, the resulting map is space-filling. Since all the individuals must appear in the (K, L)-rectangular map, constraint (2.3) ensures that at least one cell is allocated for every individual. The rectangular-shaped requirement in (C2) is stated by constraint (2.4), which forces that for every pair the cells (r, s) and (r', s') belonging to the same portion, $P_i(x)$, all the $(|r - r'| + 1) \cdot (|s - s'| + 1)$ cells in-between them must belong also to $P_i(x)$. Constraint (2.5) models the correctness of $z_{ij} = 1$, i.e., if variable z_{ij} takes the value 1, then, there must be two adjacent cells belonging to portions $P_i(x)$ and $P_j(x)$ respectively. Note that two rectangles can be only adjacent on one side, namely, from above, below, to the left or to the right. Each of those relative positions are modeled through each summation on the left hand side in constraint (2.5). On the other hand, constraints (2.6)–(2.9) model the correctness of $z_{ij} = 0$, this means that if two portions are not adjacent neither from above, below, left or right, there must not exist contiguous cells belonging to those portions. Constraints (2.10)–(2.21) model the fact that variables u are the product of two x variables, as noted in Section 2.2.1 (McCormick, 1976). Constraint (2.22) ensures the correctness of the absolute value in the area deviation in the objective function. Finally, the variables' type is modeled with constraints (2.23) and (2.24).

2.2.4 Writing the problem as an MILP

Thus, given a weighted graph $G = (V, E, \boldsymbol{\omega})$, Problem $(RM)_{\boldsymbol{\lambda}}$ can be formulated as the following MILP

$$\begin{array}{ll} \max & (2.1) \\ \text{s.t.} & (2.2)\text{–}(2.24). \end{array} \qquad (RML)_{\boldsymbol{\lambda}}$$

In a first attempt, we solved $(RML)_{\boldsymbol{\lambda}}$ using a commercial MILP solver. However, even very small instances of $(RML)_{\boldsymbol{\lambda}}$ turned out to be too hard for this solver. In the following section we propose a matheuristic for our visualization problem, which achieves a good fit in the adjacencies and the areas for the three datasets used in our experimental section. The matheuristic has $(RML)_{\boldsymbol{\lambda}}$ at its heart, since this MILP formulation, with a few decision variables fixed to a given value, is solved in each iteration.

2.3 Algorithmic approach

The formulation $(RML)_{\boldsymbol{\lambda}}$ has a hard combinatorial structure which mainly comes from the lack of information about how the N portions could fit together into Ω to form

a (K, L)-rectangular map. If valuable knowledge about the relative positions among the portions were provided, the number of possible layouts would be dramatically reduced and Problem $(RML)_{\boldsymbol{\lambda}}$ would become computationally tractable. Similar ideas can be found in Facility Layout, where customized procedures are designed to determine a reliable relative positioning among the facilities (Anjos and Vieira, 2016), and Cartography, where it is customary to impose that each portion must contain a point, which is usually the centroid of the geographical region (Duarte et al., 2014; Heilmann et al., 2004; Wood and Dykes, 2008).

In a similar fashion, our solution approach to tackle $(RML)_{\boldsymbol{\lambda}}$ is based on finding a set of points, called hereafter *locating points*, which has valuable information about the frequencies and the adjacency relation between individuals. Due to the grid structure of our visualization model, we determine a set of *locating cells* instead. Thus, let us assume that we have an external procedure that generates the locating points, $\mathbf{q} = \{q_1, \ldots, q_N\}$ such that $q_i \in P_i$, $i = 1, \ldots, N$. We define the set of locating cells $\mathcal{C}$ as,

$$\begin{aligned} \mathcal{C} \;=\; \{(i,r,s) : \exists q_i \in \mathbf{q} \text{ which lies inside the cell } (r,s), \\ 1 \le i \le N,\, 1 \le r \le K,\, 1 \le s \le L\}\,. \end{aligned}$$

Thus, solving Problem $(RML)_{\boldsymbol{\lambda}}$ with locating cells becomes

$$\begin{array}{lll} \max & (2.1) & \\ \text{s.t.} & (2.2)\text{–}(2.24) & (RML)_{\boldsymbol{\lambda},\mathcal{C}} \\ & x_{irs} = 1 \qquad (i,r,s) \in \mathcal{C}. & \end{array}$$

The constraints related to the locating cells are heuristic, i.e., for arbitrary locating cells we cannot guarantee that the optimal solution obtained for $(RML)_{\boldsymbol{\lambda},\mathcal{C}}$ is also optimal to $(RML)_{\boldsymbol{\lambda}}$. In order to obtain a good solution to $(RML)_{\boldsymbol{\lambda}}$, we construct an initial set of locating cells and perturbe them via an iterative procedure to further improve the solution. The initial set of locating cells is built by a new approach based on MDS (Kruskal, 1964; Torgerson, 1958), the MultiDimensional Scaling for (K, L)-rectangular maps, which can be applied as long as a dissimilarity measure is given (Carrizosa et al., 2007).

2.3.1 Multidimensional Scaling for (K, L)-rectangular maps

In order to find a set of points $\mathbf{q}$ that yields good (K, L)-rectangular maps, we propose a new approach based on solving a nonsmooth continuous optimization problem. This strategy arises from adapting the MDS framework to the special features of our problem by providing the points information about adjacencies and individuals' frequencies. Thus, our tailored MDS takes into account that the locating points are to be

used by $(RML)_{\boldsymbol{\lambda},\mathcal{C}}$, i.e., they have to lie in the unit square Ω and be part of the non-overlapping rectangular portions P_i whose areas are close to ω_i and which are related through an adjacency relation. Abbiw-Jackson et al. (2006), Klimenta and Brandes (2013) and Liu et al. (2013) also use MDS approaches for planar visualization maps.

Let $D = (d_{ij})$ be the shortest path distance matrix between all nodes of graph $G = (V, E, \boldsymbol{\omega})$. We want to find N points which lie in Ω, $q_i = (q_i^1, q_i^2)$, contained in N rectangles defined by their NW and SE corners, (a_i^{NW}, b_i^{NW}) and (a_i^{SE}, b_i^{SE}) respectively. These rectangles, called in what follows MDS rectangles, are surrogate of the rectangular portions P_i in $(RML)_{\boldsymbol{\lambda}}$, with some important differences. First, we do not impose that they are made of cells of the region Ω, avoiding the difficulties of the combinatorial part of Problem $(RML)_{\boldsymbol{\lambda}}$. Second, the MDS rectangles do not necessarily cover Ω. Third, they may overlap. A related approach is developed by Anjos and Vieira (2016) in Facility Layout context to determine the relative positions between the departments.

The locating points $\mathbf{q}$ are expected to be somehow central points of portions $\mathbf{P}$ in $(RML)_{\boldsymbol{\lambda},\mathcal{C}}$, and thus the distance $\|q_i - q_j\|_1$ between locating points q_i and q_j should follow the same pattern than the distance d_{ij}. Hence, we impose that

$$\|q_i - q_j\|_1 \approx \kappa d_{ij}, \tag{2.25}$$

for some κ, to be optimized. Observe that the distances between the locating points are measured according to the ℓ_1 norm. Although the usual choice of distance in MDS is the ℓ_2 norm, considering the ℓ_1 norm has the advantage that our MDS model deals with rectangles with sides parallel to the coordinate axes. Hubert et al. (1992), Leung and Lau (2004), Žilinskas and Žilinskas (2009) and Žilinskas (2012) also develop MDS applications using the ℓ_1 norm.

We require two conditions to the MDS rectangles. We want the area of MDS rectangle P_i to approximate ω_i, i.e.,

$$(a_i^{SE} - a_i^{NW})(b_i^{NW} - b_i^{SE}) \approx \omega_i. \tag{2.26}$$

Moreover, we want the MDS rectangles not to overlap, but this is imposed as a soft constraint, forcing the area of each intersection being close to zero:

$$\begin{aligned} \max\left\{0, \min\{a_i^{SE}, a_j^{SE}\} - \max\{a_i^{NW}, a_j^{NW}\}\right\} \cdot \\ \max\left\{0, \min\{b_i^{NW}, b_j^{NW}\} - \max\{b_i^{SE}, b_j^{SE}\}\right\} \quad \approx 0. \end{aligned} \tag{2.27}$$

With this notation, the MDS for (K, L)-rectangular maps $(MDSRM)$ is stated as the problem of finding rectangles, identified by their corner coordinates (a_i^{NW}, b_i^{NW}) and (a_i^{SE}, b_i^{SE}), and points q_i within minimal violation of soft constraints (2.25)–(2.27). This

is expressed as the following nonlinear nonsmooth continuous optimization problem:

$$
\begin{aligned}
\min \quad & \gamma_1 \sum_{i,j=1}^{N} \left(d_{ij} - \kappa \|q_i - q_j\|_1\right)^2 \\
& + \gamma_2 \sum_{i=1}^{N} \left(\left(a_i^{SE} - a_i^{NW}\right)\left(b_i^{NW} - b_i^{SE}\right) - \omega_i\right)^2 \\
& + \gamma_3 \sum_{i,j=1}^{N} \Big(\max\left\{0, \min\{a_i^{SE}, a_j^{SE}\} - \max\{a_i^{NW}, a_j^{NW}\}\right\} \cdot \\
& \qquad \max\left\{0, \min\{b_i^{NW}, b_j^{NW}\} - \max\{b_i^{SE}, b_j^{SE}\}\right\}\Big) \qquad (MDSRM) \\
\text{s.t.} \quad & \\
& 0 \le a_i^{NW} \le q_i^1 \le a_i^{SE} \le 1, \qquad i = 1, \ldots, N \\
& 0 \le b_i^{SE} \le q_i^2 \le b_i^{NW} \le 1, \qquad i = 1, \ldots, N \\
& \kappa > 0, \\
& a_i^{SE}, b_i^{SE}, a_i^{NW}, b_i^{NW}, q_i^1, q_i^2 \in \mathbb{R}, \qquad i = 1, \ldots, N,
\end{aligned}
$$

where κ is a scaling variable and γ_1, γ_2, $\gamma_3 \geq 0$ are scaling constants. Note that we can use a hyperbolic smoothing to approximate the absolute value and max and min functions

$$|y| \approx \sqrt{y^2 + \varepsilon},$$

$$\max\{y, y'\} = \frac{y + y' + |y' - y|}{2} \approx \frac{y + y' + \sqrt{(y' - y)^2 + \varepsilon}}{2},$$

$$\min\{y, y'\} = \frac{y + y' - |y' - y|}{2} \approx \frac{y + y' - \sqrt{(y' - y)^2 + \varepsilon}}{2},$$

where $\varepsilon > 0$.

Observe than in case there exist $i \neq j$, where q_i and q_j belong to the same cell in the (K, L)-grid, then $(RM)_{\boldsymbol{\lambda},\mathcal{C}}$ is infeasible. If this happens, several strategies are possible to recover a feasible problem. For instance, one could randomly perturb the locating points q_i and q_j until they lie in different cells. Alternatively, one can replace the constraint in $(RM)_{\boldsymbol{\lambda},\mathcal{C}}$ related with locating points by a weaker constraint of the form

$$q_i \in P_i(x), \ \forall i \in R, \tag{2.28}$$

where the set $R \subset \{1, \ldots, N\}$ is such that the different locating points belong to different cells.

2.3.2 Cell Perturbing Algorithm

In order to find a good solution to Problem $(RML)_{\boldsymbol{\lambda}}$, we propose an iterative algorithm that solves $(RML)_{\boldsymbol{\lambda},\mathcal{C}}$ for different set of locating cells $\mathcal{C}$. Let $\overline{RML_{\boldsymbol{\lambda},\mathcal{C}}}$ be the optimal solution to Problem $(RML)_{\boldsymbol{\lambda},\mathcal{C}}$, $v\left(\overline{RML_{\boldsymbol{\lambda},\mathcal{C}}}\right)$ its objective value, and $\mathcal{C}^0$ the incumbent set of locating cells.

We start with $\mathcal{C}^0 = \mathcal{C}^{MDS}$, the set of locating cells obtained by solving Problem $(MDSRM)$. At each iteration of the procedure, the incumbent set is perturbed, yielding $\mathcal{C}^*$, and $(RML)_{\boldsymbol{\lambda},\mathcal{C}^*}$ is solved. If the objective value improves, i.e., $v(\overline{RML_{\boldsymbol{\lambda},\mathcal{C}^*}}) > v(\overline{RML_{\boldsymbol{\lambda},\mathcal{C}^0}})$, we update $\mathcal{C}^0$. We refer to this procedure as the Cell Perturbing Algorithm (CPA), whose pseudocode is provided in Algorithm 2.1.

Algorithm 2.1 Cell Perturbing Algorithm (CPA)

Input: The set of locating cells derived from locating points obtained with $(MDSRM)$, $\mathcal{C}^{MDS}$, and a perturbing procedure, $perturb(\cdot)$.

```
1: C^0 ← C^MDS;
2: Solve RML_{λ,C^0};
3: repeat
4:     C* ← perturb(C^0);
5:     Solve RML_{λ,C*};
6:     if v(RML_{λ,C*}) > v(RML_{λ,C^0}) then
7:         RML_{λ,C^0} ← RML_{λ,C*};
8:         C^0 ← C*;
9:     end if
10: until stop condition is met
```

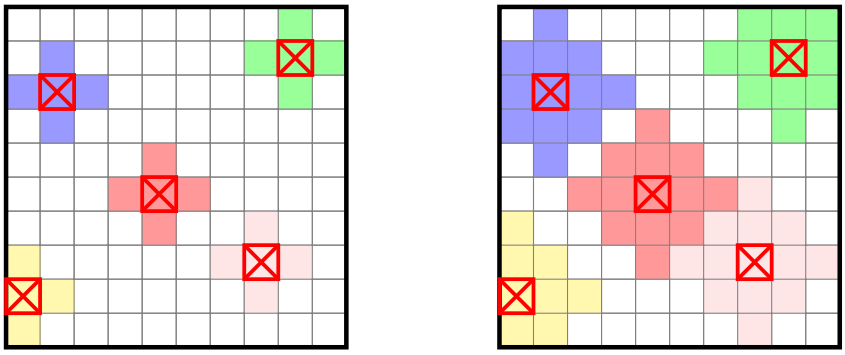
Output: $\mathcal{C}^0$, $\overline{RML_{\boldsymbol{\lambda},\mathcal{C}^0}}$

The $perturb(\cdot)$ procedure in CPA admits different designs and ours uses a neighborhood structure in the (K, L)-grid around the current set of locating cells. We define the ρ-neighborhood of a cell (i, j) as the set of cells formed by those which are at distance lower or equal than ρ, namely

$$\mathcal{N}_\rho\left((r,s)\right) = \left\{(r',s') : |r-r'| + |s-s'| \leq \rho\right\}.$$

Figure 2.4 illustrates the $\mathcal{N}_1$ and $\mathcal{N}_2$ neighborhoods (shaded cells) of the set of locating cells $\mathcal{C} = \{(3,2),(2,9),(6,5),(9,2),(8,9)\}$ (marked with "×") in Figures 2.4 (a) and 2.4 (b), respectively, on a $(10, 10)$-grid. Observe that the ℓ_1 norm is considered to measure the distance between a pair of cells.

The $perturb(\cdot)$ procedure we have used in our experimental results consists of, given a set of locating cells $\mathcal{C}$, N new locating cells are selected randomly, with uniform probabilities, each one belonging to its corresponding ρ-neighborhood. It is worth noting that only movements which are consistent with constraint (2.2) are allowed, namely there cannot be a locating cell belonging to two rectangles simultaneously. Other more sophis-



(a) $\mathcal{N}_1$ (b) $\mathcal{N}_2$

Figure 2.4: $\mathcal{N}_1$ and $\mathcal{N}_2$ neighborhoods of locating cells $\mathcal{C} = \{(3,2),(2,9),(6,5),(9,2),(8,9)\}$.

ticated designs of the *perturb*$(\cdot)$ procedure are possible, such as assigning nonuniform probabilities the cells in the neighborhood, but our experimental results are satisfactory with the choice above.

Having a good initial set of cells, as the one given by our tailored MDS, is essential to ensure a good solution to $(RML)_{\boldsymbol{\lambda}}$ in a few iterations of the CPA. Note that if the optimal solution to Problem $(RML)_{\boldsymbol{\lambda}}$ were known and the set of locating cells $\mathcal{C}$ is chosen by taking N cells of such solution, one per rectangle, then the optimal solution of Problem $(RML)_{\boldsymbol{\lambda},\mathcal{C}}$ would have the same objective value than the optimal solution of $(RML)_{\boldsymbol{\lambda}}$, although the layout might change. Thus, CPA would achieve the global optimum if the whole space of possible locating cells were explored. Nevertheless, the size of such space explodes with the dimension of the grid.

2.3.3 Embedded Cell Perturbing Algorithm

Solving the MILPs involved in the CPA, namely $(RML)_{\boldsymbol{\lambda},\mathcal{C}}$, for a tight grid might be too time-consuming, and thus performing many iterations of the CPA becomes a long task. In order to speed up the algorithm for tight grids, we design the Embedded Cell Perturbing Algorithm (ECPA), which successively inserts coarser grids into tighter ones performing some iterations of CPA in-between. The ECPA pseudocode is outlined in Algorithm 2.2.

The *subdivide*$(\cdot)$ procedure arises from the requirement of transforming the set of locating cells from a coarser grid to a tighter one when the grids are embedded. Our choice is making such transformation in the simplest way, namely we randomly sample, with equal probabilities, in the space of cells resulting from dividing the locating cells on the coarser grid to become cells on the tight one. Since we consider embeddings in which each cell is subdivided into four new cells (each row and each column is split into two to form the tighter grid), one of those four cells is selected randomly to become

Algorithm 2.2 Embedded Cell Perturbing Algorithm (ECPA)

Input: The number of levels in the hierarchy T. A set of embedded grids $\{(K_t, L_t)\}_{t=1,\dots,T}$. The set of locating cells arising from locating points obtained with $(MDSRM)$ on the (K_1, L_1)-grid, $\mathcal{C}^{MDS}_{(K_1,L_1)}$. A perturb and subdivide procedures, $perturb(\cdot)$ and $subdivide(\cdot)$.

1: $\left(\mathcal{C}^*_{(K_1,L_1)}, \overline{RML_{\boldsymbol{\lambda},\mathcal{C}^*_{(K_1,L_1)}}}\right) \leftarrow CPA\left(\mathcal{C}^{MDS}_{(K_1,L_1)}, perturb(\cdot)\right)$
2: **for** $t \leftarrow 2$ to T **do**
3: $\quad \mathcal{C}^*_{(K_t,L_t)} \leftarrow subdivide(\mathcal{C}^*_{(K_{t-1},L_{t-1})})$;
4: $\quad \left(\mathcal{C}^*_{(K_t,L_t)}, \overline{RML_{\boldsymbol{\lambda},\mathcal{C}^*_{(K_t,L_t)}}}\right) \leftarrow CPA\left(\mathcal{C}^*_{(K_{t-1},L_{t-1}1)}, perturb(\cdot)\right)$;
5: **end for**

Output: $\mathcal{C}^*_{(K_T,L_T)}$, $\overline{RML_{\boldsymbol{\lambda},\mathcal{C}^*_{(K_T,L_T)}}}$

locating cell in the tighter one. Other splitting procedures might be considered as well as nonuniform probabilities on the choice of the locating cells in the tighter grid.

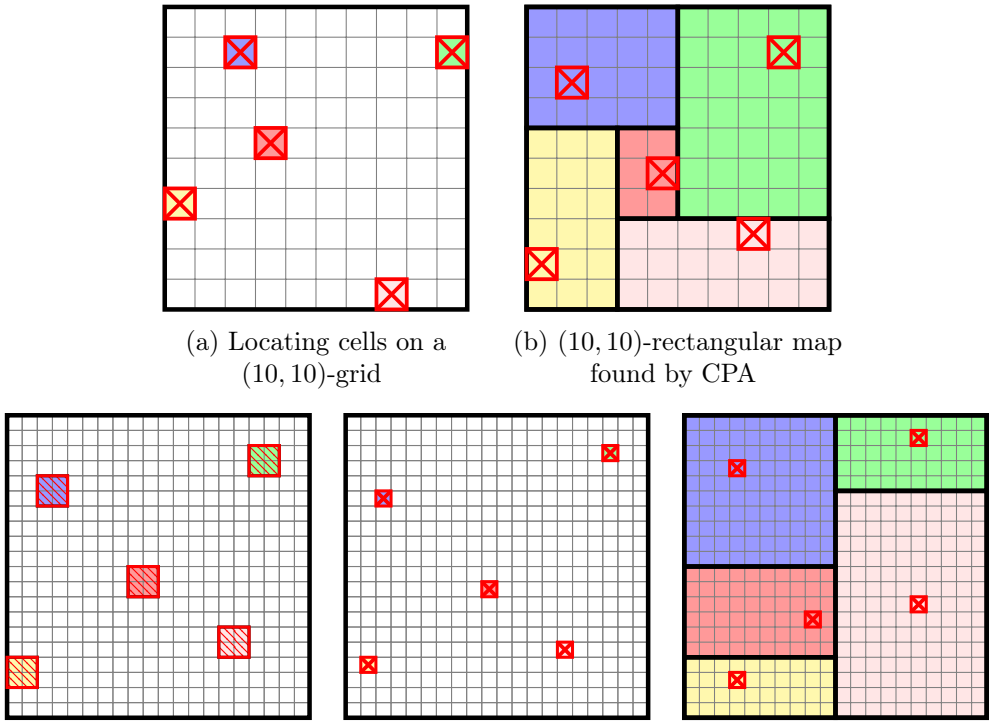
Figure 2.5 illustrates the ECPA algorithm with a $(10, 10)$ and $(20, 20)$-grids and 5 individuals. In Figure 2.5 (a), the set of 5 locating cells, found via the MDS procedure, are depicted as "×" on a $(10, 10)$-grid. A $(10, 10)$-rectangular map obtained by performing some iterations of CPA is shown in Figure 2.5 (b). Observe how the locating cells have changed via the $perturb(\cdot)$ procedure in CPA in Figures 2.5 (a) and 2.5 (b). In Figure 2.5 (c), the candidates to become locating cells on a $(20, 20)$-grid are dashed, whereas Figure 2.5 (d) contains the resulting locating cells from the subdividing procedure. Finally, Figure 2.5 (e) includes a $(20, 20)$-rectangular map obtained by some iterations of CPA, where the set of locating cells on the $(20, 20)$-grid are highlighted with a "×".

2.4 Computational experience

In this section we illustrate the ECPA approach to generate (K, L)-rectangular maps using three examples of diverse nature. The first one consists of visualizing the proportion of people in each blood group in the U.S. and the compatibility between the groups. The other two examples are cartographic applications. A (K, L)-rectangular map is presented for each dataset with $K = L = 20$. In Section 2.4.1 we describe the three datasets used in the experiments and in Section 2.4.2 how the ECPA has been implemented. We then discuss the fit of the $(20, 20)$-rectangular maps generated by ECPA in terms of the adjacency relation and the areas.

2.4.1 Datasets

The first example, `Blood`, consists of the weighted graph which models the proportion of people in the U.S. in each blood group (Stanford Blood Center, 2014), taking



(a) Locating cells on a $(10, 10)$-grid

(b) $(10, 10)$-rectangular map found by CPA

(c) Candidates locating cells for the $(20, 20)$-rectangular map

(d) Locating cells for the $(20, 20)$-rectangular map

(e) $(20, 20)$-rectangular map found by CPA

Figure 2.5: Illustration of ECPA.

into account the blood compatibility between donor and recipient. In the `Blood` graph, the nodes, and thus the individuals, are the blood groups, and two groups v_i and v_j are adjacent if either v_i can donate blood to v_j, or viceversa. In the second example, `Netherlands`, the individuals are the provinces of The Netherlands, and the data represented is their (normalized) population (Statistics Netherlands, 2013). The proximity measure considered is the geographical location, namely, two nodes are adjacent if the corresponding provinces are adjacent in the geographical map. The third example, `Germany`, is analogous to `Netherlands` but with a larger amount of individuals and adjacencies and frequencies of a different nature. The individuals are the 16 German federal states, and the frequencies to be represented are the (normalized) geographical area (Destatis, Statistisches Bundesamt, 2015). Figure 2.6 shows the `Blood`, `Netherlands` and `Germany` graphs.

2.4.2 Experiments details

A $(20, 20)$-grid is considered to build the rectangular maps, each cell thus representing a 0.25% of the area of the visualization region. In order to obtain $(20, 20)$-rectangular maps, we optimize the fit in adjacencies and areas. These are modeled by means of the

number of adjacencies reproduced in the $(20, 20)$-rectangular map ($|E \cap E^{\mathbf{P}}|$), the number of false adjacencies added in the $(20, 20)$-rectangular map ($|\overline{E} \cap E^{\mathbf{P}}|$), and the area deviation measure ($\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i|$), as stated in conditions (C3) and (C4) in Section 2.2. Finally, we consider $\boldsymbol{\lambda} \in \left\{ \left(\frac{1}{|E|}, \frac{1}{|\overline{E}|}, 1 \right), (1,0,0), (0,1,0), (0,0,1) \right\}$.

The locating points are obtained by solving $(MDSRM)$ in Section 2.3.1 with $\gamma_1 = \gamma_3 = 1$ and $\gamma_2 = 1000$. Since it is a multimodal problem, a multistart with 50 runs is executed. These continuous nonlinear problems have been solved with the IPOPT solver (Wächter and Biegler, 2006).

The ECPA has been coded in AMPL (Fourer et al., 1993) and all the MILPs involved have been solved with CPLEX v12.6 (CPLEX, IBM ILOG, 2014) on a PC Intel® Core™ i7-2600K, 16GB of RAM. The time has been limited to ten minutes for the two smallest datasets, `Blood` and `Netherlands`, and to fifteen minutes for the largest one, `Germany`. The algorithm has been performed with a hierarchy of $T = 2$ levels, where a $(10, 10)$-grid is used for $t = 1$ and the $(20, 20)$-grid for $t = 2$. We have set the radius of perturbation $\rho = 1$. We have set a maximum number of iterations of CPA on the $(10, 10)$-grid for the three datasets equal to 50, and equal to 10 for the $(20, 20)$-grid in the `Blood` example. For the two largest datasets, `Netherlands` and `Germany`, no cell perturbation was performed on the $(20, 20)$-grid. Please note that, for all datasets, the optimal $(10, 10)$-rectangular map was obtained in each step of the algorithm in a few seconds, and thus within the time limit.

2.4.3 Results

The performance of ECPA can be found in Table 2.1 for $\boldsymbol{\lambda} = \left(\frac{1}{|E|}, \frac{1}{|\overline{E}|}, 1 \right)$ as well as for its extreme values, namely $\boldsymbol{\lambda} \in \{(1,0,0), (0,1,0), (0,1,0)\}$.

For the `Blood` graph, ECPA obtained a $(20, 20)$-rectangular map in which 17 out of 19 adjacencies are reproduced, no false adjacencies are added and with an area deviation of 0.072 when $\boldsymbol{\lambda} = \left(\frac{1}{|E|}, \frac{1}{|\overline{E}|}, 1 \right)$. The directions of the edges between the blood groups have been depicted with arrows on the $(20, 20)$-rectangular map. We note here that our model does not take into account the nature of the graph (directed or undirected). Observe that the relations between the different groups are well represented through the adjacency relation in the $(20, 20)$-rectangular map, at the same time that the percentage of people belonging to each group is very accurately depicted. Varying the values of $\boldsymbol{\lambda}$ we have been able to obtain $(20, 20)$-rectangular maps which either reproduce up to 17 adjacencies, which do not introduce any false adjacency or with a total area deviation of 0.027, for $\boldsymbol{\lambda}$ equal to $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ respectively.

For the `Netherlands` graph, we obtained a $(20, 20)$-rectangular map in which 22 out of 22 adjacencies are reproduced, 3 false adjacencies are added and with an area

Table 2.1: ECPA matheuristic approach.

	$\boldsymbol{\lambda}$	$\lvert E \cap E^{\mathbf{P}}\rvert$	$\lvert \overline{E} \cap E^{\mathbf{P}}\rvert$	$\sum_{i=1}^{N} \lvert\omega_i^{\mathbf{P}} - \omega_i\rvert$	Figure
`Blood`	$\left(\frac{1}{\lvert E\rvert}, \frac{1}{\lvert\overline{E}\rvert}, 1\right)$	17	0	0.072	2.7
	$(1,0,0)$	17	0	0.148	2.8
	$(0,1,0)$	15	0	0.052	2.9
	$(0,1,0)$	13	1	0.027	2.10
`Netherlands`	$\left(\frac{1}{\lvert E\rvert}, \frac{1}{\lvert\overline{E}\rvert}, 1\right)$	22	3	0.122	2.11
	$(1,0,0)$	22	0	0.228	2.12
	$(0,1,0)$	22	0	0.228	2.13
	$(0,1,0)$	16	7	0.070	2.14
`Germany`	$\left(\frac{1}{\lvert E\rvert}, \frac{1}{\lvert\overline{E}\rvert}, 1\right)$	28	7	0.290	2.15
	$(1,0,0)$	28	2	0.738	2.16
	$(0,1,0)$	27	2	0.632	2.17
	$(0,1,0)$	19	14	0.119	2.18

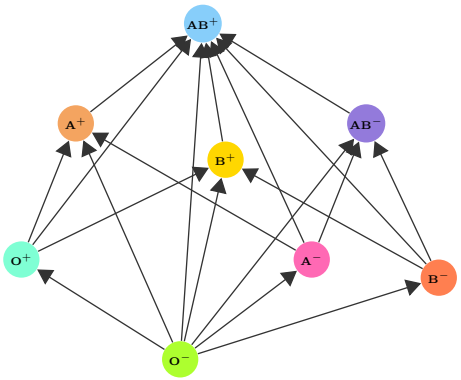
deviation of 0.122 when $\boldsymbol{\lambda} = \left(\frac{1}{|E|}, \frac{1}{|\overline{E}|}, 1\right)$. Varying the value of $\boldsymbol{\lambda}$ to its extreme values we have been able to reproduce all the adjacencies involved in the graph, i.e., 22 adjacencies without introducing any false adjacency. The lowest area deviation we have found is equal to 0.070.

For the `Germany` graph, we obtained a $(20, 20)$-rectangular map in which 28 out of 28 adjacencies are reproduced, 7 false adjacencies are added and with an area deviation of 0.290 when $\boldsymbol{\lambda} = \left(\frac{1}{|E|}, \frac{1}{|\overline{E}|}, 1\right)$. For extreme values of $\boldsymbol{\lambda}$, the maximum number of true adjacencies we are able to reproduce is 28 out of 28, while the minimum number of false adjacencies added is 2, and the minimum total area deviation is 0.119. Augmenting the number of individuals to represent yields worse error incurred in the representation of the areas when the size of the grid is maintained.

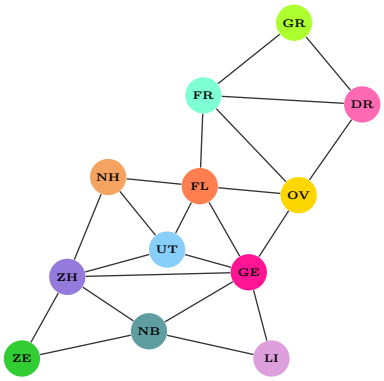
In view of the results obtained for the `Blood`, `Netherlands`, and `Germany`, we conclude that our model and solution approach are able to obtain good-quality (K, L)-rectangular maps, in the sense that a good fit in the adjacencies and areas as stated in (C3) and (C4) are obtained. In two out of three cases, `Netherlands` and `Germany`, we are able to reproduce 100% of adjacencies, whereas a very small number of false adjacencies is introduced. Indeed, in `Blood` and `Netherlands` the minimum area error obtained is in an order of magnitude of 10^{-2}.

The output of our experimental results for ECPA is presented in Figures 2.7–2.18. Figure 2.6 (a) depicts the `Blood` graph G, Figures 2.7 (a)– 2.10 (a) the $(20, 20)$-rectangular maps obtained as detailed in Section 2.4.2 with the locating cells marked

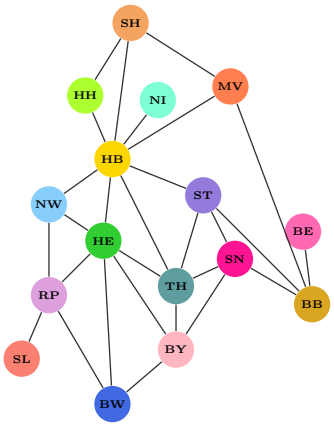
with a "$\times$", and Figures Figures 2.7 (b)– 2.10 (b) the graphs associated to the $(20, 20)$-rectangular maps, $G^{\mathbf{P}}$, in which the edges which are reproduced in the $(20, 20)$-rectangular map $(E \cap E^{\mathbf{P}})$ are depicted as a full line and those adjacent rectangles which are not edges in G $(\overline{E} \cap E^{\mathbf{P}})$ are depicted as dashed lines. The same representation is used for `Netherlands` and `Germany` datasets, which can be found in Figures 2.11–2.14 and Figures 2.15–2.18, respectively.



(a) `Blood` graph, G

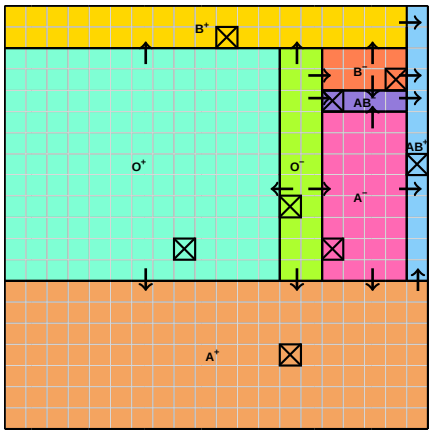


(b) `Netherlands` graph, G

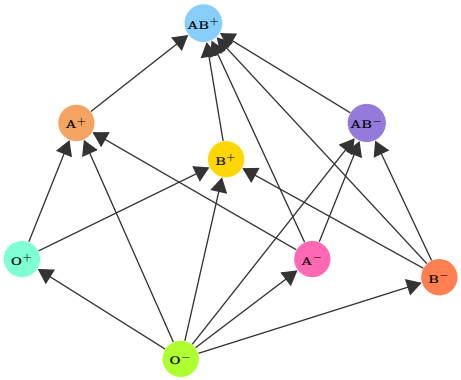


(c) `Germany` graph, G

Figure 2.6: Graphs of `Blood`, `Netherlands` and `Germany` datasets.

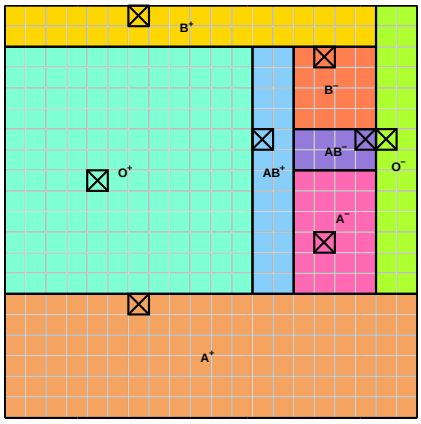


(a) $(20, 20)$-rectangular map

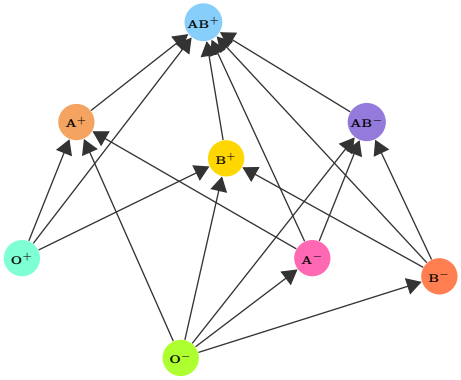


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.7: `Blood` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 17$, $|\overline{E} \cap E^{\mathbf{P}}| = 0$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.072$.

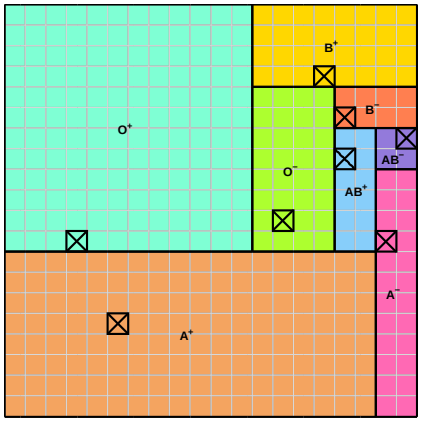


(a) $(20, 20)$-rectangular map

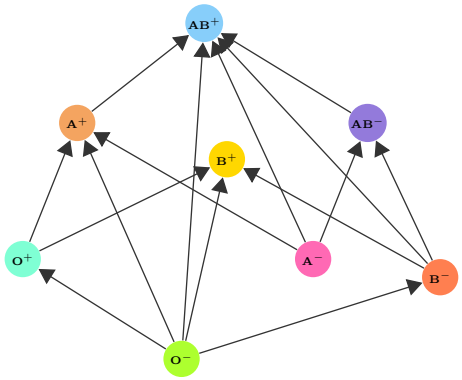


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.8: `Blood` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 17$, $|\overline{E} \cap E^{\mathbf{P}}| = 0$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.148$.

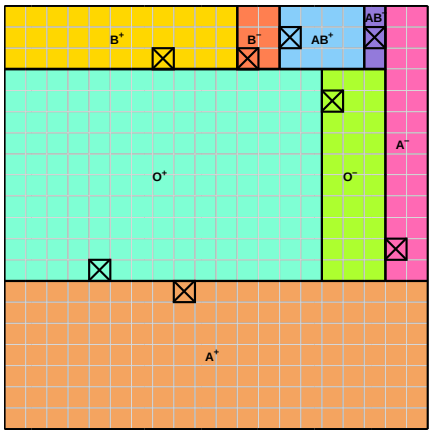


(a) $(20, 20)$-rectangular map

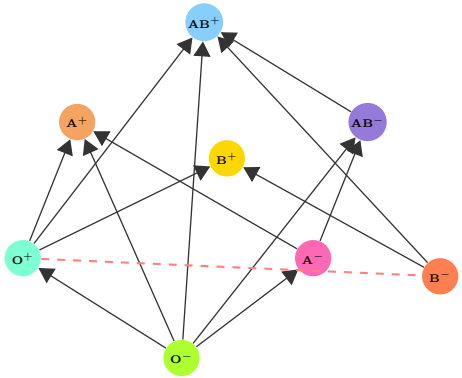


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.9: `Blood` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 15$, $|\overline{E} \cap E^{\mathbf{P}}| = 0$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.052$.

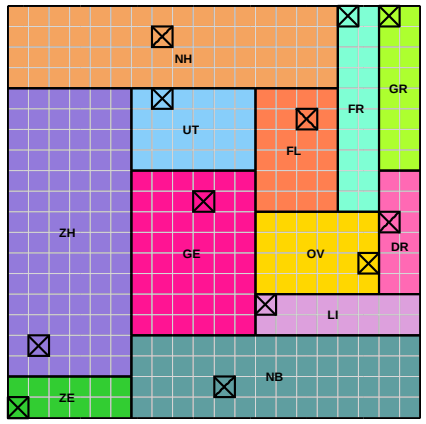


(a) $(20, 20)$-rectangular map

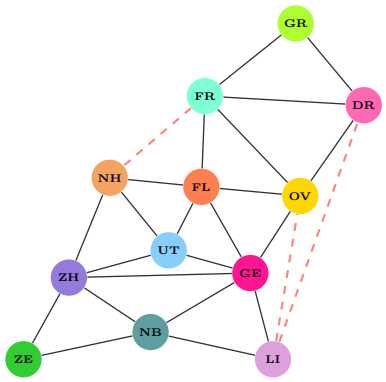


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.10: `Blood` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 13$, $|\overline{E} \cap E^{\mathbf{P}}| = 1$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.027$.

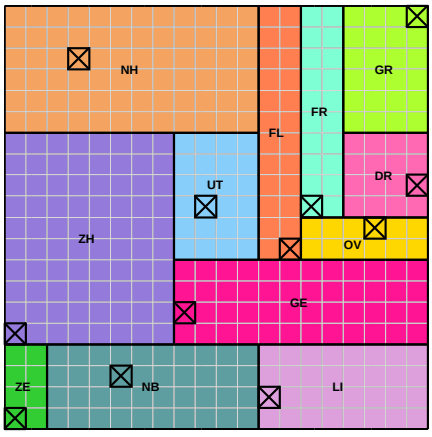


(a) $(20, 20)$-rectangular map

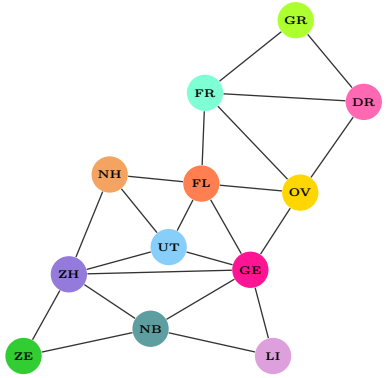


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.11: `Netherlands` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 22$, $|\overline{E} \cap E^{\mathbf{P}}| = 3$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.122$.

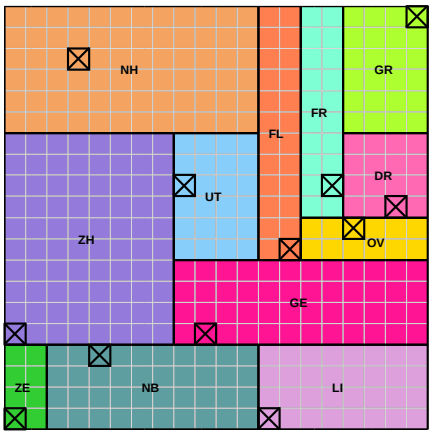


(a) $(20, 20)$-rectangular map

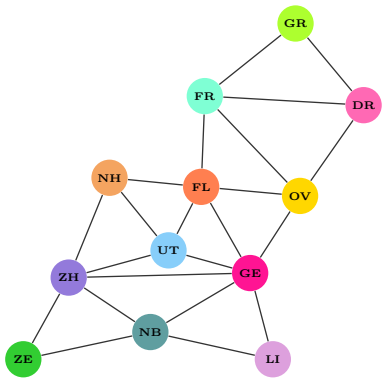


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.12: `Netherlands` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 22$, $|\overline{E} \cap E^{\mathbf{P}}| = 0$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.228$.

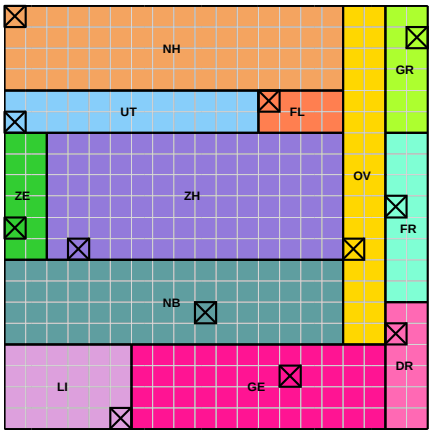


(a) $(20, 20)$-rectangular map

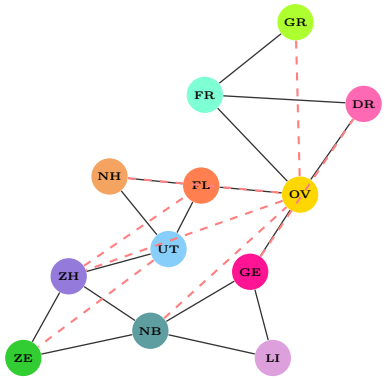


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.13: `Netherlands` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 22$, $|\overline{E} \cap E^{\mathbf{P}}| = 0$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.228$.

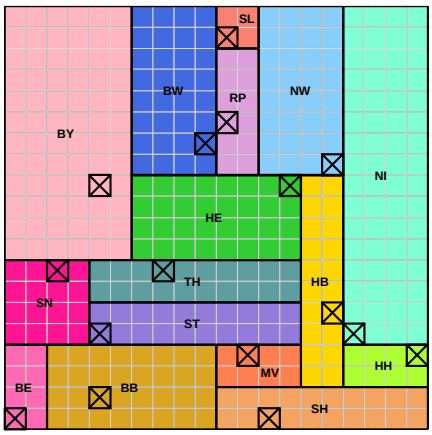


(a) $(20, 20)$-rectangular map

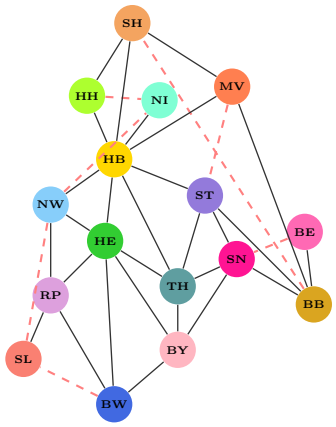


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.14: `Netherlands` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 16$, $|\overline{E} \cap E^{\mathbf{P}}| = 7$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.070$.

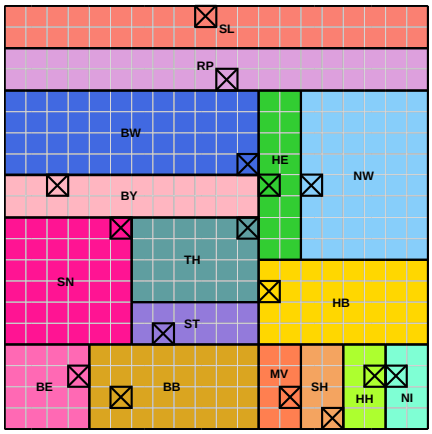


(a) $(20, 20)$-rectangular map

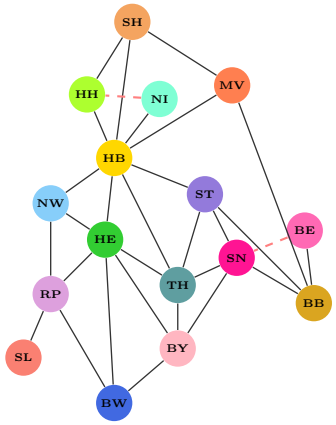


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.15: `Germany` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 28$, $|\overline{E} \cap E^{\mathbf{P}}| = 7$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.290$.

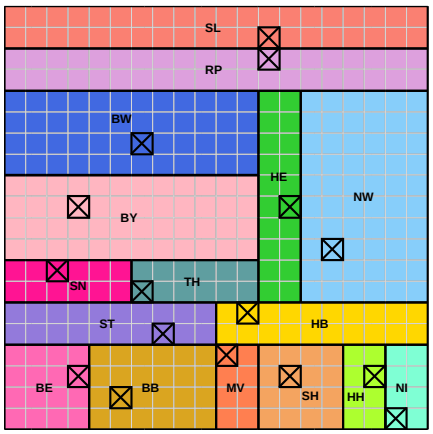


(a) $(20, 20)$-rectangular map

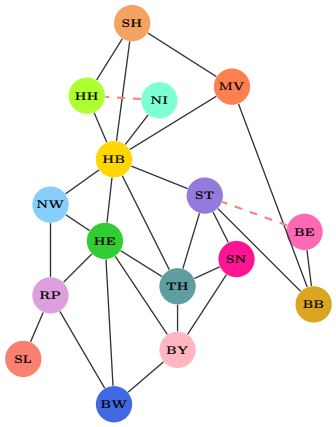


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.16: `Germany` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 28$, $|\overline{E} \cap E^{\mathbf{P}}| = 2$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.738$.

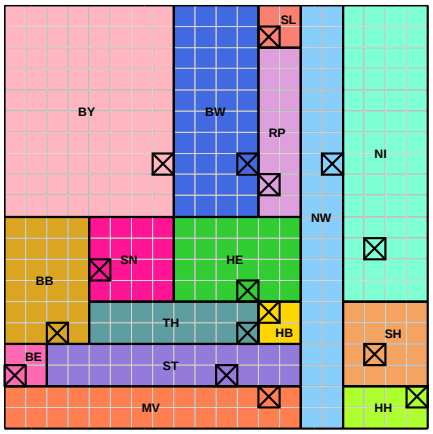


(a) $(20, 20)$-rectangular map

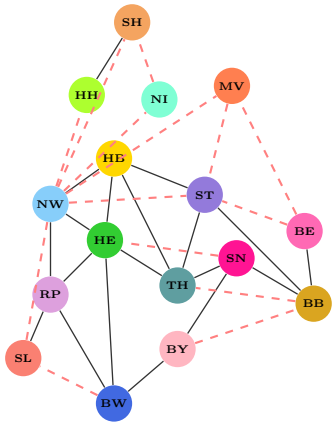


(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.17: `Germany` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 27$, $|\overline{E} \cap E^{\mathbf{P}}| = 2$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.632$.



(a) $(20, 20)$-rectangular map



(b) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

Figure 2.18: `Germany` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 19$, $|\overline{E} \cap E^{\mathbf{P}}| = 14$, $\sum_{i=1}^{N} |\omega_i^{\mathbf{P}} - \omega_i| = 0.119$.

2.5 Conclusions

In this chapter we have developed a new Mathematical Optimization approach to address the problem of visualizing by means of rectangular maps a frequency distribution and an adjacency relation attached to a set of individuals. This kind of data can be

modeled as a weighted graph and thus, our aim is to obtain rectangular maps in which the adjacencies in the graph are correctly reproduced, whereas as few false adjacencies as possible are introduced and the error incurred by approximating the frequencies by the rectangles' areas is as small as possible. The problem is formulated as an MILP. Due to its hard combinatorial structure, a tailored Multidimensional Scaling (MDS) has been designed to determine the relative positions of the rectangles in the map, and thus to reduce the number of possible layouts. This MDS acts as a surrogate of the problem, whose partial solution (locating cells) becomes a starting point for an iterative algorithm to improve the set of locations cells. Our approach has been illustrated using three examples, showing that our results are competitive, most of the true adjacencies (the ones in the original weighted graph) can be reproduced by the rectangular map, introducing only a few false ones, and with low area deviations.

Chapter 3

Visualizing frequencies and dissimilarities as Space-filling Box-connected Maps: A Mixed Integer Nonlinear Programming approach

In this chapter we go a step further in the visualization of a frequency distribution and a proximity relation. Assuming that proximities between individuals in the dataset are measured as dissimilarities, the (K, L)-rectangular maps in Chapter 2 are generalized by considering more flexible portions than rectangles, the so-called *box-connected rectangles*, and alternative proximity measures to adjacencies, namely the Single, Complete and Average Linkage. Similarly, the box-connected rectangles must form a partition of the unit square, yielding a Space-filling Box-connected Map (SBM). The construction of an SBM is formally stated as a Mathematical Optimization problem, which is solved heuristically by using Large Neighborhood Search. Our experimental results demonstrate that our procedure provides SBMs with a good fit in both the frequency distribution and the proximity relation.

3.1 Introduction

Information Visualization usually has to deal with the proper representation of a set of individuals, $V = \{v_1, \ldots, v_N\}$, to which there is attached a frequency distribution $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_N)$, such that $\sum_{i=1}^{N} \omega_i = 1$ and $\omega_i \geq 0$, $i = 1, \ldots, N$ (Spence and Lewandowsky, 1991). Besides this piece of information, datasets usually enclose knowledge about proximity relations between the individuals. Assuming that proximities between individuals in V are measured as dissimilarities, $\boldsymbol{\delta} = (\delta_{ij})_{i,j=1,\ldots,N}$, this chapter is devoted to develop a general optimization based-framework to visualize both $\boldsymbol{\omega}$ and $\boldsymbol{\delta}$. Following Chapter 2 philosophy, we seek a visualization framework consisting on a partition of the unit square in the plane $\Omega = [0, 1] \times [0, 1]$ subdivided into portions $\boldsymbol{P} = (P_1, \ldots, P_N)$ whose areas represent the frequencies and the proximities between the portions, to be properly defined, depict the dissimilarities, yielding a planar space-filling visualization map.

As discussed in Chapter 1, combining simultaneously the visualization of a frequency distribution and a dissimilarity relation is still a challenging problem to which very specific (ad-hoc) approaches exist. For instance, tools to build cartograms (Dorling, 1996; de Berg et al., 2010; Heilmann et al., 2004; Kreveld and Speckmann, 2007; Tobler, 2004) and proportional symbol maps (Cano et al., 2013; Kunigami et al., 2014) take advantage of the geographical information to properly depict dissimilarities, yielding very specific approaches that usually cannot be extended to more general examples such as the ones handled in Section 3.4. Other attempts in the Information Visualization field either disregard dissimilarities or frequencies, or they are not space-filling visualization frameworks (Carrizosa et al., 2015b; Dörk et al., 2012; Fried et al., 2015; Gómez-Nieto et al., 2014; Liu et al., 2015; Strong and Gong, 2014).

In Chapter 2, space-filling rectangular maps are proposed, which take into consideration adjacencies as measure of proximity and frequencies. However, the geometrical

shape used to represent each individual (a rectangle) seems to be too rigid to give a good fit in both types of information. In Kreveld and Speckmann (2007), there is an example illustrating that it may be impossible to represent accurately both dissimilarities and frequencies in space-filling rectangular maps. Therefore, in order to overcome this drawback, the space-filling visualization framework developed in this chapter considers portions in $\boldsymbol{P}$ more flexible than rectangles, namely connected union of rectangles which verify the so-called *box-connectivity* property (to be formally stated later). We refer to such representation hereafter as a Space-filling Box-connected Map (SBM).

The remainder of this chapter is structured as follows. In Section 3.2, we first establish the conditions that an SBM must satisfy. After defining the concept of box-connectivity, the problem of constructing an SBM is formally stated as a biobjective Mixed Integer Nonlinear Problem (MINLP) and this is then reformulated into two single-objective Mixed Integer Linear Programs (MILP). Finally, an equivalent formulation to these MILPs is provided, which, although less natural, is shown to be tighter in general. Due to the high computational effort needed to solve such optimization problems, a matheuristic is to be used. In Section 3.3, Large Neighborhood Search (LNS) (Pisinger and Ropke, 2010; Shaw, 1998) is adapted to our problem. Section 3.4 contains the numerical experiments of our methodology on three datasets of different nature. Finally, Section 3.5 closes the chapter with some conclusions.

3.2 The Mathematical Optimization model

Given a set of individuals $V = \{v_1, \ldots, v_N\}$ with a distribution of frequencies $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_N)$ and a dissimilarity measure $\boldsymbol{\delta} = (\delta_{ij})_{i,j=1,\ldots,N}$ attached, our aim is to build a Space-filling Box-connected Map (SBM) which depicts both pieces of information. In what follows we state the conditions that an SBM must satisfy.

In order to construct an SBM, we consider the unit square Ω partitioned into K rows and L columns, called in what follows (K, L)-grid. Pairs of cells forming a (K, L)-grid are considered adjacent if they share one full side, which implies that a cell can have at most four adjacent cells. Grid structures are commonly used in Reserve Network design to arrange a connected union of sites to protect ecosystems and species (Jafari and Hearne, 2013; Önal and Briers, 2006; Önal et al., 2016). Since the grid layout provides a well-structured and compact representation, it has been also used in Information Visualization frameworks (Abbiw-Jackson et al., 2006; Eppstein et al., 2015; Fried et al., 2015; Liu et al., 2015; Strong and Gong, 2014), and in Facility Layout as a tool to easily measure the area of the facilities (Kochhar et al., 1998).

Definition 3.1. *Given two cells (r, s) and (r', s') in a (K, L)-grid, we define the box*

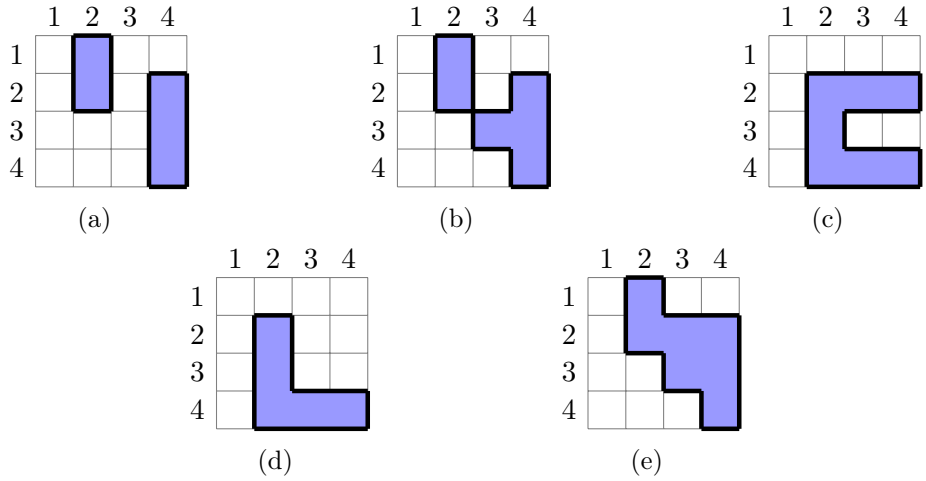


Figure 3.1: Shaded cells in (a), (b) and (c) are not box-connected regions, while shaded cells in (d) and (e) are box-connected regions.

generated by (r, s) *and* (r', s') *as*

$$B\left((r,s),(r',s')\right) = \{(r'',s'') \in (K,L)\text{-}grid: \quad \min\{r,r'\} \leq r'' \leq \max\{r,r'\}, \\ \min\{s,s'\} \leq s'' \leq \max\{s,s'\}\}.$$

Definition 3.2. *Let* S *be a subset of cells of a* (K, L)*-grid and let* $|S|$ *denote its cardinality.* S *is said to be box-connected if one of these conditions holds:*

1. $|S| = 1$.
2. $|S| = 2$ *and its two cells are adjacent.*
3. $|S| \geq 3$ *and for all non-adjacent* $(r, s), (r', s') \in S$*, there exists* $(r'', s'') \in B\left((r, s), (r', s')\right) \cap S$ *such that* $(r'', s'') \neq (r, s)$ *and* $(r'', s'') \neq (r', s')$.

Each individual v_i must be depicted in the SBM as a unique box-connected portion P_i made of cells in the (K, L)-grid. Shaded cells in Figures 3.1 (a) - 3.1 (c) represent portions that are not allowed in our representation, since they do not verify the box-connectivity stated in Definition 3.2. Regarding Figure 3.1 (a), there are no shaded cells in the intersection of the set containing the shaded cells and the box $B((2, 2), (2, 4))$, except for $(2, 2)$ and $(2, 4)$. Analogously, in Figure 3.1 (b) considering $B((2, 2), (3, 3))$. Thus, we observe that disconnected portions are obviously not box-connected. Finally, Figure 3.1 (c) violates box-connectivity when considering, for instance, the box $B((2, 4), (4, 4))$. Possible shapes allowed to represent the individuals in V are depicted in Figures 3.1 (d) and 3.1 (e).

The construction of an SBM regards the partition of Ω into N box-connected portions, which are made of cells in the (K, L)-grid in which Ω is subdivided, and in such a

way that the areas of the portions depict the frequencies, $\boldsymbol{\omega}$, and the distances between the portions are proportional to the dissimilarities, $\boldsymbol{\delta}$. These statements can be summed up in the following conditions:

(C1) The portions in $\boldsymbol{P} = (P_1, \ldots, P_N)$ form a partition of $\Omega = [0,1] \times [0,1]$,

(C2) P_i is a box-connected region, made up of a collection of cells of the (K, L)-grid in which Ω is divided, $i = 1, \ldots, N$,

(C3) distance$(P_i, P_j) \propto \delta_{ij}$, $i, j = 1, \ldots, N$, $i \neq j$,

(C4) area$(P_i) = \omega_i$, namely $\dfrac{1}{K \times L}|P_i| = \omega_i$, $i = 1, \ldots, N$, where $|P_i|$ denotes the number of cells in P_i, $i = 1, \ldots, N$.

Since the portions in $\boldsymbol{P}$ are made of connected unions of cells and they form a partition of Ω, the grid structure helps to easily measure the proximity and the area of the portions. In order to measure the distance between portions P_i and P_j in (C3), those usually used in Cluster Analysis like the Single, Complete and Average Linkage (Hansen and Jaumard, 1997) seem suitable for an SBM.

An SBM which satisfies conditions (C1) and (C2) can be obtained straightforwardly by allocating cells belonging to the same portion sequentially until filling Ω. However, including conditions (C3) and (C4) as hard requirements might make the problem unfeasible (Kreveld and Speckmann, 2007). Thus, our model consists of building SBMs in which the errors made by approximating the scaled dissimilarities, through a real positive variable κ, by the distances between the portions, and the frequencies by their areas, both measured in absolute value, are minimized. Indeed, to compare those distances with dissimilarities we follow the Multdidimensional Scaling approach (Kruskal, 1964; Torgerson, 1958). Our model considers then the violation of conditions (C3) and (C4) as objectives to be minimized, yielding a Biobjective Space-filling Box-Connected Map (BSBM) model, which reads as follows

$$
\begin{array}{ll}
\min & \displaystyle\sum_{\substack{i,j=1,\ldots,N \\ i \neq j}} |distance(P_i, P_j) - \kappa\delta_{ij}| \\
\min & \displaystyle\sum_{i=1,\ldots,N} |area(P_i) - \omega_i| \\
\text{s.t.} & \mathbf{P} = (P_1, \ldots, P_N) \text{ satisfying (C1) and (C2)} \\
& \kappa \geq 0.
\end{array}
\qquad (BSBM)
$$

Problem $(BSBM)$ can be reformulated as a parametric problem, parametrized by a real positive number α, in which the error between the distances in the SBM and the scaled dissimilarities is minimized among those maps whose area error is less or equal than α, yielding the single objective problem $(\alpha - SBM)$. $(BSBM)$ can be

also parametrized by a real positive number β, in which the error between the areas depicted in the SBM is minimized among those maps whose error between the scaled dissimilarities and the distances in the map is less or equal than β, yielding the single objective problem $(\beta - SBM)$:

$$
\begin{array}{ll}
\min & \displaystyle\sum_{\substack{i,j=1,\dots,N \\ i\neq j}} |distance(P_i,P_j) - \kappa\delta_{ij}| \\
\text{s.t.} & \mathbf{P} = (P_1,\dots,P_N) \text{ satisfying (C1) and (C2)} \\
& \displaystyle\sum_{i=1,\dots,N} |area(P_i) - \omega_i| \leq \alpha \\
& \kappa \geq 0.
\end{array}
$$

$(\alpha - SBM)$

$$
\begin{array}{ll}
\min & \displaystyle\sum_{i=1,\dots,N} |area(P_i) - \omega_i| \\
\text{s.t.} & \mathbf{P} = (P_1,\dots,P_N) \text{ satisfying (C1) and (C2)} \\
& \displaystyle\sum_{\substack{i,j=1,\dots,N \\ i\neq j}} |distance(P_i,P_j) - \kappa\delta_{ij}| \leq \beta \\
& \kappa \geq 0.
\end{array}
$$

$(\beta - SBM)$

In what follows, we formally state Problem $(\alpha - SBM)$ as a Mixed Integer Nonlinear Program (MINLP), and a similar approach can be used for $(\beta - SBM)$.

3.2.1 Decision variables

Let x_{irs}, $i = 1,\dots,N$, $r = 1,\dots,K$ and $s = 1,\dots,L$, be binary variables which determine if cell (r,s) belongs to portion P_i or not, namely

$$x_{irs} = \begin{cases} 1 & \text{if cell } (r,s) \text{ belongs to portion } P_i \\ 0 & \text{otherwise.} \end{cases}$$

Thanks to these variables, we express portion P_i as $P_i(x) = \{(r,s) : x_{irs} = 1, r = 1,\dots K, s = 1,\dots,L\}$.

Finally, let κ be a positive real variable which scales the dissimilarities $\boldsymbol{\delta}$.

3.2.2 Objective function

The expression of the objective function in $(\alpha - SBM)$ depends on the choice of the distance function. With the choices mentioned above related to Cluster Analysis, namely the Single Linkage (SL), the Complete Linkage (CL) and the Average Linkage (AvL), these distances can be easily expressed through the binary variables of our mathematical optimization problem, namely x_{irs}. Firstly, we need to define how to measure the distance between two single cells. Since the grid structure naturally calls for the use of the ℓ_1-norm, given two cells (r,s) and (r',s'), the distance between them is equal to $|r-r'| + |s-s'|$, where $r,r' = 1,\dots,K$ and $s,s' = 1,\dots,L$. Thus, the distance between two portions $P_i(x)$ and $P_j(x)$ on an SBM, $distance(P_i(x),P_j(x))$, can be expressed respectively as $SL(P_i(x),P_j(x))$, or $CL(P_i(x),P_j(x))$, or $AvL(P_i(x),P_j(x))$,

defined as

$$SL(P_i(x), P_j(x)) = \min_{\substack{r,r'=1,\dots,K \\ s,s'=1,\dots,L}} \left\{ |r-r'| + |s-s'| : \, x_{irs} = x_{jr's'} = 1 \right\}, \tag{3.1}$$

$$CL(P_i(x), P_j(x)) = \max_{\substack{r,r'=1,\dots,K \\ s,s'=1,\dots,L}} \left\{ |r-r'| + |s-s'| : \, x_{irs} = x_{jr's'} = 1 \right\}, \tag{3.2}$$

$$AvL(P_i(x), P_j(x)) = \frac{1}{|P_i(x)||P_j(x)|} \sum_{\substack{r,r'=1,\dots,K \\ s,s'=1,\dots,L}} (|r-r'| + |s-s'|) \cdot x_{irs} \cdot x_{jr's'}, \tag{3.3}$$

where $|P_i(x)|$ (respectively $|P_j(x)|$) denotes the number of cells of the portion $P_i(x)$, i.e., $|P_i(x)| = \sum_{\substack{r=1,\dots,K \\ s=1,\dots,L}} x_{irs}$.

3.2.3 Constraints

We now write the constraints in Problem $(\alpha - SBM)$ using the decision variables above, and give a brief explanation of each group of constraints.

$$\sum_{i=1,\dots,N} x_{irs} = 1, \qquad r = 1,\dots,K, \; s = 1,\dots,L, \tag{3.4}$$

$$\sum_{\substack{r=1,\dots,K \\ s=1,\dots,L}} x_{irs} \geq 1, \qquad i = 1,\dots,N, \tag{3.5}$$

$$x_{irs} \in \{0,1\}, \qquad i = 1,\dots,N, \tag{3.6}$$

$$\kappa \geq 0 \tag{3.7}$$

$$\sum_{\substack{(r'',s'') \in B((r,s),(r',s')) \\ (r'',s'') \neq (r,s) \\ (r'',s'') \neq (r',s')}} x_{ir''s''} \geq x_{irs} + x_{ir's'} - 1, \quad \begin{array}{l} i = 1,\dots,N, \; r,r' = 1,\dots,K, \\ s,s' = 1,\dots,L, \text{ such that cells} \\ (r,s) \text{ and } (r',s') \text{ are non-adjacent,} \end{array} \tag{3.8}$$

$$\sum_{i=1,\dots,N} \left| \left(\frac{1}{KL} \sum_{\substack{r=1,\dots,K \\ s=1,\dots,L}} x_{irs} \right) - \omega_i \right| \leq \alpha. \tag{3.9}$$

Constraint (3.4) models condition (C1), since it forces that every cell must belong to exactly one portion. In order to have at least one cell assigned to every portion, constraint (3.5) is considered. Constraints (3.6) and (3.7) establish the type of the variables. The box-connectivity in Definition 3.2, and required in (C2), is enforced

through constraint (3.8). The box-connectivity of $P_i(x)$ is enforced by imposing that the box generated by each pair of non-adjacent cells belonging to $P_i(x)$ (two cells that do not share a common boundary) must contain also cells of $P_i(x)$, namely the intersection between such box (excluding its two generator cells) and the portion must be nonempty. Finally, the error incurred by approximating the frequencies $\boldsymbol{\omega}$ by the area of the portions is modeled through constraint (3.9).

3.2.4 Writing the problem as an MILP

Thus, given a frequency distribution $\boldsymbol{\omega}$ and a dissimilarity measure δ, Problem $(\alpha - SMB)$ is stated as MINLP as

$$\begin{array}{ll} \min & \displaystyle\sum_{\substack{i,j=1,\dots,N \\ i\neq j}} |distance(P_i(x), P_j(x)) - \kappa\delta_{ij}| \\ \text{s.t.} & (3.4)\text{–}(3.9), \end{array} \qquad (\alpha - SBM)$$

where $distance(P_i(x), P_j(x))$ in the objective function is replaced by either $SL(P_i(x), P_j(x))$, $CL(P_i(x), P_j(x))$ or $AvL(P_i(x), P_j(x))$, as in (3.1)-(3.3) respectively.

One has that, for the SL and CL distances, $(\alpha - SBM)$ can be easily reformulated as an MILP, and as an approximation to an MILP in the case of AvL. These three reformulations of $(\alpha - SBM)$, one for each distance function, are described in what follows, and they are called $(\alpha - SBM)^L_{SL}$, $(\alpha - SBM)^L_{CL}$ and $(\alpha - SBM)^L_{AvL}$, respectively. Note that the statement of Problem $(\beta - SBM)$ as an MINLP and its pertinent reformulation as an MILP or an approximation to an MILP, yielding $(\beta - SBM)^L_{SL}$, $(\beta - SBM)^L_{CL}$ and $(\beta - SBM)^L_{AvL}$, are straightforward from the work done for $(\alpha - SBM)$.

In all three cases, we use the usual techniques to linearize the product of binary variables proposed by McCormick (1976) and the absolute values both in the objective function and constraint (3.9). Thus, we consider the following reformulations **R1**, **R2** and **R3** stated as follows:

R1: let $u_{ijrsr's'}$ be defined as $u_{ijrsr's'} = x_{irs} \cdot x_{ir's'}$, which implies that

$$u_{ijrsr's'} = \begin{cases} 1 & \text{if cell } (r,s) \text{ belongs to portion } P_i(x) \text{ and cell } (r',s') \text{ belongs to portion } P_j(x) \\ 0 & \text{otherwise.} \end{cases}$$

The following set of constraints is also needed to be included in the linear reformulation to properly linearize the product:

$$u_{ijrsr's'} \leq x_{irs} \qquad i,j=1,\dots N,\ r,r'=1,\dots,K,\ s,s'=1,\dots,L \quad (3.10)$$

$$u_{ijrsr's'} \leq x_{jr's'} \qquad i,j=1,\dots N,\ r,r'=1,\dots,K,\ s,s'=1,\dots,L \quad (3.11)$$

$$x_{irs} + x_{jr's'} \leq u_{ijrsr's'} + 1 \qquad i,j=1,\dots N,\ r,r'=1,\dots,K,\ s,s'=1,\dots,L \quad (3.12)$$

$$u_{ijrsr's'} \in \{0,1\} \qquad i,j=1,\dots N,\ r,r'=1,\dots,K,\ s,s'=1,\dots,L. \quad (3.13)$$

R2: The objective function of $(\alpha - SBM)$ is written as a linear objective subject to additional constraints by considering the positive continuous variables $\varphi_{ij}, \psi_{ij} \geq 0$, $i,j=1,\dots,N$, $i \neq j$:

$$\min \left\{ \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} |distance(P_i(x),P_j(x)) - \kappa\delta_{ij}| : \text{ s.t. } (3.4)-(3.9) \right\} =$$

$$= \min \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} (\varphi_{ij} + \psi_{ij})$$

$$\text{s.t.} \begin{cases} (3.4)-(3.9) & \\ distance(P_i(x),P_j(x)) - \kappa\delta_{ij} = \varphi_{ij} - \psi_{ij} & i,j=1,\dots,N,\ i\neq j \\ \varphi_{ij} \geq 0 & i,j=1,\dots,N,\ i\neq j \\ \psi_{ij} \geq 0 & i,j=1,\dots,N,\ i\neq j \end{cases}$$

R3: Constraint (3.9) is written as the following set of linear constraints by adding the positive continuous variables y_i, $i=1,\dots,N$:

$$\left(\frac{1}{KL} \sum_{\substack{r=1,\dots,K \\ s=1,\dots,L}} x_{irs} \right) - \omega_i \leq y_i \qquad i=1,\dots,N \quad (3.9a)$$

$$\left(\frac{1}{KL} \sum_{\substack{r=1,\dots,K \\ s=1,\dots,L}} x_{irs} \right) - \omega_i \geq -y_i \qquad i=1,\dots,N \quad (3.9b)$$

$$\sum_{i=1,\dots,N} y_i = \alpha \quad (3.9c)$$

$$y_i \geq 0 \qquad i=1,\dots,N. \quad (3.9d)$$

Single Linkage

Let z_{ij} and $\eta_{ijrsr's'}$ be defined as follows:

$$z_{ij} = \min_{\substack{r,r'=1,\dots,K \\ s,s'=1,\dots,L}} \left\{ |r-r'| + |s-s'| : x_{irs} = x_{jr's'} = 1 \right\}$$

$$\eta_{ijrsr's'} = \begin{cases} 1 & \text{if the minimum in } SL(P_i(x), P_j(x)) \text{ is attained at pair } (r,s),\ (r',s') \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $(\alpha - SBM)_{SL}$ is stated by using these set of variables plus the convenient constraints to ensure that the distances between portions correspond with the Single Linkage.

$$\begin{aligned}
&(\alpha - SBM)_{SL} = \\
&= \min \left\{ \sum_{\substack{i,j=1,\dots,N \\ i \neq j}} |SL(P_i(x), P_j(x)) - \kappa\delta_{ij}| : \text{ s.t. } (3.4) - (3.9) \right\} \\
&= \min \sum_{\substack{i,j=1,\dots,N \\ i \neq j}} |z_{ij} - \kappa\delta_{ij}| \\
&\quad \text{s.t.} \begin{cases}
(3.4) - (3.9) & \\
z_{ij} \geq (|r-r'| + |s-s'|) \cdot \eta_{ijrsr's'} & i,j = 1,\dots,N,\ i \neq j \\
 & r,r' = 1,\dots,K,\ s,s' = 1,\dots,L \\
z_{ij} \leq (|r-r'| + |s-s'|) + (K+L-2)(1 - x_{irs} \cdot x_{jr's'}) & i,j = 1,\dots,N,\ i \neq j \\
 & r,r' = 1,\dots,K,\ s,s' = 1,\dots,L \\
\sum_{\substack{i,j=1,\dots,N \\ i \neq j \\ r,r'=1,\dots,K \\ s,s'=1,\dots,L}} \eta_{ijrsr's'} \geq 1 & \\
\eta_{ijrsr's'} \leq x_{irs} \cdot x_{jr's'} & i,j = 1,\dots,N,\ i \neq j \\
 & r,r' = 1,\dots,K,\ s,s' = 1,\dots,L \\
\eta_{ijrsr's'} \in \{0,1\} & i,j = 1,\dots,N,\ i \neq j \\
 & r,r' = 1,\dots,K,\ s,s' = 1,\dots,L.
\end{cases}
\end{aligned}$$

First and second added constraints ensure that variables z_{ij} are well-defined, i.e., if the minimum is achieved in cells (r,s) and (r',s'), then z_{ij} is equal to the minimum distance between portions P_i and P_j. Therefore, these constraints are inactive either if $\eta_{ijrsr's'} = 0$ or if (r,s) or (r',s') do not belong to P_i and P_j, respectively. Third constraint is to impose that there must exist at least one pair of cells which give the minimum distance, and fourth that each pair must belong to the corresponding portion. Finally, we impose the binary requirement for $\eta_{ijrsr's'}$. Observe that, by applying **R1**-**R3** reformulations, the formulation of $(\alpha - SBM)_{SL}$ as an MILP, namely $(\alpha - SBM)^L_{SL}$,

is straightforward.

Complete Linkage

Similar reasoning applied to Single Linkage leads to the statement of $(\alpha - SBM)_{CL}$ as an MILP. Let z_{ij} and $\eta_{ijrsr's'}$ be defined as follows:

$$z_{ij} = \max_{\substack{r,r'=1,\dots,K \\ s,s'=1,\dots,L}} \left\{ |r-r'| + |s-s'| : x_{irs} = x_{jr's'} = 1 \right\}$$

$$\eta_{ijrsr's'} = \begin{cases} 1 & \text{if the maximum in } CL(P_i(x), P_j(x)) \text{ is attained at pair } (r,s),\ (r',s') \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $(\alpha - SBM)_{CL}$ is stated by using these set of variables plus the convenient constraints to ensure that the distances between portions correspond with the Complete Linkage.

$$\begin{aligned}
&(\alpha - SBM)_{CL} = \\
&= \min \left\{ \sum_{\substack{i,j=1,\dots,N \\ i \neq j}} |CL(P_i(x), P_j(x)) - \kappa\delta_{ij}| : \text{ s.t. } (3.4) - (3.9) \right\} \\
&= \min \sum_{\substack{i,j=1,\dots,N \\ i \neq j}} |z_{ij} - \kappa\delta_{ij}| \\
&\quad \text{s.t.} \begin{cases}
(3.4) - (3.9) & \\
z_{ij} \geq (|r-r'| + |s-s'|) \cdot x_{irs} \cdot x_{jr's'} & i,j = 1,\dots,N,\ i \neq j \\
& r,r' = 1,\dots,K,\ s,s' = 1,\dots,L \\
z_{ij} \leq (|r-r'| + |s-s'|) \cdot \eta_{ijrsr's'} + (K+L-2)(1-\eta_{ijrsr's'}) & i,j = 1,\dots,N,\ i \neq j \\
& r,r' = 1,\dots,K,\ s,s' = 1,\dots,L \\
\displaystyle\sum_{\substack{i,j=1,\dots,N \\ r,r'=1,\dots,K \\ s,s'=1,\dots,L}} \eta_{ijrsr's'} \geq 1 & \\
\eta_{ijrsr's'} \leq x_{irs} \cdot x_{jr's'} & i,j = 1,\dots,N,\ i \neq j \\
& r,r' = 1,\dots,K,\ s,s' = 1,\dots,L \\
\eta_{ijrsr's'} \in \{0,1\} & i,j = 1,\dots,N,\ i \neq j \\
& r,r' = 1,\dots,K,\ s,s' = 1,\dots,L
\end{cases}
\end{aligned}$$

First added constraint ensures that for any pair of cells belonging to two different portions, the distance between those portions is greater or equal than the distance between such cells. Second one is to ensure that the distance between two portions is exactly the maximum distance between all possible pairs of cells, since $\eta_{ijrsr's'}$ are precisely forcing that. Third constraint is to impose that there must exist at least one pair of cells which give the maximum distance, and fourth that each pair must belong

to the corresponding portion. Finally, we impose the binary requirement for $\eta_{ijrsr's'}$. Observe that, by applying **R1-R3** reformulations, the formulation of $(\alpha - SBM)_{CL}$ as an MILP, namely $(\alpha - SBM)^L_{CL}$, is straightforward.

Average Linkage

Since the frequency associated to individual v_i is represented through the area of portion P_i, we consider a surrogate of the expression of the average distance, (3.3), by approximating the number of cells of P_i by $|P_i| = \omega_i KL$, yielding the following expression for the Approximated Average Linkage (AvL_{app})

$$AvL_{app}(P_i(x), P_j(x)) = \frac{1}{\omega_i \omega_j K^2 L^2} \sum_{\substack{r,r'=1,\dots,K \\ s,s'=1,\dots,L}} (|r - r'| + |s - s'|) \cdot x_{irs} \cdot x_{jr's'}. \quad (3.14)$$

$$(\alpha - SBM)_{AvL} =$$
$$= \min \left\{ \sum_{\substack{i,j=1,\dots,N \\ i \neq j}} |AvL_{app}(P_i(x), P_j(x)) - \kappa \delta_{ij}| : \text{ s.t. } (3.4) - (3.9) \right\}$$

By applying **R1-R3** reformulations, the formulation of $(\alpha - SBM)_{AvL}$ as an MILP, namely $(\alpha - SBM)^L_{AvL}$, is obtained.

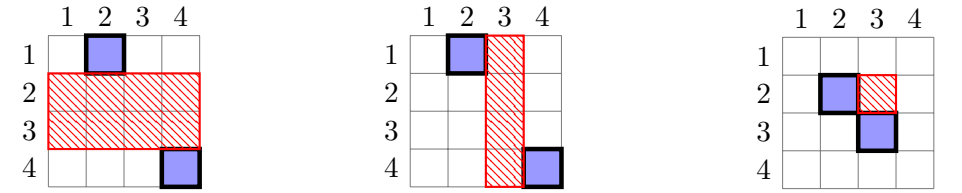
3.2.5 A tighter model

Problems $(\alpha - SBM)$ and $(\beta - SBM)$ enforce the box-connectivity of the portions $\boldsymbol{P}$ in the SBM through constraint (3.8). There exist several attempts in the literature which deal with the problem of modeling connectivity with integer programming, for instance using graph theory (Jafari and Hearne, 2013; Önal and Briers, 2006), designing the connected portions according to fixed locations (Önal et al., 2016), or considering *node-cut sets* (Carvajal et al., 2013; Wang et al., 2015).

Definition 3.3. *Let $G = (U, E)$ be a graph, whose nodes U are the cells of a (K, L)-grid. If two nodes $u, u' \in U$ are adjacent on the grid, then there exists an edge $(u, u') \in E$ linking the two nodes.*

Given two non-adjacent nodes $u, v \in U$, a set of nodes $C \subseteq U \setminus \{u, v\}$ is a node-cut separating u and v if there is no path between u and v in the subgraph $G' = (U \setminus C, E \setminus E')$, where $E' = \{(w, w') : w, w' \in C \cup \{u, v\}\}$.

Figure 3.2 illustrates some possible node-cuts on a $(4, 4)$-grid: a horizontal cut for cells $(1, 2)$ and $(4, 4)$ in Figure 3.2 (a), a vertical cut for the same cells in Figure 3.2 (b)



(a) Horizontal node-cut (b) Vertical node-cut (c) Diagonal node-cut

Figure 3.2: Dashed cells illustrate node-cuts of shaded cells on a $(4, 4)$-grid

and a diagonal cut in Figure 3.2 (b) for cells $(2, 2)$ and $(3, 3)$.

Thanks to Definition 3.3, the box-connectivity constraint, namely (3.8), can be modeled through intersections of node-cuts yielding an equivalent formulation for Problems $(\alpha - SBM)$ and $(\beta - SBM)$. In what follows, we prove that box-connectivity can be equivalently modeled through the following set of constraints:

$$\sum_{\substack{r<r''<r' \\ \min\{s,s'\}\leq s''\leq \max\{s,s'\}}} x_{ir''s''} \geq x_{irs} + x_{ir's'} - 1, \qquad \begin{array}{l} i = 1, \dots, N, \\ r = 1, \dots, K-2, \\ r' = r+2, \dots, K, \\ s, s' = 1, \dots, L, \end{array} \tag{3.8a}$$

$$\sum_{\substack{\min\{r,r'\}\leq r''\leq \max\{r,r'\} \\ s<s''<s'}} x_{ir''s''} \geq x_{irs} + x_{ir's'} - 1, \qquad \begin{array}{l} i = 1, \dots, N, \\ r, r' = 1, \dots, K, \\ s = 1, \dots, L-2, \\ s' = s+2, \dots, L \end{array} \tag{3.8b}$$

$$x_{i,r-1,s} + x_{i,r,s-1} \geq x_{irs} + x_{i,r-1,s-1} - 1, \qquad \begin{array}{l} i = 1, \dots, N, \\ r = 2, \dots, K, \ s = 2, \dots, L \end{array} \tag{3.8c}$$

$$x_{i,r-1,s} + x_{i,r,s+1} \geq x_{irs} + x_{i,r-1,s+1} - 1, \qquad \begin{array}{l} i = 1, \dots, N, \\ r = 2, \dots, K, \ s = 1, \dots, L-1 \end{array} \tag{3.8d}$$

$$x_{i,r+1,s} + x_{i,r,s-1} \geq x_{irs} + x_{i,r+1,s-1} - 1, \qquad \begin{array}{l} i = 1, \dots, N, \\ r = 1, \dots, K-1, \ s = 2, \dots, L \end{array} \tag{3.8e}$$

$$x_{i,r+1,s} + x_{i,r,s+1} \geq x_{irs} + x_{i,r+1,s+1} - 1, \qquad \begin{array}{l} i = 1, \dots, N, \\ r = 1, \dots, K-1, \ s = 1, \dots, L-1 \end{array} \tag{3.8f}$$

Constraint (3.8a) considers bounded horizontal node-cuts, in the sense that it considers that if two cells belong to the same portion and are placed in different and non-contiguous rows, namely there is at least a row in between, then there must exist cells belonging also to that portion in the rows strictly in between them. On the other hand, constraint (3.8b) addresses bounded vertical node-cuts, which analogous to the

horizontal ones but considering columns instead of rows. Finally, constraints (3.8c)-(3.8f) model diagonal node-cuts, as this in Figure 3.2 (c), namely if two cells belong to the same portion and they touch in one corner (shaded cells), then one of the two cells in the diagonal crossing that corner must also belong to such portion (striped cells).

Let us consider Problem $(\alpha - SBM)$ stated as

$$(\alpha - SBM)_{box} = \min \left\{ \sum_{\substack{i,j=1,\dots,N \\ i \neq j}} |distance(P_i(x), P_j(x)) - \kappa\delta_{ij}| : \text{s.t. } (3.4) - (3.9) \right\}$$

and

$$(\alpha - SBM)_{cut} = \min \left\{ \sum_{\substack{i,j=1,\dots,N \\ i \neq j}} |distance(P_i(x), P_j(x)) - \kappa\delta_{ij}| : \text{s.t. } (3.4) - (3.7), (3.8a) - (3.8f), (3.9) \right\},$$

where $distance(P_i(x), P_j(x))$ in the objective function is replaced by either $SL(P_i(x), P_j(x))$, $CL(P_i(x), P_j(x))$ or $AvL(P_i(x), P_j(x))$, as in (3.1)-(3.3) respectively.

Proposition 3.1. *One has that Problems $(\alpha - SBM)_{box}$ and $(\alpha - SBM)_{cut}$ are equivalent.*

Proof. Observe that constraints (3.4)-(3.7) and (3.9) appear in both problems, and thus it remains to prove that constraint (3.8) is equivalent to constraints (3.8a)-(3.8f). Let us prove such statement for a portion $P_i(x)$, denoted S to simplify.

Let us prove that, if S is box-connected, then (3.8a)-(3.8f) hold.

Suppose that (3.8a) fails. Then, the summation on the left-hand-side is equal to zero, which means that there exist two cells $(r, s), (r', s') \in S$, such that $r < r' - 1$ and

$$\big\{(r'', s'') : r < r'' < r', \min\{s, s'\} \leq s'' \leq \max\{s, s'\}\big\} \cap S = \emptyset. \tag{3.15}$$

In particular, (3.15) is also true when $s' = s$. This means

$$\big\{(r'', s) : s < s'' < s'\big\} \cap S = \emptyset. \tag{3.16}$$

On the other hand, S is box-connected and then,

$$\exists (r'', s) \in B\big((r, s), (r', s)\big) \cap S, \text{ such that } r'' \neq r \text{ and } r'' \neq r',$$

which is clearly in contradiction with (3.16).

Using a similar procedure, we can prove that (3.8b)-(3.8f) also hold.

Conversely, let us prove that, if (3.8a)-(3.8f) are satisfied, then S is box-connected.

Let $(r,s),(r',s') \in S$ such that $r < r'-1$. Since (3.8a) is satisfied, there exists $(r'',s'') \in S$ such that $r < r'' < r'$ and $\min\{s,s'\} \leq s'' \leq \max\{s,s'\}$, which implies that $(r'',s'') \in B((r,s),(r',s')) \cap S$, and it is different from (r,s) and (r',s'). This proves the desired result.

The case in which $s < s'-1$, constraint (3.8b) applies, showing that S is box-connected. Finally, for pairs of cells belonging to S such that $\{(r,s),(r-1,s-1)\}$, $\{(r,s),(r-1,s+1)\}$, $\{(r,s),(r+1,s-1)\}$ and $\{(r,s),(r+1,s+1)\}$, constraint (3.8c)-(3.8f) yield, respectively, the box-connectivity of S.

□

In order to choose between the $(\alpha-SBM)_{box}$ or the $(\alpha-SBM)_{cut}$ formulations, we study their continuous relaxation tightness. The formulation with the tightest continuous relaxation will be the preferred. Thus, let us consider the continuous relaxations of the MINLP Problems $(\alpha - SBM)_{box}$ and $(\alpha - SBM)_{cut}$, obtained by substituting constraint (3.6) by $0 \leq x_{irs} \leq 1$, $i = 1,\ldots,N$, $r = 1,\ldots,K$, $s = 1,\ldots,L$, and denoted as $R((\alpha - SBM)_{box})$ and $R((\alpha - SBM)_{cut})$ respectively. We show in what follows that, while the MINLP Problems $(\alpha - SBM)_{box}$ and $(\alpha - SBM)_{cut}$ are equivalent, as stated in Proposition 3.1, it turns out that $R((\alpha - SBM)_{cut})$, is, in general, tighter than $R((\alpha - SBM)_{box})$.

Let $v(R((\alpha - SBM)_{box}))$ and $v(R((\alpha - SBM)_{cut}))$ be the optimal objective values of Problems $R((\alpha - SBM)_{box})$ and $R((\alpha - SBM)_{cut})$, respectively.

Proposition 3.2. *One has that*

$$v(R((\alpha - SBM)_{box})) \leq v(R((\alpha - SBM)_{cut})). \tag{3.17}$$

Proof. In order to show inequality (3.17), we prove that each solution x to $R((\alpha - SBM)_{cut})$ is also feasible to $R((\alpha - SBM)_{box})$. Let x be a feasible solution to $R(\alpha - (SBM)_{cut})$, we need to prove that x satisfies constraints (3.4)-(3.9). First, constraints (3.4)-(3.7) and (3.9) are satisfied, since they appear in both problems. Then, let us check that (3.8) also holds. Since x verifies (3.8a), one has for $i = 1,\ldots,N$, $r = 1,\ldots,K-2$, $r' = r+2,\ldots,L$, $s,s' = 1,\ldots,L$

$$x_{irs} + x_{ir's'} - 1 \leq \sum_{\substack{r<r''<r' \\ \min\{s,s'\}\leq s''\leq\max\{s,s'\}}} x_{ir''s''} \leq \sum_{\substack{(r'',s'')\in B((r,s),(r',s')) \\ (r'',s'')\neq(r,s) \\ (r'',s'')\neq(r',s')}} x_{ir''s''}.$$

In other words, x is shown to satisfy (3.8) because it also satisfies (3.8b)-(3.8f), and then the result holds. □

It turns out that the feasible region of $R((\alpha - SBM)_{cut})$ is smaller than the feasible

region of $R((\alpha - SBM)_{box})$, due to problem instances for which feasible solutions to $R(\alpha - (SBM)_{box})$ are unfeasible for $R((\alpha - SBM)_{cut})$. For instance, let us consider $N = 2$, $K = 3$ and $L = 2$. One has that the following solution is feasible for $R(\alpha - (SBM)_{box})$ but unfeasible for $R((\alpha - SBM)_{cut})$, taking $\alpha = \infty$ and $\kappa \geq 0$:

$$x_{111} = \frac{3}{4} \quad x_{112} = \frac{1}{5} \quad x_{121} = \frac{1}{5} \quad x_{122} = \frac{1}{5} \quad x_{131} = \frac{1}{5} \quad x_{132} = \frac{3}{4}$$

$$x_{211} = \frac{1}{4} \quad x_{212} = \frac{4}{5} \quad x_{221} = \frac{4}{5} \quad x_{222} = \frac{4}{5} \quad x_{231} = \frac{4}{5} \quad x_{232} = \frac{1}{4}$$

The reasoning applied for finding a tighter reformulation to Problem $(\alpha - SBM)$ also holds for Problem $(\beta - SBM)$. MILPs reformulations described in Section 3.2.4 also hold in case that Problems $(\alpha - SBM)$ and $(\beta - SBM)$ are modeled through connectivity constraints (3.8a)-(3.8f) instead of (3.8). These MILPs reformulations let us solve to optimality small instances of the optimization problems, i.e., when the number of individuals is not very large as well as the (K, L)-grid is coarse enough. However, big instances remain hard to solve by standard MILPs optimizers, and heuristic techniques seem to be necessary to handle real-world examples. In Section 3.3 we describe how the Large Neighborhood Search can be integrated in our methodology as a matheuristic to construct SBMs.

3.3 Algorithmic approach

In this section, we adapt the Large Neighborhood Search (LNS) methodology to the problem of building an SBM. Since small instances of the MILPs reformulations of Problems $(\alpha - SBM)$ and $(\beta - SBM)$ can be quickly solved by standard MILP optimizers and thanks to the grid structure considered, LNS seems to be a good candidate to take advantage from such facts. The LNS metaheuristic was first proposed by Shaw (1998). It has been successfully applied in recent years to problems of different nature, for instance Vehicle Routing Problems (Demir et al., 2012; Lodi et al., 2015; Ribeiro and Laporte, 2012), Scheduling, (Korsvik et al., 2011; Pacino and Van Hentenryck, 2011) and Information Visualization (Yoghourdjian et al., 2016). Roughly speaking, LNS performs a search for good solutions in a neighborhood of a starting point. A neighborhood of a certain point contains all the solutions that can be reached from the starting one by a *destroy* procedure, which erases part of the solution, and a *repair* method, which rebuilds the previously destroyed solution in order to obtain a new and better one. The good performance of LNS does not only depend on the quality of the starting point, but also on the *destroy* and *repair* methods. A tradeoff between the degree of destruction and the rebuilding process needs to be established: the degree of destruction should be

such that a large part of the search space can be explored, and the rebuilding process should be efficient. Algorithm 3.1 contains the pseudocode of the LNS metaheuristic. For further details on LNS see Pisinger and Ropke (2010) and references therein.

Algorithm 3.1 LNS pseudocode by Pisinger and Ropke (2010)

Input: A feasible solution x, an objective function f, *destroy* and *repair* procedures

1: $x^* \leftarrow x$
2: **repeat**
3: $\quad x^t \leftarrow repair(destroy(x^*))$;
4: $\quad$ **if** $f(x^t) < f(x^*)$ **then**
5: $\quad\quad x^* \leftarrow x^t$;
6: $\quad$ **end if**
7: **until** stop condition is met

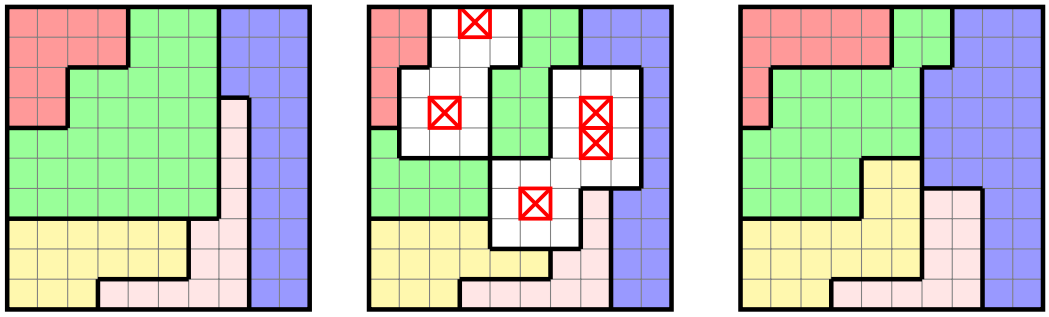
Output: x^*

In order to explain the *destroy* and *repair* procedures for our model, we introduce some necessary concepts. We define the *incidence degree* γ of a cell (r, s) in portion P_i as the number of connected cells surrounding (r, s) that also belong to P_i, namely $(r-1, s)$, $(r, s+1)$, $(r+1, s)$ and $(r, s-1)$. We say that a cell (r, s) is *redundant* if its removal keeps the portion which it belongs still connected. Observe that a cell is redundant if $\gamma = 0, 1, 2$ or 3, while $\gamma = 4$ implies that it is non-redundant.

Given an SBM, $\boldsymbol{P}$, the *destroy* phase consists of selecting randomly a number μ of redundant cells in $\boldsymbol{P}$, and removing them as well as their eight surrounding neighbors, yielding an incomplete SBM, $destroy(\boldsymbol{P})$. Figure 3.3 (b) contains $\mu = 5$ selected redundant cells (crossed bold) from the initial solution in Figure 3.3 (a), and in white their surrounded neighbors, which are also deleted. The repair step consists of rebuilding the destroyed solution $destroy(\boldsymbol{P})$ by assigning the free cells to portions satisfying (C1)-(C4), yielding a new SBM, $repair(destroy(\boldsymbol{P}))$, see Figure 3.3 (c).

The decision of selecting redundant cells in the destroy stage instead of non-redundant ones is not arbitrary. If only non-redundant cells were chosen, the risk of getting stuck on the initial SBM, $\mathbf{P}$, increases. The reason comes from imposing box-connectivity in the portions forming the $repair(destroy(\boldsymbol{P}))$ SBM, which may force to reconstruct $\mathbf{P}$ time after time because any other partition is feasible. On the other hand, selecting only redundant cells yields to a $destroy(\boldsymbol{P})$ configuration with more freedom (regarding box-connectivity) to obtain a new allocation of the free cells distinct from $\boldsymbol{P}$.

As we highlighted in the previous section, small instances of the MILP reformulations of $(\alpha - SBM)$ and $(\beta - SBM)$ can be effortlessly solved by standard MILPs optimization routines, and that is the key of our repairing procedure. To obtain the repaired solution, cells which have not been removed in the *destroy* stage become constants in the MILP



(a) Initial solution: $\boldsymbol{P}$ (b) $destroy(\boldsymbol{P})$ (c) $repair(destroy(\boldsymbol{P}))$

Figure 3.3: LNS algorithm for SBM.

problem, in the sense that their values are known and they are thus fixed. Those cells that have been removed in the *destroy* step remain decision variables when solving the MILP problem. Thanks to this variable-fixing procedure, the number of constraints is also substantially reduced, since there are already cells assigned to portions (reducing the number of constraints in (3.4) and (3.5)), there are also connected parts in the portions (reducing then the number of constraints in (3.8a)-(3.8f)), and even the number of extra constraints included in the linear reformulations. Therefore, the size of the MILPs to solve in the repairing procedure is considerably reduced compared to the full-size problems. Thus, depending on which criteria is to be improved in the current iteration of the LNS algorithm, either the MILP version of $(\alpha - SBM)$ or $(\beta - SBM)$ is solved when repairing the previously destroyed SBM to find a better one.

3.4 Computational experience

The performance of the approach described in previous sections is tested in three data sets of different sizes and nature on a (K, L)-grid with $K = L = 40$: first concerning finantial markets, second one about the letters in the English alphabet, and finally the provinces of The Netherlands as a geographic application. In Section 3.4.1 we describe the three datasets used in the experiments. Section 3.4.2 includes the experiments details and, finally, the obtained SBMs are presented in Section 3.4.3.

3.4.1 Datasets

The first dataset, `Markets`, consists of $N = 11$ financial markets across Europe and Asia. The frequency distribution $\boldsymbol{\omega}$ relates to the importance of each market relative to the world market portfolio (Flavin et al., 2002), and the dissimilarity $\boldsymbol{\delta}$ is based on the correlation between markets (Borg and Groenen, 2005). The second dataset, `Morse`, comes from a study regarding the confusion between the acoustic Morse signals used for the $N = 26$ letters of the English alphabet (Rothkopf, 1957). The dissimilarities

$\boldsymbol{\delta}$ between two signals are computed as the average percentage with which the answer 'Same!' was given in each combination of those signals (Borg and Groenen, 2005). The values $\boldsymbol{\omega}$ come from the relative frequencies of letters in the English language (Lewand, 2000). Last dataset, `Netherlands`, the individuals are the $N = 12$ provinces of The Netherlands, and the frequencies $\boldsymbol{\omega}$ are their population rates (Statistics Netherlands, 2013). The dissimilarity between two provinces is related to their geographical location, and this is measured as the length of the shortest path in the graph obtained from considering two nodes adjacent if the corresponding provinces are adjacent in the geographical map.

3.4.2 Experiments details

The matheuristic based on LNS described in Section 3.3 has been coded in AMPL (Fourer et al., 1993) and all the MILPs problems have been solved with CPLEX v12.6 (CPLEX, IBM ILOG, 2014) with a time limit of five minutes, on a PC Intel® Core™ i7-2600K, 16GB of RAM.

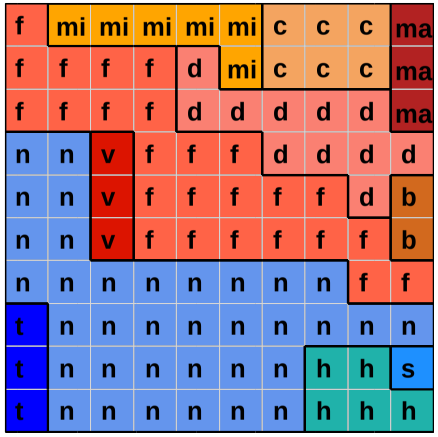
In order to compute an SBM with the methodology described in this chapter, some decisions should be made in advance. Firstly, we need to choose between any of the distances proposed in Section 3.2, namely Single, Complete or Average Linkage. Due to the characteristics of the problem we are dealing with, we consider that the Average Linkage reflects properly the visualization aim of the SBM, since it summarizes the distance information between every pair of cells belonging to the portions and thus, they become something global instead of something intrinsic of a single cell as the Single or Complete Linkage would do. Thus, the mathematical optimization models considered throughout this section are $(\alpha{-}SBM)^L_{AvL}$ and $(\beta{-}SBM)^L_{AvL}$, in which box-connectivity has been modeled through constraints (3.8a)-(3.8f). In order to perform the LNS procedure described in Section 3.3, we need to determine how the redundant cells are selected in the *destroy* stage and which mathematical optimization problem, either $(\alpha - SBM)^L_{AvL}$ and $(\beta - SBM)^L_{AvL}$, is solved in the repair phase. On one hand, in the destroy step, μ redundant cells are selected with nonuniform probabilities, depending on their incidence degree: a redundant cell (r, s) with incidence degree equal to $\gamma = 0, \ldots, 3$, will be selected with probability proportional to $2^{3-\gamma}$. This way, cells with a low incidence degree, and thus those which have more chances to be allocated to different portions, are selected with higher probability. On the other hand, when repairing the destroyed solution, we assume that we have an SBM for which the error when approximating $\boldsymbol{\omega}$ by the portions' areas is pretty low, and therefore we can use a small value of parameter α. Then, Problem $(\alpha - SBM)^L_{AvL}$ is solved in this phase to improve the dissimilarities fit. We made this decision because we considered that it is crucial that areas are very well fitted, whereas the dissimilarities admit more flexibility when interpreting the SBM. As a stopping condition for LNS we establish a maximum

number of iterations.

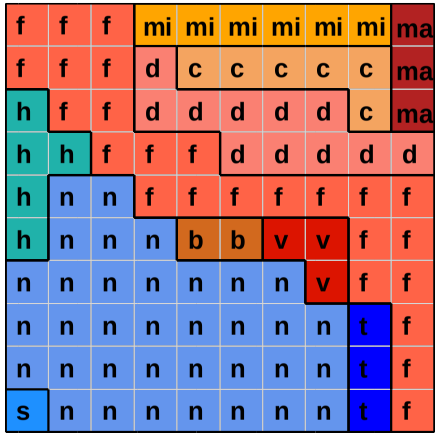
In order to construct an SBM on a $(40, 40)$-grid for the three previously described datasets, we firstly obtain an initial SBM on a $(10, 10)$-grid by solving a surrogate of Problem $(\beta - SBM)^L_{AvL}$, which has an accurate representation of the frequency distribution as the area of the portions and also some information about the dissimilarities. To do that, Problem $(\beta - SBM)^L_{AvL}$ is solved, setting $\beta = \infty$, including an extra set of constraints that imposes that some cells are already assigned to some portions. These cells are found accordingly to the dissimilarities between individuals by using the Multidimensional Scaling for (K, L)-rectangular maps described in Section 2.3.1. Then, 100 iterations of the LNS algorithm are performed, with $\mu = 4$ and α equal to the objective value obtained when solving the surrogate $(\beta - SBM)^L_{AvL}$. The so-obtained SBM is embedded into a $(20, 20)$-grid, and it is considered as an initial SBM for 50 iterations of the LNS algorithm with $\mu = 8$ and α a 15% smaller than the objective value obtained when solving the surrogate $(\beta - SBM)^L_{AvL}$ in the first stage. Finally, the so-obtained SBM is embedded into a $(40, 40)$-grid, and it is considered as an initial SBM for 25 iterations of the LNS algorithm with $\mu = 16$ and α a 5% smaller than the objective value obtained when solving the surrogate $(\beta - SBM)^L_{AvL}$ in the previous stage.

3.4.3 Results

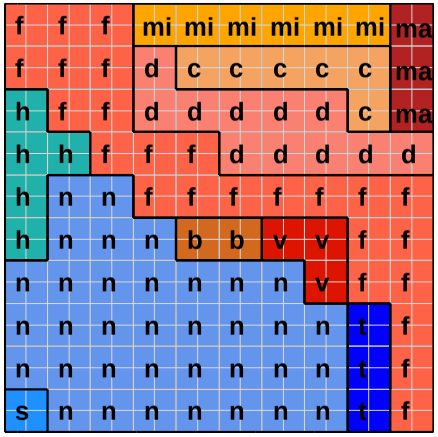
The process to construct an SBM for `Markets` dataset is illustrated in Figures 3.4-3.5. The initial SBM on a $(10, 10)$-grid obtained by solving the surrogate $(\beta - SBM)^L_{AvL}$ is depicted in Figure 3.4 (a). After 100 iterations of LNS the SBM in Figure 3.4 (b) is obtained. Then, this SBM is embedded into a $(20, 20)$-grid, see Figure 3.4 (c), and it is set as the initial solution for the LNS algorithm. After 50 iterations and a reduction of a 15% of the parameter α, the solution depicted in Figure 3.4 (d) is obtained. This SBM is embedded into a $(40, 40)$-grid, see Figure 3.4 (e), and it is set as the initial solution for the LNS algorithm. After 25 iterations and imposing a 5% reduction of the parameter α, the solution in Figure 3.5 is obtained. Figures 3.6-3.7 show the SBMs for `Morse` and `Netherlands` datasets, which are also obtained with the procedure described above.



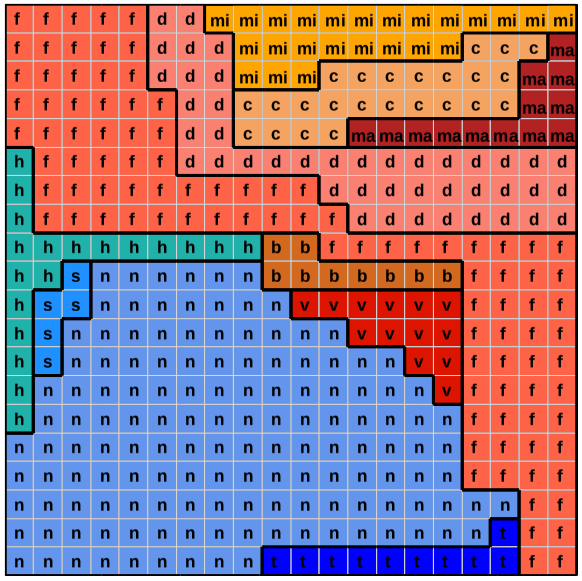
(a) Initial SBM on a $(10, 10)$-grid



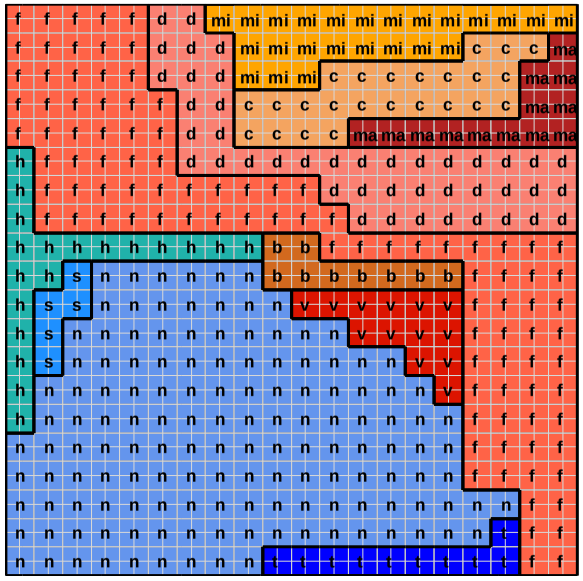
(b) SBM on a $(10, 10)$-grid after LNS



(c) Initial SBM on a $(20, 20)$-grid



(d) SBM on a $(20, 20)$-grid after LNS



(e) Initial SBM on a $(40, 40)$-grid

Figure 3.4: Illustration of the methodology proposed to construct an SBM for `Markets` dataset.

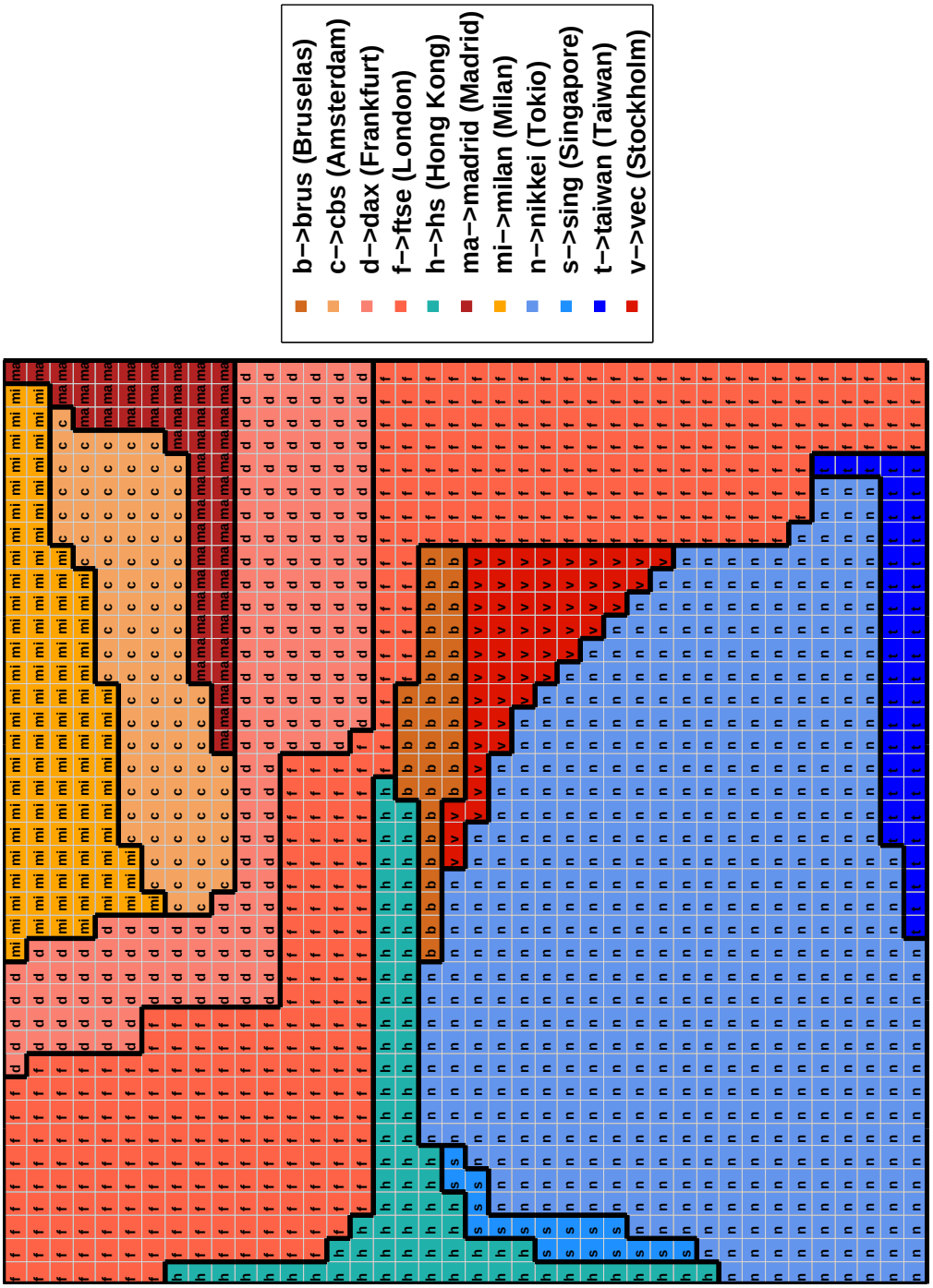


Figure 3.5: SBM for `Markets` using a $(40, 40)$-grid.

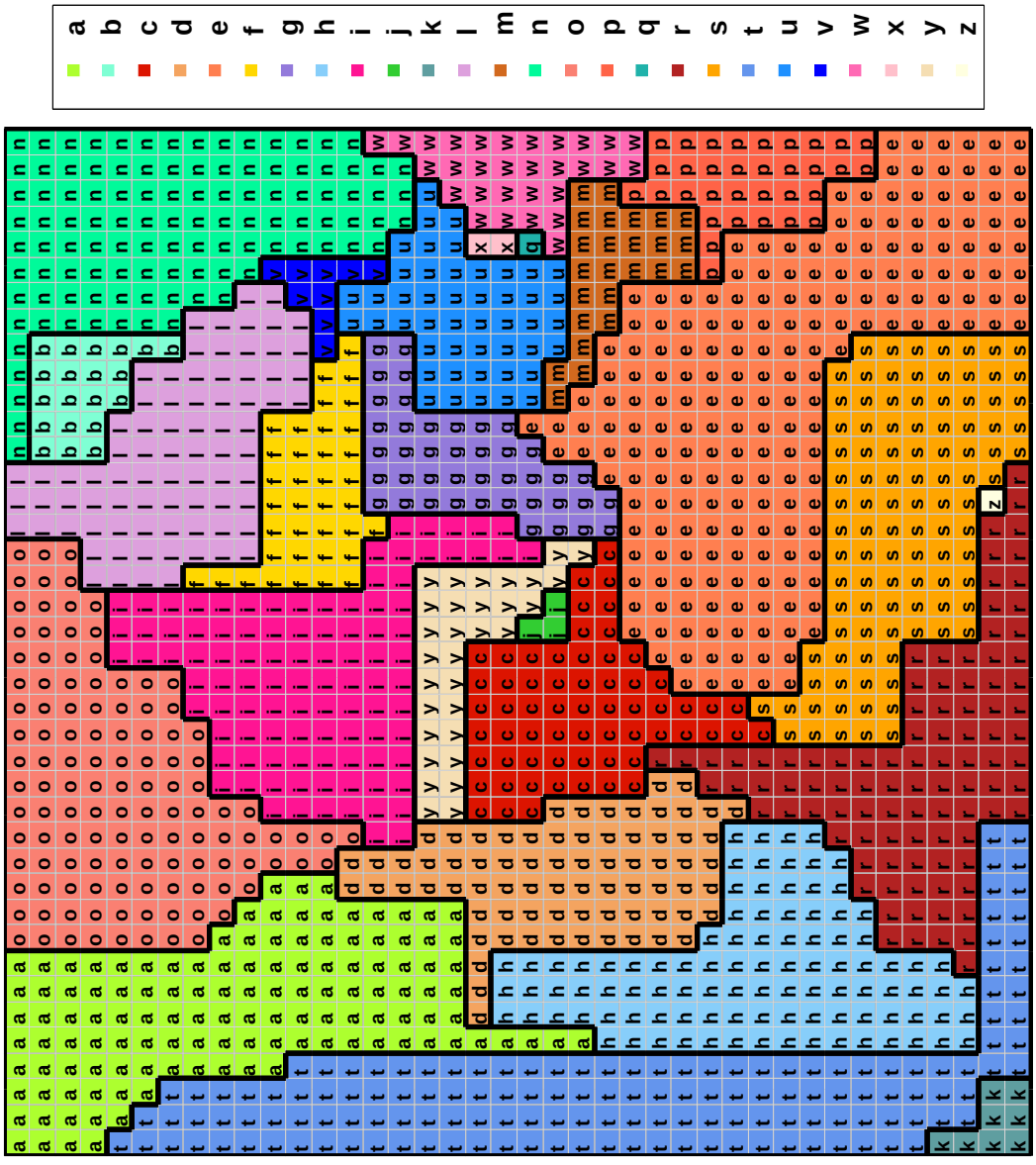


Figure 3.6: SBM for `Morse` using a $(40, 40)$-grid.

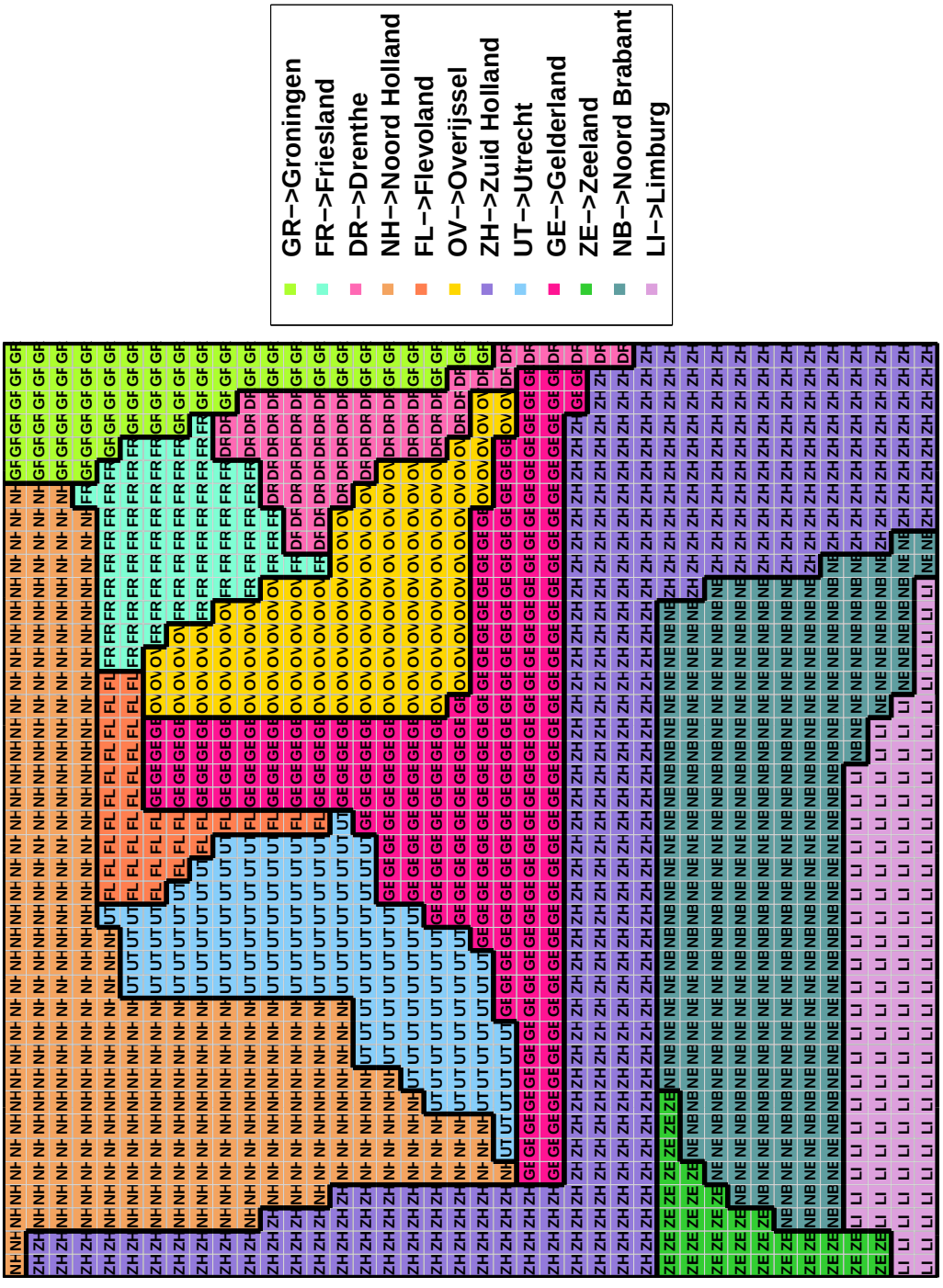


Figure 3.7: SBM for `Netherlands` using a $(40, 40)$-grid.

3.5 Conclusions

In this chapter, we have formally introduced a mathematical optimization formulation for the problem of visualizing as a Space-filling Box-connected Map a frequency distribution and a dissimilarity relation attached to a set of individuals. This visualization problem is addressed by simultaneously optimizing the error incurred when approximating the frequencies by the area of the portions and the dissimilarities by the distances among them, satisfying the condition that the portions are box-connected and they must form a partition of the unit square. Such approach yields a biobjective optimization problem, which is solved as a single-objective problem that optimizes the fit in distances ensuring a very accurate fit in the sizes of the portions. The Large Neighborhood Search has been proposed to deal with big instances of such problems, due to the fact that the combinatorial structure underlying in the optimization problems makes solving them to optimality too time-demanding. The usefulness of our approach has been illustrated in a variety of data sets, related to market indices, the Morse code and a geographical map.

Chapter 4

Visualizing frequencies and dissimilarities as geometric objects: A Difference of Convex optimization approach

The visualization frameworks presented in Chapters 2 and 3 seek an approximate representation of the frequency distribution attached to a set of individuals while preserving a proximity structure in a space-filling layout. In this chapter, we relax the space-filling requirement, and seek an accurate visualization of those frequencies, while still preserving the proximity structure. The individuals involved in the dataset are visualized as convex bodies in a given visualization region, whose sizes depict the frequencies and their locations their proximity. This problem, which extends the standard Multidimensional Scaling Analysis, is written as a global optimization problem whose objective is the difference of two convex functions (DC). Suitable DC decompositions allow us to use the Difference of Convex Algorithm (DCA) in a very efficient way. Our algorithmic approach is used to visualize two real-world datasets.

4.1 Introduction

In this chapter we present a new mathematical programming framework to build a visualization map, in which the individuals in a set $V = \{v_1, \ldots, v_N\}$ are depicted as convex bodies in a bounded region $\Omega \subset \mathbb{R}^n$, usually $n \leq 3$. These objects must have a volume proportional to a given frequency distribution associated with the individuals, $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_N)$, and they should be placed accordingly to a dissimilarity measure attached to the individuals, $\boldsymbol{\delta} = (\delta_{ij})_{i,j=1,\ldots,N}$. In order to locate the objects in Ω, a reference convex body $\mathcal{B}$ is used, to be translated and expanded. However, since our final goal is to obtain a visualization map which allows the analysts to understand the data they are working with, a criterion which somehow controls the appearance of the plot needs to be also considered. We will deal with this paradigm by focusing on how the objects are spread out over Ω.

The visualization model presented in this chapter may seem very close to Multidimensional Scaling (MDS) (Kruskal, 1964; Torgerson, 1958). However, it has the special feature of representing in the bounded region Ω not only dissimilarities as distances between convex bodies, but also the frequencies $\boldsymbol{\omega}$ through the volumes of these objects in Ω. Moreover, contrary to proportional symbol maps (Cabello et al., 2010) our visualization tool is able to deal with dissimilarities of any nature, not necessarily coming from a geographical application, and rescale both the dissimilarities and the frequencies associated to the individuals to fit in Ω. Observe that fitting the convex bodies into Ω may yield representations in which the objects intersect if their sizes are not small enough, but, on the other hand, too small objects obstruct the visualization of the frequencies (Cano et al., 2013; Kunigami et al., 2014). Ideally the objects should be spread out across the visualization region. This aim will be also taken into account when modeling the problem.

The methodology proposed in this chapter has applications in fields others than

Information Visualization, such as for instance, Location Analysis or Distance Geometry. In location problems, the facilities to be located are usually considered as points. However, a natural extension is to consider facilities as dimensional structures (Díaz-Báñez et al., 2004), and Difference of Convex (DC) techniques have been specifically applied to this generalization (Blanquero et al., 2009; Carrizosa et al., 1998a). Ours can also be seen as a problem in Distance Geometry optimization, as carefully reviewed by Liberti et al. (2014). In Distance Geometry, a graph realization problem consists of finding a configuration of points such that their (Euclidean) distances fit a given dissimilarity matrix. Among them is the Sensor Network Location problem (Pong and Tseng, 2011; So and Ye, 2007; Tseng, 2007; Wang et al., 2008), in which one assumes that some individuals are anchors (their location is known) and the remaining ones are mobile sensors, whose location is to be obtained so that their Euclidean distances fit the dissimilarities. Thus, our method can also be applied to the Sensor Network Location problem, in which sensors and anchors have a nonnegligible area.

In this chapter, the construction of a visualization map with the three characteristics mentioned above, namely the individuals are represented as convex bodies whose volumes are proportionals to a frequency distribution, which are located according to a dissimmilarity measure and which are spread out in the visualization region Ω, is written as a global biobjective optimization problem with convex constraints. We show that the objective function of the aggregate problem can be expressed as a DC function, and thus DC optimization tools can be used to solve the optimization program (Le Thi and Pham Dinh, 2005).

The rest of the chapter is organized as follows. In Section 4.2 the biobjective optimization program to build the visualization map is formalized. In Section 4.3, structural properties of the optimization problem are analyzed. In Section 4.4, we present our algorithmic approach. Numerical results for two datasets of different size and nature are included in Section 4.5. Finally, Section 4.6 closes the chapter with some conclusions.

4.2 The Mathematical Optimization model

Let us consider a set of individuals $V = \{v_1, \ldots, v_N\}$, which have attached a frequency distribution $\boldsymbol{\omega} \in \mathbb{R}^N_+$ and a dissimilarity matrix $\boldsymbol{\delta} = (\delta_{ij})_{\substack{i=1,\ldots,N \\ j=1,\ldots,N}}$. Let $\Omega \subset \mathbb{R}^n$ be a closed convex set and let $\mathcal{B} \subset \mathbb{R}^n$ be a convex body, namely a compact convex set, with nonempty interior, which is symmetric with respect to the origin (which belongs to $\mathcal{B}$), called reference object. Each individual v_i is associated with a set of the form $\boldsymbol{c}_i + \tau r_i \mathcal{B}$, where $r_i \geq 0$ is chosen so that the volume of $r_i\mathcal{B}$ is proportional to the frequency ω_i, $\boldsymbol{c}_i \in \mathbb{R}^n$ is a translation vector and τ is a common positive rescaling for all objects. We seek the values of the variables $\boldsymbol{c}_i$, $i = 1, \ldots, N$, and τ so that the convex bodies $\boldsymbol{c}_i + \tau r_i \mathcal{B}$ are contained in Ω. Figure 4.1 illustrates the previously described

representation.

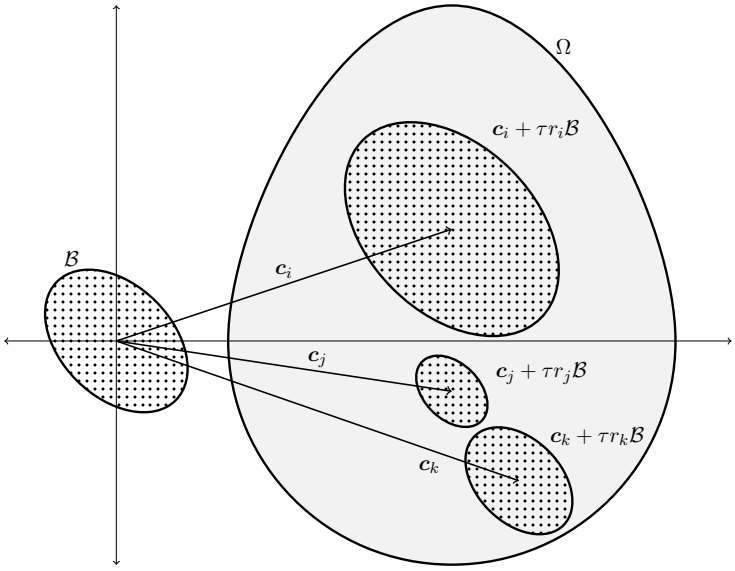


Figure 4.1: Example in $\mathbb{R}^2$ of a visualization region Ω, a reference object $\mathcal{B}$ and three individuals v_i, v_j and v_k defined through the translation vectors $\boldsymbol{c}_i$, $\boldsymbol{c}_j$ and $\boldsymbol{c}_k$, which are scaled via τr_i, τr_j and τr_k.

Hereafter, we deal with a biobjective optimization problem: the distances between the convex bodies representing the individuals v_i and v_j must resemble the dissimilarities δ_{ij} between such individuals, and these objects must be spread out in Ω to make the visualization easier. The two criteria are formalized in what follows.

Regarding the first objective, a function d, which gives us a strictly positive distance between two non-intersecting objects representing individuals v_i and v_j and zero otherwise, needs to be considered. Thus, we define the function g_{ij}, which assigns such distance to two individuals v_i and v_j, as follows

$$\begin{array}{rccl} g_{ij}: & \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ & (\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) & \longmapsto & d(\boldsymbol{c}_i + \tau r_i \mathcal{B}, \boldsymbol{c}_j + \tau r_j \mathcal{B}). \end{array} \tag{4.1}$$

Then, to quantify the resemblance between the distances in the visualization map and the dissimilarities, the summation over all the individuals of the squared differences between the distances and the rescaled dissimilarities through a positive variable κ needs to be minimized. Thus, we consider as first objective the function F_1 defined as

$$\begin{array}{rccl} F_1: & \mathbb{R}^n \times \ldots \times \mathbb{R}^n \times \mathbb{R}^+ \times \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ & (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N, \tau, \kappa) & \longmapsto & \displaystyle\sum_{\substack{i,j=1,\ldots,N \\ i \neq j}} \left[g_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) - \kappa \delta_{ij} \right]^2. \end{array}$$

Observe that for simplicity all pairs (i, j) are considered in the summation in F_1, but our analysis remains valid if only some pairs (i, j) are taken into consideration, as done e.g. by Trosset (2002).

To avoid that the convex bodies representing the individuals collapse in a small subregion of Ω, we encourage objects to be spread out all over Ω. There are several ways to model spread. For instance, we could use the overall volume covered by the objects, the amount of intersections between them, or the distances between the objects. This last option is the one analyzed in detail in this chapter, and therefore, our aim is to maximize the sum over all the individuals of the distances between the objects representing them. Let F_2 be a function which, given the translation vectors, $\boldsymbol{c}_i$, and the rescaling parameter, τ, computes the spread of the visualization map in such way. Then, written in minimization form, one has

$$\begin{array}{rccl} F_2: & \mathbb{R}^n \times \ldots \times \mathbb{R}^n \times \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ & (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N, \tau) & \longmapsto & -\sum\limits_{\substack{i,j=1,\ldots,N \\ i \neq j}} g_{ij}^2(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau). \end{array} \tag{4.2}$$

Note that F_2 does not distinguish between how much the objects intersect, since it penalizes in the same way two convex bodies one on top of the other and two tangent objects. A possible way to quantify the amount of intersection between two objects is by measuring the minimum-norm translation of such objects which makes them not to intersect. This leads to the concept of penetration depth (Elkeran, 2013; Ong and Gilbert, 1996; Umetani et al., 2009).

Let $\|\cdot\|$ be a norm, A_1 and A_2 two convex compact sets with nonempty interior, and $\boldsymbol{p}$ a direction in $\mathbb{R}^n$. Let $int(\cdot)$ be the interior of a set and '$+$' the translator operator. The penetration depth of A_1 and A_2 is defined as the minimum norm vector $\boldsymbol{p} \in \mathbb{R}^n$ such that A_1 translated through the direction $\boldsymbol{p}$, namely $\boldsymbol{p} + A_1$, is disjoint with $int(A_2)$, i.e.

$$\pi(A_1, A_2) = \min_{\boldsymbol{p} \in \mathbb{R}^n} \{\|\boldsymbol{p}\| : (\boldsymbol{p} + A_1) \cap int(A_2) = \emptyset\}.$$

Computing the penetration depth between two sets is costly, in general. Nevertheless, exact and heuristic algorithms exist for specific types of convex bodies such as discs and convex polytopes in $\mathbb{R}^2$ and $\mathbb{R}^3$ (Cameron and Culley, 1986; Lin and Manocha, 2004).

The amount of intersection between the objects in the visualization map can be quantified as the sum over all the individuals of the squared penetration depth between

pairs of them, yielding the function F_2^{Π} defined as

$$\begin{array}{rccl} F_2^{\Pi}: & \mathbb{R}^n \times \ldots \times \mathbb{R}^n \times \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ & (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N, \tau) & \longmapsto & \displaystyle\sum_{\substack{i,j=1,\ldots,N \\ i \neq j}} \pi^2 \left(\boldsymbol{c}_i + \tau r_i \mathcal{B}, \boldsymbol{c}_j + \tau r_j \mathcal{B}\right). \end{array} \tag{4.3}$$

However, the penetration depth does not measure how separated the convex bodies are. Then, an alternative to the two previous spread criteria, namely F_2 and F_2^{Π}, which does take into account both the amount of intersection and the separation of the objects, consists of measuring the distance between the centers of the objects. Maximizing the sum over all the individuals of the squared distances between the centers gives an alternative spread criterion, namely

$$\begin{array}{rccl} F_2^{\boldsymbol{c}}: & \mathbb{R}^n \times \ldots \times \mathbb{R}^n \times \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ & (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N, \tau) & \longmapsto & -\displaystyle\sum_{\substack{i,j=1,\ldots,N \\ i \neq j}} \|\boldsymbol{c}_i - \boldsymbol{c}_j\|^2. \end{array} \tag{4.4}$$

Thus, the problem of building a visualization map in which a set of convex bodies in the form $\boldsymbol{c}_i + \tau r_i \mathcal{B}$ are represented in the region Ω, satisfying that the distances between the objects resemble the dissimilarities between the individuals and the map is spread enough, can be stated as a biobjective optimization problem. By proceeding in the usual way, we consider the convex combination of the objectives and solve the aggregate problem, see Ehrgott (2006). Thus, given $\lambda \in [0, 1]$, the Visualization Map problem, $(VM)^*$, is stated as follows

$$\begin{array}{ll} \displaystyle\min_{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_N,\tau,\kappa} & \lambda F_1(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N, \tau, \kappa) + (1-\lambda) F_2^*(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N, \tau) \\ \text{s.t.} & \boldsymbol{c}_i + \tau r_i \mathcal{B} \subseteq \Omega, \ i = 1, \ldots, N \\ & \tau \in T \\ & \kappa \in K, \end{array} \tag{VM)*}$$

where $K, T \subset \mathbb{R}^+$ and F_2^* refers to either F_2, F_2^{Π} or $F_2^{\boldsymbol{c}}$ stated in (4.2), (4.3) and (4.4), yielding the models (VM), $(VM)^{\Pi}$ or $(VM)^{\boldsymbol{c}}$ respectively.

4.3 Properties

In this section we study the structure of Problem $(VM)^*$ for the three different choices proposed as the spread criterion, namely the three possibilities for F_2^* given in (4.2), (4.3) and (4.4). For each choice, we will prove that their objective functions are DC, by considering distance functions d, defined in the space of compact convex sets of

$\mathbb{R}^n$, which satisfy the following:

Assumption 1. *The function d, defined on pairs of compact convex sets of $\mathbb{R}^n$, satisfies for any A_1 and A_2*

(i) $d \geq 0$ *and* d *is symmetric*

(ii) $d(A_1, A_2) = d(A_1 + z, A_2 + z)$, $\forall z \in \mathbb{R}^n$

(iii) The function $d_z : z \in \mathbb{R}^n \longmapsto d(z + A_1, A_2)$ *is convex and satisfies for all* $\theta > 0$ *that* $d_z(\theta A_1, \theta A_2) = \theta d_{\frac{1}{\theta}z}(A_1, A_2)$.

Typical instances of d satisfying (i)-(iii) are

1. The infimum distance, defined as

$$d(A_1, A_2) = \inf\{\|a_1 - a_2\| : a_1 \in A_1, a_2 \in A_2\} \tag{d1}$$

2. The supremum distance, defined as

$$d(A_1, A_2) = \sup\{\|a_1 - a_2\| : a_1 \in A_1, a_2 \in A_2\} \tag{d2}$$

3. The average distance, defined as

$$d(A_1, A_2) = \int \|a_1 - a_2\| d\mu(a_1) d\nu(a_2), \tag{d3}$$

4. The Hausdorff distance, defined as

$$d(A_1, A_2) = \max\left\{ \sup_{a_1 \in A_1} \inf_{a_2 \in A_2} \|a_1 - a_2\|, \sup_{a_2 \in A_2} \inf_{a_1 \in A_1} \|a_1 - a_2\| \right\}, \tag{d4}$$

where μ, ν are probability distributions in $\mathbb{R}^n$ with support A_1 and A_2 (Carrizosa et al., 1995, 1998a,b; Koshizuka and Kurita, 1991).

Observe that, thanks to Assumption 1, the distance between two convex bodies representing individuals v_i and v_j, given by the function g_{ij} in (4.1), can be expressed as

$$g_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) = \tau d_{\frac{1}{\tau}(\boldsymbol{c}_i - \boldsymbol{c}_j)}(r_i\mathcal{B}, r_j\mathcal{B}), \tag{4.5}$$

and thus g_{ij} is the perspective of the convex function $f_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j) = d_{\boldsymbol{c}_i - \boldsymbol{c}_j}(r_i\mathcal{B}, r_j\mathcal{B})$. Hence, g_{ij} is convex as well (Hiriart-Urruty and Lemaréchal, 1993).

Elementary tools of DC optimization enable us to show that objective function in (VM), namely $\lambda F_1 + (1 - \lambda)F_2$, is DC, and a DC decomposition can be given. The result is presented in Proposition 4.1.

Proposition 4.1. *The function $\lambda F_1 + (1-\lambda)F_2$ is DC, $\lambda \in [0,1]$, and a decomposition is given by*

$$\lambda F_1 + (1-\lambda)F_2 = u - (u - \lambda F_1 - (1-\lambda)F_2)\,,$$

where

$$u = \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \left\{\max\{3\lambda - 1, 0\} g_{ij}^2(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) + 2\lambda(\kappa\delta_{ij})^2\right\}.$$

Proof.

$$\begin{aligned}
&\lambda F_1 + (1-\lambda)F_2 = \\
&\quad = \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \left\{\lambda\left[g_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) - \kappa\delta_{ij}\right]^2 - (1-\lambda) g_{ij}^2(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau)\right\} \\
&\quad = \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \left\{(3\lambda - 1) g_{ij}^2(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) + 2\lambda\kappa^2\delta_{ij}^2 - \lambda(g_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) + \kappa\delta_{ij})^2\right\}
\end{aligned}$$

Thanks to the convexity of the function g_{ij} and, since g_{ij}, λ, $\delta_{ij} \geq 0$, then $g_{ij}^2(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau)$, $2\lambda\kappa^2\delta_{ij}^2$ and $(g_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) + \kappa\delta_{ij})^2$ are convex. Finally, $(3\lambda - 1) g_{ij}^2(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau)$ is convex for $3\lambda - 1 \geq 0$ and concave otherwise. □

In the same vein, we can prove that the objective functions in $(VM)^{\Pi}$ and $(VM)^{\mathbf{c}}$ are DC as well, as stated in the following results.

Proposition 4.2. *Let h_{ij} be defined as the penetration depth between $\boldsymbol{c}_i + \tau r_i\mathcal{B}$ and $\boldsymbol{c}_j + \tau r_j\mathcal{B}$, namely*

$$\begin{array}{rccl}
h_{ij}: & \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\
& (\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) & \longmapsto & \pi\left(\boldsymbol{c}_i + \tau r_i\mathcal{B}, \boldsymbol{c}_j + \tau r_j\mathcal{B}\right).
\end{array}$$

Denoting as $\sigma_{\mathcal{B}}$ the support function of $\mathcal{B}$, one has that h_{ij} is DC, and a decomposition is given by $h_{ij} = u_{ij} - (u_{ij} - h_{ij})$, where

$$u_{ij} \;=\; \max\left\{\max_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \left\{\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) - \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\}, 0\right\}.$$

Proof. For convex sets A_1 and A_2 with nonempty interior, the condition in the definition of penetration depth stated in Section 4.2 is equivalent to the existence of a separating hyperplane between the sets $\boldsymbol{p} + A_1$ and A_2, i.e., of some $\boldsymbol{\xi} \neq 0$, such that

$$\boldsymbol{\xi}^\top(\boldsymbol{p} + \boldsymbol{a}_1) \leq \boldsymbol{\xi}^\top\boldsymbol{a}_2 \qquad \forall \boldsymbol{a}_1 \in A_1, \boldsymbol{a}_2 \in A_2.$$

Without loss of generality, we can consider $\|\boldsymbol{\xi}\| = 1$ and thus we have

$$\begin{array}{rcll} \pi(A_1, A_2) & = & \min\limits_{\boldsymbol{p},\boldsymbol{\xi}\in\mathbb{R}^n} & \|\boldsymbol{p}\| \\ & & \text{s.t.} & \boldsymbol{\xi}^\top(\boldsymbol{p}+\boldsymbol{a}_1) \leq \boldsymbol{\xi}^\top \boldsymbol{a}_2 \qquad \forall \boldsymbol{a}_1 \in A_1, \boldsymbol{a}_2 \in A_2 \\ & & & \|\boldsymbol{\xi}\| = 1. \end{array}$$

Thus, h_{ij} can be written as follows

$$\begin{array}{rcll} h_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) & = & \min\limits_{\boldsymbol{p},\boldsymbol{\xi}\in\mathbb{R}^n} & \|\boldsymbol{p}\| \\ & & \text{s.t.} & \boldsymbol{\xi}^\top(\boldsymbol{p}+\boldsymbol{c}_i+\tau r_i \boldsymbol{x}_i) \leq \boldsymbol{\xi}^\top(\boldsymbol{c}_j+\tau r_j \boldsymbol{x}_j) \qquad \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{B} \\ & & & \|\boldsymbol{\xi}\| = 1. \end{array}$$

Equivalently, the first constraint, i.e.,

$$\boldsymbol{\xi}^\top(\boldsymbol{p}+\boldsymbol{c}_i+\tau r_i \boldsymbol{x}_i) \leq \boldsymbol{\xi}^\top(\boldsymbol{c}_j+\tau r_j \boldsymbol{x}_j) \quad \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{B},$$

can be written as follows,

$$\boldsymbol{\xi}^\top(\boldsymbol{p}+\boldsymbol{c}_i) + \tau r_i \max_{\boldsymbol{x}\in\mathcal{B}} \boldsymbol{\xi}^\top \boldsymbol{x} \leq \boldsymbol{\xi}^\top \boldsymbol{c}_j + \tau r_j \min_{\boldsymbol{x}\in\mathcal{B}} \boldsymbol{\xi}^\top \boldsymbol{x}.$$

Let $\sigma_{\mathcal{B}}$ be the support function of $\mathcal{B}$, i.e.,

$$\sigma_{\mathcal{B}}(z) = \max_y \{y^\top z : y \in \mathcal{B}\}$$

Since $\mathcal{B}$ is assumed to be symmetric with respect to the origin, we have

$$\begin{array}{rcl} \max\limits_{x\in\mathcal{B}} \boldsymbol{\xi}^\top x & = & \sigma_{\mathcal{B}}(\boldsymbol{\xi}) \\ \min\limits_{x\in\mathcal{B}} \boldsymbol{\xi}^\top x & = & -\sigma_{\mathcal{B}}(\boldsymbol{\xi}). \end{array}$$

Hence, by replacing the expression of the support function in the constraint above, one has

$$\begin{array}{rcll} h_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) & = & \min\limits_{\boldsymbol{p},\boldsymbol{\xi}\in\mathbb{R}^n} & \|\boldsymbol{p}\| \\ & & \text{s.t.} & \boldsymbol{\xi}^\top \boldsymbol{p} \leq \boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) - \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi}) \\ & & & \|\boldsymbol{\xi}\| = 1. \end{array}$$

For $\boldsymbol{\xi}$ fixed with $\|\boldsymbol{\xi}\| = 1$, let $\eta(\boldsymbol{\xi}) = \boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) - \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})$. It follows that the inner minimum in $h_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau)$, is the distance from the origin to the halfspace $\boldsymbol{\xi}^\top \boldsymbol{p} \leq \eta(\boldsymbol{\xi})$, and such distance equals 0, if 0 belongs to the halfspace, i.e., if

$0 \leq \boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) - \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})$, and $-\eta(\boldsymbol{\xi})$ else. Hence

$$\begin{aligned} h_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) &= \min_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \max\left\{0, -\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) + \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\} \\ &= \max\left\{0, \min_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \left\{-\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) + \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\}\right\} \end{aligned}$$

But, for $\boldsymbol{\xi}$ fixed, the function $(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) \longmapsto -\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) + \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})$ is affine, and thus the function $(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) \longmapsto \min_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \left\{-\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) + \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\}$ is the minimum of affine functions, and is thus concave. Hence, h_{ij} is the maximum between 0 and a concave function, which is DC, whose decomposition is

$$\begin{aligned} &h_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) = \\ &\quad = \max\left\{0, \min_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \left\{-\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) + \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\}\right\} \\ &\quad = \max\left\{-\min_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \left\{-\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) + \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\}, 0\right\} \\ &\qquad + \min_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \left\{-\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) + \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\} \\ &\quad = \max\left\{\max_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \left\{\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) - \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\}, 0\right\} \\ &\qquad - \max_{\substack{\boldsymbol{\xi}\in\mathbb{R}^n \\ \|\boldsymbol{\xi}\|=1}} \left\{\boldsymbol{\xi}^\top(\boldsymbol{c}_j - \boldsymbol{c}_i) - \tau(r_i + r_j)\sigma_{\mathcal{B}}(\boldsymbol{\xi})\right\} \\ &\quad = u_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) - (u_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) - h_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau)). \end{aligned}$$

□

Corollary 4.1. *The function $\lambda F_1 + (1-\lambda)F_2^{\Pi}$ is DC, $\lambda \in [0, 1]$.*

Proof. The function F_1 is DC. Indeed, it is sufficient to take $\lambda = 1$ in Proposition 4.1. F_2^{Π} is also DC by using Proposition 4.2 and Proposition 3.7 in Tuy (1998). Then, since the summation of DC functions is also DC, the result holds. □

Corollary 4.2. *The function $\lambda F_1 + (1-\lambda)F_2^{\boldsymbol{c}}$ is DC, $\lambda \in [0, 1]$.*

Proof. Since the function F_1 is DC (take $\lambda = 1$ in Proposition 4.1) and $F_2^{\boldsymbol{c}}$ is concave, since it is minus the summation of squares of a nonnegative convex function, the result holds. □

Therefore, DC decompositions for objective functions in $(VM)^{\Pi}$ and $(VM)^{\mathbf{c}}$ are readily available from the DC decomposition of F_1 in Proposition 4.1 ($\lambda = 1$), Proposition 4.2 and the concavity of $F_2^{\mathbf{c}}$.

Showing that a function is DC and giving explicitly a DC decomposition enables us to use DC optimization algorithms. It is well known that the performance of the procedures may strongly depend on the choice of the DC decomposition (Blanquero and Carrizosa, 2009; Bomze et al., 2008; Ferrer and Martínez-Legaz, 2009). Particularly, we seek a DC decomposition involving a quadratic convex separable function as those addressed by Le Thi (2000) and Pham Dinh and Le Thi (1998) for the special case where (*d1*) is used. We will show in Section 4.4 that such alternative decomposition yields a simple DC Algorithm (DCA) first stated in Pham Dinh and El Bernoussi (1986, 1988), whose convergence follows from the general convergence results of DCA (Le Thi and Pham Dinh, 2013, 2005; Pham Dinh and Le Thi, 1997).

The following result, Proposition 4.3, states a different DC decomposition for the objective function in (VM) from the one given in Proposition 4.1, when the infimum distance given in (*d1*) is considered. In fact, this alternative decomposition involves a quadratic convex separable function.

Before giving the proof of Proposition 4.3, the following technical result is needed.

Lemma 4.1. *Let $\beta_{ij} \in \mathbb{R}$ be such that $\beta_{ij} \geq 2\|r_i\mathbf{b}_i - r_j\mathbf{b}_j\|^2$, $\forall \mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}$. Then, g_{ij}^2 can be expressed as a DC function, $g_{ij}^2 = u_{ij} - (u_{ij} - g_{ij}^2)$, where*

$$u_{ij}(\mathbf{c}_i, \mathbf{c}_j, \tau) \quad = \quad 2\|\mathbf{c}_i - \mathbf{c}_j\|^2 + \beta_{ij}\tau^2.$$

Proof.

$$\begin{aligned}
&g_{ij}^2(\mathbf{c}_i, \mathbf{c}_j, \tau) = \\
&\quad = \min_{\mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}} \|\mathbf{c}_i - \mathbf{c}_j + \tau(r_i\mathbf{b}_i - r_j\mathbf{b}_j)\|^2 \\
&\quad = \min_{\mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}} \left\{ \|\mathbf{c}_i - \mathbf{c}_j\|^2 + \tau^2\|r_i\mathbf{b}_i - r_j\mathbf{b}_j\|^2 + 2\tau(\mathbf{c}_i - \mathbf{c}_j)^\top(r_i\mathbf{b}_i - r_j\mathbf{b}_j) \right\} \\
&\quad = \min_{\mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}} \{ \|\mathbf{c}_i - \mathbf{c}_j\|^2 + \tau^2\|r_i\mathbf{b}_i - r_j\mathbf{b}_j\|^2 \\
&\qquad\quad + \|\mathbf{c}_i - \mathbf{c}_j\|^2 + \tau^2\|r_i\mathbf{b}_i - r_j\mathbf{b}_j\|^2 - \|\mathbf{c}_i - \mathbf{c}_j - \tau(r_i\mathbf{b}_i - r_j\mathbf{b}_j)\|^2 \} \\
&\quad = 2\|\mathbf{c}_i - \mathbf{c}_j\|^2 + \beta_{ij}\tau^2 + \min_{\mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}} \{ -\beta_{ij}\tau^2 + 2\tau^2\|r_i\mathbf{b}_i - r_j\mathbf{b}_j\|^2 - \|\mathbf{c}_i - \mathbf{c}_j - \tau\left(r_i\mathbf{b}_i - r_j\mathbf{b}_j\right)\|^2 \} \\
&\quad = 2\|\mathbf{c}_i - \mathbf{c}_j\|^2 + \beta_{ij}\tau^2 + \min_{\mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}} \{ \tau^2\left(2\|r_i\mathbf{b}_i - r_j\mathbf{b}_j\|^2 - \beta_{ij}\right) - \|\mathbf{c}_i - \mathbf{c}_j - \tau\left(r_i\mathbf{b}_i - r_j\mathbf{b}_j\right)\|^2 \} \\
&\quad = 2\|\mathbf{c}_i - \mathbf{c}_j\|^2 + \beta_{ij}\tau^2 - \max_{\mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}} \{ \|\mathbf{c}_i - \mathbf{c}_j - \tau\left(r_i\mathbf{b}_i - r_j\mathbf{b}_j\right)\|^2 - \tau^2\left(2\|r_i\mathbf{b}_i - r_j\mathbf{b}_j\|^2 - \beta_{ij}\right) \}
\end{aligned}$$

Observe that taking $\beta_{ij} \in \mathbb{R}$ such that

$$2\|r_i\boldsymbol{b}_i - r_j\boldsymbol{b}_j\|^2 - \beta_{ij} \leq 0 \qquad \forall \boldsymbol{b}_i, \boldsymbol{b}_j \in \mathcal{B},$$

the function

$$(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) \longmapsto \|\boldsymbol{c}_i - \boldsymbol{c}_j - \tau\left(r_i\boldsymbol{b}_i - r_j\boldsymbol{b}_j\right)\|^2 - \tau^2\left(2\|r_i\boldsymbol{b}_i - r_j\boldsymbol{b}_j\|^2 - \beta_{ij}\right)$$

is convex. Since the maximum of convex functions is convex, we have obtained a DC decomposition for g_{ij}^2 as in the statement taking $u_{ij} = 2\|\boldsymbol{c}_i - \boldsymbol{c}_j\|^2 + \beta_{ij}\tau^2$. □

Proposition 4.3. *The function $\lambda F_1 + (1-\lambda)F_2$, where d is the infimum distance (d1), can be expressed as a DC function, $\lambda F_1 + (1-\lambda)F_2 = u - (u - \lambda F_1 - (1-\lambda)F_2)$, where the quadratic separable convex function u is given by*

$$u = \max\{3\lambda - 1, 0\} \cdot \left[\sum_{i=1,\dots,N} \left\{8(N-1)\|\boldsymbol{c}_i\|^2\right\} + \tau^2 \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \beta_{ij}\right] + 2\lambda\kappa^2 \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \delta_{ij}^2,$$

where β_{ij} satisfies $\beta_{ij} \geq 2\|r_ib_i - r_jb_j\|^2$ for all $b_i, b_j \in \mathcal{B}$.

Proof. If $\lambda < \dfrac{1}{3}$, considering Proposition 4.1, one has

$$\lambda F_1 + (1-\lambda)F_2 = \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \left\{2\lambda\kappa^2\delta_{ij}^2 - \left[\lambda(g_{ij} + \kappa\delta_{ij})^2 - (3\lambda - 1)g_{ij}^2(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau)\right]\right\},$$

and thus $u = \displaystyle\sum_{\substack{i,j=1,\dots,N \\ i\neq j}} 2\lambda\kappa^2\delta_{ij}^2$ holds.

If $\lambda \geq \dfrac{1}{3}$, by using the DC decomposition for g_{ij}^2 obtained in Lemma 4.1 and Proposition 4.1, one has

$$\begin{aligned}
&\lambda F_1 + (1-\lambda)F_2 = \\
&\quad = \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \left\{(3\lambda - 1)g_{ij}^2(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) + 2\lambda\kappa^2\delta_{ij}^2 - \lambda(g_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) + \kappa\delta_{ij})^2\right\} \\
&\quad = \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \Big\{2(3\lambda - 1)\|\boldsymbol{c}_i - \boldsymbol{c}_j\|^2 + (3\lambda - 1)\beta_{ij}\tau^2 + 2\lambda\kappa^2\delta_{ij}^2 - \big[\lambda(g_{ij}(\boldsymbol{c}_i, \boldsymbol{c}_j, \tau) + \kappa\delta_{ij})^2 \\
&\qquad\qquad + (3\lambda - 1)\max_{\boldsymbol{b}_i, \boldsymbol{b}_j \in \mathcal{B}} \left\{\|\boldsymbol{c}_i - \boldsymbol{c}_j - \tau\left(r_i\boldsymbol{b}_i - r_j\boldsymbol{b}_j\right)\|^2 - \tau^2\left(2\|r_i\boldsymbol{b}_i - r_j\boldsymbol{b}_j\|^2 - \beta_{ij}\right)\right\}\Big]\Big\}
\end{aligned}$$

$$= \sum_{i=1,\dots,N} \left\{8(3\lambda-1)(N-1)\|\boldsymbol{c}_i\|^2\right\} + (3\lambda-1)\tau^2 \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \beta_{ij} + 2\lambda\kappa^2 \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \delta_{ij}^2$$
$$- \sum_{\substack{i,j=1,\dots,N \\ i\neq j}} \Big[2(3\lambda-1)\|\boldsymbol{c}_i+\boldsymbol{c}_j\|^2 + \lambda(g_{ij}(\boldsymbol{c}_i,\boldsymbol{c}_j,\tau)+\kappa\delta_{ij})^2$$
$$+ (3\lambda-1) \max_{\boldsymbol{b}_i,\boldsymbol{b}_j\in\mathcal{B}} \left\{\|\boldsymbol{c}_i-\boldsymbol{c}_j-\tau\left(r_i\boldsymbol{b}_i-r_j\boldsymbol{b}_j\right)\|^2 - \tau^2\left(2\|r_i\boldsymbol{b}_i-r_j\boldsymbol{b}_j\|^2-\beta_{ij}\right)\right\}\Big]$$

□

Section 4.4 is mainly devoted to show how Problem (VM) with the decomposition given in Proposition 4.3 can be efficiently solved by DCA.

4.4 Algorithmic approach

Propositions 4.1-4.3 and Corollaries 4.1-4.2 show that $(VM)^*$ is an optimization problem with a DC objective function with a DC decomposition available and simple constraints. Then, DC optimization tools can be used, either of exact nature for very low dimensional problems (Blanquero and Carrizosa, 2009; Blanquero et al., 2009), or heuristics, as the DCA (Le Thi and Pham Dinh, 2013, 2005; Pham Dinh and Le Thi, 1997). The latter is the approach we are following in this chapter.

Roughly speaking, DCA consists of an iterative process in which a sequence of convex programs are solved. Given a DC program of the form $\min\{f(x) = u(x)-v(x) : x \in X\}$, where X is a convex feasible region, at each iteration, the concave part $(-v(x))$ is replaced by its affine majorization at a certain $x_0 \in X$, and the resulting convex problem is then solved. Let $\langle\cdot,\cdot\rangle$ be an inner product in $\mathbb{R}^n$, and let $\partial v(x_0)$ the subdifferential of v at x_0. A general scheme of DCA is outlined in Algorithm 4.1.

Algorithm 4.1 DCA scheme (Le Thi and Pham Dinh, 2005)

Input: $x_0 \in X$.

1: $t \leftarrow 0$
2: **repeat**
3: Compute some $y_t \in \partial v(x_t)$;
4: Compute $x_{t+1} \in \arg\min \{u(x) - (v(x_t) + \langle x - x_t, y_t\rangle) : x \in X\}$;
5: $t \leftarrow t + 1$;
6: **until** stop condition is met.

Output: x_t

However, running times would be dramatically reduced if a DC decomposition of the objective function were available so that the convex optimization problems to be solved in line 4 of Algorithm 4.1 were trivial, in the sense that an explicit expression

for the optimal solution is available. This idea has been studied by Le Thi (2000) and Pham Dinh and Le Thi (1998) and it will be customized to Problem (VM), considering the infimum distance given in *(d1)*, in what follows.

When the DCA scheme is applied to Problem (VM) with the DC decomposition given in Proposition 4.3, we see that the convex subproblems to be solved at line 4 of Algorithm 4.1 have the form

$$\begin{aligned} \min_{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_N,\tau,\kappa} \quad & \left\{ \sum_{i=1,\ldots,N} \left\{ M^{\boldsymbol{c}_i} \|\boldsymbol{c}_i\|^2 \right\} + M^{\kappa}\kappa^2 + M^{\tau}\tau^2 + \sum_{i=1,\ldots,N} \left\{ \boldsymbol{c}_i{}^{\top} \boldsymbol{q}^{\boldsymbol{c}_i} \right\} + p^{\kappa}\kappa + p^{\tau}\tau \right\} \\ \text{s.t.} \quad & \boldsymbol{c}_i + \tau r_i \mathcal{B} \subseteq \Omega, \ i = 1, \ldots, N \\ & \tau \in T \\ & \kappa \in K, \end{aligned}$$

for scalars $M^{\boldsymbol{c}_i}$, M^{κ}, $M^{\tau} \in \mathbb{R}^+$, which come from the coefficients that multiply each term in the u part of the DC decomposition given in Proposition 4.3, and vectors $\boldsymbol{q}^{\boldsymbol{c}_i} \in \mathbb{R}^n$ and scalars p^{κ} and $p^{\tau} \in \mathbb{R}$, which come for the computation of the subgradients at a given point of the $u - \lambda F_1 - (1-\lambda)F_2$ part.

Such problem can be written as a summation of two separate problems,

$$\min_{\kappa \in K} \ \left\{ M^{\kappa}\kappa^2 + p^{\kappa}\kappa \right\} + \min_{\substack{\boldsymbol{c}_i + \tau r_i \mathcal{B} \subseteq \Omega \\ \tau \in T}} \left\{ \sum_{i=1,\ldots,N} \left\{ M^{\boldsymbol{c}_i} \|\boldsymbol{c}_i\|^2 + \boldsymbol{c}_i{}^{\top} \boldsymbol{q}^{\boldsymbol{c}_i} \right\} + M^{\tau}\tau^2 + p^{\tau}\tau \right\}. \tag{4.6}$$

The first problem in (4.6) is a quadratic problem in one variable, for which a closed form can be given for its optimal value. The second problem in (4.6) is separable in the variables $\boldsymbol{c}_i$ if the linking variable τ were fixed at τ_0. For this reason, an alternating strategy seems to be plausible, in which one alternates the optimization of τ for $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N$ fixed (and this is a one-dimensional quadratic problem and thus a closed formula for the optimal solution is readily obtained), and then for τ fixed, the centers $\boldsymbol{c}_i$ are to be optimized. This is done by solving separately N optimization problems of the form

$$\begin{aligned} \min_{\boldsymbol{c}_i} \quad & \left\{ M^{\boldsymbol{c}_i} \|\boldsymbol{c}_i\|^2 + \boldsymbol{c}_i{}^{\top} \boldsymbol{q}^{\boldsymbol{c}_i} \right\} \\ \text{s.t.} \quad & \boldsymbol{c}_i \in \Omega - \tau r_i \mathcal{B}. \end{aligned} \tag{4.7}$$

In order to solve Problem (VM), we propose an alternating procedure which integrates a DCA strategy, to obtain $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N$ as stated in Algorithm 4.1, into an outer loop to get τ and κ. The alternating scheme to solve (VM) is stated in Algorithm 4.2: lines 3–8 contain the DCA as outlined in Algorithm 4.1 to find $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N$, which is embedded in a main loop to get κ and τ (lines 1–14). We point out that line 6 in

Algorithm 4.2 contains the convex optimization problems to be solved in each iteration of DCA (line 4 of Algorithm 4.1), and explicit expressions for the optimal solution of the optimization problems in lines 10 and 12 are known. Therefore, if the optimal solution of the optimization problems in line 6 of Algorithm 4.2 could be optimally computed without calling any external numerical optimization routine, then each iteration of the inner DCA in lines 3–8 would be computationally cheap.

Algorithm 4.2 Alternating scheme for (VM)

Input: $\boldsymbol{c}_1^0, \ldots, \boldsymbol{c}_N^0 \in \Omega$, $\kappa^0 \in K$, $\tau^0 \in T$.

1: $s \leftarrow 0$;
2: **repeat**
3: $t \leftarrow 0$;
4: **repeat**
5: Compute $M^{\boldsymbol{c}_i{}^t}$ and $\boldsymbol{q}^{\boldsymbol{c}_i{}^t}$, $i = 1, \ldots, N$;
6: Compute $\boldsymbol{c}_1^{t+1}, \ldots, \boldsymbol{c}_N^{t+1}$ by solving (4.7) for τ fixed at τ^s;
7: $t \leftarrow t + 1$;
8: **until** stop condition is met.
9: Compute M^{κ^s} and p^{κ^s};
10: Compute κ^{s+1} by solving the first optimization problem in (4.6);
11: Compute M^{τ^s} and p^{τ^s};
12: Compute τ^{s+1} by solving the second optimization problem in (4.6) for $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N$ fixed
at $\boldsymbol{c}_1^t, \ldots, \boldsymbol{c}_N^t$;
13: $s \leftarrow s + 1$;
14: **until** stop condition is met.

Output: $\boldsymbol{c}_1^t, \ldots, \boldsymbol{c}_N^t, \kappa^t, \tau^s$

Two particular cases of (4.7) have an amenable structure, yielding a closed formula for the optimal solution, and thus avoiding any call to external numerical optimization routines in line 6 of Algorithm 4.2. Indeed, suppose Ω is a rectangle, for simplicity taken as $[0,1]^n$, τ is fixed to a real positive value τ_0, and $\mathcal{B}$ is the disc centered at the origin with radius r_0. Then, the constraint in (4.7) can be rewritten as

$$\tau_0 r_0 r_i \leq c_{ij} \leq 1 - \tau_0 r_0 r_i, \ j = 1, \ldots, n,$$

and thus (4.7) is expressed as

$$\sum_{j=1,\ldots,n} \min_{c_{ij}} \left\{ M^{\boldsymbol{c}_i} c_{ij}^2 + \boldsymbol{q}_j^{\boldsymbol{c}_i} c_{ij} : \ \tau_0 r_0 r_i \leq c_{ij} \leq 1 - \tau_0 r_0 r_i \right\} \tag{4.8}$$

In other words, (4.7) is decomposed into n one dimensional quadratic problems on

an interval, and thus a closed formula is readily obtained for the optimal solution of each problem of the form (4.8), and thus also for (4.7).

Similarly, suppose Ω and $\mathcal{B}$ are discs centered at the origin of radius 1 and r_0 respectively. Then, (4.7) is rewritten as

$$\begin{array}{ll} \min\limits_{\boldsymbol{c}_i} & \left\{ M^{\boldsymbol{c}_i} \|\boldsymbol{c}_i\|^2 + \boldsymbol{c}_i{}^\top \boldsymbol{q}^{\boldsymbol{c}_i} \right\} \\ \text{s.t.} & \|\boldsymbol{c}_i\| \leq 1 - \tau_0 r_0 r_i. \end{array} \tag{4.9}$$

Karush-Kuhn-Tucker conditions immediately yield an expression for the optimal solution of (4.9).

Summarizing, while the alternating strategy stated in Algorithm 4.2, which contains a DCA scheme, could be applied to solve $(VM)^*$ for an arbitrary DC decomposition of the objective function, we see that the DC decomposition given in Proposition 4.3 for (VM) is particularly attractive. We have shown that some convenient choices of Ω (a rectangle or a disc) and $\mathcal{B}$ (a disc) yield a closed formula for the optimal solution of the subproblems to be addressed at each stage of the inner DCA, thus avoiding the need of using numerical optimization routines. Le Thi (2000) successfully applied this strategy for other problems.

4.5 Computational experience

The methodology proposed in Section 4.4 is illustrated using two real-world datasets of diverse nature, for which both the frequency distribution and the dissimilarities are readily available in the literature. In particular, the dissimilarity measures come from a correlation matrix and a shortest paths matrix in a directed graph, whereas the frequencies represent a proportion (continuous variable) and the outdegree of a set of nodes (discrete variable), respectively.

4.5.1 Datasets

The first dataset consists of $N = 11$ financial markets across Europe and Asia. The frequency distribution $\boldsymbol{\omega}$ relates to the importance of each market relative to the world market portfolio (Flavin et al., 2002), and the dissimilarity $\boldsymbol{\delta}$ is based on the correlation between pairs of markets (Borg and Groenen, 2005). The second dataset is a social network of $N = 200$ musicians, modeled as a graph, where there is an arc connecting two nodes if one musician was influential on the other (Dörk et al., 2012). The frequency distribution $\boldsymbol{\omega}$ represents the outdegree of each node and the dissimilarity between pairs of musicians is based on the shortest distance from pairs of nodes (Dörk et al., 2012).

4.5.2 Experiments details

Algorithm 4.2 has been coded in C and the experiments have been carried out in a Windows 8.1 PC Intel® Core™ i7-4500U, 16GB of RAM. We set $\lambda = 0.9$ and $\mathcal{B}$ equal to the circle centered at $(0, 0)$ with radius equal to one. Since (VM) is a multimodal problem and the DCA may get stuck at a local optimum, 100 runs of a multistart are executed. Initial values for $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N$ are uniformly generated in Ω, whereas the initial values for κ and τ are chosen as the midpoint of intervals K and T, respectively. At each run of the multistart procedure, index s in Algorithm 4.2 takes a maximum value of 3 and index t a maximum value of 50.

4.5.3 Results

Figure 4.2 plots the financial markets dataset on the visualization region $\Omega = [0, 1] \times [0, 1]$, with the scaling parameters ranging in the intervals $K = T = [0.4, 0.6]$. Observe that, the European markets are clustered above the Asian ones, covering the upper half rectangle. These two clusters are represented with different colours. Figure 4.3 plots the musicians' social network taking a circular visualization region, namely $\Omega = \mathcal{B}$, with the scaling parameters ranging in the intervals $K = [0.075, 0.100]$ and $T = [0.015, 0.030]$, respectively. In the plot at the top, we find all musicians. In the plot at the bottom, we have highlighted one of the most influential nodes, the Rolling Stones, and the connected nodes: musicians influencing the Rolling Stones (respectively, those influenced by them) can be found in a lighter (respectively darker) colour.

4.6 Conclusions

In this chapter we have addressed the problem of representing, in a visualization region Ω, a set of individuals by means of convex bodies so that the distance between the objects fits as close as possible a given dissimilarity matrix, the volume of the objects represents a frequency distribution, and, at the same time, the spread of the objects within Ω is maximized.

The problem has been formulated as a DC optimization problem, and the powerful heuristic DCA has been proposed as solution approach. For particular choices of the visualization region Ω (a rectangle and a disc), the reference object (a disc) and the function d (the infimum distance), closed formulas for the optimal solutions of the DCA subproblems are obtained, thus avoiding the need to use numerical optimization routines. The examples presented demonstrate the usefulness of our approach.

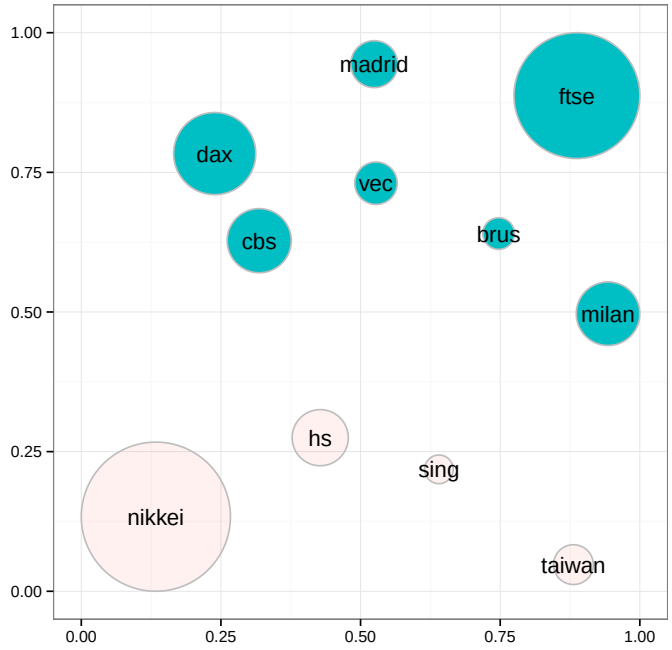


Figure 4.2: Visualizing financial markets.

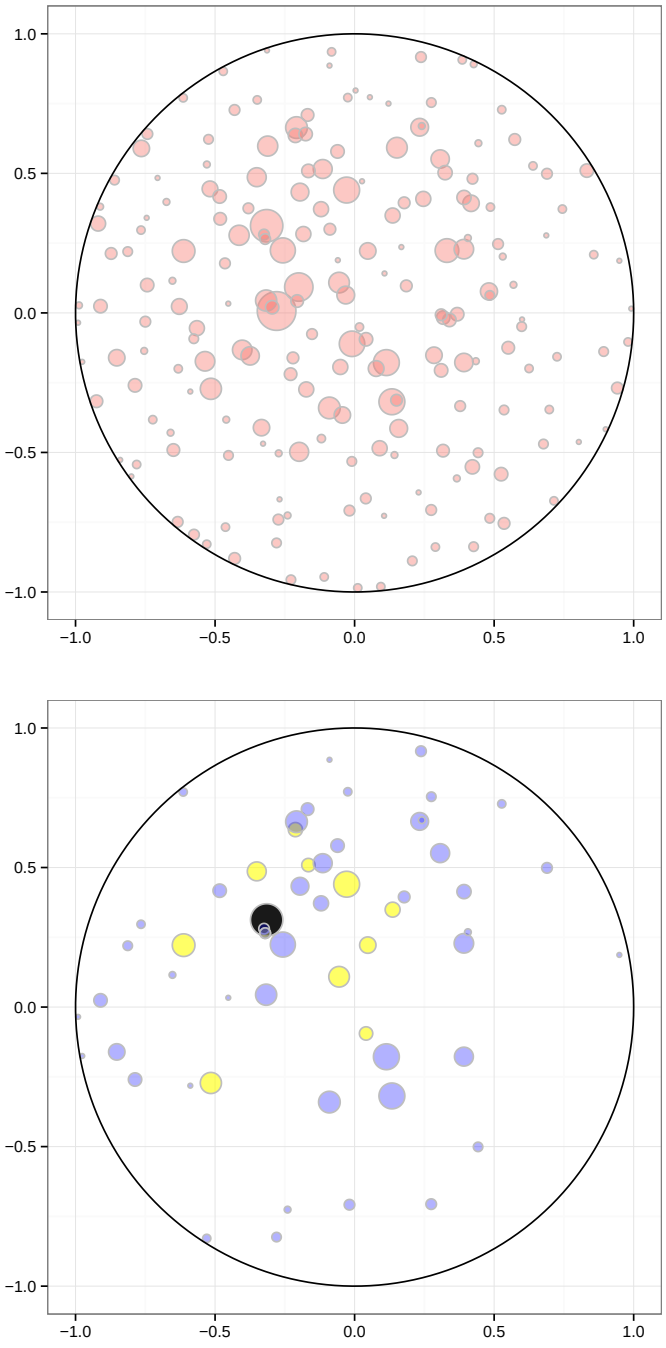


Figure 4.3: Visualizing the musicians' social network.

Chapter 5

Visualizing dynamic frequencies and dissimilarities as geometric objects: A Mixed Integer Nonlinear Programming approach

In this chapter the visualization model developed in Chapter 4 is extended to the time-varying case, namely we develop a new framework to visualize datasets which are made up of individuals observed along different time periods. These individuals have attached a time-dependent frequency distribution and a dissimilarity measure, which may vary over time as well. A Mathematical Optimization model is proposed and solved by means of difference of convex optimization techniques and nonconvex quadratic binary optimization. This way a visualization framework is obtained, the so-called Dynamic Visualization Map, which faithfully represents the dynamic frequencies by means of the areas of convex bodies, whereas it trades off the correct representation of the dissimilarities as the distances between those objects, their spread in the visual region and the preservation of the mental map. Our procedure is successfully tested on dynamic geographic and linguistic datasets.

5.1 Introduction

This chapter deals with datasets which consist of N individuals, $V = \{v_1, \ldots, v_N\}$, for which a distribution of frequencies has been historically observed during T time periods, and a measure of proximity between individuals, given as a time-dependent dissimiliarity matrix, is also given.

Visualizing dissimilarities has been historically done by Multidimensional Scaling (MDS) (Kruskal, 1964; Torgerson, 1958). As discussed in Chapter 1, a straightforward approach to visualize dynamic multivariate datasets, observed along T time periods, would consist of executing T independent MDS, one per period. Nevertheless, this approach might yield difficult-to-interpret visualizations (Groenen and Franses, 2000; Xu et al., 2013) especially when the dissimilarities change abruptly in consecutive periods or, since MDS results are invariant under rotations and reflections, and thus the snapshots may turn upside-down, as shown in Figure 1.5. This drawback calls for the construction of visualization frameworks which preserve the *mental map* (Misue et al., 1995), i.e., the transitions in the layouts in two consecutive time periods should be smooth, in the sense that the individuals do not suffer big displacements in the shift from one period to the next one.

On top of the challenge posed by the fact that data are time-varying, we also want to visualize the frequencies attached at each time period to each individual. To do this, following the proportional symbol map approach (Cabello et al., 2010), a first approach might consist of executing an MDS, and then replacing points by symmetric objects, say, discs or rectangles, centered at the MDS points, and whose area is proportional to the corresponding frequency. Nevertheless, the scale chosen by the user to (proportionally) depict such information may yield either too small objects or excess of overlapping between them, (Kunigami et al., 2014). Indeed, the changes in the

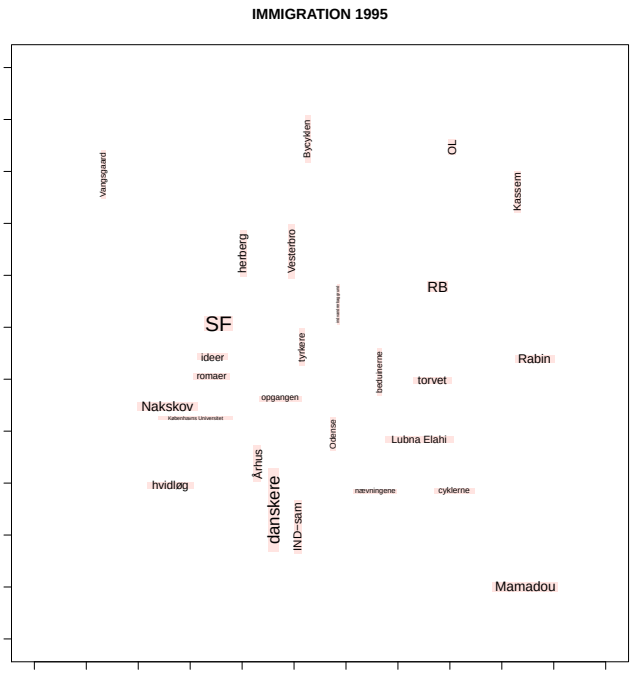


Figure 5.1: Visualization of Danish words by means of different rectangles, scaled according to their frequencies, and distances representing words co-occurrence as stated in Chapter 4.

perception of distances, induced by these (a posteriori) depicted objects, might yield misleading conclusions about the proximity between the individuals. In Chapter 4, we developed a generalization of MDS to visualize non-dynamic data, which simultaneously incorporates the information about dissimilarities and a frequency distribution by (a priori) deciding which convex body (disc, rectangle, etc.) represents each individual. See e.g. Figure 5.1 for an example in which N = 27 Danish words are depicted as rectangles. Here weights represent the importance of such words in the Danish news in 1995 and dissimilarities measure their co-occurrence.

This work presents a new Mathematical Optimization model to visualize dynamic datasets involving a frequency distribution and dissimilarities. We design a one-stage procedure, which involves a non-trivial generalization of the approach presented in Chapter 4. This new visualization framework simultaneously builds a collection of T snapshots, each representing a time period, in which the individuals under consideration are depicted as convex bodies located in a visualization region Ω, whose areas represent the frequencies and the distances between the objects depict the dissimilarities. The novelty of the presented model is twofold. First, the preservation of the mental map is incorporated into the optimization model by pursuing smooth transitions between two

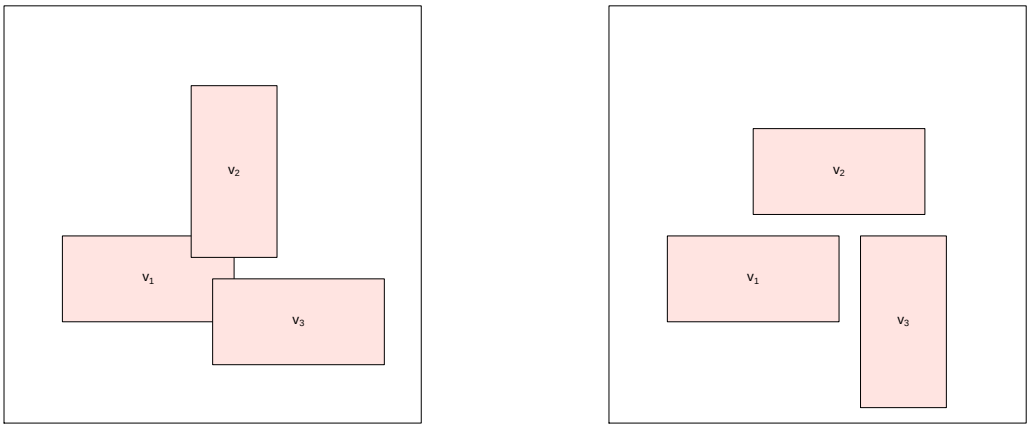


Figure 5.2: Importance of the choice of the convex bodies depicting each individual.

consecutive snapshots. Second, we assume that each individual has attached a catalogue of candidate convex bodies to be depicted and the choice of the object becomes a decision of the model, making it more flexible. For instance, each individual in Figure 5.2 could be depicted by means of two different rectangles: one whose basis has two units and one unit of height, and its 90 degrees rotation. Whereas the rectangles appear collapsed in Figure 5.2 (left), a different choice from the catalogue makes the visualization clearer: the same individuals are shown, but different rectangles have been chosen for v2 and v3, Figure 5.2 (right) .

The optimization model can be formulated as a Mixed Integer Nonlinear Problem (MINLP), which is handled through an alternating procedure. This procedure combines the use of Difference of Convex (DC) optimization techniques, such as the Difference of Convex Algorithm (DCA) (Pham Dinh and El Bernoussi, 1986, 1988; Le Thi and Pham Dinh, 2013, 2005; Pham Dinh and Le Thi, 1997) and nonconvex quadratic binary optimization. Our approach is clearly different from existing techniques in the literature, mostly ad-hoc multi-stage procedures, which often depend on user's manual tuning or exploit the nature of the data, and thus cannot be used for arbitrary datasets. Some examples are found in Graph Drawing (Battista et al., 1999; Beck et al., 2016; Lin et al., 2011; Van Vlasselaer et al., 2016; Xu et al., 2013), or geographical applications (Andrienko et al., 2003; Mashima et al., 2012) and others (Aigner et al., 2011; Gomez-Nieto et al., 2016).

The remainder of this chapter is organized as follows. Section 5.2 is devoted to present the model to visualize the dynamic complex dataset under study. In Section 5.3, we present a solution approach based on DC optimization tools and nonconvex

quadratic binary optimization. Some computational tests, involving data of different nature, are included in Section 5.4. Section 5.5 contains some conclusions.

5.2 The Mathematical Optimization model

In what follows, the problem of visualizing a dynamic dataset by means of convex bodies is formally stated and written as a mathematical optimization program.

Let $\Omega \subseteq \mathbb{R}^2$ be a visualization region, which acts as the computer screen. Let us consider a set of N individuals, $V = \{v_1, \ldots, v_N\}$, which have been observed over a time horizon of T time periods. For each $t = 1, \ldots, T$, let $V(t) \subseteq V$ be the subset of individuals to be represented in time period t, and let $|V(t)|$ be its cardinality. The elements in $V(t)$ have attached a vector of frequencies $\boldsymbol{\omega}(t) = (\omega_{i,t})_{i \in V(t)} \in \mathbb{R}_+^{|V(t)|}$ and a dissimilarity measure $\boldsymbol{\delta}(t) = (\delta_{ij,t})_{i,j \in V(t)} \in \mathbb{R}_+^{|V(t)| \times |V(t)|}$. Let $\mathscr{B}_i = \left\{\mathcal{B}_i^1, \ldots, \mathcal{B}_i^{s_i}\right\}$ be a catalogue of convex bodies, called *reference objects*, which are assumed to be symmetric closed convex sets, centered in the origin of the coordinate system in $\mathbb{R}^2$. The elements in $\mathscr{B}_i$ are the candidates to represent each $v_i \in V$, although just one of them is chosen for the whole time horizon. Let $\tau, r_{i,t}^p \in \mathbb{R}_+$ be real positive numbers, which scale the area of the reference objects in $\mathscr{B}_i$. The scaling of the reference objects is made in such a way that the area of $r_{i,t}^p \mathcal{B}_i^p$ is equal to $\omega_{i,t}$. Besides this scaling, τ is a positive parameter to be chosen by the user, which rescales all the objects in all periods in order to make sure they fit into Ω.

The dynamic dataset described above is visualized by means of a collection of T snapshots, each containing the individuals in $V(t)$ depicted as convex bodies, such as discs or rectangles. In order to properly visualize the frequencies and dissimilarities attached to the data, five conditions are considered:

(C1) Each individual $v_i \in V$ is represented by means of the same reference object, chosen from the corresponding catalogue $\mathscr{B}_i$, throughout the whole time horizon.

(C2) In each time period t, the area of the convex body used to represent each individual in $V(t)$ is proportional to its frequency $\boldsymbol{\omega}(t)$.

(C3) In each time period t, the proximity between the convex bodies representing the individuals in $V(t)$ resemble the dissimilarities $\boldsymbol{\delta}(t)$.

(C4) In each time period, the convex bodies are spread over the visualization region Ω.

(C5) The transitions between two consecutive snapshots are smooth.

In what follows, we introduce a Mathematical Optimization model which considers (C1) and (C2) as hard conditions, whereas the violation of conditions (C3)–(C5) is

minimized. A visualization framework satisfying these conditions is called in what follows a *Dynamic Visualization Map*.

Let $\mathbf{x} = (x_i^p)_{i=1,\dots,N,\, p=1,\dots,s_i}$ be decision variables defined as

$$x_i^p = \begin{cases} 1 & \text{if individual } v_i \text{ is represented by } \mathcal{B}_i^p \in \mathscr{B}_i \\ 0 & \text{otherwise,} \end{cases}$$

and let $\boldsymbol{c}_{i,t} \in \mathbb{R}^2$ be continuous variables, which translate the reference objects in $\mathscr{B}_i$ and determine their positions. In other words, if the reference object $\mathcal{B}_i^p \in \mathscr{B}_i$ is chosen to represent individual v_i, then v_i will be represented at time period t by the convex body $\boldsymbol{c}_{i,t} + \tau r_{i,t}^p \mathcal{B}_i^p$. Therefore, constructing a Dynamic Visualization Map is stated as a Mixed Integer Nonlinear Optimization Problem (MINLP), whose aim is to find the choice $\mathbf{x}$ of reference objects and the values of the translation vectors $\boldsymbol{c}_{1,1}, \dots, \boldsymbol{c}_{N,T}$ to obtain a good fit in criteria (C3)–(C5) modeled through an objective function F to be detailed later. Hence, one has to solve a problem of the form:

$$\begin{aligned} \min_{\boldsymbol{c}_{1,1},\dots,\boldsymbol{c}_{N,T},\mathbf{x}} \quad & F(\boldsymbol{c}_{1,1}, \dots, \boldsymbol{c}_{N,T}, \mathbf{x}) \\ \text{s.t.} \quad & \sum_{p=1,\dots,s_i} x_i^p = 1,\ i = 1, \dots, N, \\ & \boldsymbol{c}_{i,t} + \tau r_{i,t}^p x_i^p \mathcal{B}_i^p \subseteq \Omega,\ i = 1, \dots, N,\ p = 1, \dots, s_i,\ t = 1, \dots, T, \\ & \boldsymbol{c}_{i,t} \in \mathbb{R}^2,\ i = 1, \dots, N; t = 1, \dots, T, \\ & x_i^p \in \{0, 1\},\ i = 1, \dots, N,\ p = 1, \dots, s_i. \end{aligned} \tag{$DyViMap$}$$

The first constraint in ($DyViMap$) ensures condition (C1) is satisfied, namely, for each individual, only one reference object is chosen, among the candidates in its catalogue, to represent the individual in all time periods. The second constraint when $x_i^p = 1$ ensures that the whole convex body representing v_i, constructed by translating and scaling the reference object $\mathcal{B}_i^p$, must fit into Ω. In particular, this means that $\boldsymbol{c}_{i,t} \in \Omega$. The second constraint when $x_i^p = 0$ does not add any new information, given the observation we just made on $\boldsymbol{c}_{i,t}$. Observe that thanks to the choice of parameters $r_{i,t}^p$, namely the area of $r_{i,t}^p \mathcal{B}_i^p$ is equal to $\omega_{i,t}$, condition (C2) is satisfied, independently of the choice of reference object. Finally, the type of the variables is modeled through the third and fourth constraints.

The objective function in ($DyViMap$), F, models the violation of conditions (C3)–(C5) considering a weighted sum of three functions, F_{MDS}, F_{spread} and F_{smooth}, through a vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$, such that $\lambda_k \geq 0$ and $\sum_{k=1}^3 \lambda_k = 1$, yielding

$$F = \lambda_1 F_{MDS} + \lambda_2 F_{spread} + \lambda_3 F_{smooth}. \tag{5.1}$$

The first term, F_{MDS}, measures the discrepancy between the given dissimilarities

and the proximity between the objects (condition (C3)), by considering the *stress* expression in MDS (Borg and Groenen, 2005; Cox and Cox, 2000). The second one, F_{spread}, quantifies the spread of the objects in the visualization region (condition (C4)). Finally, F_{smooth} models the smoothness in the transition between two consecutive periods (condition (C5)).

In order to measure the proximity between two convex bodies in the t-th time period, namely the distance between $\boldsymbol{c}_{i,t} + \tau r^p_{i,t}\mathcal{B}^p_i$ and $\boldsymbol{c}_{j,t} + \tau r^q_{j,t}\mathcal{B}^q_j$, we consider the infimum distance between two closed convex sets, which is a convex function (Hiriart-Urruty and Lemaréchal, 1993). Let $\|\cdot\|$ denote the Euclidean norm, then the infimum distance between $\boldsymbol{c}_{i,t} + \tau r^p_{i,t}\mathcal{B}^p_i$ and $\boldsymbol{c}_{j,t} + \tau r^q_{j,t}\mathcal{B}^q_j$ is defined as

$$\begin{array}{rccl} d_{\tau r^p_{i,t}\mathcal{B}^p_i;\tau r^q_{j,t}\mathcal{B}^q_j}: & \mathbb{R}^2\times\mathbb{R}^2 & \longrightarrow & \mathbb{R}^+ \\ & (\boldsymbol{c}_{i,t},\boldsymbol{c}_{j,t}) & \longmapsto & \inf\limits_{\substack{\boldsymbol{b}_{i,t}\in\mathcal{B}^p_i \\ \boldsymbol{b}_{j,t}\in\mathcal{B}^q_j}} \left\| \left(\boldsymbol{c}_{i,t}+\tau r^p_{i,t}\boldsymbol{b}_{i,t}\right) - \left(\boldsymbol{c}_{j,t}+\tau r^q_{j,t}\boldsymbol{b}_{j,t}\right)\right\|, \end{array}$$

Then, the expressions of F_{MDS}, F_{spread} and F_{smooth} are:

$$F_{MDS}(\boldsymbol{c}_{1,1},\ldots,\boldsymbol{c}_{N,T},\mathbf{x}) = \sum_{t=1}^{T}\sum_{i,j\in V(t)}\sum_{\substack{p=1,\ldots,s_i \\ q=1,\ldots,s_j}} \left[d(\boldsymbol{c}_{i,t}+\tau r^p_{i,t}\mathcal{B}^p_i, \boldsymbol{c}_{j,t}+\tau r^q_{j,t}\mathcal{B}^q_j) - \kappa\delta_{ij,t}\right]^2 x^p_i x^q_j,$$

$$F_{spread}(\boldsymbol{c}_{1,1},\ldots,\boldsymbol{c}_{N,T},\mathbf{x}) = -\sum_{t=1}^{T}\sum_{i,j\in V(t)}\sum_{\substack{p=1,\ldots,s_i \\ q=1,\ldots,s_j}} d^2(\boldsymbol{c}_{i,t}+\tau r^p_{i,t}\mathcal{B}^p_i, \boldsymbol{c}_{j,t}+\tau r^q_{j,t}\mathcal{B}^q_j) x^p_i x^q_j$$

$$F_{smooth}(\boldsymbol{c}_{1,1},\ldots,\boldsymbol{c}_{N,T}) = \sum_{t=1}^{T-1}\sum_{i=1,\ldots,N} \|\boldsymbol{c}_{i,t} - \boldsymbol{c}_{i,t+1}\|^2.$$

Note that F_{MDS} and F_{spread} are generalizations from those presented in Chapter 4 for the particular case in which the dataset is not dynamic, i.e., just one time period is considered. The expression of F_{MDS} includes a positive parameter κ (to be chosen by the user), which scales the dissimilarities to make them comparable with the distances between objects measured by means of the infimum distance. Contrary to the model stated in Chapter 4, in which κ and τ are variables of the optimization problem, both are here parameters of the problem. The spread criterion, modeled through F_{spread}, aims to separate the convex bodies representing the individuals as much as possible by means of the squared infimum distance between them. The preservation of the mental map is modeled through F_{smooth}, which imposes that the locations of the objects, given by their translation vectors, do not suffer big changes from one time period to the next one (Xu et al., 2013), and by the fact that an individual is depicted by means of the

same reference object throughout the whole time horizon, since $\mathbf{x}$ does not depend on the time period.

The aim of the presented model is to obtain a trade-off between the criteria involved to enhance the interpretability of the dynamic complex data structure under consideration.

5.3 Algorithmic approach

Section 5.2 states the problem of building a Dynamic Visualization Map as a Mixed Integer Nonlinear Problem (MINLP). This section is devoted to present a solution approach to solve $(DyViMap)$, in which continuous and binary variables are optimized in an alternating fashion: the choice $\mathbf{x}$ of reference objects depicting the individuals, belonging to their corresponding catalogue, is optimized for translations $\boldsymbol{c}_{1,1}, \ldots, \boldsymbol{c}_{N,T}$ fixed, then the translation vectors $\boldsymbol{c}_{1,1}, \ldots, \boldsymbol{c}_{N,T}$ are optimized for $\mathbf{x}$ fixed, and the process is repeated until a stopping criterion is satisfied.

On one hand, observe that if the continuous variables $\boldsymbol{c}_{1,1}, \ldots, \boldsymbol{c}_{N,T}$ in $(DyViMap)$ are fixed, the resulting problem is a nonconvex binary quadratic optimization problem with assignment constraints of the form

$$\begin{array}{lll} \min\limits_{\mathbf{x}} & \displaystyle\sum_{i,j\in V} \sum_{\substack{p=1,\ldots,s_i \\ q=1,\ldots,s_j}} a_{ij}^{pq}(\boldsymbol{c})\, x_i^p x_j^q & \\ & & (DyViMap)_{\boldsymbol{c}} \\ \text{s.t.} & \sum_{p=1}^{s_i} x_i^p = 1, & i = 1, \ldots, N, \\ & x_i^p \in \{0,1\}, & i = 1, \ldots, N,\ p = 1, \ldots, s_i, \end{array}$$

where the coefficients $a_{ij}^{pq}(\boldsymbol{c})$ take the form

$$a_{ij}^{pq}(\boldsymbol{c}) = \sum_{t:\, i,j\in V(t)} \left(\lambda_1 \left[d(\boldsymbol{c}_{i,t} + \tau r_{i,t}^p \mathcal{B}_i^p, \boldsymbol{c}_{j,t} + \tau r_{j,t}^q \mathcal{B}_j^q) - \kappa \delta_{ij,t} \right]^2 - \lambda_2 d^2(\boldsymbol{c}_{i,t} + \tau r_{i,t}^p \mathcal{B}_i^p, \boldsymbol{c}_{j,t} + \tau r_{j,t}^q \mathcal{B}_j^q) \right)$$

Problem $(DyViMap)_{\boldsymbol{c}}$ can thus be solved by standard MINLP Global Optimization solvers, at the expense of high running times. However, when, at most, two reference objects are in the catalogue of each individual, e.g., just two rectangles are allocated to each individual, the problem can be rewritten as a convex quadratic problem, speeding up convergence. Indeed, in this special case, the nonconvex binary quadratic problem $(DyViMap)_{\boldsymbol{c}}$ can be rewritten as an unconstrained convex quadratic 0–1 problem by setting $x_i^2 = 1 - x_i^1$ for all $i = 1, \ldots, N$, and then one can solve the problem via e.g. any of the convexifications of the objective function described by Billionnet and Elloumi

(2007).

On the other hand, for fixed values of $\mathbf{x}$, ($DyViMap$) becomes a nonlinear continuous optimization problem, which involves a Difference of Convex (DC) function to be minimized. Indeed, when $\mathbf{x}$ is fixed, F is DC due to the fact that F_{MDS} and F_{smooth} are convex, F_{spread} is concave, and $\lambda_k \geq 0$, $k = 1, 2, 3$. It is worth noting that, since $\mathbf{x}$ is fixed, each individual is allocated to one reference object, and the model is analogous to the single-object case studied in Chapter 4 with the additional convex term F_{smooth} in the objective.

Thus, the Difference of Convex Algorithm (DCA) is a suitable tool to find good quality solutions which requires a DC decomposition of the objective function. The performance of the DCA strongly depends on the choice of the DC decomposition (Blanquero and Carrizosa, 2009; Bomze et al., 2008; Ferrer and Martínez-Legaz, 2009). In this chapter, as done in Chapter 4 and in Le Thi (2000) and Pham Dinh and Le Thi (1998), we seek a DC decomposition of F, with fixed $\mathbf{x}$, whose expression is formed by a quadratic separable convex function minus a convex function, as stated in Proposition 5.1. We prove first the following result concerning a DC decomposition of the distance function $d_{\tau r_{i,t}^p \mathcal{B}_i^p; \tau r_{j,t}^q \mathcal{B}_j^q}$.

Lemma 5.1. *Let $d_{\tau r_{i,t}^p \mathcal{B}_i^p; \tau r_{j,t}^q \mathcal{B}_j^q}$ be the infimum distance between the two closed, convex, and symmetric with respect the origin, sets $\boldsymbol{c}_{i,t} + \tau r_{i,t}^p \mathcal{B}_i^p$ and $\boldsymbol{c}_{j,t} + \tau r_{j,t}^q \mathcal{B}_j^q$. Then, $d^2_{\tau r_{i,t}^p \mathcal{B}_i^p; \tau r_{j,t}^q \mathcal{B}_j^q}$ can be expressed as a DC function, $d^2_{\tau r_{i,t}^p \mathcal{B}_i^p; \tau r_{j,t}^q \mathcal{B}_j^q} = u - \left(u - d^2_{\tau r_{i,t}^p \mathcal{B}_i^p; \tau r_{j,t}^q \mathcal{B}_j^q}\right)$, where*

$$u\left(\boldsymbol{c}_{i,t}, \boldsymbol{c}_{j,t}\right) = 2\left(\|\boldsymbol{c}_{i,t}\|^2 + \|\boldsymbol{c}_{j,t}\|^2\right).$$

Proof.

$$\begin{aligned}
&d^2_{\tau r_{i,t}^p \mathcal{B}_i^p; \tau r_{j,t}^q \mathcal{B}_j^q}\left(\boldsymbol{c}_{i,t}, \boldsymbol{c}_{j,t}\right) = \\
&\quad = \inf_{\substack{\boldsymbol{z}_i \in \boldsymbol{c}_{i,t} + \tau r_{i,t}^p \mathcal{B}_i^p \\ \boldsymbol{z}_j \in \boldsymbol{c}_{j,t} + \tau r_{j,t}^q \mathcal{B}_j^q}} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 \\
&\quad = \inf_{\substack{\boldsymbol{y}_i \in \tau r_{i,t}^p \mathcal{B}_i^p \\ \boldsymbol{y}_j \in \tau r_{j,t}^q \mathcal{B}_j^q}} \|(\boldsymbol{c}_{i,t} + \boldsymbol{y}_i) - (\boldsymbol{c}_{j,t} + \boldsymbol{y}_j)\|^2 \\
&\quad = \inf_{\substack{\boldsymbol{y}_i \in \tau r_{i,t}^p \mathcal{B}_i^p \\ \boldsymbol{y}_j \in \tau r_{j,t}^q \mathcal{B}_j^q}} \left\{\|\boldsymbol{c}_{i,t} - \boldsymbol{c}_{j,t}\|^2 + \|\boldsymbol{y}_{i,t} - \boldsymbol{y}_{j,t}\|^2 + 2\left(\boldsymbol{c}_{i,t} - \boldsymbol{c}_{j,t}\right)^\top \left(\boldsymbol{y}_{i,t} - \boldsymbol{y}_{j,t}\right)\right\} \\
&\quad = \|\boldsymbol{c}_{i,t} - \boldsymbol{c}_{j,t}\|^2 + \inf_{\substack{\boldsymbol{y}_i \in \tau r_{i,t}^p \mathcal{B}_i^p \\ \boldsymbol{y}_j \in \tau r_{j,t}^q \mathcal{B}_j^q}} \left\{\|\boldsymbol{y}_{i,t} - \boldsymbol{y}_{j,t}\|^2 + 2\left(\boldsymbol{c}_{i,t} - \boldsymbol{c}_{j,t}\right)^\top \left(\boldsymbol{y}_{i,t} - \boldsymbol{y}_{j,t}\right)\right\} \\
&\quad = 2\left(\|\boldsymbol{c}_{i,t}\|^2 + \|\boldsymbol{c}_{j,t}\|^2\right) - \left(\|\boldsymbol{c}_{i,t} + \boldsymbol{c}_{j,t}\|^2 + \sup_{\substack{\boldsymbol{y}_i \in \tau r_{i,t}^p \mathcal{B}_i^p \\ \boldsymbol{y}_j \in \tau r_{j,t}^q \mathcal{B}_j^q}} \left\{\|\boldsymbol{y}_{j,t} - \boldsymbol{y}_{i,t}\|^2 + 2\left(\boldsymbol{c}_{i,t} - \boldsymbol{c}_{j,t}\right)^\top \left(\boldsymbol{y}_{j,t} - \boldsymbol{y}_{i,t}\right)\right\}\right) \\
&\quad = 2\left(\|\boldsymbol{c}_{i,t}\|^2 + \|\boldsymbol{c}_{j,t}\|^2\right) - \varphi\left(\boldsymbol{c}_{i,t}, \boldsymbol{c}_{j,t}\right)
\end{aligned}$$

Function $\varphi(\boldsymbol{c}_{i,t}, \boldsymbol{c}_{j,t})$ is the sum of two convex functions, and it is thus convex. Therefore, the result in the statement holds. □

Proposition 5.1. *Let F as in (5.1). For a given $\mathbf{x}$, function F can be expressed as a DC function, $F = u - (u - F)$, where the quadratic separable convex function u is given by*

$$\begin{aligned} u &= \sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ 2\max\{\lambda_1 - \lambda_2, 0\} \left(\|\boldsymbol{c}_{i,t}\|^2 + \|\boldsymbol{c}_{j,t}\|^2 \right) + \lambda_1 \kappa^2 \delta_{ij,t}^2 \right\} \\ &\quad + 2\lambda_3 \sum_{i=1}^{N} \left(\|\boldsymbol{c}_{i,1}\|^2 + \|\boldsymbol{c}_{i,T}\|^2 + 2 \sum_{t=2,\ldots,T-1} \|\boldsymbol{c}_{i,t}\|^2 \right) \end{aligned}$$

Proof. Note that since $\mathbf{x}$ is fixed by assumption, we know in advance the reference object used to represent each individual. Therefore, we can drop the dependence on p and q, and we rename the distance function as $d_{\tau r_{i,t}^p \mathcal{B}_i^p; \tau r_{j,t}^q \mathcal{B}_j^q} = d_{ij,t}$.

If $\lambda_1 \geq \lambda_2$, one has

$$\begin{aligned} F = & \\ &= \sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ \lambda_1 \left(d_{ij,t}^2 + \kappa^2 \delta_{ij,t}^2 - 2\kappa \delta_{ij,t} d_{ij,t} \right) - \lambda_2 d_{ij,t}^2 \right\} \\ &\quad + \lambda_3 \sum_{i=1}^{N} \sum_{t=1}^{T-1} \left\{ 2\|\boldsymbol{c}_{i,t}\|^2 + 2\|\boldsymbol{c}_{i,t+1}\|^2 - \|\boldsymbol{c}_{i,t} + \boldsymbol{c}_{i,t+1}\|^2 \right\} \\ &= \left[\sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ (\lambda_1 - \lambda_2) d_{ij,t}^2 + \lambda_1 \kappa^2 \delta_{ij,t}^2 \right\} + 2\lambda_3 \sum_{i=1}^{N} \left\{ \|\boldsymbol{c}_{i,1}\|^2 + \|\boldsymbol{c}_{i,T}\|^2 + 2 \sum_{t=1}^{T-2} \|\boldsymbol{c}_{i,t}\|^2 \right\} \right] \\ &\quad - \left[\sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ 2\lambda_1 \kappa \delta_{ij,t} d_{ij,t} \right\} + \lambda_3 \sum_{i=1}^{N} \sum_{t=1}^{T-1} \|\boldsymbol{c}_{i,t} + \boldsymbol{c}_{i,t+1}\|^2 \right] \\ &\overset{\text{Lemma 5.1}}{=} \left[\sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ 2(\lambda_1 - \lambda_2)(\|\boldsymbol{c}_{i,t}\|^2 + \|\boldsymbol{c}_{j,t}\|^2) + \lambda_1 \kappa^2 \delta_{ij,t}^2 \right\} \right. \\ &\quad \left. + 2\lambda_3 \sum_{i=1}^{N} \left\{ \|\boldsymbol{c}_{i,1}\|^2 + \|\boldsymbol{c}_{i,T}\|^2 + 2 \sum_{t=1}^{T-2} \|\boldsymbol{c}_{i,t}\|^2 \right\} \right] \\ &\quad - \left[\sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ 2(\lambda_1 - \lambda_2)(\|\boldsymbol{c}_{i,t}\|^2 + \|\boldsymbol{c}_{j,t}\|^2) - (\lambda_1 - \lambda_2) d_{ij,t}^2 + 2\lambda_1 \kappa \delta_{ij,t} d_{ij,t} \right\} \right. \\ &\quad \left. + \lambda_3 \sum_{i=1}^{N} \sum_{t=1}^{T-1} \|\boldsymbol{c}_{i,t} + \boldsymbol{c}_{i,t+1}\|^2 \right] \end{aligned}$$

If $\lambda_1 < \lambda_2$, one has

$$
\begin{aligned}
F = & \\
& = \sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ \lambda_1 \left(d_{ij,t}^2 + \kappa^2 \delta_{ij,t}^2 - 2\kappa \delta_{ij,t} d_{ij,t} \right) - \lambda_2 d_{ij,t}^2 \right\} \\
& \quad + \lambda_3 \sum_{i=1}^{N} \sum_{t=1}^{T-1} \left\{ 2\|\boldsymbol{c}_{i,t}\|^2 + 2\|\boldsymbol{c}_{i,t+1}\|^2 - \|\boldsymbol{c}_{i,t} + \boldsymbol{c}_{i,t+1}\|^2 \right\} \\
& = \left[\sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ \lambda_1 \kappa^2 \delta_{ij,t}^2 \right\} + 2\lambda_3 \sum_{i=1}^{N} \left\{ \|\boldsymbol{c}_{i,1}\|^2 + \|\boldsymbol{c}_{i,T}\|^2 + 2 \sum_{t=1}^{T-2} \|\boldsymbol{c}_{i,t}\|^2 \right\} \right] \\
& \quad - \left[\sum_{t=1}^{T} \sum_{i,j \in V(t)} \left\{ (\lambda_2 - \lambda_1) d_{ij,t}^2 + 2\lambda_1 \kappa \delta_{ij,t} d_{ij,t} \right\} + \lambda_3 \sum_{i=1}^{N} \sum_{t=1}^{T-1} \|\boldsymbol{c}_{i,t} + \boldsymbol{c}_{i,t+1}\|^2 \right].
\end{aligned}
$$

□

Recall that DCA is an iterative process in which a sequence of convex programs are solved. At each iteration, the concave part is replaced by its affine majorization at a certain feasible point, and the resulting convex problem is then solve (Le Thi and Pham Dinh, 2013, 2005; Pham Dinh and Le Thi, 1997). Thanks to the DC decomposition of F given in Proposition 5.1, for $\mathbf{x}$ fixed, one needs to solve $N \times T$ convex quadratic problems with simple constraints:

$$
\begin{aligned}
\min_{\boldsymbol{c}_{i,t}} \quad & \left\{ M_{i,t} \|\boldsymbol{c}_{i,t}\|^2 + \boldsymbol{c}_{i,t}^{\top} \boldsymbol{\gamma}_{i,t}^{\bar{\boldsymbol{c}}} \right\} \\
\text{s.t.} \quad & \boldsymbol{c}_{i,t} + \tau r_{i,t}^p x_i^p \mathcal{B}_i^p \subseteq \Omega, \\
& \boldsymbol{c}_{i,t} \in \mathbb{R}^2,
\end{aligned}
\qquad (DyViMap)_{DCA_{i,t}}
$$

for scalars $M_{i,t} \in \mathbb{R}_+$, which follow from the coefficients that multiply each term in the u part (after grouping terms) in Proposition 5.1. Vectors $\boldsymbol{\gamma}_{i,t}^{\bar{\boldsymbol{c}}} \in \mathbb{R}^2$ are subgradients of the function $u - F$ evaluated in the locations obtained in the previous iteration of DCA, $\bar{\boldsymbol{c}} = (\bar{\boldsymbol{c}}_{1,1}, \ldots, \bar{\boldsymbol{c}}_{N,T})$. When Ω has an amenable form, for instance a box or a disc, the optimal solution of Problems $(DyViMap)_{DCA_{i,t}}$, $i = 1, \ldots, N$, $t = 1, \ldots, T$, can be readily obtained by differentiating and equating the gradient to zero (considering correctly the constraints). In this case, as noted in Chapter 4, running times will be strongly reduced since the convex optimization problems to be solved at each stage of DCA have a closed expression for their optimal value.

The DCA scheme for solving $(DyViMap)$ with fixed $\mathbf{x}$ is outlined in Algorithm 5.1.

Algorithm 5.1 DCA scheme for $(DyViMap)$ with fixed $\mathbf{x}$

Input: $\boldsymbol{c}^{ini} = \left(\boldsymbol{c}^{ini}_{1,1}, \ldots, \boldsymbol{c}^{ini}_{N,T}\right)$, such that $\boldsymbol{c}^{ini}_{i,t} + \tau r^p_{i,t} x^p_i \mathcal{B}^p_i \subseteq \Omega$, $i = 1, \ldots, N$, $t = 1, \ldots, T$.

1: $\bar{\boldsymbol{c}} \leftarrow \boldsymbol{c}^{ini}$

2: **repeat**

3: Compute $\boldsymbol{\gamma}^{\bar{\boldsymbol{c}}}_{i,t} \in \partial\left(u - F\right)(\bar{\boldsymbol{c}})$;

4: Compute $\boldsymbol{c} = (\boldsymbol{c}_{1,1}, \ldots, \boldsymbol{c}_{N,T})$ as the solution of Problem $(DyViMap)_{DCA_{i,t}}$, for all $i = 1, \ldots, N$, $t = 1, \ldots, T$;

5: $\bar{\boldsymbol{c}} \leftarrow \boldsymbol{c}$;

6: **until** stop condition is met.

Output: $\bar{\boldsymbol{c}} = (\bar{\boldsymbol{c}}_{1,1}, \ldots, \bar{\boldsymbol{c}}_{N,T})$.

Summarizing, solving Problem $(DyViMap)$ by means of an alternating algorithm which optimizes its continuous and binary variables, respectively, requires the call to a DCA subroutine in the first case (Algorithm 5.1 to optimize $\boldsymbol{c}_{1,1}, \ldots, \boldsymbol{c}_{N,T}$) and an integer nonconvex quadratic solver to optimize $\mathbf{x}$. The alternating routine to solve $(DyViMap)$ is given in Algorithm 5.2.

Algorithm 5.2 Alternating scheme for $(DyViMap)$

Input: $\boldsymbol{x}^{ini} = \left((x^{ini})^p_i\right) \in \{0,1\}^S$, where $S = \sum_{k=1}^N s_k$, and $\boldsymbol{c}^{ini} = \left(\boldsymbol{c}^{ini}_{1,1}, \ldots, \boldsymbol{c}^{ini}_{N,T}\right)$, such that $\boldsymbol{c}^{ini}_{i,t} + \tau r^p_{i,t} (x^{ini})^p_i \mathcal{B}^p_i \subseteq \Omega$, $i = 1, \ldots, N$, $p = 1, \ldots, s_i$, $t = 1, \ldots, T$.

1: $\bar{\boldsymbol{c}} \leftarrow \boldsymbol{c}^{ini}$;

2: $\bar{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{ini}$;

3: **repeat**

4: $\bar{\boldsymbol{c}} \leftarrow$ Algorithm 5.1$(\bar{\boldsymbol{c}})$;

5: $\bar{\boldsymbol{x}} \leftarrow$ solve $(DyViMap)_{\boldsymbol{c}}$;

6: **until** stop condition is met.

Output: $\bar{\boldsymbol{c}} = (\bar{\boldsymbol{c}}_{1,1}, \ldots, \bar{\boldsymbol{c}}_{N,T})$, $\bar{\boldsymbol{x}} = \left(\bar{x}^p_i\right)$, $i = 1, \ldots, N$, $p = 1, \ldots, s_i$.

5.4 Computational experience

The methodology proposed in Section 5.3 is illustrated in four datasets, one geographic a three linguistic examples. In Section 5.4.1 we describe the four datasets used in the experiments and in Section 5.4.2 how Algorithms 5.1 and 5.2 have been implemented. Finally, Section 5.4.3 includes the results.

5.4.1 Datasets

Our first example consists of visualizing the evolution of the population in the states forming the U.S. across three time periods. The U.S. dataset consists of $N = 50$ individuals, the states forming the country, for which the population in $T = 3$ time periods (three years: 1890, 1950 and 2010), $\boldsymbol{\omega}(t)$, has been recorded, (Census Bureau, 2012). The dissimilarities, $\boldsymbol{\delta}(t)$, depict the geodesic distance between the centroids of the states. These dissimilarities have been computed in R, (R Core Team, 2016), running the `gdist` function to the latitude/longitud coordinates given in `state.center` data. The dataset measurement is incomplete, in the sense that there are no data available for Alaska (AK) and Hawaii (HI) in 1890. In other words, $V(1)$ consists of all states excepting AK and HI, whereas $V(2)$ and $V(3)$ contain the 50 states.

The linguistic datasets consist of the most popular words arising in Danish news around the three different topics along a time period: *internet* between 1994 and 1997, *terror* between 2001 and 2015 and *immigration* between 1995 and 2015 (Carrizosa et al., 2017a). The relevance of each word per year, $\boldsymbol{\omega}(t)$, is measured by means of the *term frequency inverse document frequency (tf-idf)* weighting factor, whereas $\boldsymbol{\delta}(t)$ depicts the co-occurrence of pairs of words using the cosine vector similarity formula (Salton and Buckley, 1988). We have a corpus of words V, and we set $V(t) = \{v \in V : v \text{ is relevant in time period } t\}$.

5.4.2 Experiments details

Algorithms 5.1 and 5.2 have been coded in AMPL (Fourer et al., 1993), and the quadratic binary problems have been solved with CPLEX 12.6, (CPLEX, IBM ILOG, 2014). The computational experiments have been carried out on a PC Intel® Core™ i7-2600K, 16GB of RAM. Since Problem ($DyViMap$) is expected to be multimodal, since it extends standard MDS (known to be multimodal, Trosset and Mathar (1997) and Žilinskas and Podlipskytė (2003)), DCA may get stuck in local optima. For this reason, Algorithm 5.2 has been embedded in a multistart routine.

We set $\Omega = [0,1] \times [0,1]$, and $\lambda_1 = 0.7$, $\lambda_2 = 0.2$ and $\lambda_3 = 0.1$. The choices of τ and κ are made dependent on the dataset following the expressions in (5.2) and (5.3), respectively.

$$\tau = \frac{1}{\max_t \sum_{i=1}^{N} \omega_{i,t}} \cdot 0.05 \tag{5.2}$$

$$\kappa = \frac{N(N-1)T}{\sum_{t=1}^{T} \sum_{\substack{i,j=1,\dots,N \\ i \neq j}} \delta_{ij,t}} \cdot 0.30 \tag{5.3}$$

The unit square in $\mathbb{R}^2$ is considered as the reference object for all the U.S. dataset, this is $\mathscr{B}_i = \{\Omega = [0,1] \times [0,1]\}$, $i = 1, \dots, N$. Since there exists only one reference object per individual, variables $\mathbf{x}$ are known (fixed) a priori and the alternating strategy in Algorithm 5.2 reduces to Algorithm 5.1. The maximum number of iterations in Algorithm 5.1 is set to 100 and the number of iterations of the multistart routine to 50. Initial values of the translation vectors, $\boldsymbol{c}^{ini}$ are uniformly generated in Ω, taking into account the feasible region.

For the linguistic datasets related to *internet* and *terror* topics, the unit disc in $\mathbb{R}^2$ is considered as the reference object for all the words, this is $\mathcal{B}_i = \{(a,b) : a^2 + b^2 \leq 1\}$, $i = 1, \dots, N$, and thus $\mathscr{B}_i = \{\mathcal{B}_i\}$. Similarly to the U.S. example, variables $\mathbf{x}$ are known a priori and the alternating strategy in Algorithm 5.2 reduces to Algorithm 5.1.

For the linguistic datasets related to *immigration* topic, we consider that each word $v_i \in V$ has associated as a set of potential reference objects $\mathscr{B}_i = \left\{\mathcal{R}_i^1, \mathcal{R}_i^2\right\}$, where $\mathcal{R}_i^p$ are rectangles parallel to the coordinate axes, $p = 1, 2$. The height of $\mathcal{R}_i^1$ is equal to one unit, whereas its basis corresponds to the number of letters of the word. $\mathcal{R}_i^2$ is obtained using a 90 degree rotation of $\mathcal{R}_i^1$. Since two reference objects per word are considered, the 0–1 quadratic optimization problems to be solved in Algorithm 5.2 can be rewritten as 0–1 unconstrained convex quadratic problems. The convexification of the problem is made following the first approach described by Billionnet and Elloumi (2007), namely computing the smallest eigenvalue of the matrix by means of the Gerschgorin Theorem (Gershgorin, 1931). We set the maximum number of iterations in Algorithm 5.2 to 10, where the inside call to Algorithm 5.1 is made with a maximum number of iterations of 10 as well, and the number of runs of the multistart routine to 50. Initial values of the translation vectors, $\boldsymbol{c}^{ini}$ are randomly generated in Ω.

5.4.3 Results

Figures 5.3–5.5 show the Dynamic Visualization Map representing the U.S. dataset. Besides the good fitting of dissimilarities, we observe how the squares representing the states are spread over region Ω and they do not suffer big shifts from one period to another.

Figure 5.6 shows the Dynamic Visualization Map representing the linguistic dataset related to *internet*, whereas Figures 5.7–5.9 relate to *terror* topic and Figures 5.10–5.13 with *immigration*. Words that are not in a period t but they are in $t+1$ have been underlined to highlight their first appearance. Despite the continuous changing in the

words in each time period, the preservation of the mental map (smoothness criterion) is achieved for those words that are present in (almost) the whole time horizon.

5.5 Conclusions

In this chapter we have addressed the problem of visualizing a dynamic complex dataset, by preserving its underlying structure as well as the mental map to make the interpretability easier. A Mathematical Optimization model has been proposed, which yields a Mixed Integer Nonlinear Program. We assume that there exists a catalogue of candidate reference objects to represent each individual in the dataset, and the choice should be made according to three goodness of fit criteria that pursue the preservation of the data structure: first, the distance between the convex bodies must resemble a given dissimilarity; second, the objects must be spread over the visualization region; finally, the mental map should be preserved. An alternating algorithm, which combines the use of DC optimization tools (Difference of Convex Algorithm) as well as 0–1 quadratic programming, has been proposed as solution approach. Our numerical experience shows the suitability of our algorithm. Indeed, the stated model for dynamic complex datasets can be straightforwardly generalized to $\mathbb{R}^n$, $n \geq 3$. Moreover, different distances than the infimum could be considered as well.

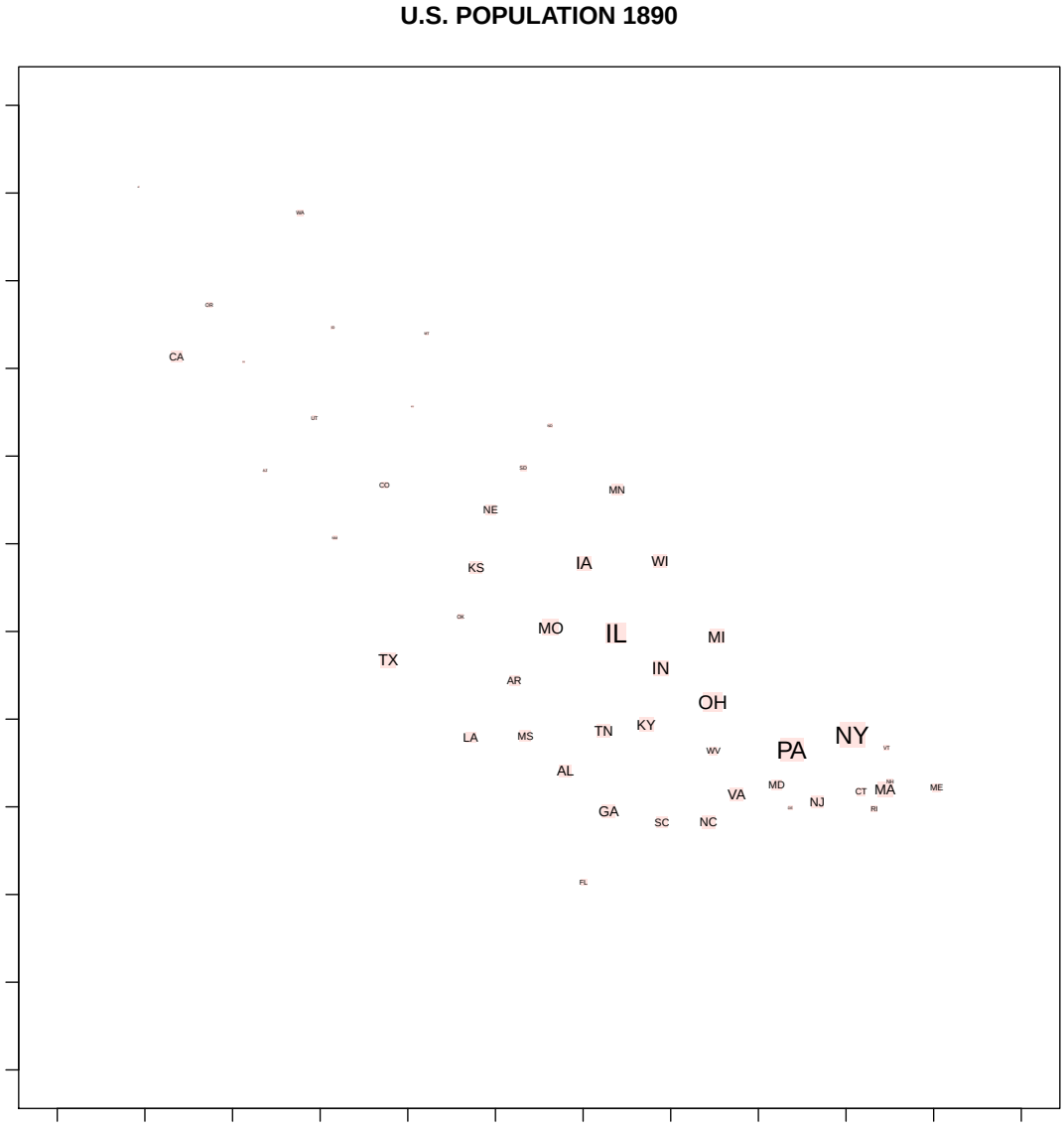


Figure 5.3: U.S. dataset visualization in $T = 1$.

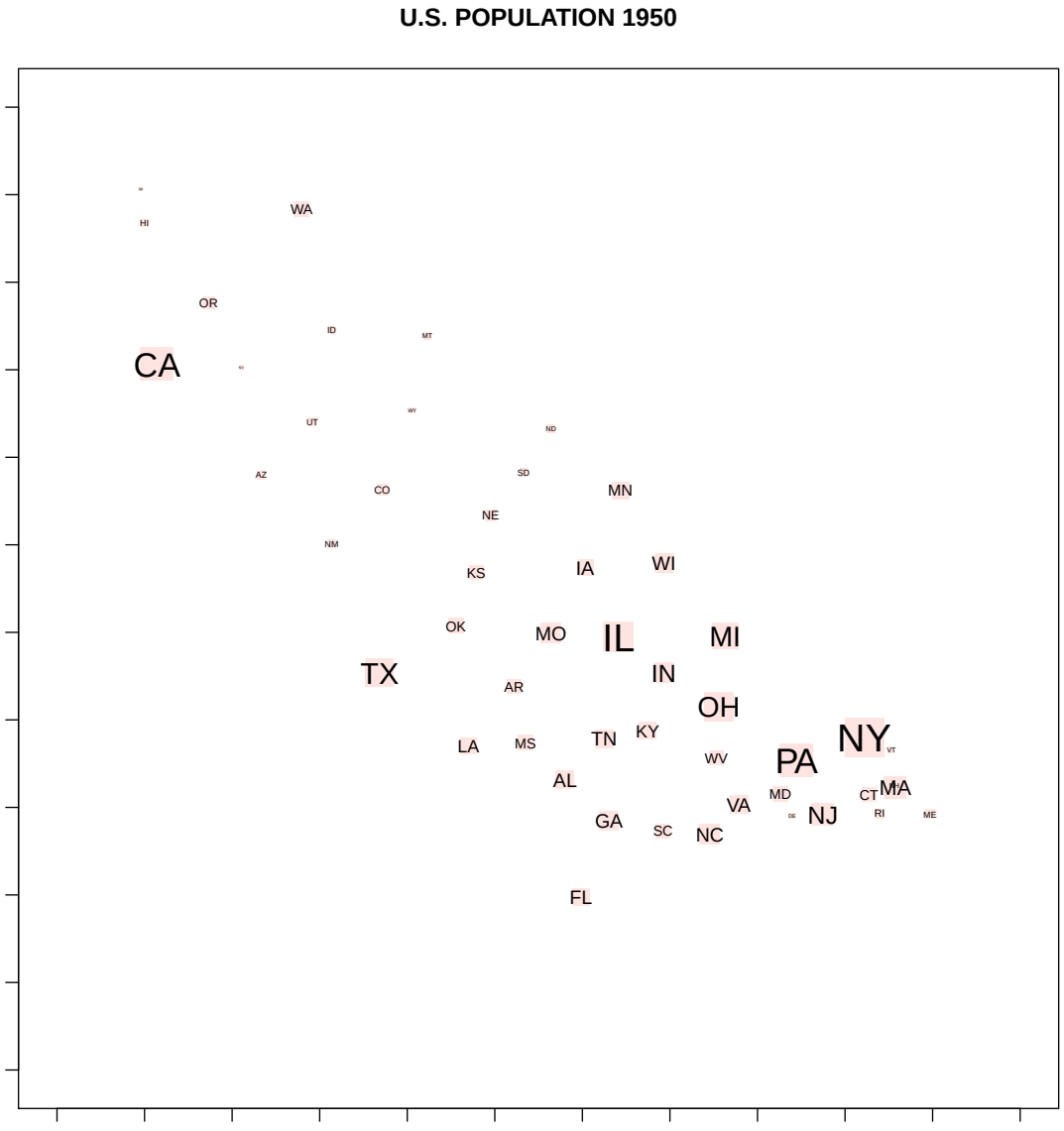


Figure 5.4: U.S. dataset visualization in $T = 2$.

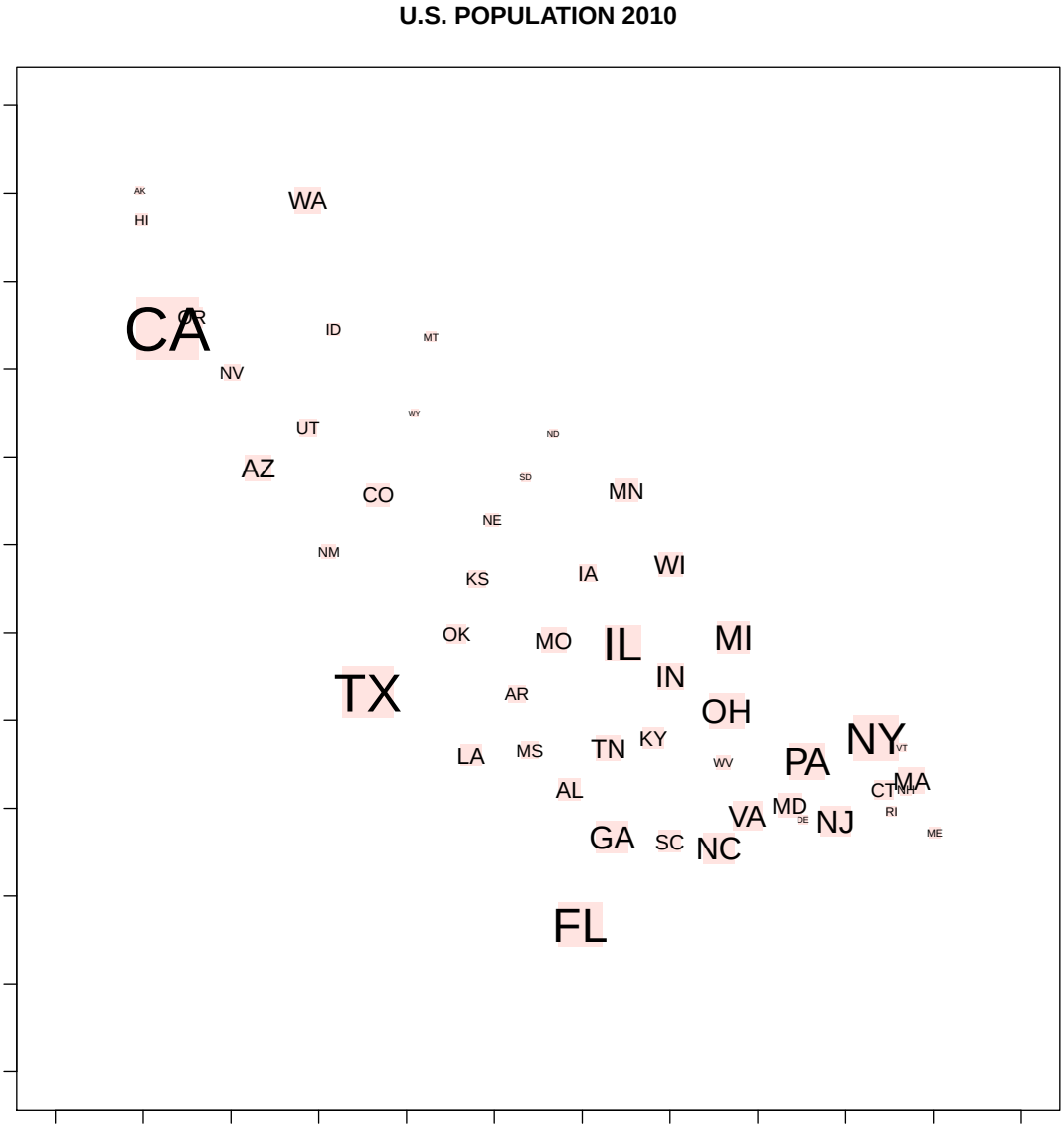


Figure 5.5: U.S. dataset visualization in $T = 3$.

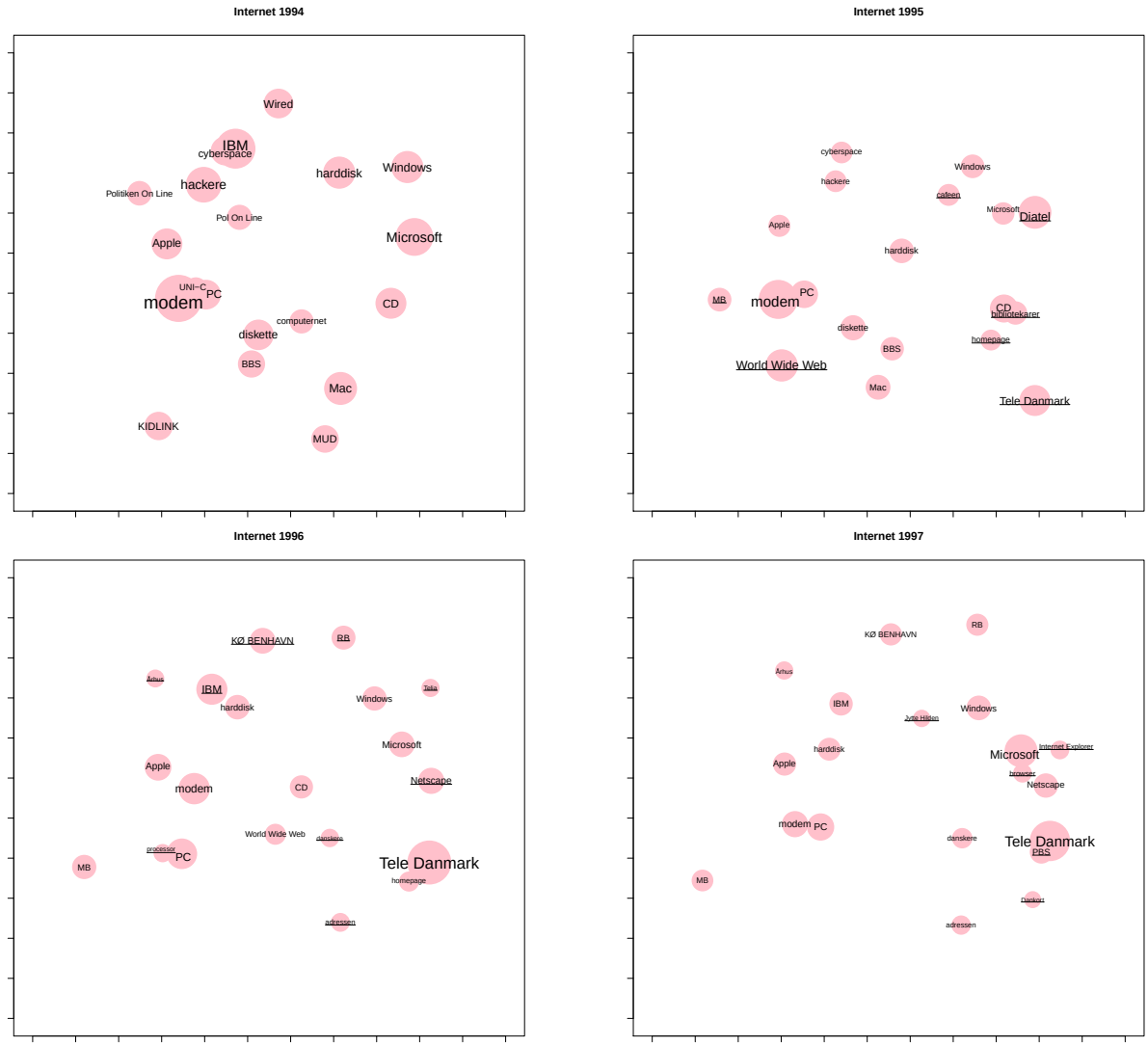


Figure 5.6: Dynamic Visualization Map of *internet* topic.

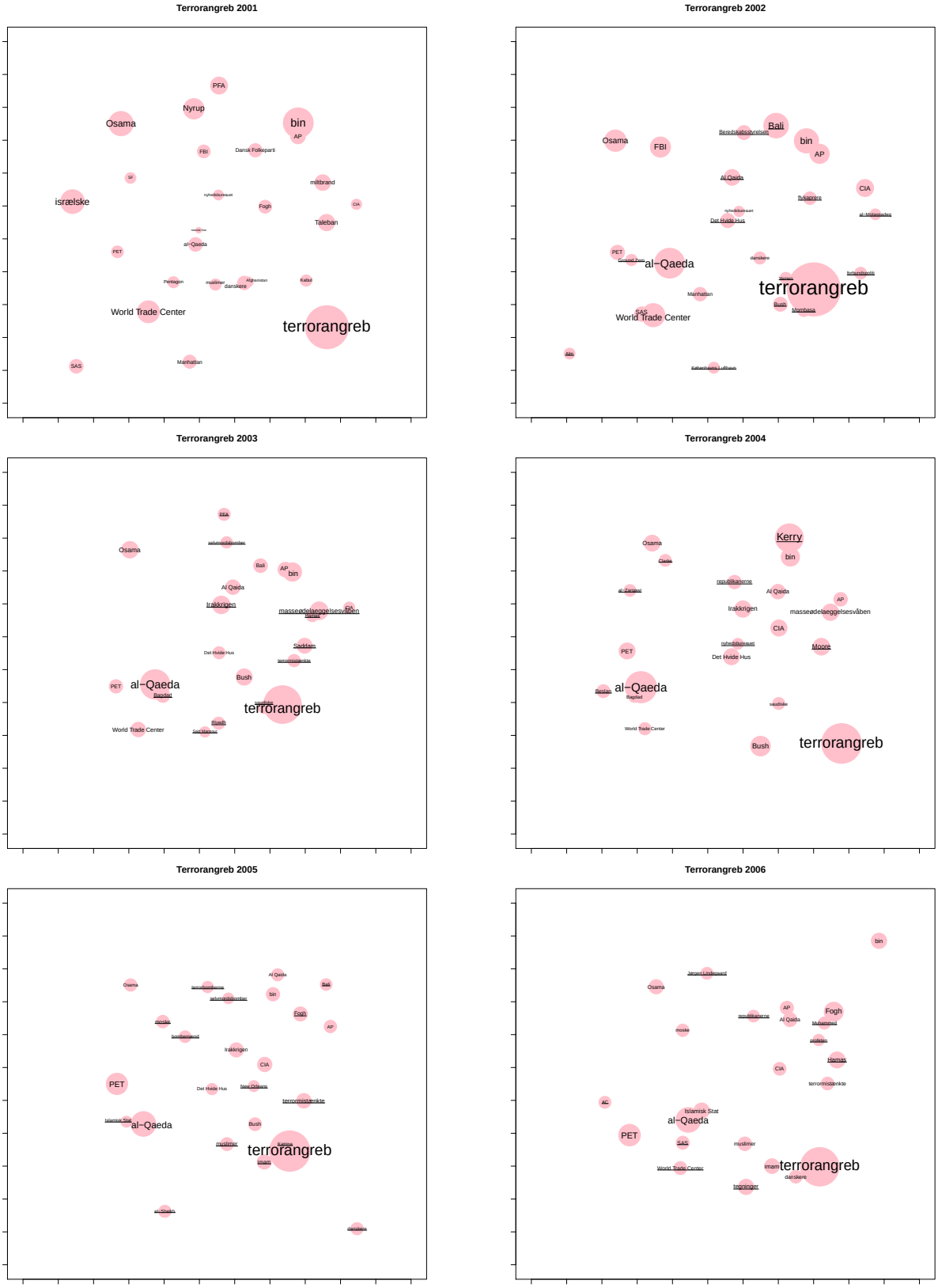


Figure 5.7: Dynamic Visualization Map of *terror* topic in $T = 1, \ldots, 6$.

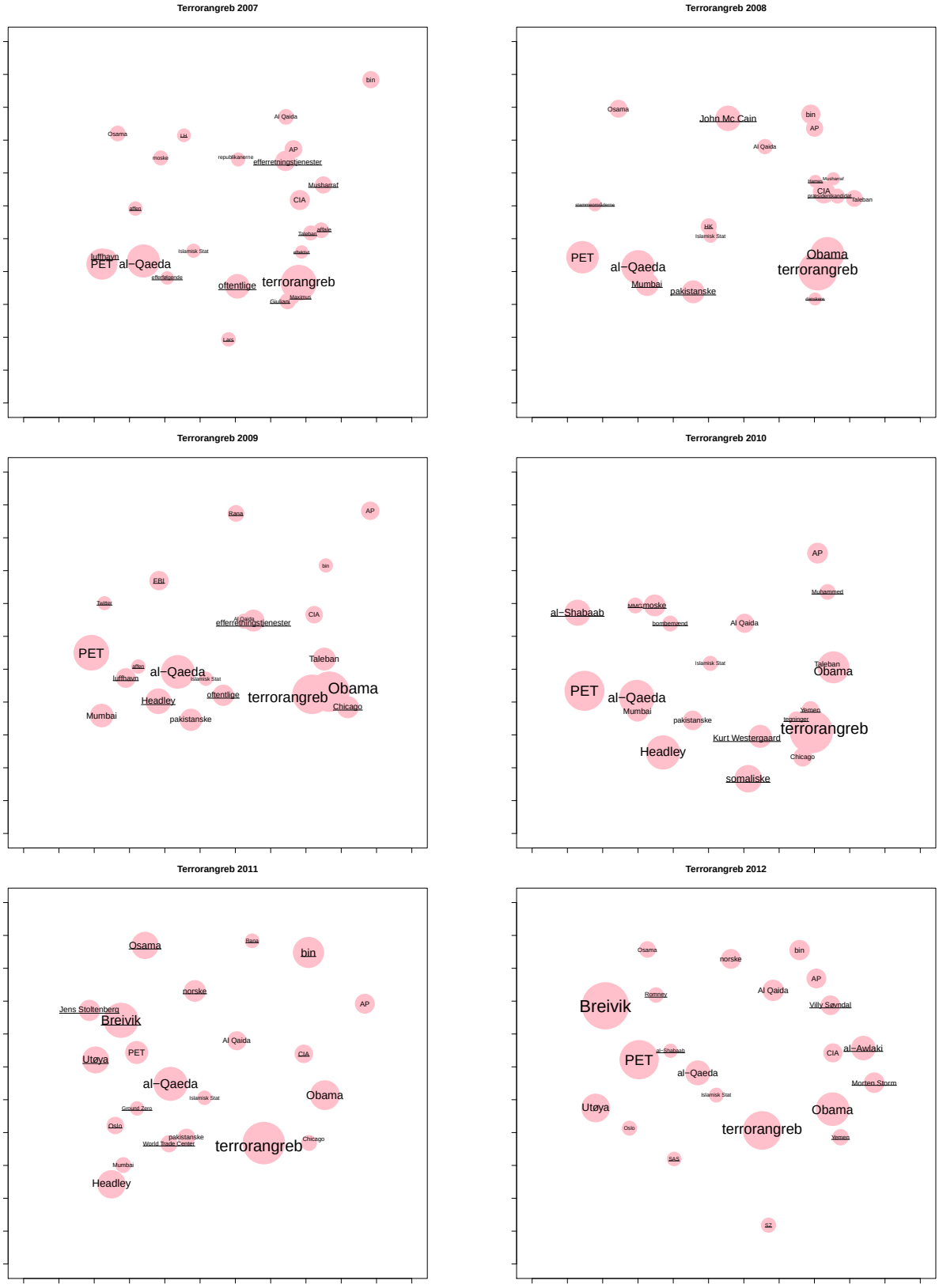


Figure 5.8: Dynamic Visualization Map of *terror* topic in $T = 7, \ldots, 12$.

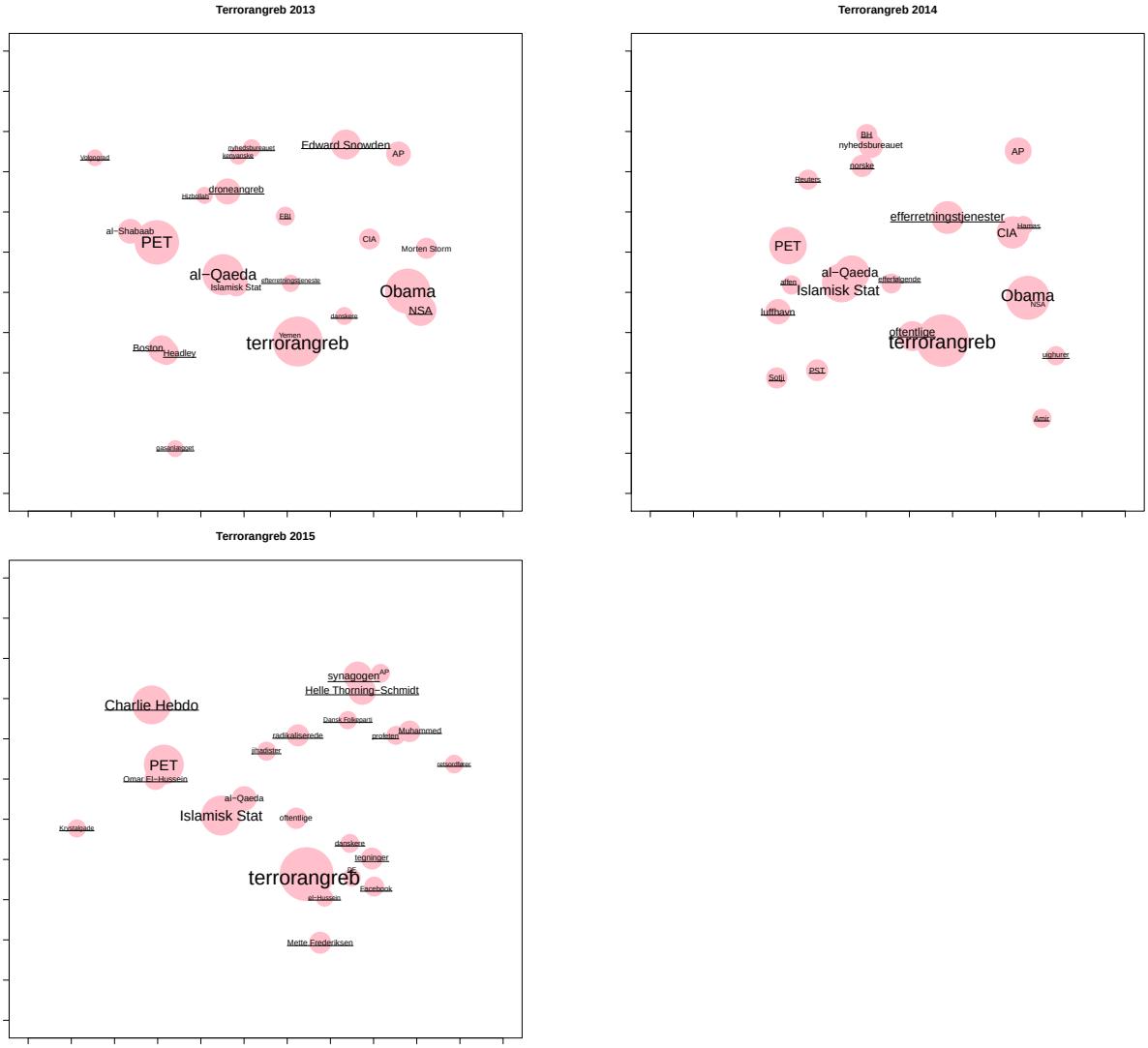


Figure 5.9: Dynamic Visualization Map of *terror* topic in $T = 13, \ldots, 15$.

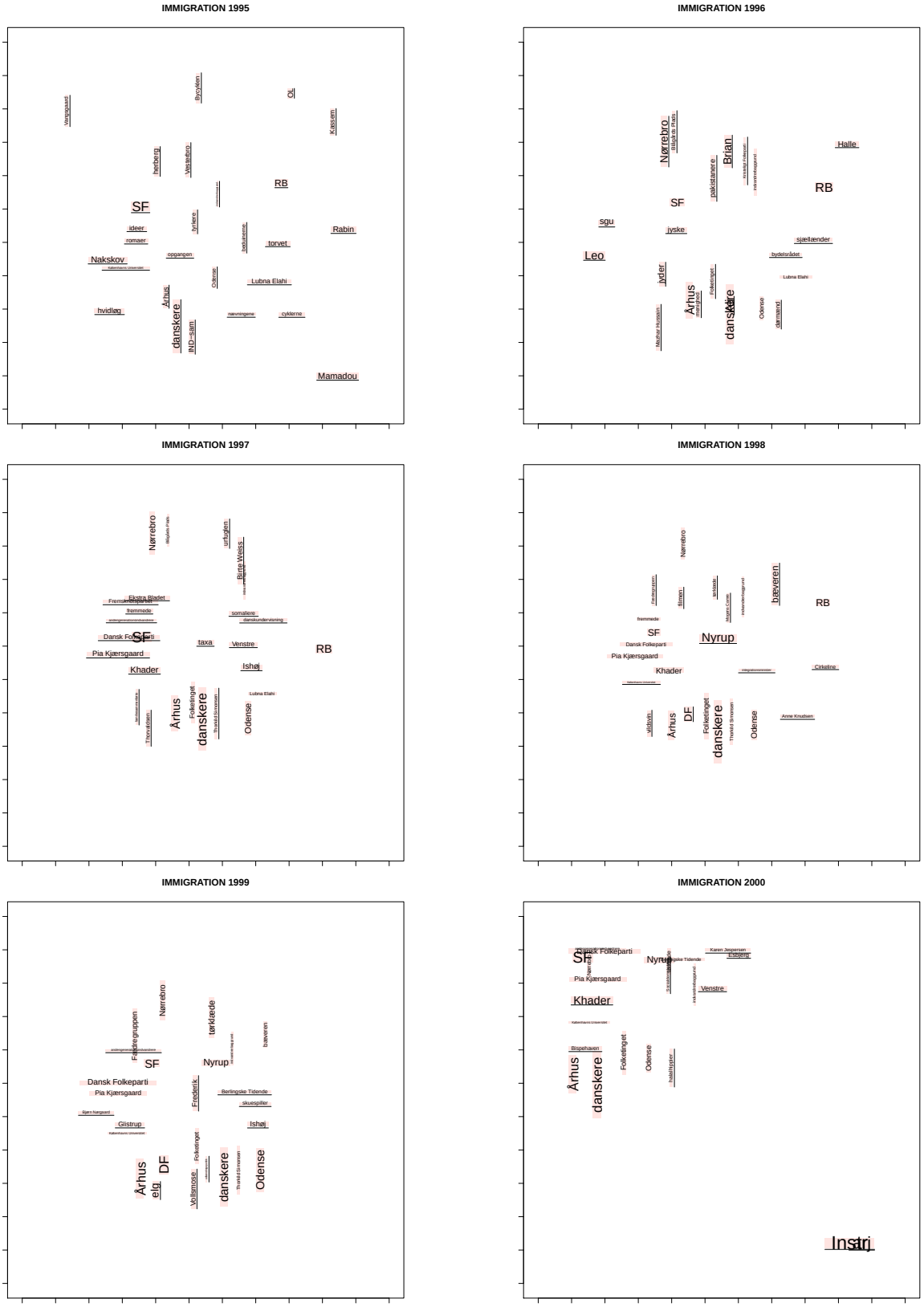


Figure 5.10: Linguistic dataset visualization in $T = 1, \ldots, 6$.

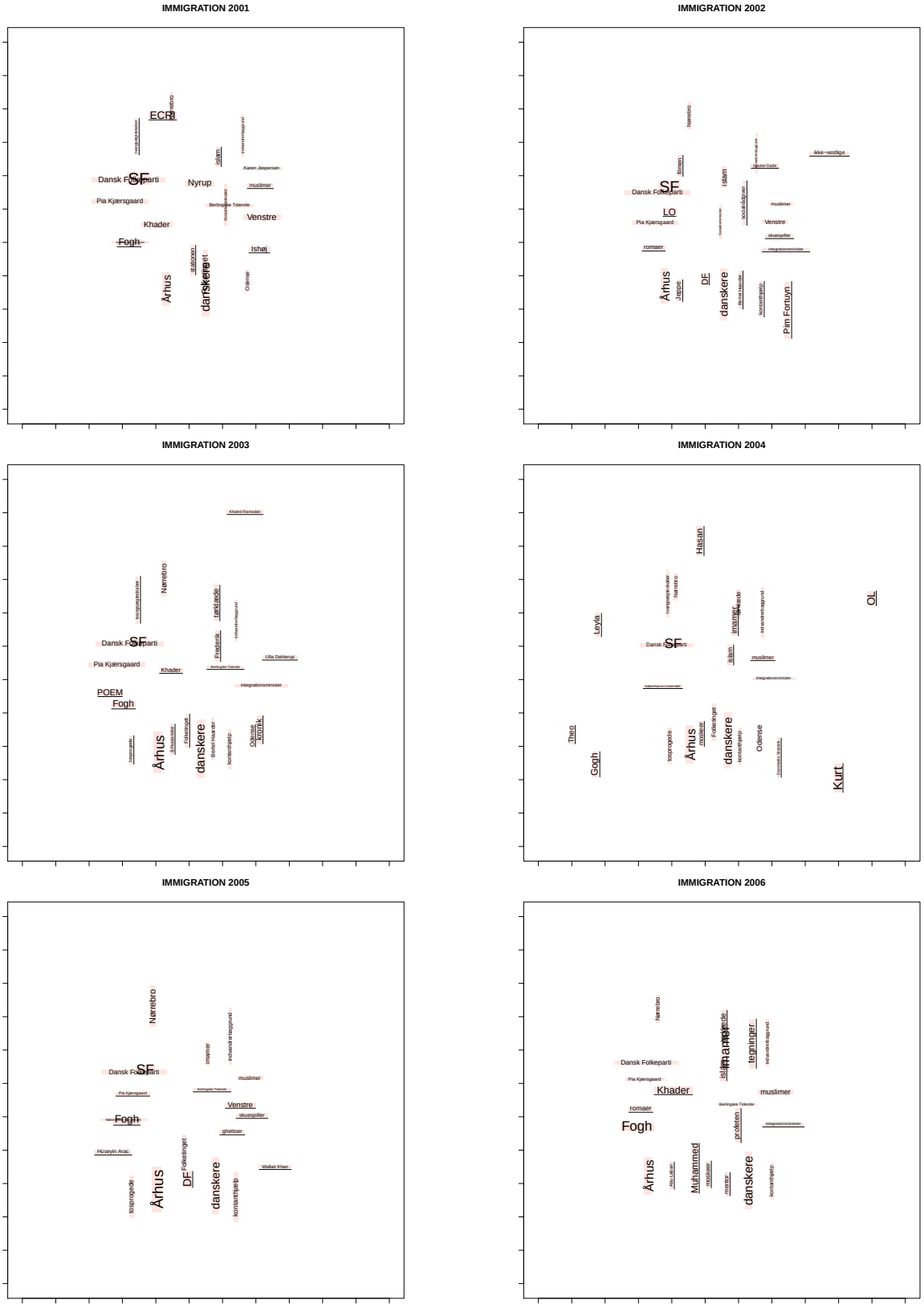


Figure 5.11: Linguistic dataset visualization in $T = 7, \ldots, 12$.

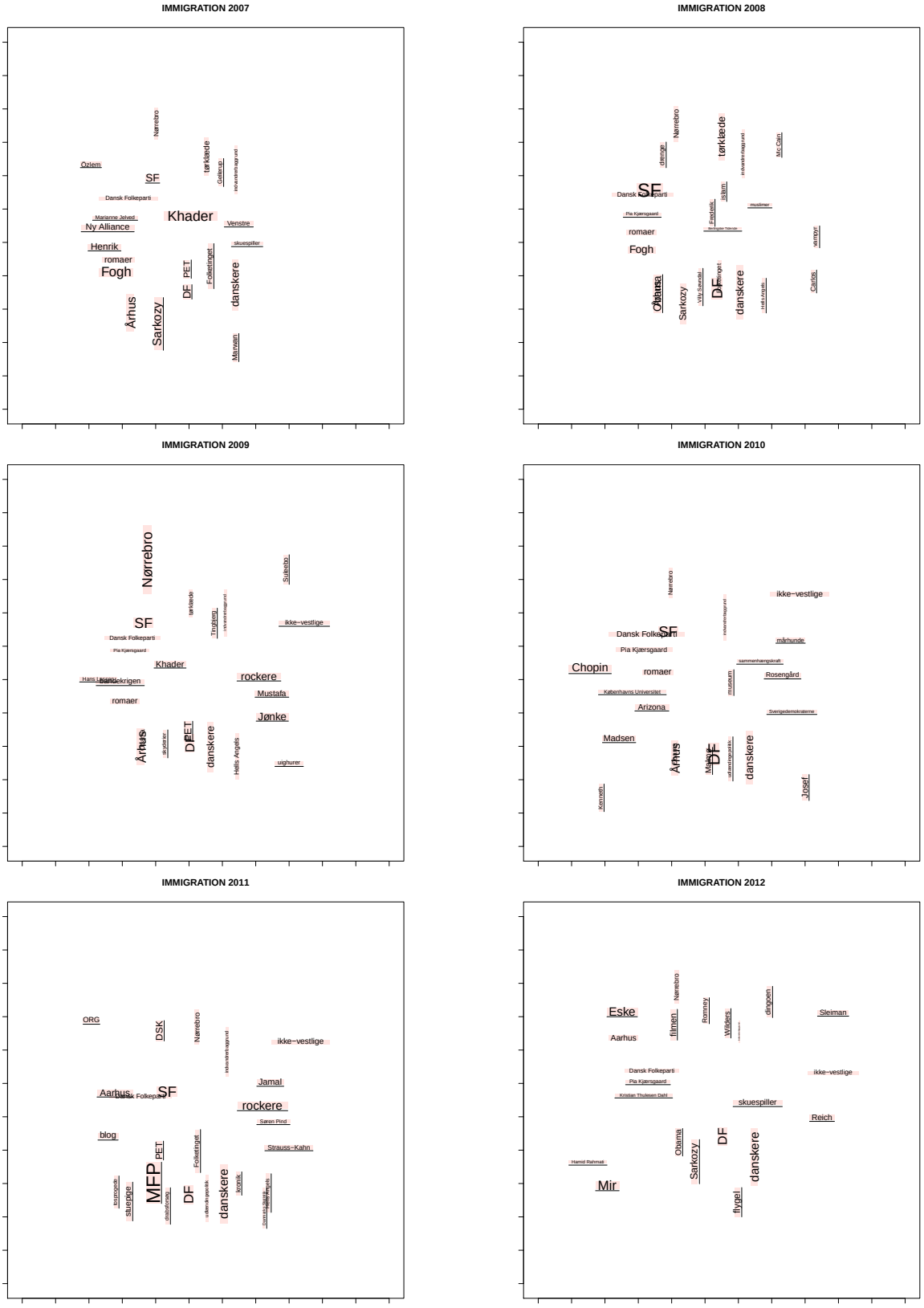


Figure 5.12: Linguistic dataset visualization in $T = 13, \ldots, 18$.

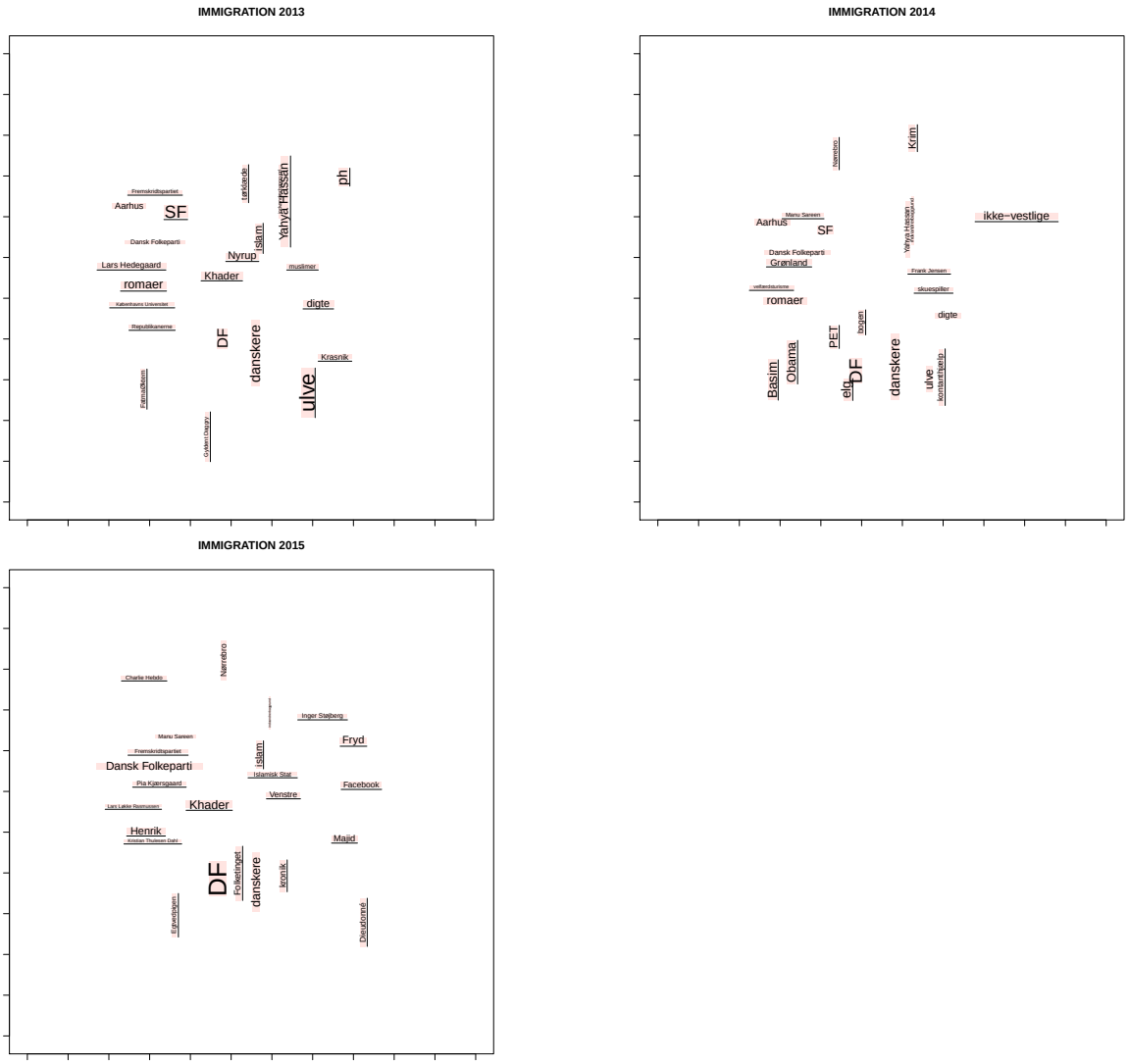


Figure 5.13: Linguistic dataset visualization in $T = 19, \ldots, 21$.

Chapter 6

General conclusions and further work

In this PhD dissertation, Mathematical Optimization has been the common thread to deal with Information Visualization challenges. Different ways of visualizing a complex dataset by maintaining its underlying structure have been presented through the chapters forming this thesis. Specifically, we have dealt with data involving a frequency distribution and a proximity relationship, in terms of adjacencies or dissimilarities. Moreover, these magnitudes may vary over time as well. Roughly speaking, the visualization frameworks presented in this thesis consist of representing each individual involved in the dataset as a geometric object, which is scaled according to the frequency distribution and located in a given visualization region according to the proximity relationship. Indeed, each of these geometric objects may contain extra information such as frequencies, hierarchies or proximity relations, represented e.g. by means of histograms, treemaps, or graphs, respectively, as well as additional numerical or categorical variables, which can be visualized by means of colors, hues, etc. (Benbasat and Dexter, 1985; Gomez-Nieto et al., 2016; Shmueli et al., 2016).

In order to build visualization frameworks for these datasets, new optimization models based on Mixed Integer (Non) Linear Programming (Chapters 2, 3 and 5) and Difference of Convex tools (Chapters 4 and 5) have been formally stated. In addition, solution approaches consisting of matheuristics that exploit the structure of the problems have been either specifically designed (Cell Perturbing Algorithm and Embedded Cell Perturbing Algorithm in Chapter 2) or adapted (Large Neighborhood Search in Chapter 3 and Difference of Convex Algorithm in Chapters 4 and 5) to obtain high-quality solutions.

There are several interesting lines for future research. Regarding Chapters 2 and 3, the so-called "segment moving heuristic" (Kreveld and Speckmann, 2007), could be customized to our problem in order to improve the approximation made in the areas after having a (K, L)-rectangular map or a Space-filling Box-connected Map. Nevertheless, even if we were able to detect the portions whose sizes can be changed, and thus, the segments that can be moved without destroying either the rectangular shapes or the box-connectivity, the proximities' structure could be dramatically changed by such movements. Hence, local changes are difficult to detect due to the rigid structure of the visualization frameworks and this approach deserves further study. In addition, modeling dynamic (K, L)-rectangular map and Space-filling Box-connected Maps which can handle temporal changes seems to be also a very interesting problem (Cui et al., 2010; de Pinho et al., 2010; Mashima et al., 2012). Finally, considering visualization regions different from a rectangle, in which a regular grid cannot be obtained in a straightforward manner, is a challenging problem when modeling portions' connectivity (Carvajal et al., 2013; Wang et al., 2015).

Regarding Chapters 4 and 5, we have considered the infimum distance (*d1*). Instead, one can consider other classical distances in Cluster Analysis such as the supremum

distance (*d2*) or the average one (*d3*), Hansen and Jaumard (1997), as well as other approaches such as the Hausdorff distance (*d4*). First, it should be observed that the average distance between two convex sets may not have an easy expression, and thus approximations may be needed (Koshizuka and Kurita, 1991; Vaughan, 1984). Second, we have assumed the reference objects $\mathcal{B}$ and $\mathcal{B}_i^p$ to be convex, to guarantee the convexity of the function giving the infimum distance and thus allowing us to express $(VM)^*$ and $(DyViMap)$ as DC optimization problems. However, for arbitrary reference objects the infimum distance (*d1*) and the Hausdorff distance (*d4*) functions may not be DC (Blanquero and Carrizosa, 2009). Nevertheless, as discussed e.g. by Blanquero et al. (2009), important classes of nonconvex sets (e.g. finite union of convex sets) make the infimum distance a DC function, and thus the analysis in these chapters extend gracefully to such cases. It should be observed that if the supremum distance or the average distance are used instead, then the distance function is convex for arbitrary reference objects. This follows from the fact that the supremum in (*d2*) and the integral in (*d3*) preserve the convexity of the norm function used in their definitions. Thus, the objective function of $(VM)^*$ and $(DyViMap)$ would be DC regardless of the shape of the reference objects. Third, working on different ways of addressing the spread criterion as well as the overlapping between the objects depicting the individuals might be interesting. Dealing with new functions, which may have good properties in terms of convexity or DC decompositions as those studied in these chapters, sets a new challenge in terms of modeling.

In general, there exist interesting extensions of the work presented in this PhD dissertation as well as the possibility of studying other applications of Mathematical Optimization to Information Visualization. First, the solution approaches used in this thesis are heuristic, and thus, the convergence of the algorithm to the global optima is not guaranteed. A better performance can be obtained if these algorithms are plugged, as local search routines, within a strategy which avoids local optima, such as (continuous) Variable Neighborhood Search (Carrizosa et al., 2012; Mladenović and Hansen, 1997; Mladenović et al., 2008). Second, all visualization frameworks developed can also be adapted to visualize hierarchical data, in which inside every object representing the individuals, namely rectangles, box-connected rectangles or general geometric objects, a new set of individuals have to be depicted. However, this extension is not straightforward since nested proximity relationships need to be taken into account and consequently depicted: the individuals in the same group should respect their in-between proximities but also the relationship among the members of other groups. Third, the online version of these problems is a challenging extension of the studies made in this PhD dissertation: instead of considering that the dataset is fully known in advance for the whole time horizon, it is natural to assume that in forthcoming time periods new individuals may appear. These new individuals should be integrated in the previously

constructed visualization framework in real-time.

List of Figures

References

Abbiw-Jackson, R., Golden, B., Raghavan, S., and Wasil, E. (2006). A divide-and-conquer local search heuristic for data visualization. *Computers & Operations Research*, 33(11):3070–3087.

Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011). *Visualization of time-oriented data*. Springer Science & Business Media.

Alam, M. J., Biedl, T., Felsner, S., Kaufmann, M., Kobourov, S. G., and Ueckerdt, T. (2013). Computing cartograms with optimal complexity. *Discrete & Computational Geometry*, 50(3):784–810.

Andrienko, N., Andrienko, G., and Gatalsky, P. (2003). Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503–541.

Anjos, M. F. and Liers, F. (2012). Global approaches for facility layout and VLSI floorplanning. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 849–877. Springer US, Boston, MA.

Anjos, M. F. and Vieira, M. V. C. (2016). An improved two-stage optimization-based framework for unequal-areas facility layout. *Optimization Letters*, 10(7):1379–1392.

Archambault, D. and Purchase, H. C. (2016). Can animation support the visualisation of dynamic graphs? *Information Sciences*, 330:495–509.

Baesens, B., Setiono, R., Mues, C., and Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, 49(3):312–329.

Bao, Y. and Datta, A. (2014). Simultaneously discovering and quantifying risk types from textual risk disclosures. *Management Science*, 60(6):1371–1391.

Battista, G. D., Eades, P., Tamassia, R., and Tollis, I. G. (1999). *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall, Upper Saddel River, NJ.

Baudel, T. and Broeksema, B. (2012). Capturing the design space of sequential space-filling layouts. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2593–2602.

Beck, F., Burch, M., Diehl, S., and Weiskopf, D. (2016). A taxonomy and survey of dynamic graph visualization. Forthcoming in *Computer Graphics Forum*.

Benbasat, I. and Dexter, A. S. (1985). An experimental evaluation of graphical and color-enhanced information presentation. *Management Science*, 31(11):1348–1364.

Bertsimas, D., O'Hair, A., Relyea, S., and Silberholz, J. (2016). An analytics approach to designing combination chemotherapy regimens for cancer. *Management Science*, 62(5):1511–1531.

Biedl, T. C. and Genç, B. (2005). Complexity of octagonal and rectangular cartograms. In *17th Canadian Conference on Computational Geometry*, pages 117–120.

Billionnet, A. and Elloumi, S. (2007). Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68.

Blanquero, R. and Carrizosa, E. (2009). Continuous location problems and big triangle small triangle: constructing better bounds. *Journal of Global Optimization*, 45(3):389–402.

Blanquero, R., Carrizosa, E., and Hansen, P. (2009). Locating objects in the plane using global optimization techniques. *Mathematics of Operations Research*, 34(4):837–858.

Bomze, I. M., Locatelli, M., and Tardella, F. (2008). New and old bounds for standard quadratic optimization: dominance, equivalence and incomparability. *Mathematical Programming*, 115(1):31–64.

Borg, I. and Groenen, P. J. F. (2005). *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag, New York.

Buchin, K., Speckmann, B., and Verdonschot, S. (2012). Evolution strategies for optimizing rectangular cartograms. In *Geographic Information Science*, volume 7478, pages 29–42. Springer.

Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H., and Chen, L. (2008). Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472.

Cabello, S., Haverkort, H., Van Kreveld, M., and Speckmann, B. (2010). Algorithmic aspects of proportional symbol maps. *Algorithmica*, 58(3):543–565.

Cameron, S. and Culley, R. (1986). Determining the minimum translational distance between two convex polyhedra. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 591–596.

Cano, R. G., Kunigami, G., De Souza, C. C., and De Rezende, P. J. (2013). A hybrid grasp heuristic to construct effective drawings of proportional symbol maps. *Computers & Operations Research*, 40(5):1435–1447.

Carrizosa, E., Conde, E., Muñoz-Márquez, M., and Puerto, J. (1995). The generalized weber problem with expected distances. *Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle*, 29(1):35–57.

Carrizosa, E., Dražić, M., Dražić, Z., and Mladenović, N. (2012). Gaussian variable neighborhood search for continuous optimization. *Computers & Operations Research*, 39(9):2206–2213.

Carrizosa, E. and Guerrero, V. (2014a). Biobjective sparse principal component analysis. *Journal of Multivariate Analysis*, 132:151–159.

Carrizosa, E. and Guerrero, V. (2014b). rs-Sparse principal component analysis: A mixed integer nonlinear programming approach with VNS. *Computers & Operations Research*, 52:349–354.

Carrizosa, E., Guerrero, V., Hardt, D., Romero Morales, D., and Valverde Martínez, J. M. (2017a). On building dynamic visualization maps for news story topic data. Technical report, Copenhagen Business School.

Carrizosa, E., Guerrero, V., and Romero Morales, D. (2015a). A multi-objective approach to visualize adjacencies in weighted graphs by rectangular maps. Technical report, Optimization Online. `http://www.optimization-online.org/DB_HTML/2015/12/5226.html`.

Carrizosa, E., Guerrero, V., and Romero Morales, D. (2015b). Visualizing data as objects by DC (difference of convex) optimization. Technical report, Optimization Online. `http://www.optimization-online.org/DB_HTML/2015/12/5227.html`.

Carrizosa, E., Guerrero, V., and Romero Morales, D. (2017b). Visualization of complex dynamic datasets by means of mathematical optimization. Technical report, IMUS.

Carrizosa, E., Guerrero, V., and Romero Morales, D. (2017c). Visualizing proportions and dissimilarities by Space-filling maps: a Large Neighborhood Search approach. *Computers & Operations Research*, 78:369–380.

Carrizosa, E., Martín-Barragán, B., and Morales, D. R. (2010). Binarized support vector machines. *INFORMS Journal on Computing*, 22(1):154–167.

Carrizosa, E., Martín-Barragán, B., Plastria, F., and Romero Morales, D. (2007). On the selection of the globally optimal prototype subset for nearest-neighbor classification. *INFORMS Journal on Computing*, 19(3):470–479.

Carrizosa, E., Muñoz-Márquez, M., and Puerto, J. (1998a). Location and shape of a rectangular facility in $\mathbb{R}^n$. Convexity properties. *Mathematical Programming*, 83(1-3):277–290.

Carrizosa, E., Muñoz-Márquez, M., and Puerto, J. (1998b). The Weber problem with regional demand. *European Journal of Operational Research*, 104(2):358–365.

Carrizosa, E. and Romero Morales, D. (2013). Supervised classification and mathematical optimization. *Computers & Operations Research*, 40(1):150–165.

Carvajal, R., Constantino, M., Goycoolea, M., Vielma, J. P., and Weintraub, A. (2013). Imposing connectivity constraints in forest planning models. *Operations Research*, 61(4):824–836.

Census Bureau, U. S. (2012). *Cartograms of State Populations in 1890, 1950 and 2010*. United States. `https://www.census.gov/dataviz/visualizations/021/508.php`.

Chen, C. P. and Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275:314–347.

Chen, L. and Buja, A. (2009). Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219.

Cottam, J. A., Lumsdaine, A., and Weaver, C. (2012). Watch this: A taxonomy for dynamic data visualization. In *IEEE Conference on Visual Analytics Science and Technology*, pages 193–202.

Cox, T. F. and Cox, M. A. A. (2000). *Multidimensional scaling*. CRC Press.

CPLEX, IBM ILOG (2014). `http://www.ilog.com/products/cplex/`.

Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M. X., and Qu, H. (2010). Context preserving dynamic word cloud visualization. In *IEEE Pacific Visualization Symposium*, pages 121–128.

de Berg, M., Mumford, E., and Speckmann, B. (2009). On rectilinear duals for vertex-weighted plane graphs. *Discrete Mathematics*, 309(7):1794–1812.

de Berg, M., Mumford, E., and Speckmann, B. (2010). Optimal BSPs and rectilinear cartograms. *International Journal of Computational Geometry & Applications*, 20(02):203–222.

De Leeuw, J. (1977). Applications of convex analysis to multidimensional scaling. In Barra, J. R., Brodeau, F., Romier, G. V. C., and B., eds., *Recent Developments in Statistics*, pages 133–146. North Holland Publishing Company.

De Leeuw, J. and Heiser, W. (1977). Convergence of correction matrix algorithms for multidimensional scaling. In Lingoes, J., Roskam, E., and Borg, I., eds., *Geometric Representations of Relational Data*, pages 735–752. Mathesis Press, Ann Arbor, MI.

De Leeuw, J. and Heiser, W. J. (1980). Multidimensional scaling with restrictions on the configuration. *Multivariate analysis*, 5:501–522.

de Pinho, R. D., de Oliveira, M. C. F., and de Andrade Lopes, A. (2010). An incremental space to visualize dynamic data sets. *Multimedia Tools and Applications*, 50(3):533–562.

Demir, E., Bektaş, T., and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359.

Destatis, Statistisches Bundesamt (2015). Area and population. `www.destatis.de`. Retrieved on: 2015-01-14.

Díaz-Báñez, J., Mesa, J. A., and Schöbel, A. (2004). Continuous location of dimensional structures. *European Journal of Operational Research*, 152(1):22–44.

Dörk, M., Carpendale, S., and Williamson, C. (2012). Visualizing explicit and implicit relations of complex information spaces. *Information Visualization*, 11(1):5–21.

Dorling, D. (1996). Area cartograms: their use and creation. In *Concepts and Techniques in Modern Geography*. Environmental Publications, Norwich.

Duarte, F. S., Sikansi, F., Fatore, F. M., Fadel, S. G., and Paulovich, F. V. (2014). Nmap: A novel neighborhood preservation space-filling algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2063–2071.

Dzemyda, G., Kurasova, O., and Žilinskas, J. (2013). *Multidimensional Data Visualization: Methods and Applications*. Springer-Verlag New York.

Ehrgott, M. (2006). A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360.

Elkeran, A. (2013). A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. *European Journal of Operational Research*, 231(3):757–769.

Eppstein, D., Mumford, E., Speckmann, B., and Verbeek, K. (2012). Area-universal rectangular layouts. *SIAM Journal on Computing*, 41(3):537–564.

Eppstein, D., van Kreveld, M., Speckmann, B., and Staals, F. (2015). Improved grid map layout by point set matching. *International Journal of Computational Geometry & Applications*, 25(02):101–122.

Ferrer, A. and Martínez-Legaz, J. E. (2009). Improving the efficiency of DC global optimization methods by improving the DC representation of the objective function. *Journal of Global Optimization*, 43(4):513–531.

Flavin, T., Hurley, M., and Rousseau, F. (2002). Explaining stock market correlation: A gravity model approach. *The Manchester School*, 70:87–106.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

Fourer, R., Gay, D. M., and Kernighan, B. W. (1993). *AMPL: A Modeling Language for Mathematical Programming*. Scientific Press.

Freitas, A. (2013). Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10.

Fried, O., DiVerdi, S., Halber, M., Sizikova, E., and Finkelstein, A. (2015). IsoMatch: Creating informative grid layouts. *Computer Graphics Forum*, 34(2):155–166.

Gershgorin, S. A. (1931). Über die Abgrenzung der Eigenwerte einer Matrix. *Bulletin de l'Académie des Sciences de l'URSS*, 6:749–754.

Gomez-Nieto, E., Casaca, W., Motta, D., Hartmann, I., Taubin, G., and Nonato, L. G. (2016). Dealing with multiple requirements in geometric arrangements. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1223–1235.

Gómez-Nieto, E., San Roman, F., Pagliosa, P., Casaca, W., Helou, E. S., de Oliveira, M. C. F., and Nonato, L. G. (2014). Similarity preserving snippet-based visualization of web search results. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):457–470.

Groenen, P. J. F. (1993). *The Majorization Approach to Mmultidimensional Scaling: Some Problems and Extensions*. DSWO Press, Leiden.

Groenen, P. J. F. and Franses, P. H. (2000). Visualizing time-varying correlations across stock markets. *Journal of Empirical Finance*, 7(2):155–172.

Hahsler, M. (2017). An experimental comparison of seriation methods for one-mode two-way data. *European Journal of Operational Research*, 257(1):133–143.

Hall, K. W., Perin, C., Kusalik, P. G., Gutwin, C., and Carpendale, S. (2016). Formalizing emphasis in information visualization. *Computer Graphics Forum*, 35(3):717–737.

Hansen, P. and Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3):191–215.

Hartigan, J. A. (1967). Representation of similarity matrices by trees. *Journal of the American Statistical Association*, 62(320):1140–1158.

Heer, J., Bostock, M., and Ogievetsky, V. (2010). A tour through the visualization zoo. *Communications of the ACM*, 53:59–67.

Heilmann, R., Keim, D. A., Panse, C., and Sips, M. (2004). RecMap: Rectangular map approximations. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 33–40. IEEE Computer Society.

Hiriart-Urruty, J. and Lemaréchal, C. (1993). *Convex Analysis and Minimization Algorithms*. Springer.

Hubert, L., Arabie, P., and Hesson-Mcinnis, M. (1992). Multidimensional scaling in the city-block metric: A combinatorial approach. *Journal of Classification*, 9(2):211–236.

Jafari, N. and Hearne, J. (2013). A new method to solve the fully connected reserve network design problem. *European Journal of Operational Research*, 231(1):202–209.

Jankovits, I., Luo, C., Anjos, M. F., and Vannelli, A. (2011). A convex optimisation framework for the unequal-areas facility layout problem. *European Journal of Operational Research*, 214(2):199–215.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. Wiley, New York.

Keim, D., Andrienko, G., Fekete, J. D., Görg, C., Kohlhammer, J., and Melançon, G. (2008). Visual analytics: Definition, process, and challenges. In *Information Visualization: Human-Centered Issues and Perspectives*, pages 154–175. Springer, Berlin, Heidelberg.

Klimenta, M. and Brandes, U. (2013). Graph drawing by classical multidimensional scaling: new perspectives. In *International Symposium on Graph Drawing*, volume 7704, pages 55–66. Springer.

Kochhar, J. S., Foster, B. T., and Heragu, S. S. (1998). HOPE: a genetic algorithm for the unequal area facility layout problem. *Computers & Operations Research*, 25(7):583–594.

Korsvik, J. E., Fagerholt, K., and Laporte, G. (2011). A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research*, 38(2):474–483.

Koshizuka, T. and Kurita, O. (1991). Approximate formulas of average distances associated with regions and their applications to location problems. *Mathematical Programming*, 52(1-3):99–123.

Koźmiński, K. and Kinnen, E. (1985). Rectangular duals of planar graphs. *Networks*, 15(2):145–157.

Kreveld, M. and Speckmann, B. (2007). On rectangular cartograms. *Computational Geometry*, 37(3):175–187.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Kunigami, G., de Rezende, P. J., de Souza, C. C., and Yunes, T. (2014). Optimizing the layout of proportional symbol maps: Polyhedra and computation. *INFORMS Journal on Computing*, 26(2):199–207.

Le Thi, H. A. (2000). An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints. *Mathematical Programming*, 87:401–426.

Le Thi, H. A. and Pham Dinh, T. (2001). DC programming approach to the multidimensional scaling problem. In Migdalas, A., Pardalos, P. M., and Värbrand, P., eds., *From Local to Global Optimization*, volume 53 of *Nonconvex Optimizations and Its Applications*, pages 231–276. Springer.

Le Thi, H. A. and Pham Dinh, T. (2005). The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133(1-4):23–46.

Le Thi, H. A. and Pham Dinh, T. (2013). DC programming approaches for distance geometry problems. In Mucherino, A., Lavor, C., Liberti, L., and Maculan, N., eds., *Distance Geometry*, pages 225–290. Springer.

Leung, P. L. and Lau, K. (2004). Estimating the city-block two-dimensional scaling model with simulated annealing. *European Journal of Operational Research*, 158(2):518–524.

Lewand, R. E. (2000). *Cryptological Mathematics*. The Mathematical Association of America, Washington, D. C.

Liberti, L., Lavor, C., Maculan, N., and Mucherino, A. (2014). Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69.

Lin, C.-C., Lee, Y.-Y., and Yen, H.-C. (2011). Mental map preserving graph drawing using simulated annealing. *Information Sciences*, 181(19):4253–4272.

Lin, M. C. and Manocha, D. (2004). Collision and proximity queries. In O'Rourke, J. and Goodman, E., eds., *Handbook of Discrete and Computational Geometry*, pages 787–807. CRC Press.

Liu, S., Cui, W., Wu, Y., and Liu, M. (2014). A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393.

Liu, X., Hu, Y., North, S., and Shen, H. (2013). Compactmap: A mental map preserving visual interface for streaming text data. In *IEEE International Conference on Big Data*, pages 48–55.

Liu, X., Hu, Y., North, S., and Shen, H. W. (2015). CorrelatedMultiples: Spatially coherent small multiples with constrained multi-dimensional scaling. *Computer Graphics Forum*, pages 1–12.

Lodi, A., Malaguti, E., Stier-Moses, N. E., and Bonino, T. (2015). Design and control of public-service contracts and an application to public transportation systems. *Management Science*, 62(4):1165–1187.

Martens, D., Baesens, B., Gestel, T. V., and Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, 183(3):1466–1476.

Mashima, D., Kobourov, S. G., and Hu, Y. (2012). Visualizing dynamic data with maps. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1424–1437.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. *Mathematical Programming*, 10(1):147–175.

Misue, K., Eades, P., Lai, W., and Sugiyama, K. (1995). Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183–210.

Mladenović, N., Dražić, M., Kovačevic-Vujčić, V., and Čangalović, M. (2008). General variable neighborhood search for the continuous optimization. *European Journal of Operational Research*, 191(3):753–770.

Mladenović, N. and Hansen, P. (1997). Variable Neighborhood Search. *Computers & Operations Research*, 24(11):1097–1100.

Nocaj, A. and Brandes, U. (2012). Organizing search results with a reference map. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2546–2555.

Olafsson, S., Li, X., and Wu, S. (2008). Operations research and data mining. *European Journal of Operational Research*, 187(3):1429–1448.

Önal, H. and Briers, R. A. (2006). Optimal selection of a connected reserve network. *Operations Research*, 54(2):379–388.

Önal, H., Wang, Y., Dissanayake, S. T. M., and Westervelt, J. D. (2016). Optimal design of compact and functionally contiguous conservation management areas. *European Journal of Operational Research*, 251(3):957–968.

Ong, C. J. and Gilbert, E. G. (1996). Growth distances: New measures for object separation and penetration. *IEEE Transactions on Robotics and Automation*, 12(6):888–903.

Owen-Smith, J., Riccaboni, M., Pammolli, F., and Powell, W. W. (2002). A comparison of U.S. and European university-industry relations in the life sciences. *Management Science*, 48(1):24–43.

Pacino, D. and Van Hentenryck, P. (2011). Large neighborhood search and adaptive randomized decompositions for flexible jobshop scheduling. In *International Joint Conference on Artificial Intelligence*, pages 1997–2002.

Panse, C. (2016). Rectangular Statistical Cartograms in R: The recmap Package. *arXiv preprint arXiv:1606.00464*.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572.

Pham Dinh, T. and El Bernoussi, S. (1986). Algorithms for solving a class of nonconvex optimization problems. methods of subgradients. In *Fermat Days 85: Mathematics for Optimization*, volume 129, pages 249 – 271. North-Holland.

Pham Dinh, T. and El Bernoussi, S. (1988). Duality in DC (difference of convex functions) optimization. Subgradient methods. In *Trends in Mathematical Optimization: 4th French-German Conference on Optimization*, pages 277–293. Basel.

Pham Dinh, T. and Le Thi, H. A. (1997). Convex analysis approach to DC programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355.

Pham Dinh, T. and Le Thi, H. A. (1998). A branch-and-bound method via DC optimization algorithm and ellipsoidal technique for box constrained nonconvex quadratic programming problems. *Journal of Global Optimization*, 13:171–206.

Pisinger, D. and Ropke, S. (2010). Large neighborhood search. In Gendreau, M. and Potvin, J. Y., eds., *Handbook of Metaheuristics*, volume 146, chapter 13, pages 399–419. Springer US.

Pong, T. K. and Tseng, P. (2011). (Robust) edge-based semidefinite programming relaxation of sensor network localization. *Mathematical Programming*, 130(2):321–358.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. `https://www.R-project.org/`. Retrieved on: 2017-01-17.

Raisz, E. (1934). The rectangular statistical cartogram. *Geographical Review*, 24(2):292–296.

Ribeiro, G. M. and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3):728–735.

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired-associate learning tasks. *Journal of Experimental Psychology*, 53(2):94.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming*, volume 1520, pages 417–431. Springer, Berlin Heidelberg.

Sherali, H. D., Fraticelli, B. M. P., and Meller, R. D. (2003). Enhanced model formulations for optimal facility layout. *Operations Research*, 51(4):629–644.

Shmueli, G., Patel, N. R., and Bruce, P. C. (2016). *Data Mining for Business Analytics: Concepts, Techniques, and Applications in XLMiner*. John Wiley & Sons.

Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99.

So, A. M.-C. and Ye, Y. (2007). Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2-3):367–384.

Spence, I. and Lewandowsky, S. (1991). Displaying proportions and percentages. *Applied Cognitive Psychology*, 5(1):61–77.

Stanford Blood Center (2014). Blood type in U.S. `http://bloodcenter.stanford.edu/`. Retrieved on: 2014-11-19.

Statistics Netherlands (2013). Population; gender, age, marital status and region, January 1. `www.cbs.nl`. Retrieved on: 2013-10-31.

Strehl, A. and Ghosh, J. (2003). Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS Journal on Computing*, 15(2):208–230.

Strong, G. and Gong, M. (2014). Self-sorting map: An efficient algorithm for presenting multimedia data in structured layouts. *IEEE Transactions on Multimedia*, 16(4):1045–1058.

Takane, Y., Young, F. W., and De Leeuw, J. (1977). Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika*, 42(1):7–67.

Tamassia, R., eds. (2013). *Handbook of Graph Drawing and Visualization*. CRC press.

Tani, K., Tsukiyama, S., Shinoda, S., and Shirakawa, I. (1991). On area-efficient drawings of rectangular duals for VLSI floor-plan. *Mathematical Programming*, 52(1-3):29–43.

Tobler, W. (2004). Thirty five years of computer cartograms. *Annals of the Association of American Geographers*, 94(1):58–73.

Torgerson, W. S. (1958). *Theory and Methods of Scaling*. Wiley.

Trosset, M. and Mathar, R. (1997). On the existence of nonglobal minimizers of the stress criterion for metric multidimensional scaling. In *Proceedings of the Statistical Computing Section*, pages 158–162. American Statistical Association.

Trosset, M. W. (2002). Extensions of classical multidimensional scaling via variable reduction. *Computational Statistics*, 17:147–163.

Tseng, P. (2007). Second-order cone programming relaxation of sensor network localization. *SIAM Journal on Optimization*, 18(1):156–185.

Tufte, E. R. (1986). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA.

Tuy, H. (1998). *Convex Analysis and Global Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Umetani, S., Yagiura, M., Imahori, S., Imamichi, T., Nonobe, K., and Ibaraki, T. (2009). Solving the irregular strip packing problem via guided local search for overlap minimization. *International Transactions in Operational Research*, 16(6):661–683.

Ustun, B. and Rudin, C. (2016). Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391.

Van Vlasselaer, V., Eliassi-Rad, T., Akoglu, L., Snoeck, M., and Baesens, B. (2016). GOTCHA! Network-based fraud detection for social security fraud. Forthcoming in *Management Science*.

Vaughan, R. (1984). Approximate formulas for average distances associated with zones. *Transportation Science*, 18(3):231–244.

Viégas, F. B. and Wattenberg, M. (2008). Tag clouds and the case for vernacular visualization. *Interactions*, 15(4):49–52.

von Landesberger, T., Fellner, D., and Ruddle, R. (2016). Visualization system requirements for data processing pipeline design and optimization. Forthcoming in *IEEE Transactions on Visualization and Computer Graphics*.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.

Wang, W., Wang, H., Dai, G., and Wang, H. (2006). Visualization of large hierarchical data by circle packing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 517–520, New York, NY, USA. ACM.

Wang, Y., Buchanan, A., and Butenko, S. (2015). On imposing connectivity constraints in integer programs. Technical report, Optimization Online. `http://www.optimization-online.org/DB_HTML/2015/02/4768.html`.

Wang, Z., Zheng, S., Ye, Y., and Boyd, S. (2008). Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM Journal on Optimization*, 19(2):655–673.

Witten, D. M. and Tibshirani, R. (2011). Supervised multidimensional scaling for visualization, classification, and bipartite ranking. *Computational Statistics & Data Analysis*, 55(1):789–801.

Wolfe, P. (2013). Making sense of Big Data. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 18031–18032.

Wood, J. and Dykes, J. (2008). Spatially ordered treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1348–1355.

Xu, K. S., Kliger, M., and Hero III, A. O. (2013). A regularized graph layout framework for dynamic network visualization. *Data Mining and Knowledge Discovery*, 27(1):84–116.

Yoghourdjian, V., Dwyer, T., Gange, G., Kieffer, S., Klein, K., and Marriott, K. (2016). High-quality ultra-compact grid layout of grouped networks. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):339–348.

Žilinskas, A. and Podlipskytė, A. (2003). On multimodality of the sstress criterion for metric multidimensional scaling. *Informatica*, 14(1):121–130.

Žilinskas, A. and Žilinskas, J. (2009). Branch and bound algorithm for multidimensional scaling with city-block metric. *Journal of Global Optimization*, 43(2):357–372.

Žilinskas, J. (2012). Parallel branch and bound for multidimensional scaling with city-block distances. *Journal of Global Optimization*, 54(2):261–274.